The Role of Lexico-Semantic Feedback in Open-Domain Textual Question-Answering

Sanda Harabagiu, Dan Moldovan
Marius Paşca, Rada Mihalcea, Mihai Surdeanu,
Răzvan Bunescu, Roxana Gîrju, Vasile Rus and **Paul Morărescu**
Department of Computer Science and Engineering
Southern Methodist University
Dallas, TX 75275-0122
{sanda}@engr.smu.edu

Abstract

This paper presents an open-domain textual Question-Answering system that uses several feedback loops to enhance its performance. These feedback loops combine in a new way statistical results with syntactic, semantic or pragmatic information derived from texts and lexical databases. The paper presents the contribution of each feedback loop to the overall performance of 76% human-assessed precise answers.

1 Introduction

Open-domain textual Question-Answering (Q&A), as defined by the TREC competitions[1], is the task of identifying in large collections of documents a text snippet where the answer to a natural language question lies. The answer is constrained to be found either in a *short* (50 bytes) or a *long* (250 bytes) text span. Frequently, keywords extracted from the natural language question are either within the text span or in its immediate vicinity, forming a *text paragraph*. Since such paragraphs must be identified throughout voluminous collections, automatic and autonomous Q&A systems incorporate an index of the collection as well as a paragraph retrieval mechanism.

Recent results from the TREC evaluations ((Kwok et al., 2000) (Radev et al., 2000) (Allen et al., 2000)) show that Information Retrieval (IR) techniques alone are not sufficient for finding answers with high precision. In fact, more and more systems adopt architectures in which the semantics of the questions are captured prior to paragraph retrieval (e.g. (Gaizauskas and Humphreys, 2000) (Harabagiu et al., 2000)) and used later in extracting the answer (cf. (Abney et al., 2000)). When processing a natural language question two goals must be achieved. First we need to know what is the *expected answer type*; in other words, we need to know what we are looking for. Second, we need to know where to look for the answer, e.g. we must identify the question keywords to be used in the paragraph retrieval.

The expected answer type is determined based on the question stem, e.g. *who*, *where* or *how much* and eventually one of the question concepts, when the stem is ambiguous (for example *what*), as described in (Harabagiu et al., 2000) (Radev et al., 2000) (Srihari and Li, 2000). However finding question keywords that retrieve all candidate answers cannot be achieved only by deriving some of the words used in the question. Frequently, question reformulations use different words, but imply the same answer. Moreover, many equivalent answers are phrased differently. In this paper we argue that the answer to complex natural language questions cannot be extracted with significant precision from large collections of texts unless several lexico-semantic feedback loops are allowed.

In Section 2 we survey the related work whereas in Section 3 we describe the feedback loops that refine the search for correct answers.

[1]The Text REtrieval Conference (TREC) is a series of workshops organized by the National Institute of Standards and Technology (NIST), designed to advance the state-of-the-art in information retrieval (IR)

Section 4 presents the approach of devising keyword alternations whereas Section 5 details the recognition of question reformulations. Section 6 evaluates the results of the Q&A system and Section 7 summarizes the conclusions.

2 Related work

Mechanisms for open-domain textual Q&A were not discovered in the vacuum. The 90s witnessed a constant improvement of IR systems, determined by the availability of large collections of texts and the TREC evaluations. In parallel, Information Extraction (IE) techniques were developed under the TIPSTER Message Understanding Conference (MUC) competitions. Typically, IE systems identify information of interest in a text and map it to a predefined, target representation, known as *template*. Although simple combinations of IR and IE techniques are not practical solutions for open-domain textual Q&A because IE systems are based on domain-specific knowledge, their contribution to current open-domain Q&A methods is significant. For example, state-of-the-art Named Entity (NE) recognizers developed for IE systems were readily available to be incorporated in Q&A systems and helped recognize names of people, organizations, locations or dates.

