Proceedings of the EACL 2003 Workshop on Natural Language
Processing for Question Answering, April 2003, Budapest, Hungary

Selectively Using Relations to Improve Precision in Question Answering

Boris Katz and **Jimmy Lin**
MIT Artificial Intelligence Laboratory
200 Technology Square
Cambridge, MA 02139
{boris,jimmylin}@ai.mit.edu

Abstract

Despite the intuition that linguistically sophisticated techniques should be beneficial to question answering, real gains in performance have yet to be demonstrated empirically in a reliable manner. Systems built around sophisticated linguistic analysis generally perform worse than their linguistically-uninformed cousins. We believe that the key to effective application of natural language processing technology is to selectively employ it only when helpful, without abandoning simpler techniques. To this end, we identify two linguistic phenomena that current information extraction driven systems have difficulty with, and demonstrate how syntactic processing can help. By indexing syntactic relations that can be reliably extracted from corpus text and matching questions with documents at the relation level, we demonstrate that syntactic analysis enables a question answering system to successfully handle these phenomena, thereby improving precision.

1 Introduction

Most current question answering systems utilize a combination of information retrieval and information extraction techniques to find short answers to fact-based questions such as "Who killed Lincoln?" The dominant approach, driven by IE technology, is to first find a set of potentially relevant passages and then "pinpoint" the exact location of the answer by searching for an entity whose semantic type matches the question type.

Although respectable performance can be achieved with this relatively simple two-stage process, there exist empirical limits on the effectiveness of this approach. By analyzing a subset of TREC-9 and CBC questions, Light et al. (2001) established an expected upper bound on the performance of a question answering system with perfect passage retrieval, named-entity detection, and question classification at around 70%. The primary reason for this limit is that many named entities of the same semantic type often occur close together, and a QA system, without the aid of any additional knowledge, would be forced to choose randomly.

Although we are still years away from systems that can provide accurate semantic analysis on open-domain text, significant headway has been made in the syntactic analysis of documents. Matching questions and passages based on syntactically-derived relations offers an interim solution for overcoming the limitation of IE-based question answering systems. Although previous systems have employed similar techniques (to be discussed further in Section 2), they generally did not perform as well as systems that utilize linguistically-uninformed techniques. We attribute these results to the reliance on NLP techniques as the fundamental machinery for question answering, despite their brittleness. Instead, we suggest a more pragmatic approach: continue to use linguistically-uninformed techniques as the foundation of a question answering system, and apply sophisticated NLP approaches *only when they are known to improve performance.* To this end, we identify two linguistic phenomena that current IE-driven systems have difficulty with, and

(1) The bird ate the snake.
(1′) The snake ate the bird.
(2) the largest planet's volcanoes
(2′) the planet's largest volcanoes
(3) the house by the river
(3′) the river by the house
(4) The Germans defeated the French.
(4′) The Germans were defeated by the French.

Figure 1: Semantic differences that cannot be captured by lexical content alone

demonstrate how syntactic processing can help. Additionally, to overcome the usual brittleness associated with syntactic parsing, we utilize syntactic relations, captured in terms of ternary expressions, that can be reliably and easily extracted from complex real-world text.

The fragment pairs in Figure 1 illustrate the elusive nature of "meaning"; although fragments in each pair are nearly indistinguishable in terms of lexical content, their meanings are vastly different. Naturally, because one text fragment may be an appropriate answer to a question while the other fragment may not be, a question answering system seeking to achieve high precision must differentiate the semantic content of the pairs. Ideally, question answering should be based on the semantics of questions and documents, but unfortunately, full semantic analysis is beyond current technology in anything but highly-restricted domains. Instead, as a compromise, we propose to match questions and answers at the syntactic level.

