Open-Domain Voice-Activated Question Answering

Sanda Harabagiu and **Dan Moldovan**
Department of Computer Science
University of Texas at Dallas
sanda@cs.utdallas.edu
moldovan@utdallas.edu

Joe Picone
Institute of Signal
and Information Processing
Mississippi State University
picone@isip.msstate.edu

Abstract

Voice-Activated Question Answering (VAQA) systems represent the next generation capability for universal access by integrating state-of-the-art in question answering Q&A and automatic speech recognition (ASR) in such a way that the performance of the combined system is better than the individual components. This paper presents an implemented VAQA system and describes the techniques that enable the terative refinement of both Q&A and ASR. The results of our experiments show that spoken questions can be processed with surprising accuracy when using our VAQA implementation.

1 Introduction

Open-domain question answering (ODQA) is a critical technology for the next generation of Internet applications. Text-based Q&A technology has been making vast inroads into the public consciousness through web sites such as www.askjeeves.com. It is clear the next step is to integrate voice input (and output) to alleviate the keyboard bottleneck. Because the amount of information on the Internet is growing exponentially, standard word statistic-based search engines are rapidly becoming obsolete due to the large number of irrelevant matches returned. Further, the explosion of web-based portable computing devices with limited display capabilities (e.g., cellular phones) has created a serious need for advanced information access technologies that interact with the Internet using voice and other modalities.

Voice-Activated Question Answering (VAQA) systems represent the next generation capability for universal access by integrating state-of-the-art in question answering and automatic speech recognition (ASR) in such a way that the performance of the combined system is better than the individual components. An overview of our approach is given in Figure 1.

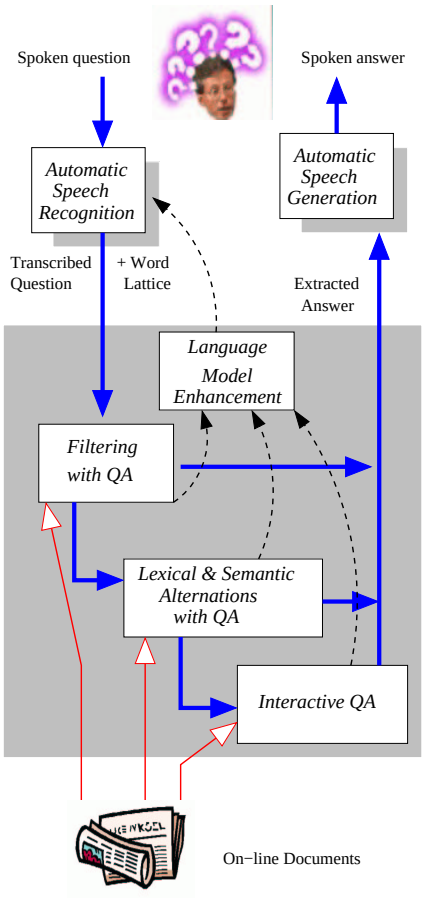


Figure 1: Global view of a Voice Activated Question Answering System

The system attempts to answer correctly a spoken question by first filtering the many possible ill-formed questions from the word lattice, and if this fails it performs lexical and semantic alternations to the remaining questions in the

reduced word lattice. If the answer is not found by using alternations of the keywords, then finally, the question will be answered through an interactive Q&A module. By allowing the Q&A and ASR systems to interact and pass information back and forth, and by allowing each system to reprocess the data based on iterative feedback from the other, we can converge on a better hypothesis that is something neither system could have achieved in isolation. We refer to this process as *iterative refinement*, and it is a technical cornerstone of our method for VAQA.

The remaining of the paper is organized as follows. Section 2 describes the motivation of the iterative refinement of voice and question processing and details our methodology. Section 3 presents the filtering mechanisms implemented in VAQA whereas Section 4 describes the enhanced language model for question processing. The lexico-semantic alternations enabled for Q&A are not described in this paper, as they are similar to those presented in (Harabagiu et al. 2001). Section 5 describes the interactive QA part of our VAQA system and Section 6 evaluates the results. Section 7 summarizes the conclusions.