Assuming that it is very likely that the answer is a named entity, (Srihari and Li, 2000) describes a NE-supported Q&A system that functions quite well when the expected answer type is one of the categories covered by the NE recognizer. Unfortunately this system is not fully autonomous, as it depends on IR results provided by external search engines. Answer extractions based on NE recognizers were also developed in the Q&A presented in (Abney et al., 2000) (Radev et al., 2000) (Gaizauskas and Humphreys, 2000). As noted in (Voorhees and Tice, 2000), Q&A systems that did not include NE recognizers performed poorly in the TREC evaluations, especially in the short answer category. Some Q&A systems, like (Moldovan et al., 2000) relied both on NE recognizers and some empirical indicators.

However, the answer does not always belong to a category covered by the NE recognizer. For such cases several approaches have been developed. The first one, presented in (Harabagiu et al., 2000), the answer type is derived from a large answer taxonomy. A different approach, based on statistical techniques was proposed in (Radev et al., 2000). (Cardie et al., 2000) presents a method of extracting answers as noun phrases in a novel way. Answer extraction based on grammatical information is also promoted by the system described in (Clarke et al., 2000).

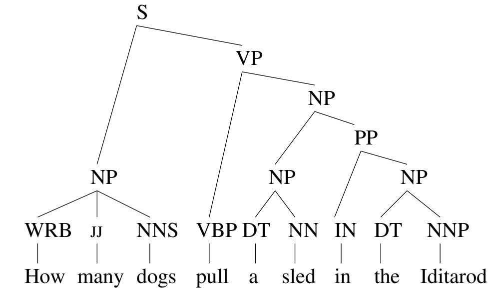
One of the few Q&A systems that takes into account morphological, lexical and semantic alternations of terms is described in (Ferret et al., 2000). To our knowledge, none of the current open-domain Q&A systems use any feedback loops to generate lexico-semantic alternations. This paper shows that such feedback loops enhance significantly the performance of open-domain textual Q&A systems.

3 Textual Q&A Feedback Loops

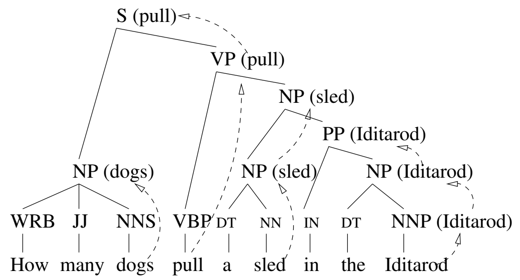
Before initiating the search for the answer to a natural language question we take into account the fact that it is very likely that the same question or a very similar one has been posed to the system before, and thus those results can be used again. To find such *cached questions*, we measure the similarity to the previously processed questions and when a reformulation is identified, the system returns the corresponding cached correct answer, as illustrated in Figure 1.

When no reformulations are detected, the search for answers is based on the conjecture that the eventual answer is likely to be found in a text paragraph that (a) contains the most representative question concepts and (b) includes a textual concept of the same category as the expected answer. Since the current retrieval technology does not model semantic knowledge, we break down this search into a boolean retrieval, based on some question keywords and a filtering mechanism, that retains only those passages containing the expected answer type. Both the *question keywords* and the *expected answer type* are identified by using the dependencies derived from the question parse.

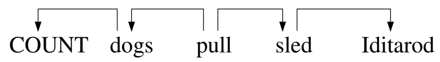
By implementing our own version of the publicly available Collins parser (Collins, 1996), we also learned a *dependency* model that enables the mapping of parse trees into sets of binary relations between the head-word of each constituent and its sibling-words. For example, the parse tree of TREC-9 question *Q210: "How many dogs pull a sled in the Iditarod ?"* is:



For each possible constituent in a parse tree, rules first described in (Magerman, 1995) and (Jelinek et al., 1994) identify the head-child and propagate the head-word to its parent. For the parse of question *Q210* the propagation is:



When the propagation is over, head-modifier relations are extracted, generating the following dependency structure, called *question semantic form* in (Harabagiu et al., 2000).



