Experiments with Open-Domain Textual Question Answering

Sanda M. Harabagiu and **Marius A. Paşca** and **Steven J. Maiorano**
Department of Computer Science and Engineering
Southern Methodist University
Dallas, TX 75275-0122
{sanda,marius,steve}@renoir.seas.smu.edu

Abstract

This paper describes the integration of several knowledge-based natural language processing techniques into a Question Answering system, capable of mining textual answers from large collections of texts. Surprizing quality is achieved when several lightweight knowledge-based NLP techniques complement mostly shallow, surface-based approaches.

1 Background

The last decade has witnessed great advances and interest in the area of Information Extraction (IE) from real-world texts. Systems that participated in the TIPSTER MUC competitions have been quite successful at extracting information from newswire messages and filling templates with information pertaining to events or situations of interest. Typically, the templates model queries regarding *who* did *what* to *whom*, *when* and *where*, and eventually *why*.

Recently, a new trend in information processing from texts has emerged. Textual Question Answering (Q/A) aims at identifying the answer of a question in large collections of on-line documents. Instead of extracting all events of interest and their related entities, a Q/A system highlights only a short piece of text, accounting for the answer. Moreover, questions are expressed in natural language, are not constrained to a specific domain and are not limited to the six question types sought by IE systems (i.e. who_1 did $what_2$ to $whom_3$, $when_4$ and $where_5$, and eventually why_6).

In open-domain Q/A systems, the finite-state technology and domain knowledge that made IE systems successful are replaced by a combination of (1) knowledge-based question processing, (2) new forms of text indexing and (3) lightweight abduction of queries. More generally, these systems combine creatively components of the NLP basic research infrastructure developed in the 80s (e.g. the computational theory of Q/A reported in (Lehnert 1978) and the theory of abductive interpretation of texts reported in (Hobbs et al.1993)) with other shallow techniques that make possible the open-domain processing on real-world texts.

The idea of building open-domain Q/A systems that perform on real-world document collections was initiated by the eighth Text REtrieval Conference (TREC-8), by organizing the first competition of answering fact-based questions such as *"Who came up with the name, El Nino?"*. Resisting the temptation of merely porting and integrating existing IE and IR technologies into Q/A systems, the developers of the TREC Q/A systems have not only shaped new processing methods, but also inspired new research in the challenging integration of surface-text-based methods with knowledge-based text inference. In particular, two clear knowledge processing needs are presented: (1) capturing the semantics of open-domain questions and (2) justifying the correctness of answers.

In this paper, we present our experiments with integrating knowledge-based NLP with shallow processing techniques for these two aspects of Q/A. Our research was motivated by the need to enhance the precision of an implemented Q/A system and by the requirement to prepare it for scaling to more complex questions than those presented in the TREC competition. In the remaining of the paper, we describe a Q/A architecture that allows the integration of knowledge-based NLP processing with shallow processing and we detail their interactions. Section 2 presents the functionality of several knowledge processing modules and describes the NLP techniques for question and answer processing. Section 3 explains the semantic and logical interactions of processing questions and answers whereas Section 4 highlights the inference aspects that implement the justification option of a Q/A system. Section 5 presents the results and the evaluations whereas Section 6 concludes the paper.

2 The NLP Techniques

Surprising quality for open-domain textual Q/A can be achieved when several lightweight knowledge-based NLP techniques complement mostly shallow, surface-based approaches. The processing imposed by Q/A systems must be distinguished, on the one hand, from IR techniques, that locate sets of doc-