2 Iterative Refinement of Voice and Question Processing

When integrating a state-of-the-art question answering system with an automatic speech recognition (ASR), the experiments show the need to devise a combined system that is better than the individual components. Our experiments involved the best performing Q&A system in the recent Text Retrieval Conferences (TREC) (*http://trec.nist.gov*) and an Automatic Speech Recognition (ASR) system publicly available from the Institute of Signal and Information Processing (ISSP) (*http://www.isip.msstate.edu*). We collected a small database of three speakers who read 200 questions from the TREC-8 competition and the 700 questions evaluated in TREC-9. These questions were processed through the ISSP's state-of-the-art LVCSR system trained on a diverse database of broadcast news. The 1-best output from this system was then sent to a high performance Q&A system. The baseline performance of the Q&A system from text input was 76%. Performance of the Q&A system on the output of the speech recognizer, which operated at a 30% WER, was only 7%. Examination of the results exposes several fundamental flaws of this simple combination of an ASR and QA system, including the importance of named entity information, and the inadequacies of current speech recognition technology based on N-gram language models.

Our solution to the problem of Open-Domain Voice-Activated Question-Answering is based on several interactions between the ASR and the QA systems. Our VAQA System uses first the ASR to generate a transcribed question along with a lattice of words that are recognized with various probabilities. A special filtering mechanism then uses both the question transcription and the word lattice to filter out words that cannot be processed by a typical Q&A system due to syntactic, semantic or pragmatic inconsistencies. The result is a word lattice of smaller dimensions, useful for generating an enhanced language model, employed by the ASR. This language model is used to reprocess the spoken question before presenting it to a high-performance Q&A system capable of using lexical and semantic alternations of the question keywords when searching for the answer. (Harabagiu et al. 2001) reported a successful method of enhancing the preformance of open-domain textual Q&A systems by enabling three diffrent forms of keyword variants, that we called alternations. In our experiments, we made use of all these three forms of alternations: (1) morphological variations (e.g. the keyword *invent* is expanded into *invention* and *inventor*); (2) semantic alternations based on synonyms or hypernyms encoded in WordNet (Miller 1995) (e.g. *murderer* and *killer*); and (3) lexical paraphrases using one or multiple terms to express the same concept (e.g. *linke better* and *prefer*). However, there are cases when none of the syntactic, semantic or pragmatic information can improve the interpretation of the question because either all the words are incorrectly recognized or the question was very short, asking about a single concept that is misunderstood. Allowing a follow-up and engaging in a dialog with the user enables the system to negotiate the meaning of the question and therefore provide with the expected answer. In this latter case, the transcription of the original question

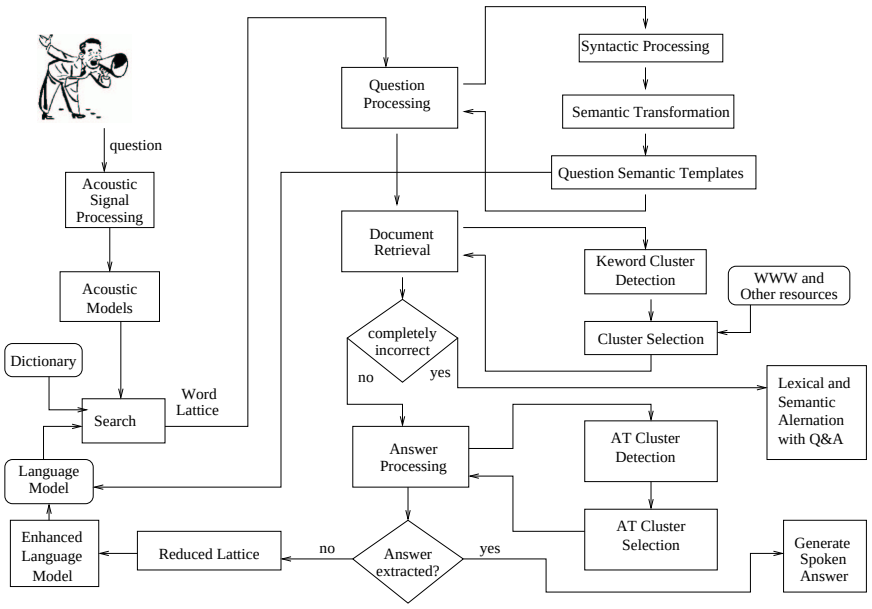


Figure 2: Architecture of the Filtering component of a Voice-Activated Question/Answering System

can be recovered and the language model may be further enhanced to capture the missing linguistic information.