In the structure above, COUNT represents the *expected answer type*, replacing the question stem *"how many"*. Few question stems are unambiguous (e.g. *who*, *when*). If the question stem is ambiguous, the expected answer type is determined by the concept from the question semantic form that modifies the stem. This concept is searched in an ANSWER TAXONOMY comprising several tops linked to a significant number of WordNet noun and verb hierarchies. Each top represents one of the possible expected answer types implemented in our system (e.g. PERSON, PRODUCT, NUMERICAL VALUE, COUNT, LOCATION). We encoded a total of 38 possible answer types.

In addition, the question keywords used for paragraph retrieval are also derived from the question semantic form. The question keywords are organized in an ordered list which first enumerates the named entities and the question quotations, then the concepts that triggered the recognition of the expected answer type followed by all adjuncts, in a left-to-right order, and finally the question head. The conjunction of the keywords represents the boolean query applied to the document index. (Moldovan et al., 2000) details the empirical methods used in our system for transforming a natural language question into an IR query.

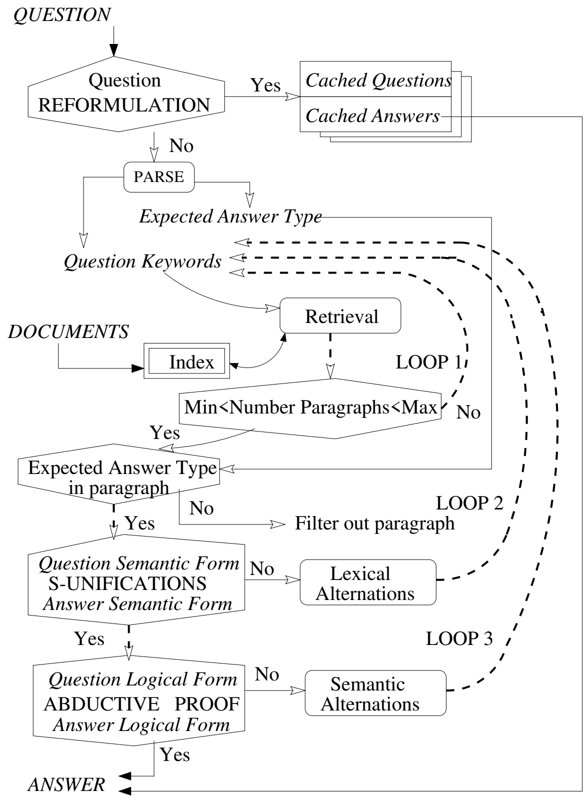


Figure 1: Feedbacks for the Answer Search.

It is well known that one of the disadvantages of boolean retrieval is that it returns either too many or too few documents. However, for question answering, this is an advantage, exploited by the first feedback loop represented in Figure 1. **Feedback loop 1** is triggered when the number of retrieved paragraphs is either smaller than a minimal value or larger than a maximal value determined beforehand for each answer type. Alternatively, when the number of paragraphs is within limits, those paragraphs that do not contain at least one concept of the same semantic category as the expected answer type are filtered out. The remaining paragraphs are parsed and their dependency structures, called *answer semantic forms*, are derived.

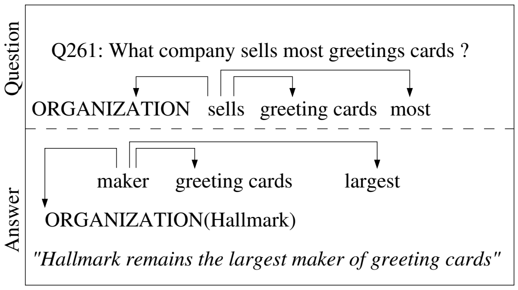
Feedback loop 2 illustrated in Figure 1 is activated when the question semantic form and the answer semantic form cannot by unified. The unification involves three steps:

∘ *Step 1: The recognition of the expected answer type*. The first step marks all possible concepts that are answer candidates. For example, in the case of TREC -9 question *Q243: "Where did the ukulele originate ?"*, the expected answer type is LOCATION. In the paragraph "*the ukulele introduced from Portugal into the Hawaiian islands*" contains two named entities of the category LOCATION and both are marked accordingly.