2 Hasn't this been tried before?

The concept of marrying NLP techniques with large-scale IR is not new, but effective integration of the two remains an open research question. Fagan (1987) experimented with indexing noun phrases and prepositional phrases. More recently, various researchers have experimented with indexing syntactically derived word pairs (Strzalkowski et al., 1996; Zhai et al., 1996; Arampatzis et al., 1998); the types of constructions examined in the context of indexing include linguistically motivated pairs such as head/modifier and adjective/noun. In addition, full linguistic trees (Smeaton et al., 1994) and case frames (Croft and Lewis, 1987) as units of indexing have been tried. However, none of these experiments resulted in dramatic improvement in precision or recall, and often even resulted in degraded performance. In all of theses studies, the word-level index was directly augmented with linguistically-derived representations. Often, this caused performance issues because the creation of an index is limited by the speed of the parser, and because sophisticated linguistic representations were not amenable to large-scale indexing.

The current generation of question answering systems that employ NLP alleviate performance problems by delaying linguistic analysis until the corpus has been narrowed down to a small set of candidate documents or passages. The MURAX System (Kupiec, 1993) is an early example of such an approach. More recently, Litkowski (1999) described a system that utilizes a combination of syntactic relations, e.g., subject-verb-object, and some semantic relations, e.g., time and location. After initially retrieving a set of candidate documents, the system then parses both the question and the passages and attempts matching at the relation level. Unfortunately, this and similar techniques that depend heavily on syntactic analysis, e.g., PIQASso (Attardi et al., 2001), yielded relatively poor performance. A drawback of this two-step paradigm is low recall: if the keyword-based document retrieval system does not return *any* relevant documents due to such problems as synonymy, anaphora, or argument alternations, any amount of additional processing is useless. The current work-around to this problem is to implement feedback loops that relax the query set if the results are too restrictive (Moldovan et al., 2002). Not only does this introduce complex dependencies in a system's architecture, but it also necessitates the addition of new modules to assess the quality of the result sets.

In the domain of information access, we attribute the mediocre track record of sophisticated linguistic techniques not to the impotence of NLP technology in general, but rather to the manner in which it has been applied. Results appear to demonstrate that the current level of natural language technology is still too brittle to be applied in all situations. Because existing linguistically-impoverished methods have proven to be robust and capable of delivering useful levels of performance, we propose a more pragmatic approach: recognize situations where linguistic techniques would help and employ them only when necessary.

With this approach, the critical question becomes: under what circumstances can natural language processing techniques improve question answering? In reply, we describe two broad linguistic phenomena that are difficult to handle with the information extraction driven paradigm. Within these two areas, the use of syntactic relations results in a dramatic improvement in the precision of question answering systems.

3 Two Phenomena

We have identified two broad phenomena that cannot be easily handled by linguistically uninformed question answering systems: *semantic symmetry* and *ambiguous modification*. Examples representing typical results from current QA systems (Figure 2) help illustrate the phenomena.

The first example (Q1) demonstrates the problem of semantic symmetry: although the questions "What do frogs eat?" and "What eats frogs?" are similar at the word level, they have very different meanings and should be answered differently. The second example (Q2) demonstrates the problem of ambiguous modification: adjectives like *largest* and prepositional phrases such as *in the Solar System* can modify a variety of different head nouns. Potential answers may contain the correct entities, but they may not be in the correct syntactic relations with each other, e.g., *the largest planet* instead of *the largest volcano*. Both these phenomena could benefit from a more detailed linguistic treatment to pinpoint more precise answers.

Semantic symmetry occurs when the selectional restrictions of different arguments of the same head overlap; for example, the selectional restriction for the subject of *eat* is *animate* and the selectional restriction for the object is *edible*, so semantic symmetry occurs whenever the subject and object are both animate and edible. In these cases, lexical content is insufficient to determine the meaning of the sentence—syntactic analysis is required to discover head-arguments relations.

(Q1) What do frogs eat?
(A1) Adult *frogs eat* mainly insects and other small animals, including earthworms, minnows, and spiders.
(A2) Alligators *eat* many kinds of small animals that live in or near the water, including fish, snakes, *frogs*, turtles, small mammals, and birds.
(A3) Some bats catch fish with their claws, and a few species *eat* lizards, rodents, small birds, tree *frogs*, and other bats.