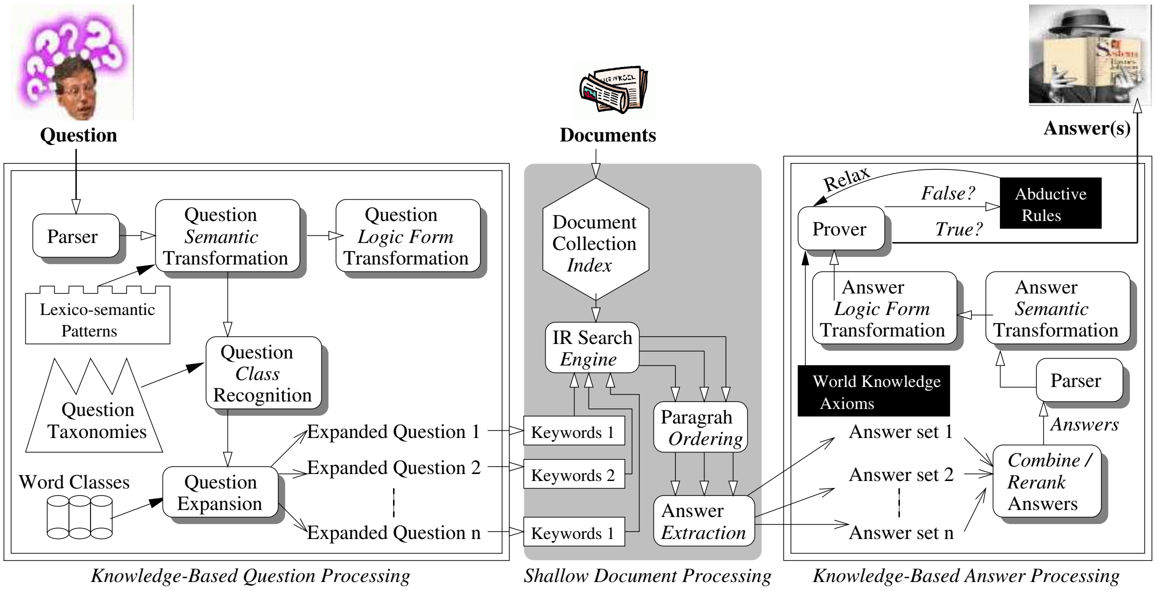

Figure 1: An architecture for knowledge-based Question/Answering

uments containing the required information, based on *keywords techniques*. Q/A systems are presented with natural language questions, far richer in semantics than a set of keywords eventually structured around some operators. Furthermore, the output of Q/A systems is either the actual answer identified in a text or small text fragments containing the answer. This eliminates the user's trouble of finding the required information in sometimes large sets of retrieved documents. Open-domain Q/A systems must also be distinguished, on the other hand, from IE systems that model the information need through database templates, thus less naturally than a textual answer. Moreover, open-domain IE is still difficult to achieve, because its linguistic pattern recognition relies on domain-dependent lexico-semantic knowledge.

To be able to satisfy the open-domain constraints, textual Q/A systems replace the linguistic pattern matching capabilities of IE systems with methods that rely on the recognition of the *question type* and of the *expected answer type*. Generally, this information is available by accessing a classification based on the *question stem* (i.e. *what*, *how much*, *who*) and the head of the first noun phrase of the question. Question processing also includes the identification of the *question keywords*. Empirical methods, based on a set of ordered heuristics operating on the phrasal parse of the question, extract keywords that are passed to the search engine. The overall precision of the Q/A system depends also on the recognition of the *question focus*, since the answer extraction, succeeding the IR phase, is centered around the question focus. Unfortunately, empirical methods for focus recognition are hard to develop without the availability of richer semantic knowledge.

Special requirements are set on the document processing component of a Q/A system. To speed-up the answer extraction, the search engine returns only those paragraphs from a document that contain all queried keywords. The paragraphs are ordered to promote the cases when the keywords not only are as close as possible, but also preserve the syntactic dependencies recognized in the question. Answers are extracted whenever the question topic and the answer type are recognized in a paragraph. Thereafter the answers are scored based on several bag-of-words heuristics. Throughout all this processing, the NLP techniques are limited to (a) named entity recognition; (b) semantic classification of the question type, based on information provided by an off-line question taxonomy and semantic class information available from WordNet (Fellbaum 1998); and (c) phrasal parsing produced by enhancing Brill's part-of-speech tagger with some rules for phrase formation.

However simple, this technology surpasses 75% precision on trivia questions, as posed in the TREC-8 competition (cf. (Moldovan et al.1999)). An impressive improvement of 14% is achieved when more knowledge-intensive NLP techniques are applied at both question and answer processing level. Figure 1 illustrates the architecture of a system that has enhanced Q/A performance.