∘ *Step 2: The identification of the question concepts*. The second step identifies the question words, their synonyms, morphological derivations or WordNet hypernyms in the answer semantic form.

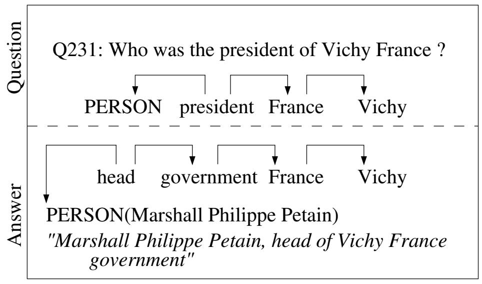
∘ *Step 3: The assessment of the similarities of dependencies*. In the third step, two classes of similar dependencies are considered, generating unifications of the question and answer semantic forms:

▷ *Class L2-1:* there is a one-to-one mapping between the binary dependencies of the question and binary dependencies from the answer semantic form. Moreover, these dependencies largely cover the question semantic form[2]. An example is:



We find an entailment between producing, or making and selling goods, derived from WordNet, since synset {*make, produce, create*} has the genus *manufacture*, defined in the gloss of its homomorphic nominalization as *"for sale"*. Therefore the semantic form of question *Q261* and its illustrated answer are similar.

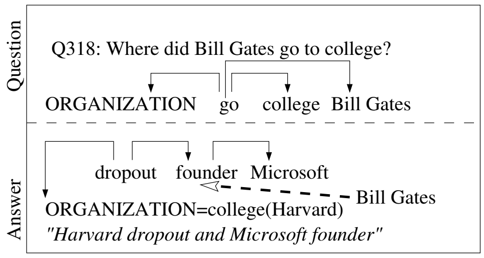
▷ *Class L2-2:* Either the question semantic form or the answer semantic form contain new concepts, that impose a bridging inference. The knowledge used for inference is of lexical nature and is later employed for abductions that justify the correctness of the answer. For example:



Nouns *head* and *government* are constituents of a possible paraphrase of *president*, i.e. *"head of government"*. However, only world knowledge can justify the answer, since there are countries where the prime minister is the head of government. Presupposing this inference, the semantic form of the question and answer are similar.

Feedback loop 3 from Figure 1 brings forward additional semantic information. Two classes of similar dependencies are considered for the abduction of answers, performed in a manner similar to the justifications described in (Harabagiu et al., 2000).

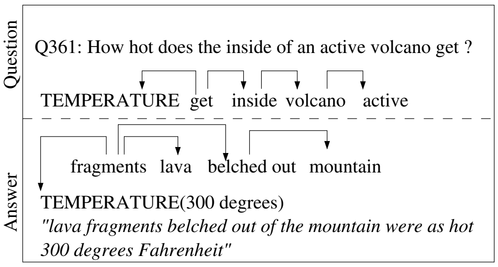
▷ *Class L3-1:* is characterized by the need for contextual information, brought forward by reference resolution. In the following example, a chain of coreference links *Bill Gates* and *Microsoft founder* in the candidate answer:



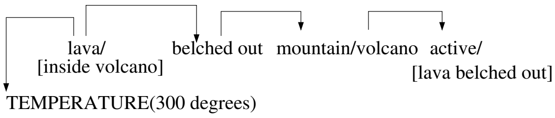
▷ *Class L3-2:* Paraphrases and additional information produce significant differences between the question semantic form and the answer semantic form. However, semantic information contributes to the normalization of the answer dependencies until they can be unified with the question dependencies. For example, if (a) a *volcano* IS-A *mountain*; (b) *lava* IS-PART of *volcano*, and moreover it is a part coming from the *inside*; and (c) *fragments* of *lava* have all the properties of *lava*, the following question semantic

[2]Some modifiers might be missing from the answer.

form and answer semantic form can be unified:



The resulting normalized dependencies are:



The semantic information and the world knowledge needed for the above unifications are available from WordNet (Miller, 1995). Moreover, this knowledge can be translated in axiomatic form and used for abductive proofs. Each of the feedback loops provide the retrieval engine with new alternations of the question keywords. Feedback loop 2 considers morphological and lexical alternations whereas Feedback loop 3 uses semantic alternations. The method of generating the alternations is detailed in Section 4.