(Q2) What is the largest volcano in the Solar System?
(B1) Mars boasts many extreme geographic features; for example, Olympus Mons, the *largest volcano in the solar system*.
(B2) The Galileo probe's mission to Jupiter, the *largest* planet *in the Solar system*, included amazing photographs of the *volcanoes* on Io, one of its four most famous moons.
(B3) Even the *largest volcanoes* found on Earth are puny in comparison to others found around our own cosmic backyard, *the Solar System*.
(B4) Olympus Mons, which spans an area the size of Arizona, is the *largest volcano in the Solar System*.

Figure 2: Examples illustrating semantic symmetry and ambiguous modification (emphasis added)

Ambiguous modification occurs when an argument's selectional restrictions are so *un*restrictive that the argument can belong to more than one head in a particular context. Since nearly anything can be *large* or *good*, syntactic analysis is necessary to pin down which head this argument actually belongs to.

In order to define the phenomena described above more formally, we shall adopt a first order predicate logic formalism. In our description, sentences are parsed into logical forms consisting of relations (*n*-ary predicates) with words as their arguments. The semantics of the predicate logic can be modeled as constraints on the domain of the arguments: a logical expression is semantically valid, or "makes sense," if and only if the arguments of every predicate type-check with constraints imposed by that predicate. For example, if R is a one place predicate whose argument is constrained on the set $s = \{a, b, c\}$, then $R(d)$ is not a semantically valid expression. Given this definition of semantic validity, we can define a function $\mathcal{S}$ on any logical expression e:

$$\mathcal{S}(e) = \begin{cases} 1 & \text{if } e \text{ is a semantically valid} \\ 0 & \text{otherwise} \end{cases}$$

Using this framework, we can then formally define semantically symmetric relations:

> **Semantic Symmetry**
> A relation is *semantically symmetric* iff there exists w, w_1, and w_2 such that $\mathcal{S}(R(w, w_1)) = \mathcal{S}(R(w_2, w)) = 1$.

A typical example of semantically symmetric relations involves sentences where one can swap the subject and object and still end up with a sentence that "makes sense," with respect to $\mathcal{S}$. This occurs when the domains of the arguments of a particular predicate (as determined by the selectional restrictions of that particular head) have a non-null intersection, e.g., some questions involving predators and prey:

> $eat(x, y)$
> $x \subset$ animate-agent
> $y \subset$ animate-agent, inanimate-object . . .

Thus, the difficulty with "What do frogs eat?" is that the question seeks entities that fulfill a certain relation, namely, all x such that $eat(frog, x)$ is true. However, statements that answer the question "What do frogs eat?" and "What eats frogs?" are likely to contain both the relevant keywords *frog* and *eat*. Since *eat* is a semantically symmetric relation, both $eat(frog, x)$ and $eat(x, frog)$ are likely to be found within the corpus.

The phenomenon of semantically symmetric relations observed above is by no means an isolated instance. Examples of such verbs abound in English, e.g., *visit, meet, defeat, kill, love, see,* and many, many more. Together, all the verbs in this class of semantically symmetric relations present a problem for any non-linguistically informed QA system.

Ambiguous modification represents another phenomenon that linguistically-uninformed QA systems have difficulty handling:

> **Ambiguous Modification**
> A word w, involving a relation R, is an ambiguous modifier iff there exist at least two words, w_1 and w_2, in the same local context as w, such that $\mathcal{S}(R(w, w_1)) = \mathcal{S}(R(w, w_2)) = 1$.

A phrase like *the planet's largest volcanoes* illustrates the ambiguous modification phenomenon. For example, the adjective *largest*, involved in an adjective-noun modification relation, is not very constrained in its possible choices of head nouns, and hence is free to "float" among nouns in its local context.[1] This means that given passages with similar lexical content containing the adjective *largest*, it is difficult to determine exactly which head noun the adjective is modifying without syntactic analysis.[2] Hence, if the relation under consideration is crucial to answering the question,[3] syntax is required to precisely pin down relations from both the question and the corpus to ensure that the relations match satisfactorily. The possessive relation involving *planet* and *volcano* is another instance of the ambiguous modification phenomenon because there are other potential choices for modifiers and modifiees, e.g., *the planet's ocean* or *the island's volcano*.