As represented in Figure 1, all three modules of the Q/A system preserve the shallow processing components that determine good performance. In the *Question Processing* module, the Question Class recognizer, working against a taxonomy of questions, still constitutes the central processing that takes place at this stage. However, a far richer representation of the question classes is employed. To be able to classify against the new question taxonomy each question is first fully parsed and transformed into a semantic representation that captures all relationships between phrase heads.

The recognition of the question class is based on the comparison of the question semantic representation with the semantic representation of the nodes from the question taxonomy. Taxonomy nodes encode also the answer type, the question focus and the semantic class of question keywords. Multiple sets of keywords are generated based on their semantic class, all pertaining to the same original question. This feature enables the search engine to retrieve multiple sets of documents, pertaining to multiple sets of answers, that are extracted, combined and ranked based on several heuristics, reported in (Moldovan et al.1999). This process of obtaining multiple sets of answers increases the likelihood of finding the correct answer.

However, the big boost in the precision of the knowledge-based Q/A system is provided by the option of enabling the justification of the extracted answer. All extracted answers are parsed and transformed in semantic representations. Thereafter, both semantic transformations for questions and answers are translated into logic forms and presented to a simplified theorem prover. The proof backchains from the question to the answer, its trace generating a justification. The prover may access a set of abduction rules that relax the justification process. Whenever an answer cannot be proven, it is discarded. This option solves multiple situations when the correct answer is not ranked as the first return, due to stronger surface-text-based indicators in some other answers, which unfortunately are not correct.

This architecture allows for simple integration of semantic and axiomatic knowledge sources in a Q/A system and determines efficient interaction of text-surface-based and knowledge-based NLP techniques.

3 Interactions

Three main interactions between text-surface-based and knowledge-based NLP techniques are designed in our Q/A architecture:

1. When multiple sets of question keywords are passed to the search engine, increasing the chance of finding the text paragraph containing the answer.
2. When the question focus and the answer type, resulting from the knowledge-based processing of the question, are used in the extraction of the answer, based on several empirical scores.
3. When the justification option of the Q/A system is available. Instead of returning answers scored by some empirical measures, a proof of the correctness of the answer is produced, by accessing the logical transformations of the question and the answer, as well as axioms encoding world knowledge.

All these interactions depend on two factors: (1) the transformations of the question or answer into semantic or logical representations; and (2) the availability of knowledge resources, e.g. the question taxonomy and the world knowledge axioms. The availability of new, high-performace parsers that operate on real world texts determines the transformation into semantic and logic formulae quite simple. In addition, the acquisition of question taxonomies is alleviated by machine learning techniques inspired from bootstrapping methods that learn linguistic patterns and semantic dictionaries for IE (cf. (Riloff and Jones, 1999)). World knowledge axioms can also be easily derived by processing the gloss definitions of WordNet (Fellbaum 1998).

3.1 Semantic and Logic Transformations

Semantic Transformations

Instead of producing only a phrasal parse for the question and answer, we make use of one of the new statistical parsers of large real-world text coverage (Collins, 1996). The parse trees produced by such a parser can be easily translated into a semantic representation that (1) comprises all the phrase heads and (2) captures their inter-relationships by anonymous links. Figure 2 illustrates both the parse tree and the associated semantic representation of a TREC-8 question.