4 Keyword Alternations

To enhance the chance of finding the answer to a question, each feedback loop provides with a different set of keyword alternations. Such alternations can be classified according to the linguistic knowledge they are based upon:

1.Morphological Alternations. When lexical alternations are necessary because no answer was found yet, the first keyword that is altered is determined by the question word that either prompted the expected answer type or is in the same semantic class with the expected answer type. For example, in the case of question *Q209: "Who invented the paper clip ?"*, the expected answer type is PERSON and so is the subject of the verb *invented* , lexicalized as the nominalization *inventor*. Moreover, since our retrieval mechanism does not stem keywords, all the inflections of the verb are also considered. Therefore, the initial query is expanded into:

QUERY(Q209):[*paper AND clip AND (invented OR inventor OR invent OR invents)*]

2. Lexical Alternations. WordNet encodes a wealth of semantic information that is easily mined. Seven types of semantic relations span concepts, enabling the retrieval of synonyms and other semantically related terms. Such alternations improve the recall of the answer paragraphs. For example, in the case of question *Q221: "Who killed Martin Luther King ?"*, by considering the synonym of *killer*, the noun *assassin*, the Q&A system retrieved paragraphs with the correct answer. Similarly, for the question *Q206: "How far is the moon ?"*, since the adverb *far* is encoded in WordNet as being an attribute of *distance*, by adding this noun to the retrieval keywords, a correct answer is found.

3. Semantic Alternations and Paraphrases. We define as semantic alternations of a keyword those words or collocations from WordNet that (a) are not members of any WordNet synsets containing the original keyword; and (b) have a chain of WordNet relations or bigram relations that connect it to the original keyword. These relations can be translated in axiomatic form and thus participate to the abductive backchaining from the answer to the question - to justify the answer. For example semantic alternations involving only WordNet relations were used in the case of question *Q258: "Where do lobsters like to live ?"*. Since in WordNet the verb *prefer* has verb *like* as a hypernym, and moreover, its glossed definition is *liking better*, the query becomes:

QUERY(Q258):[*lobsters AND (like OR prefer)* AND live]

Sometimes multiple keywords are replaced by a semantic alternation. Sometimes these alternations are similar to the relations between multi-term paraphrases and single terms, other time they simply are semantically related terms. In the case of question *Q210: "How many dogs pull a sled in the Iditarod ?"*, since the definition of WordNet sense 2 of noun *harness* contains the bigram *"pull cart"* and both *sled* and *cart* are forms of *vehicles*, the alternation of the pair of keywords [*pull*, *slide*] is rendered by *harness*. Only when this feedback is received, the paragraph containing the correct answer is retrieved.

To decide which keywords should be expanded and what form of alternations should be used we rely on a set of heuristics which complement the heuristics that select the question keywords and generate the queries (as described in (Moldovan et al., 2000)):

Heuristic 1: Whenever the first feedback loop requires the addition of the main verb of the question as a query keyword, generate all verb conjugations as well as its nominalizations.

Heuristic 2: Whenever the second feedback loop requires lexical alternations, collect from WordNet all the synset elements of the direct hypernyms and direct hyponyms of verbs and nominalizations that are used in the query. If multiple verbs are used, expand them in a left-to-right order.

Heuristic 3: Whenever the third feedback loop imposes semantic alternations expressed as paraphrases, if a verb and its direct object from the question are selected as query keywords, search for other verb-object pairs semantically related to the query pair. When new pairs are located in the glosses of a synset S, expand the query verb-object pair with all the elements from S.