In the example (Q2) in Figure 2, there are two ambiguous modifiers: *largest*, involved in an adjective-noun modification relation, and *in the Solar System*, involved in a location relation. Sentences (B2) and (B3) have the correct lexical content, but only some of the correct syntactic relations. In (B2), both *largest* and *in the Solar System* modify the incorrect head noun. In (B3), *in the Solar System* does not modify the correct head noun.

[1] By local context, we mean the set of head nouns surrounding the ambiguous modifier in question. The size of this local context is related to the granularity of the information retrieval process we are comparing against. For example, in document retrieval, where all the words in a document are considered, an ambiguous modifier can potentially modify a head noun anywhere in the document.

[2] Researchers have attempted to work around this problem by indexing consecutive word pairs, but there are simple examples in which this technique would not help, e.g., "the brown apple" vs. "the apple brown from bruising," "John's house" vs. "house of John."

[3] True in this case, because we are looking for a planet with the *largest* volcano, and not, for example, the largest planet that possesses a volcano.

Adjectives are often ambiguous modifiers: given a context with a pool of adjectives and nouns, any particular adjective could potentially modify many nouns. Under such circumstances, a question answering system cannot achieve high precision without exactly identifying the particular relation between words through detailed syntactic analysis.

4 Ternary Expressions

In the previous section, we identified two natural language phenomena that pose difficulties to traditional IE-driven QA systems, difficulties which are alleviated by endowing a system with the ability to perform syntactic analysis. To do so, we need a syntactic representation that can capture the important relations between words in text, yet is amenable to rapid processing and matching.

Although parse trees capture syntactic relations, they are difficult to generate, store, and manipulate rapidly. In a particular set of experiments, Smeaton et al. (1994) lamented the awkwardness and slow speed of processing full parse trees. Indeed, generating detailed parses is often time-consuming, and much of the parse information is not directly relevant to question answering anyway.

Similarly, logical form is not the best representation for our purposes, despite its utility in precisely and formally delineating problems. Manipulation of logical forms requires significant computational machinery, e.g., unification mechanisms, a general theorem prover, etc. Furthermore, using logic as the paradigm for matching relations requires careful and exact formulation of all axioms and allowed inferences *a priori*, which is not practical due to the ambiguous nature of language.

We believe that a more pragmatic solution to capturing the relations relevant for question answering is to distill natural language text into ternary (three-place) expressions (Katz, 1988). Ternary expressions may be intuitively viewed as subject-relation-object triples, and can easily express many types of relations, e.g., subject-verb-object relations, possession relations, etc. The START Question Answering System (Katz, 1997) has been employing such representations effectively in question answering for the last two decades, and we find that they are a good compromise between expressiveness and simplicity.

Using ternary expressions, the semantic differences between the text fragments presented in Figure 1 can be distinguished at the syntactic level:

(1) [bird eat snake]
(1′) [snake eat bird]
(2) [largest `adjmod` planet]
[planet `poss` volcano]
(2′) [largest `adjmod` volcano]
[planet `poss` volcano]
(3) [house by river]
(3′) [river by house]
(4) [Germans defeat French]
(4′) [French defeat Germans]

5 Initial Experiments

In order to demonstrate our ideas, we have implemented Sapere, a prototype natural language question answering system that retrieves answers by matching ternary expressions derived from the question with those derived from the corpus text.