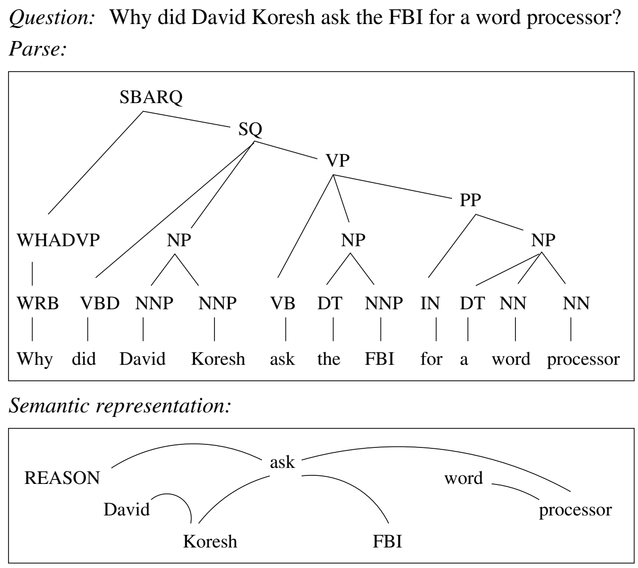

Figure 2: Question semantic transformation

The actual transformation into semantic representation of a question or an answer is obtained as a by-product of the parse tree traversal. Initially, all leaves of the parse tree are classified as *skipnodes* or *non-skipnodes*. All nouns, non-auxiliary verbs, adjectives and adverbs are categorized as non-skipnodes. All the other leaves are skipnodes. Bottom-up traversal of the parse tree entails the propagation of leaf labels whenever the parent node has more than one non-skipnode child. A rule based on the syntactic category of the father selects one of the children to propagate its label at the next level in the tree. The winning node will then be considered linked to all the other former siblings that are non-skipnodes. The propagation continues until the parse tree root receives a label, and thus a semantic graph is created as a by-product. Part of the label propagation, we also consider that whenever all children of a non-terminal are skipnodes, the parent becomes a skipnode as well.

Figure 3 represents the label propagation for the parse tree of the question represented in Figure 2. The labels of *Koresh*, *ask*, *FBI* and *processor* are propagated to the next level. This entails that *Koresh* is linked to *David*, *ask* to *FBI* and *processor* and *processor* to *word*. As *ask* becomes the label of the tree root, it is also linked to *REASON*, the question type determined by the question stem: *why*. The label propagation rules are identical to the rules for mapping from trees to dependency structures used my Michael Collins (cf. (Collins, 1996)). These rules identify the head-child, and propagate its label up in the tree.

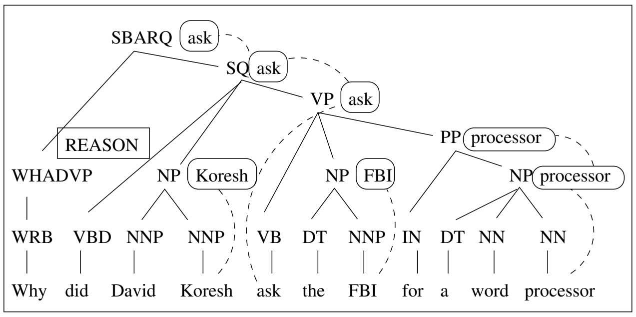

Figure 3: Parse tree traversal

Logical Transformations

The logical formulae in which questions or answers are translated are inspired by the notation proposed in (Hobbs, 1986-1) and implemented in `TACITUS` (Hobbs, 1986-2).

Based on the davidsonian treatment of action sentences, in which events are treated as individuals, every question and every answer are transformed in a first-order predicate formula for which (1) verbs are mapped in predicates *verb(e,x,y,z,...)* with the convention that variable e represents the eventuality of that action or event to take place, whereas the other arguments (e.g. x, y, z, ...) represent the predicate arguments of the verb; (2) nouns are mapped into their lexicalized predicates; and, (3) modifiers have the same argument as the predicate they modify. For example, the question illustrated in Figure 2 has the following logical form transformation (`LFT`):

```
[REASON(x)&David(y)&Koresh(y)&ask(e,x,y,z,p)&
&FBI(z)&processor(p)&word(p)]
```

The process of translating a semantic representation into a logic form has the following steps:

1. For each node in the semantic representation, create a predicate with a distinct argument.
2.a. If a noun and an adjective predicate are linked they should have the same argument.
2.b. The same for verbs and adverbs, pairs of nouns or an adjective and an adverb.
3. For each verb predicate, add arguments corresponding to each predicate to which it is directly linked in the semantic representation.

Predicate arguments can be identified because the semantic representation using anonymous relations represents uniformly adjuncts and thematic roles. However, step 2 of the translation procedure recognizes the adjuncts, making predicate arguments the remaining connections of the verb in the semantic representation.