Another set of possible alternations, defined by the existence of lexical relations between pairs of words from different question are used to detect question reformulations. The advantage of these different forms of alternations is that they enable the resolution of similar questions through *answer caching* instead of normal Q&A processing.

5 Question Reformulations

In TREC-9 243 questions were reformulations of 54 inquiries, thus asking for the same answer. The reformulation classes contained variable number of questions, ranging from two to eight questions. Two examples of reformulation classes are listed in Table 1. To classify questions in reformulation groups, we used the algorithm:

Reformulation_Classes(new_question, old_questions)
1. For each question from old_questions
2. Compute similarity(question,new_question)
3. Build a new similarity matrix $\mathcal{M}$ such that it is generated by adding to the matrix for the old_questions a new row and a new column representing the similarities computed at step 2.
4. Find the transitive closures for the set $\{$old_questions$\} \cup \{$new_question$\}$
5. Result: reformulation classes as transitive closures.

In Figure 2 we represent the similarity matrix for six questions that were successively posed to the answer engine. Since question reformulations are transitive relations, if at a step n questions Q_i and Q_j are found similar and Q_i already belongs to $\mathcal{R}$, a reformulation class previously discovered (i.e. a group of at least two similar questions), then question Q_j is also included in $\mathcal{R}$. Figure 2 illustrates the transitive closures for reformulations at each of the five steps from the succession of six questions. To be noted that at step 4 no new similarities were found , thus Q_5 is not found similar to Q_4 at this step. However, at step 5, since Q_6 is found similar to both Q_4 and Q_5, Q_4 results similar to all the other questions but Q_3.

Q397:	*When was the Brandenburg Gate in Berlin built?*
Q814:	*When was Berlin's Brandenburg gate erected?*
Q-411:	*What tourist attractions are there in Reims?*
Q-711:	*What are the names of the tourist attractions in Reims?*
Q-712:	*What do most tourists visit in Reims?*
Q-713:	*What attracts tourists to Reims?*
Q-714:	*What are tourist attractions in Reims?*
Q-715:	*What could I see in Reims?*
Q-716:	*What is worth seeing in Reims?*
Q-717:	*What can one see in Reims?*

Table 1: Two classes of TREC-9 question reformulations.

	Q1	Q2	Q3	Q4	Q5	Q6	
Q1	0	1	0	1	0	0	
Q2	1	0	0	0	0	0	Step 1: {Q1, Q2}
Q3	0	0	0	0	0	0	Step 2: {Q1, Q2} {Q3}
Q4	1	0	0	0	0	1	Step 3: {Q1, Q2, Q4} {Q3}
Q5	0	0	0	0	0	1	Step 4: {Q1, Q2, Q4} {Q3} {Q5}
Q6	0	0	0	1	1	0	Step 5: {Q1, Q2, Q4, Q5, Q6} {Q3}

Figure 2: Building reformulation classes with a similarity matrix.

The algorithm that measures the similarity between two questions is:

Algorithm Similarity(Q, Q')
Input: a pair of question represented as two word strings:
Q: $w_1\ w_2 \ldots w_n$ and *Q': $w'_1\ w'_2 \ldots w'_n \ldots w_m$*

1. Apply a part-of-speech tagger on both questions:
Tag(Q): $w_1/tag_1\ w_2/tag_2 \ldots w_n/tag_n$
Tag(Q'): $w'_1/tag'_1\ w'_2/tag'_2 \ldots w_m/tag'_m$
2. Set nr_matches=0
3. Identify quadruples $(w_i, tag_i, w'_j, tag'_j)$ such that *if w_i and w'_j are content words with $tag_i \equiv tag'_j$ and Lexical_relation(w_i, w'_j) holds then increase nr_matches*

4. Relax the Lexical_relation and goto step 3;
5. If (nr_matches/number of content words $\geq t$)
then Q and Q' are similar

The *Lexical_relation* between a pair of content words is initially considered to be a string identity. In later loops starting at step 3 one of the following three possible relaxations of *Lexical_relation* are allowed: (a) common morphological root (e.g. *owner* and *owns*, from question *Q742: "Who is the owner of CNN ?"* and question *Q417: "Who owns CNN ?"* respectively); (b) WordNet synonyms (e.g. *gestation* and *pregnancy* from question *Q763: "How long is human gestation ?"* and question *Q765: "A normal human pregnancy lasts how many months ?"*, respectively) or (c) WordNet hypernyms (e.g. the verbs *erect* and *build* from question *Q814: "When was Berlin's Brandenburg gate erected ?"* and question *Q397: "When was the Brandenburg Gate in Berlin built ?"* respectively).