As a baseline for comparison, we implemented a simple boolean retrieval engine that uses a standard inverted keyword index to index documents at the sentence level. All stopwords are discarded, and all content words are stemmed. For the baseline, a conjunctive query of all non-stopwords from the query is issued to the boolean retrieval engine; the resulting set of sentences is ranked by the number of non-stopwords that were found in each sentence. Although boolean keyword search systems do not perform as well as state-of-the-art IR engines, we believe that they serve as an adequate baseline for comparison since there is substantial empirical evidence that boolean-based passage retrieval techniques are sufficient to obtain reasonable performance in question answering tasks (Light et al., 2001).

Sapere is primarily a relations-indexing engine; it stores and indexes ternary expressions extracted from the corpus text and performs matching at the relation level between questions and sentences stored in its index. Ternary expressions are generated from text by postprocessing the results of Minipar (Lin, 1993), a fast and robust functional dependency parser. Currently, Sapere detects the following types of relations: subject-verb-object (including passive constructions), adjective-noun modification, noun-noun modification, possessive

What countries has Japan invaded?
What eats snakes?
Who defeated the Spanish Armada?
When do lions hunt?
What is the largest planet?

Figure 3: Sample questions used in the user study

relations, predicate nominatives, predicate adjectives, appositives, and prepositional phrases.

Ternary expressions are similarly derived from the question, with the *wh*-entity left as an unbound variable. Sapere attempts to match relations in the question with those found in the corpus text, thereby binding the unbound variable in the question with the actual answer. If such a match occurs, the candidate sentence is returned.

The test corpus used in our experiments was an electronic version of the Worldbook Encyclopedia, which contains approximately 20,000 articles. The entire corpus was parsed and relations extracted from it were indexed by Sapere.

The test set consisted of 16 hand-selected questions that illustrate the two linguistic phenomena discussed previously; some of these questions are shown in Figure 3. For example, "Who defeated the Spanish Armada?" probes semantically symmetric relations; "What is the largest planet?" tests ambiguous modification.

Results from both the baseline system and Sapere were collected and manually judged to be either relevant or irrelevant. The comparison between the baseline and Sapere can be seen in Table 1. For the sixteen questions in this particular test set, indexing and matching relations (Sapere) achieved a precision of 0.84 ± 0.11, while basic keyword matching (baseline) achieved only a precision of 0.29 ± 0.11. In addition, Sapere returned far fewer results, reducing the amount of reading the user must perform to obtain the correct answer.

A sample output from the baseline keywords indexer is shown in Figure 4. After removing stopwords from the query, our simple keyword search engine returned 32 results that contain the keywords *frog* and *eat*. Of all the sentences returned, only (C4) correctly answers the user query.

	Relations	Keywords
Avg. # of sentences returned	4	43.88
Avg. # of correct sentences	3.13	5.88
Avg. precision	0.84	0.29

Table 1: Comparison between relations and keyword indexing.

(Q3) What do frogs eat?
(C1) Alligators eat many kinds of small animals that live in or near the water, including fish, snakes, frogs, turtles, small mammals, and birds.
(C2) Some bats catch fish with their claws, and a few species eat lizards, rodents, small birds, tree frogs, and other bats.
(C3) Bowfins eat mainly other fish, frogs, and crayfish.
(C4) Adult frogs eat mainly insects and other small animals, including earthworms, minnows, and spiders.
(C5) Kookaburras eat caterpillars, fish, frogs, insects, small mammals, snakes, worms, and even small birds.
(C6) ...

Figure 4: Sample output from the baseline keyword indexer

By comparison, our relations matcher returns only (C4) as the correct answer.

Syntactic analysis of the question can distinguish the ternary expression derived from "What do frogs eat?" ([frog eat $?x$]) from the ternary expression representing "What eats frogs?" ([$?x$ eat frog]) and respond to the former question with sentences containing such relations as [frog eat insect] instead of sentences containing relations like [alligator eat frog] , despite the similar lexical content of all the sentences.