3.2 Question Taxonomy

The question taxonomy represents each question node as a quintuple: (1) a semantic representation of a question; (2) the question type; (3) the answer type; (4) the question focus and (5) the question keywords. By using over 1500 questions provided by Remedia, as well as other 2000 questions retrieved from FAQFinder, we have been able to learn classification rules and build a complex question taxonomy.

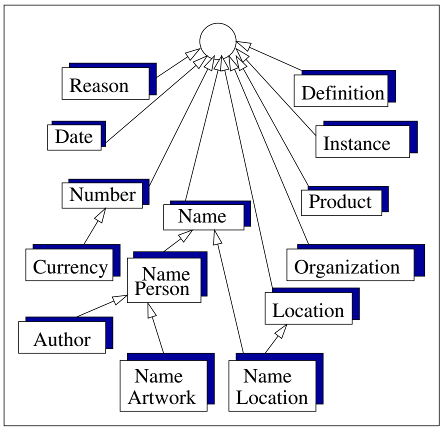

Figure 4: A snapshot of the top Question Taxonomy

Initially, we started with a seed hierarchy of 25 question classes, manually built, in which all the semantic classes of the nodes from the semantic representations were decided off-line, by a human expert. 300 questions were processed to create this seed hierarchy. Figure 4 illustrates some of the nodes of the top of this hierarchy. Later, as 500 more questions were considered, we started classifying them semi-automatically, using the following two steps: (1) first a human would decide the semantic class of each node in the semantic representation of the new question; (2) then a classification procedure would decide whether the question belongs to one of the existing classes or a new class should be considered. To be able to classify a new question in one of the existing question classes, two conditions must be satisfied: (a) all nodes from the taxonomy question must correspond to new question nodes with the same semantic classes; and (b) unifyable nodes must be linked in the same way in both representations. The hierarchy grew to 68 question nodes.

Later, 2700 more questions were classified fully automatically. To decide the semantic classes of the nodes, we used the WordNet semantic hierarchies, by simply assigning to each semantic representation node the same class as that of any other question term from its WordNet hierarchy.

The semantic representation, having the same format for questions and answers, is a case frame with anonymous relations, that allows the unification of the answer to the question regardless of the case relation. Figure 5 illustrates four nodes from the question taxonomy, two for the "currency" question type and two for the "person name" question type. The Figure also represents the mappings of four TREC-8 questions in these hierarchy nodes. The mappings are represented by dashed arcs. In this Figure, the nodes from the semantic representations that contain a question mark are place holders for the expected answer type.

An additional set of classification rules is associated with this taxonomy. Initially, all rules are based on the recognition of the *question stem* and of the *answer type*, obtained with class information from WordNet. However we could learn new rules when morphological and semantic variations of the semantic nodes are allowed. Moreover, along with the new rules, we enrich the taxonomy, because often the new questions unify only partially with the current taxonomy. All semantic and morphologic variations of the semantic representation nodes are grouped into word classes. Several of the word classes we used are listed in Table 1.

Word Class	Words
Value words	"monetary value", "money", "price"
Expenditure words	"spend", "buy", "rent", "invest"
Creation words	"author", "designer", "invent"...

Table 1: Examples of word classes.

The bootstrapping algorithm that learns new classification rules and new classes of questions is based on an information extraction measure: $score(rule_i){=}A_i * log_2(N_i)$, where A_i stands for the number of different lexicon entries for the *answer type of the question*, whereas $N_i = A_i/F_i$, where F_i is the number of different focus categories classified. The steps of the bootstrapping algorithm are:

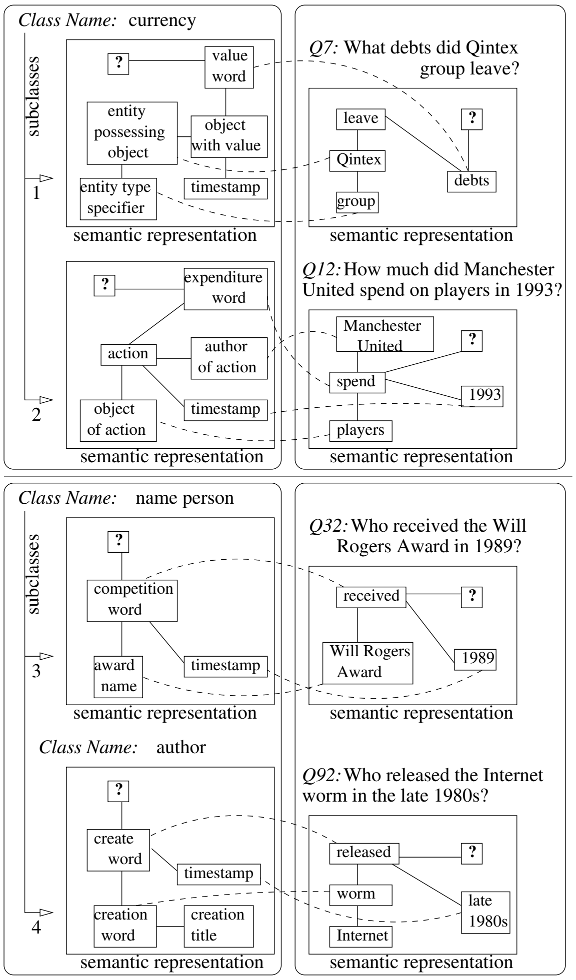

Figure 5: Mapping Questions in the Taxonomy

1. Retrieve concepts morphologically/semantically related to the semantic representations
2. Apply the classification rules to all questions that contain any newly retrieved concepts.
3. New_Classificatiton_Rules={}
MUTUAL BOOTSTRAPPING LOOP
4. Score all new classification rules
5. best_CR=the highest scoring classification rule
6. Add best_CR to the classification rules
7. Add the questions classified by best_CR to the taxonomy
8. Goto step 4 three times.
9. Discard all new rules but the best scoring three.
10. Goto 4. until the Q/A performance improves.

4 The Justification Option

A Q/A system that provides with the option of justifying the answer has the advantage that erroneous answers can be ruled out systematically. In our quest of enhancing the precision of a Q/A system by incorporating additional knowledge, we found this option very helpful. However, the generation of justifications for open-domain textual Q/A systems poses some challenges. First, we needed to develop a very efficient prover, operating on logical form transformations. Our proofs are backchaining from the questions through a mixture of axioms. We use three forms of axioms: (1) axioms derived from the facts stated in the textual answer; (2) axioms representing world knowledge; and (3) axioms determined by coreference resolution in the answer text. For example, some of the axioms employed to prove the answer to the TREC-8 question *Q6: Why did David Koresh ask the FBI for a word processor?"* are:

```
-----------------------------------------------
SET 1
Mr(71):= null. Koresh(71):=null. word(72):=null.
processor(72):=null.  sent(77 76 78 71):=null.
-----------------------------------------------
SET 2
David(1):=Mr(1).   _REASON(5):= enable(5 3 6).
-----------------------------------------------
SET 3
FBI(1):=null.
-----------------------------------------------
```

The first set represents facts extracted through LFT predicates of the textual answer: *"Over the weekend Mr Koresh sent a request for a word processor to enable him to record his revelations"*. The second set represents world knowledge axioms that we acquired semi-automatically. For instance we know that *David* is a male name, thus that person can be addressed with *Mr.*. Similarly, events are enabled for some reason. The third set of axioms represent the fact that the *FBI* is in the context of the text answer. To be noted that the axioms derived from the answer have constant arguments, represented by convention with numbers larger than 70. All the other arguments are variables.

Q52	Who invented the road traffic cone?
Answer (shallow methods)	*Smiling proudly for the cameras , Governor Pete Wilson, US Transportation Secretary Federico Pena and Mayor Richard Riordan removed a half - dozen plastic orange cones from the roadway and the first cars passed*
Answer (kb-based methods)	*David Morgan, the company's managing director and inventor of the plastic cone even collects them.*

Table 2: Examples of improved answer correctness.