6 Performance evaluation

To evaluate the role of lexico-semantic feedback loops in an open-domain textual Q&A system we have relied on the 890 questions employed in the TREC-8 and TREC-9 Q&A evaluations. In TREC, for each question the performance was computed by the reciprocal value of the rank (RAR) of the highest-ranked correct answer given by the system. Given that only the first five answers were considered in the TREC evaluations, i f the RAR is defined as $RAR = \frac{1}{rank_i}$ its value is 1 if the first answer is correct; 0.5 if the second answer was correct, but not the first one; 0.33 when the correct answer was on the third position; 0.25 if the fourth answer was correct; 0.2 when the fifth answer was correct and 0 if none of the first five answers were correct. The Mean Reciprocal Answer Rank (MRAR) is used to compute the overall performance of the systems participating in the TREC evaluation $MRAR = \frac{1}{n}(\sum_i^n \frac{1}{rank_i})$ In addition, TREC-9 imposed the constraint that an answer is considered correct only when the textual context from the document that contains it can account for it. When the human assessors were convinced this constraint was satisfied, they considered the RAR to be *strict*, otherwise, the RAR was considered *lenient*.

Table 2 summarizes the MRARs provided by NIST for the system on which we evaluated the role of lexico-semantic feedbacks. Table 3 lists the quantitative analysis of the feedback loops. Loop 1 was generated more often than any other loop. However, the small overall average number of feedback loops that have been carried out indicate that the fact they port little overhead to the Q&A system.

	MRAR *lenient*	MRAR *strict*
Short answer	0.599	0.580
Long answer	0.778	0.760

Table 2: NIST-evaluated performance

	Average number	Maximal number
Loop 1	1.384	7
Loop 2	1.15	3
Loop 3	1.07	5

Table 3: Number of feedbacks on the TREC test data

More interesting is the qualitative analysis of the effect of the feedback loops on the Q&A evaluation. Overall, the precision increases substantially when all loops were enabled, as illustrated in Table 4.

L1	L2	L3	MRAR short	MRAR long
No	No	No	0.321	0.385
Yes	No	No	0.451	0.553
No	Yes	No	0.490	0.592
Yes	Yes	No	0.554	0.676
No	No	Yes	0.347	0.419
Yes	No	Yes	0.488	0.589
No	Yes	Yes	0.510	0.629
Yes	Yes	Yes	0.568	0.737

Table 4: Effect of feedbacks on accuracy. L1=Loop 1; L2=Loop 2; L3=Loop 3.

Individually, the effect of Loop 1 has an accuracy increase of over 40%, the effect of Loop 2 had an enhancement of more than 52% while Loop 3 produced an enhancement of only 8%. Table 4 lists also the combined effect of the feedbacks, showing that when all feedbacks are enabled, for short answers we obtained an MRAR of 0.568, i.e. 76% increase over Q&A without feedbacks. The MRAR for long answers had a similar increase of 91%. Because we also used the answer caching technique, we gained more than 1% for short answers and almost 3% for long answers, obtaining the result listed in Table 2. In our experiments, from the total of 890 TREC questions, lexical alternations were used for 129 questions and the semantic alternations were needed only for 175 questions.