6 Discussion

Large precision gains were achieved in our experiments because the test set was engineered to illustrate the two phenomena that we identified: semantic symmetry and ambiguous modification. By exactly pinpointing the areas in which natural language techniques are helpful, Sapere was able to exploit syntactic parsing to dramatically increase precision. Instead of relying exclusively on sophisticated linguistic techniques, we suggest that simpler linguistically-uninformed techniques

(Q1003) What is the highest dam in the U.S.?
(D1) Extensive flooding was reported Sunday on the Chattahoochee River in Georgia as it neared its crest at Tailwater and George *Dam*, its *highest* level since 1929. (AP900319-0047)
(D2) A swollen tributary the Ganges River in the capital today reached its *highest* level in 34 years, officials said, as soldiers and volunteers worked to build *dams* against the rising waters. (AP880902-0066)
(D3) Two years ago, the numbers of steelhead returning to the river was the *highest* since the *dam* was built in 1959. (SJMN91-06144185)

Figure 5: Sample answers from TREC that illustrate problems with ambiguous modification.

should not be abandoned. The resulting combined system will see more modest, but still valuable, gains in precision.

Because Sapere currently derives relations from Minipar, the quality of the relations ultimately depends on the quality of the parser. Despite parse errors, indexing syntactic representations boosts the performance of our question answering system because the relations we chose to index are generally ones that can be reliably and accurately extracted from text, e.g., adjective-noun modification, simple subject-verb-object relations from the matrix clause, etc. However, deriving relations using off-the-shelf parsers, while convenient for current experimental purposes, might not be the ideal situation in the longer term. A custom-built lightweight parser specifically designed to extract relations might be faster and more accurate.

A quick analysis of the TREC corpus reveals that instances of the phenomena we've described occur frequently. Many real-word user questions crucially depend on particular relations between entities, e.g., "When were William Shakespeare's twins born?", taken from Q1002 at TREC-2001 QA Track. The vast majority of incorrect answers reported the birthday of Shakespeare himself, because the crucial possessive relation between the poet and his twins was neglected. Shown in Figure 5 are a few more incorrect answers returned by actual systems in TREC-2001 QA Track, attributed to ambiguous modification.

A distinct feature of Sapere is that it indexes all relations extracted from the corpus, instead of post-processing a set of candidate documents or passages from an IR system. We believe that this strategy overcomes the recall problem associated with the standard two-step approach: if the IR system doesn't produce *any* relevant documents, further processing (linguistic or otherwise) is useless. Take the example shown in Figure 5: since *highest* and *dam* co-occur frequently, it is possible that irrelevant results will "swamp out" relevant documents, so that relevant documents might not even be considered in later processing stages of a system. This problem is especially severe if the answer is only mentioned in one document from the entire corpus. Indexing relations from the ground up is a potential solution to this problem. Unfortunately, this strategy is also very computationally intensive, limiting our initial experiments to a smaller corpus. Our next goal is to apply our relations-indexing technique to the much larger TREC corpus.

Sapere's relation matcher currently attempts to match relations derived from the question against relations derived from the corpus. It does not, however, verify the absence of particular relations that may "invalidate" a response. The simplest example of this is explicit negation, although other adverbial modifiers (e.g., *never*, *barely*, *unlikely*), modals, verbs that take sentential complements (e.g., *deny*, *hallucinate*), and even certain adjective (e.g., *former*, *non-existent*) can have the same effect. We are currently building a more sophisticated relations matcher that will take these effects into account. An even more troublesome issue is that of implied relations in a user query. When a user asks "What is the tallest mountain?", the qualification "in the world" is generally assumed. The implied relation is a matter of common sense, based on shared world knowledge. Without this knowledge, a system like Sapere might return the tallest mountain *in the world*, the tallest mountain *in Asia*, the tallest mountain *in Europe*, etc. Although these are arguably all correct answers (in some sense), it might not be appropriate to list the tallest mountains in every country. Short of "solving" common sense, the best solution is to build better interfaces that allow users to iteratively refine queries and supply missing information.