The justification of this answer is provided by the following proof trace. The prover attempts to prove the LFT of the question (QLF) correct by proving from left to right each term of QLF.

```
-----------------------------------------------------------
 ->Answer:Over the weekend Mr Koresh sent a request for a word
processor to enable him to record his revelations.
->QLF:David(1)^Koresh(1)^word(2)^processor(2)^FBI(4)^
ask(3 4 2 1 5)^_REASON(5)^_PER(1)^_ORG(4)
->ALF:Mr(71)^Koresh(71)^word(72)^processor(72)^revelations(74)^
record(73 74 75)^enable(75 73 76)^request(76)^sent(77 76 78 71)
^weekend(78)^_PER(71)^_DATE(78)
...........................................................
-->Proving:David(1)^Koresh(1)^word(2)^processor(2)^FBI(4)^
ask(3 4 2 1 5)^_REASON(5)^_PER(1)^_ORG(4)
There are 1 target axioms. Selected axiom: David(1):= Mr(1).
Unifying: 1 to 1. Ok
--> Proving: Mr(1)^Koresh(1)^word(2)^processor(2)^FBI(4)
^ask(3 4 2 1 5)^_REASON(5)^_PER(1)^_ORG(4)
There are 1 target axioms. Selected axiom: Mr(71):= null.
Unifying: 1 to 71. Ok
--> Proving: Koresh(71)^word(2)^processor(2)^FBI(4)^
ask(3 4 2 71 5)^_REASON(5)^_PER(71)^_ORG(4)
There are 1 target axioms. Selected axiom: Koresh(71):= null.
Unifying: 71 to 71. Ok
--> Proving: word(2)^processor(2)^FBI(4)^ask(3 4 2 71 5)^
_REASON(5)^_PER(71)^_ORG(4)
There are 1 target axioms. Selected axiom: word(72):= null.
Unifying: 2 to 72. Ok
--> Proving: processor(72)^FBI(4)^ask(3 4 72 71 5)^_REASON(5)
^_PER(71)^_ORG(4)
There are 1 target axioms. Selected axiom: processor(72):= null.
Unifying: 72 to 72. Ok
--> Proving: FBI(4)^ask(3 4 72 71 5)^_REASON(5)^_PER(71)^_ORG(4)
There are 1 target axioms. Selected axiom: FBI(1):= null.
Unifying: 4 to 1. Ok
  --> Proving: ask(3 4 72 71 5)^_REASON(5)^_PER(71)^_ORG(4)
There are 2 target axioms. Selected axiom: ask(1 2 3 4 5):=
sent(1 6 7 4)^request(6).
Unifying: 1 to 2. 3 to 1. 5 to 5. 71 to 4. 72 to 3. Ok
  --> Proving: sent(1 6 7 71)^request(6)^_REASON(5)^_PER(71)^_ORG(2)
There are 1 target axioms. Selected axiom: sent(77 76 78 71):= null.
Unifying: 1 to 77. 6 to 76. 7 to 78. 71 to 71. Ok
--> Proving: request(76)^_REASON(5)^_PER(71)^_ORG(2)
There are 1 target axioms. Selected axiom: request(76):= null.
Unifying: 76 to 76. Ok
--> Proving: _REASON(5)^_PER(71)^_ORG(2)
There are 1 target axioms. Selected axiom: _REASON(5):= enable(5 3 6).
Unifying: 5 to 5. Ok
--> Proving: enable(5 3 6)^_PER(71)^_ORG(2)
There are 1 target axioms. Selected axiom: enable(75 73 76):= null.
Unifying: 3 to 73. 5 to 75. 6 to 76. Ok
--> Proving: _PER(71)^_ORG(2)
There are 3 target axioms. Selected axiom: _PER(71):= null.
Unifying: 71 to 71. Ok
--> Proving: _ORG(2)
There are 1 target axioms. Selected axiom: _ORG(1):= FBI(1).
Unifying: 2 to 1. Ok
--> Proving: null|||| We found:Success.

-----------------------------------------------------------------
```

There are cases when our simple prover fails to prove a correct answer. We have noticed that this happens because in the answer semantic representation, some concepts that are connected in the question semantic representation are no longer directly linked. This is due to the fact that there are either parser errors or there are new syntactic dependencies between the two concepts. To accommodate this situation, we allow different constants that are arguments of the same predicate to be unifiable. The special cases in which this relaxation of the unification procedure is allowed constitute our abduction rules.