3 Filtering for VAQA

The architecture of the filtering component of the *Voice-Activated Question-Answering* (VAQA) system is illustrated in Figure 2. The transcribed question, generated by the ASR module has usually a multitude of errors, either determined by the presence of words that were not in the vocabulary of the ASR or due to the simple language model it encodes. The spoken question signal is processed before it is used by the acoustic models that create the search space for the question words. Initially the language model from the ASR is used to produce the question transcription as well as a lattice of words recognized with different probabilities. Overall, the role of the filters is to significantly reduce the large number of outputs produced by the word lattice search module. For example, for the TREC-8 question *"Who was President Cleveland's wife?"* out of 105 outputs in the word lattice, only 18 passed all the filters described below.

The filter illustrated in Figure 2 improves the speech-recognition probabilistic model by using information acquired from several sources. First, dictionaries from the named entity recognizer included in the Q&A system enhance the limited dictionaries currently used in ASR. Many paths that are currently unexplored because of the lack of dictionaries become available when larger dictionaries are considered. Second, *question templates* are useful for reconsidering other possible alternatives rather than those selected. For example, most TREC questions start with a question stem like *What*, *Who* or *When*. While this does not always apply, it does indicate that the input from the user is more likely to start with a question stem. The enhanced language model incorporates this form of knowledge. However, the new language model cannot be trained entirely on the lattice, and the filtering component will retain only the sequences of words from the lattice that comply with most of the syntactic, semantic and pragmatic requirements of general question templates. For example, when considering the question *"Who is President Cleveland's wife?"* many incorrect alternatives like *"As President Cleveland wife"* or *"For as president Cleveland wife"* are assigned lower probabilities than those alternatives starting with a question stem, e.g. *"Who was President Cleveland wife?"*. To further reduce the word lattice on which the enhance language model is trained, three additional forms of filtering take place: (1) question processing filtering; (2) passage retrieval filter-

ing; and (3) answer processing filtering. The question filtering uses three forms of information: syntactic, semantic (answer-type based) and pragmatic (pattern based). The *syntactic filter* is based on the information provided by the probabilistic parser encoded employed by the Q&A system. Spurious alternatives like *"The was President Cleveland wife"* are easily rejected because the probability of a global parse is close to zero. The absence of a verb is also detected by the syntactic filter. For example the questions *"Whose President Cleveland life?"*, *"When President Cleveland life?"* and *"What President Cleveland Zweig?"* will are discarded at this level. The *semantic filter* identifies questions whose question stem is not recognized successfully or does not match the answer type. For example, the alternative *"It was President Cleveland lawyer"* for the question *"Who is President Cleveland's wife?"* does not have a question stem, so it will be discarded. When the question stem is recognized but it does not fit into the semantic class of the expected answers, the alternative is also rejected. For instance, *"Who's President Cleveland life?"* is rejected because there is a mismatch between the question stem *"Who"* (expecting a person or an organization's name as answer) and the question term *"life"*. The *pragmatic filter* further checks the semantics of the question, restricting the set of alternative questions to those that make sense semantically against a set of question patterns. The question *"How far is Yaroslavl from Moscow?"* constitutes an example. Even if the city names are not recognized, question patterns can identify that the first concept after the question stem should be a location, as long as the second concept (Moscow) is identified as a location name.

4 Enhanced Language Model for Question Processing

Typically, a language model (LM) provides constraints on the sequence of words that are allowed to be recognized. In particular, it provides a mechanism to estimate the probability of some word w_k in q word sequence W given the surrounding words. Ideally, the LM integrates linguistic knowledge, domain knowledge and pragmatic knowledge. For most ASR applications, none of these forms of knowledge are easy to identify, therefore N-gram models are used indirectly to encode syntax, semantics and pragmatics by concentrating on the local dependencies between words. However, our experiments have shown that N-grams are insufficient for the recognition of spoken question words.