Another important issue remains to be resolved: how do we classify questions into classes that exhibit either semantic symmetry or ambiguous modification? It should be possible to automatically construct a relational "lexicon" of symmetric relations and ambiguous modifiers. Semantically symmetric relations can be recognized by considering the domains of their arguments; if significant intersection exists between the arguments, then the relation exhibits semantic symmetry. Ambiguous modifiers can be automatically acquired in a similar way; given a relation $R(m, w)$ between modifier m and head w (extracted from the question), find all heads $c_1 \ldots c_n$ that co-occur with w; if the relation $R(m, c_i)$ holds for any of the c's, then we can conclude that the modifier m is ambiguous.

A future research direction is to expand on our catalog of linguistic phenomena; we have presented two here, but we believe that there are additional opportunities where syntactic analysis could benefit question answering. By selectively using natural language processing technology only when they are known to be beneficial, a system like Sapere can significantly boost question answering performance. Naturally, if matching user queries and corpus documents fails at the syntactic relations level (i.e., produces no matches), a question answering system should fall back on less linguistically-informed approaches.

7 Conclusions

Many members of the IR and NLP community believe that question answering is an application in which sophisticated linguistic techniques will truly shine. However, this belief has yet to be empirically verified, as linguistically-impoverished systems have generally outperformed those that attempt syntactic or semantic analysis. In support of those techniques, we have categorized and empirically verified two phenomena, semantic symmetry and ambiguous modification, in which syntactic relations prove to be extremely effective. By first identifying, and then selectively applying linguistically-sophisticated techniques, we can overcome the limitations of present-day natural language technology and increase the performance of question answering systems.

Acknowledgements

We would like to thank Sue Felshin, Gregory Marton, and Stefanie Tellex for their valuable suggestions and comments on earlier drafts of this paper. We would also like to thank the anonymous reviewers for their comments and suggestions. This research is funded by DARPA under contract number F30602-00-1-0545 and administered by the Air Force Research Laboratory. Additional funding is provided by the Oxygen Project.

References

A. Arampatzis, Th.P. van der Weide, C.H.A. Koster, and P. van Bommel. 1998. Phrase-based information retrieval. *Information Processing and Management*, 34(6):693–707, December.

G. Attardi, A. Cisternino, F. Formica, M. Simi, A. Tommasi, and C. Zavattari. 2001. PIQASso: Pisa question answering system. In *TREC 2001*.

B. Croft and D. Lewis. 1987. An approach to natural language processing for document retrieval. In *SIGIR-1987*.

J. Fagan. 1987. *Experiments in Automatic Phrase Indexing for Document Retrieval: A Comparisons of Syntactic and Non-Syntactic Methods*. Ph.D. thesis, Cornell University.

B. Katz. 1988. Using English for indexing and retrieving. In *RIAO'88*.

B. Katz. 1997. Annotating the World Wide Web using natural language. In *RIAO'97*.

J. Kupiec. 1993. MURAX: A robust linguistic approach for question answering using an an on-line encyclopedia. In *SIGIR-1993*.

M. Light, G. Mann, E. Riloff, and E. Breck. 2001. Analyses for elucidating current question answering technology. *Journal of Natural Language Engineering*, Fall–Winter.

D. Lin. 1993. Principle-based parsing without overgeneration. In *ACL-1993*.

K. Litkowski. 1999. Question-answering using semantic relation triples. In *TREC-8*.

D. Moldovan, M. Paşca, S. Harabagiu, and M. Surdeanu. 2002. Performance issues and error analysis in an open-domain question answering system. In *ACL-2002*.

A. Smeaton, R. O'Donnell, and F. Kelledy. 1994. Indexing structures derived from syntax in TREC-3: System description. In *TREC-3*.

T. Strzalkowski, L. Guthrie, J. Karlgren, J. Leistensnider, F. Lin, J. Perez-Carballo, T. Straszheim, J. Wang, and J. Wilding. 1996. Natural language information retrieval: TREC-5 report. In *TREC-5*.

C. Zhai, X. Tong, N. Milic-Frayling, and D. Evans. 1996. Evaluation of syntactic phrase indexing—CLARIT NLP track report. In *TREC-5*.