7 Conclusion

This paper has presented a Q&/A system that employs several feedback mechanisms that provide lexical and semantic alternations to the question keywords. By relying on large, open-domain linguistic resources such as WordNet we enabled a more precise approach of searching and mining answers from large collections of texts. Evaluations indicate that when all three feedback loops are enabled we reached an enhancement of almost 76% for short answers and 91% for long answers, respectively, over the case when there are no feedback loops. In addition, a small increase is produced by relying on cached answers of similar questions. Our results so far indicate that the usage of feedback loops that produce alternations is significantly more efficient than multi-word indexing or annotations of large corpora with predicate-argument information.

References

Steve Abney, Michael Collins, and Amit Singhal. Answer extraction. In *Proceedings of the 6th Applied Natural Language Processing Conference (ANLP-2000)*, pages 296–301, Seattle, Washington, 2000.

James Allen, Margaret Connell, W. Bruce Croft, Fan-Fang Feng, David Fisher and Xioayan Li. INQUERY in TREC-9. *Proceedings of the Text Retrieval Conference (TREC-9)*, pages 504–510, 2000.

Claire Cardie, Vincent Ng, David Pierce, Chris Buckley. Examining the role of statistical and linguistic knowledge sources in a general-knowledge que stion answering system. In *Proceedings of the 6th Applied Natural Language Processing Conference (ANLP-2000)*, pages 180–187, Seattle, Washington, 2000.

C.L. Clarke, Gordon V. Cormak, D.I.E. Kisman and T.R. Lynam. Question Answering by passage selection. *Proceedings of the Text Retrieval Conference (TREC-9)*, pages 65–76, 2000.

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

Olivier Ferret, Brigitte Grau, Martine Hurault-Plantet, Gabriel Illouz, Christian Jacquemin, Nicolas Masson and Paule Lecuyer. QALC- the question-answering system of LIMSI-CNRS. *Proceedings of the Text Retrieval Conference (TREC-9)*, pages 316–326, 2000.

Robert Gaizauskas and Kevin Humphreys. A combined IR/NLP approach to question answering against large text collections. In *Proceedings of the 6th Content-Based Multimedia Information Access Conference (RIAO-2000)*, pages 1288–1304, Paris, France, 2000.

Sanda Harabagiu, Marius Paşca and Steven Maiorano. Experiments with Open-Domain Textual Question Answering. In the *Proceedings of the 18th International Conference on Computational Linguistics (COLING-2000)*, pages 292–298, 2000.

Frederick Jelinek, John Lafferty, Dan Magerman, Robert Mercer, Adwait Ratnaparkhi and Selim Roukos. Decision tree parsing using a hidden derivational model. In *Proceedings of the 1994 Human Language Technology Workshop*, pages 272–277, 1994.

K.L. Kwok, L. Grunfeld, N. Dinstl and M. Chan. TREC-9 Cross Language, Web and Question-Answering Track Experiments using PIRCS. *Proceedings of the Text Retrieval Conference (TREC-9)*, pages 26–35, 2000.

Dan Magerman. Statistical decision-tree models of parsing. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics, ACL-95*, pages 276–283, 1995.

George A. Miller. WordNet: A Lexical Database. *Communication of the ACM*, vol 38: No11, pages 39–41, November 1995.

Dan Moldovan, Sanda Harabagiu, Marius Paşca, Rada Mihalcea, Richard Goodrum, Roxana Gîrju and Vasile Rus. The Structure and Performance of an Open-Domain Question Answering System. *Proceedings of the 38th Annual Meeting of the Association for Comoutational Linguistics (ACL-2000)*, pages 563–570, 2000.

Dragomir Radev, John Prager, and V. Samn. Ranking suspected answers to natural language questions using predictive annotation. In *Proceedings of the 6th Applied Natural Language Processing Conference (ANLP-2000)*, pages 150–157, Seattle, Washington, 2000.

Rohini Srihari and W. Li. A question answering system supported by information extraction. In *Proceedings of the 6th Applied Natural Language Processing Conference (ANLP-2000)*, Seattle, Washington, 2000.

Ellen M. Voorhees and Dawn Tice. Building a question answering test collection. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-2000)*, Athens, Greece, 2000.