Possible alternatives (e.g. (Chelba and Jelinek 1998), (Kuhn and de Mori 1992), (Lafferty et al. 1992) (Lau et al. 1992), (Jardino 1996) and (Bahl et al. 1989)) emphasize syntactic dependencies whereas experiments in open-domain textual Q&A have shown that semantic and pragmatic dependencies are more important. For example, given the question: *"How far is Yaroslavl from Moscow?"* and its transcription recognized by our ASR: *"AFFAIR IS YES LEVEL FROM MOSCOW"* it is obvious that the correction of *"AFFAIR"* into *"How far"* may be obtained easier if a DISTANCE semantic class is associated with the bigram [*from* LOCATION] where *Moscow* is identified as a LOCATION by a Named Entity tagger. The DISTANCE semantic class imposes the identification of the associated question stem *"How far"*. This is more straight-forward than trying to compute long-distance dependency probabilities between *AFFAIR* and *MOSCOW*. In this way, semantic information characteristic for question processing takes precedence over syntactic information. Additionally, Yaroslavl cannot be recognized simply because it is not in the vocabulary. However, a back-off model that uses lists of LOCATION-words from the text collection can be used to approximate its recognition. The list of all possible names of locations collocating with Moscow in the same paragraphs is a new form of pragmatic knowledge, readily available when retrieval systems built for Q&A are used.

Our enhanced language model is based on the *noisy channel* framework in which we consider that the transcribed question produced by the ASR contains noise on top of the information from the originally spoken question. As in any noisy channel application, we must solve three problems:

• *Source model:* We must assign to every spoken question Q_s a probability $P(Q_s)$ which quantifies the chance that it is recognized correctly.

• *Channel model:* We assign to every pair of spoken and transcribed question a probability $P(Q_t|Q_s)$ which gives the chance that when Q_s

is uttered, it will be recognized as Q_t.

• *Decoder:* When we obtain a transcription of a question, we look for the original, spoken question Q_s that maximizes $P(Q_s|Q_t)$. This is equivalent to searching for Q_s that maximizes $P(Q_s) \times P(Q_t|Q_s)$.

Our source and channel models assign probabilities to the semantic transformations of the questions rather than to the string of words. Semantic transformations of questions were first introduced in (Harabagiu et al. 2000) as graphs in which the edges are binary dependencies recognized in the syntactic constituents of the questions and the question stems are replaced by semantic classes e.g. PERSON, DISTANCE. (Harabagiu et al. 2000) describes how such semantic classes are assigned as *expected answer types* based on an extensive hierarchy of answer types. For example, given the TREC-8 question *"How far is Yaroslavl from Moscow?"*, we obtain the following semantic transformation of the question:

The source model assigns probabilities to the question semantic transformations of the source question, we consider for every question Q the semantic transformation S such that:

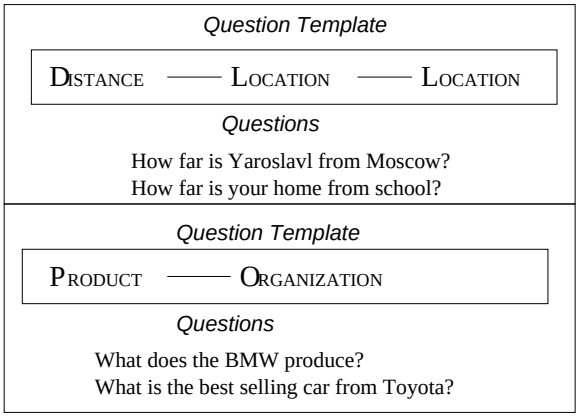
$$S_{best}(Q) = argmax_S P(S|Q)$$

The key to this statistical model is that the semantic transformation is obtained by (1) a set of binary dependencies D; (2) the base NPs recognized by the parser, denoted by B; and (3) the question semantic template information Z, combining semantic information of the classes categorizing question stems with the semantic classes categorizing possible answers. A statistical model for representing B and D was proposed in (Collins 1996) and is implemented in our current parser, used to generate the semantic transformations. However, this model needs to be re-trained to incorporate the semantic-class information available from (a) Named Entity taggers; (b) our off-line answer-type taxonomy comprising over 40 semantic classes covering more than 8000 linguistic concepts; and (c) semantic classes in which the question stems can be mapped, as first reported in (Moldovan, Harabagiu et al. 1999). In this way, our source model becomes:

$$P(S|Q) = P(B, D, Z|Q) =$$

$$= P(B|Q) \times P(D|Q, B) \times P(Z|Q, B, D)$$

The semantic template model computes the probability that a question is mapped in a semantic template, given its BaseNP and binary dependency models. Question semantic templates consist of semantic classes and unlabeled, binary dependencies between them. The semantic classes comprise either tops of the answer type hierarchies, similar to those described in (Harabagiu et al. 2000) or the semantic categories distinguished by our Named Entity tagger. Moreover, multiple questions can be mapped in the same semantic template:



We compute the semantic template probabilities as a product of the probability of recognizing the expected answer type (EAT) as the first class of the template with the probabilities of disambiguating any of the words from the question in the template semantic classes C_T:

$$P(EAT) \times \prod_i P(C_T^i)$$

The stochastic channel model performs some minimal operations on the syntactic constituents and their corresponding binary dependencies with the goal of creating a full semantic representation of the question. The expansion operations correlate entities from the questions that have semantic tags with semantic templates of questions, ranked by the probabilities of the semantic model. For example, given Q_s = *"AFFAIR IS YES LEVEL FROM MOSCOW"* we compute the factors $P(Q_s)$ and $P(Q_t|Q_s)$. We obtain only two subtrees, one spanning *"AFFAIR IS YES LEVEL"*,

the other *"LEVEL FROM MOSCOW"*. The reduced question obtained when the dependencies are known generates two sub-graphs: *affair−level* and *level−Moscow*. The back-off for question templates suggest the replacement of *affair* with several stems: *Where*, *Who* and *How far*, each with a different probability. We also compute $P(Q_t|Q_t)$, which computes the probability of the other relationship from the question template connecting DISTANCE to a LOCATION concept. This entails the replacement of *"YES LEVEL"* in Q_s with the name of a LOCATION. A list of possible locations is provided by the names recognized as LOCATIONS in the cluster selected at filtering time.

5 Interactive Question Answering

Sometimes, due to the errors generated in the ASR, the meaning of a question may be completely lost. The solution is to allow the system to obtain clarifications from the user, and import the additional information in the processing of the question. The following dialog illustrates such a clarification example:

Q: Where did the ukulele originate?
ASR: WHERE DID YOU GO A LEADER IN GENERATE?
A: Are you interested in a specific leader?.
Q': No, I am interested in ukulele, the musical instrument
ASR: NO I'M INTERESTED IN LEADER IN THE MUSICAL IN SUMMER
A': The ukulele, introduced from Portugal into the Hawaiian Islands about 1879, was first used in Canadian schools in the Maritime provinces about 20 years ago to teach music.

When the initial question is processed, only the question stem, *where* was recognized correctly, identifying the correct *expected answer type* as LOCATION. The focus of the question was not recognized correctly, as it was believed that the question asks about some leader. At this point the system generates a clarification question, thus allowing for mixed initiative. The clarification question selects *leader* as the focus concept because of the hypothesis that LOCATIONS are associated with organizations of people, and *leader* is a member of the PERSON subhierarchy. An affirmative answer from the user confirms that the question focus was recognized correctly. However, in our experiment, the user first stated *No*, indicating that the system did not comprehend the topic of the question, recognized by the head of the first NP syntactically dependent on the question stem. Additionally, the user provided a categorization of *ukulele*, defining it as a musical instrument. Fortunately, the keyword *musical* is recognized correctly, and it was used to retrieve the paragraph containing the correct answer, even if *ukulele* was still not recognized.

This was posible because the output of the ASR is not processed in isolation, but rather in the context of the prior interactions. Since the word *leader* was already rejected as the question focus, it is not employed to retrieve relevant paragraphs. Based on the empirical methods developed for selecting question keywords reported in (Moldovan, Harabagiu et al. 1999), only *musical* and *summer* are selected for retrieval. Additionally, each retrieved paragraph had to contain at least one expression identified by the Named Entity Recognizer encoded in the Q&A system as a LOCATION. When the number of such paragraphs is too small (i.e. under 20) the last keyword (*summer*) is dropped and the paragraph retrieval resumes. 384 paragraphs were retrieved, each containing either the word *musical* or any of its alternations and at least one LOCATION. The paragraphs were ordered with a comparison function learned with a method reported in (Paşca and Harabagiu 2001). The first paragraph is returned as the answer to the user's question.