5 Evaluation

Both qualitative and quantitative evaluation of the integration of surface text-based and knowledge-based methods for Q/A is imposed. Quantitatively, Table 3 summarizes the scores obtained when only shallow methods were employed, in contrast with the results when knowledge-based methods were integrated. We have separately measured the effect of the integration of the knowledge-based methods at question processing and answer processing level. We have also evaluated the precision of the system when both integrations were implemented. The results were the first five answers returned within 250 bytes of text, when approximatively half million TREC documents are mined. We have used the 200 questions from TREC-8, and the correct answers provided by NIST. The performance was measured both with the NIST scoring method employed in the TREC-8 and by simply assigning a score of 1 for the question having a correct answer, regardless of its position.

	Percentage of correct answers in top 5 returns	NIST score
Text-surface-based	77.7%	64.5%
Knowledge-based Question Processing (only)	83.2%	71.5%
Text-surface-based only with Answer Justification	77.7%	73%
Knowledge-based Question Processing with Answer Justification	89.5%	84.75%

Table 3: Accuracy performance

When using the NIST scoring method to evaluate an individual answer, we used only six values:$(1, .5, .33, .25, .2, 0)$, representing the score the answer's question obtains. If the first answer is correct, it obtains a score of 1, if the second one is correct, it is scored with .5, if the third one is correct, the score becomes .33, if the fourth is correct, the score is .25 and if the fifth one is correct, the score is .2. Otherwise, it is scored with 0. No credit is given if multiple answers are correct. Table 3 shows that both knowledge-based methods enhanced the precision, regardless of the scoring method.

To further evaluate the contribution of the justification option, we evaluated separately the precision of the prover for those questions for which the surface-text-based methods of our system, when operating alone, cannot find correct answers. We had 45 TREC-8 questions for which the evaluation of the prover was performed. Table 4 summarizes the accuracy of the prover.

	Proven correct	Proven incorrect	Precision
Incorrect answers (no knowledge)	3	210	98.5%
Correct answers (KB-based)	127	5	96.2%
Incorrect answers (KB-based)	4	38	90.04%

Table 4: Prover performance

Qualitatively, we find that the integration of knowledge-based methods is very beneficial. Table 2 illustrates the correct answer obtained with these methods, in contrast to the incorrect answer provided when only the shallow techniques are applied.

6 Conclusions

We believe that the performance of a Q/A system depends on the knowledge sources it employs. In this paper we have presented the effect of the integration of knowledge derived from question taxonomies and produced by answer justifications on the Q/A precision. Our knowledge-based methods are lightweight, since we do not generate precise semantic representations of questions or answers, but mere approximations determined by syntactic dependencies. Furthermore, our prover operates on very simple logical representations, in which syntactic and semantic ambiguities are completely ignored. Nevertheless, we have shown that these approximations are functional, since we implemented a prover that justifies answers with high precision. Similarly, our knowledge-based question processing is a mere combination of word class information and syntactic dependencies.

References

Michael Collins. A New Statistical Parser Based on Bigram Lexical Dependencies. In *Proceedings of the 34st Annual Meeting of the Association for Computational Linguistics, ACL-96*, pages 184–191, 1996.

Christiane Fellbaum (Ed). WordNet - An Electronic Lexical Database. MIT Press, 1998.

Jerry R. Hobbs. Discourse and Inference. Unpublished manuscript, 1986.

Jerry R. Hobbs. Overview of the TACITUS Project. In *Computational Linguistics*, 12:(3), 1986.

Jerry Hobbs, Mark Stickel, Doug Appelt, and Paul Martin. Interpretation as abduction. *Artificial Intelligence*, 63, pages 69–142, 1993.

Wendy Lehnert. The processing of question answering. Lawrence Erlbaum Publishers, 1978.

Dan Moldovan, Sanda Harabagiu, Marius Paşca, Rada Mihalcea, Richard Goodrum, Roxana Gîrju and Vasile Rus. Lasso: a tool for surfing the answer net. In *Proceedings of TREC-8*, 1999.

Ellen Riloff and Rosie Jones. Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping. In *Proceedings of the 16th National Conference on Artificial Intelligence, AAAI-99*, 1999.