6 Results

To evaluate the iterative refinement of voice and question processing, we measured the performance of both the Q&A and the ASR systems. To measure the performance of the Q&A system we used the same evaluation methods as those employed in TREC and used also the answer keys that were provided by the TREC organizers. 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, if 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 Q&A system for all the $n = 200$ tested questions $MRAR = \frac{1}{n}(\sum_i^n \frac{1}{rank_i})$. Table 1 shows the scores obtained when applying only some of the iterations or different possible combinations of the iterations. We trained the VAQA system on the 700 questions from TREC-9 and tested them on the 200 questions from TREC-8.

Filtering (F) only	0.124
Language Model (LM) only	0.262
Interactive QA (IQA) only	0.176
F+LM	0.406
F+IQA	0.312
LM+IQA	0.388
F+LM+IQA	0.474

Table 1: The Mean Reciprocal Answer Rank (MRAR) of the VAQA System for 200 TREC-8 open-domain questions.

We also measured the Word Error Rate (WER) of the ASR system after each iteration. The results are listed in Table 2.

Filtering (F) only	28.4%
Language Model (LM) only	17.5%
Interactive QA (IQA) only	25.3%
F+LM	15.8%
F+IQA	22.4%
LM+IQA	12.6%
F+LM+IQA	11.3%

Table 2: Word Error Rate in the VAQA System for 200 TREC-8 open-domain questions.

7 Conclusions

We believe that the performance of the voice-activated Q&A depends mostly on the enhanced language model and on the corrections enabled by the interactive Q&A module. To train the enhanced language model, the filtering component was essential, as it allowed to discard multiple questions recognized incorrectly. The results of our experiments show that the interactions enabled by our VAQA implementation improve both the accuracy of spoken Q&A and better the word error rate of the ASR.

References

L. Bahl, P.F. Brown, P.V. de Souza and R.L. Mercer. A Tree-Based Statistical Language Model for Natural Language Speech Recognition. In *IEEE Transactions on Acoustincs, Speech and Signal Processing*, pages 1001–1008, 1989.

Ciprian Chelba and Frederick Jelinek. Expoliting syntactic structure for language modeling. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING/ACL-98)*, Montreal Canada, 1998.

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.

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

S. Harabagiu, D. Moldovan, M. Paşca, R. Mihalcea, M. Surdeanu, R. Bunescu, R. Gîrju, V. Rus and P. Morărescu. The role of lexico-semantic feedback in open-domain textual question-answering. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL-2001)*, Toulouse, France, 2001, pages 274–281.

M. Jardino. Multilingual stochastic N-Gram class language models. In *Proceedings of the IEEE Conference on Acoustincs, Speech and Signal Processing*, pages 161–163, 1996.

Frederick Jelinek. Statistical Methods for Speech Recognition. MIT Press, Cambridge, Massachusetts, USA, 2000.

J. Lafferty, D. Sleator and D. Temperley. Grammatical Trigrams: A Probabilistic Model of Link Grammar. In *Proceedings of the AAAI Fall Symposium on Probabilistic Approaches to Natural Language*, pages 89–97, 1992.

R. Kuhn and R. de Mori. A cache based natural language model for speech recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 691-692, June 1992.

R. Lau, R. Rosenfeld and S. Roukos. Trigger-based language models: a maximum entropy approach. In *Proceedings of the IEEE Conference on Acoustincs, Speech and Signal Processing*, pages 45–48, 1992.

G. Miller. WordNet: a lexical database. *Communications of the ACM*, 38(11):39–41, 1995.

Dan I. Moldovan, Sanda M. Harabagiu, Marius A. Paşca, Rada Mihalcea, Roxana Gîrju, Richard Goodrum and Vasile Rus. LASSO: a tool for surfing the answer net. In *Proeceedings of the 8th Text Retrieval Conference (TREC-8)*, pages 65–73, 1999.

M. Paşca and S. Harabagiu. High Performance Question/Answering In *Proceedings of the 24th Annual International ACL SIGIR Conference on Research and Development in Information Retrieval (SIGIR-2001)*, September 2001, New Orleans LA, pages 366-374.