Voice Activated Question Answering

Sanda Harabagiu
Department of Computer Sciences
University of Texas at Austin
Austin, TX 78712
sanda@cs.utexas.edu

Joe Picone
Institute of Signal and Information Processing
Mississippi State University
Mississippi State, Mississippi 39762
picone@isip.msstate.edu

Dan Moldovan
Department of Computer Science
University of Texas at Dallas
Dallas, TX 75275
moldovan@utdallas.edu

Extended Abstract
Paper Presentation

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.

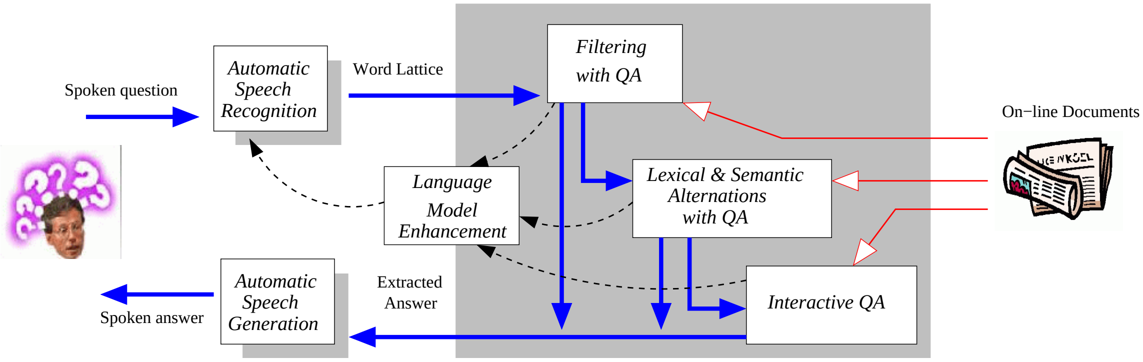
Spoken question
Automatic Speech Recognition
Word Lattice
Filtering with QA
On−line Documents
Language Model Enhancement
Lexical & Semantic Alternations with QA
Automatic Speech Generation
Extracted Answer
Interactive QA
Spoken answer

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

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. 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 this research.

2 Interative Refinement of Voice and Question Processing

Our Open-Domain Voice-Activated Question-Answering 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 re-process 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. 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 can be recovered and the language model may be further enhanced to capture the missing linguistic information.

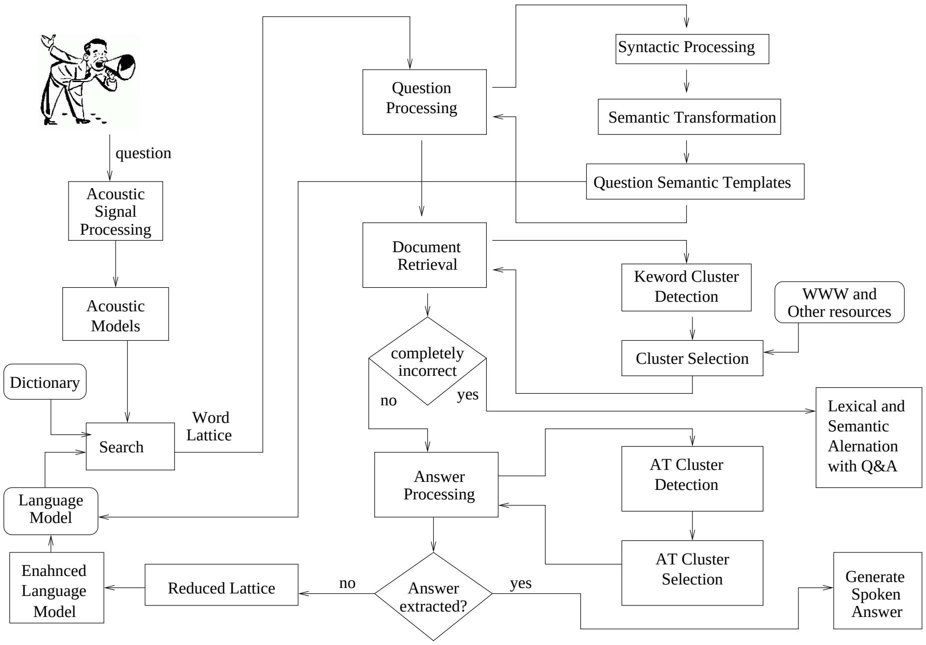

Figure 2: Architecture of the Filtering component of a Voice-Activated Question/Answering System

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 re-considering 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 latice on which the enhance language model is trained, three additional forms of filtering take place: (1) question processing filtering; (2) passage retrieval filtering; 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.

3 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 semantics and pragmatics by concentrating on the local dependencies between words. However, our preliminary experiments have shown that N-grams are insufficient for the recognition of question words.

Possible alternatives are long-range dependencies as reported in [Chelba and Jelinek 1998] or [Kupiek 1989], cache dependencies described in [Kuhn and de Mori 1992], link dependencies as introduced in [Lafferty et al.1992] and trigger models reported in [Lau et al.1992], class grammars as in [Jardino 1996] and decision-tree clustered language models described in [Bahl et al.1989]. All these language models 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.

4 Interactive Question Answering

The ability to interact with the user in real-world systems that seek information from large collections of texts is an important challenge because it is the only way of obtaining clarifications, confirmations or additional information from the user. In IR for example, the interaction is enabled by a mechanism called *relevance feedback*. Relevance feedback allows the users to review preliminary search results and indicate which text fragments they find particularly relevant. The search query is subsequently modified to reflect these preferences (usually by adding or re-weighting some keywords) and the search is repeated. We note that relevance feedback is a form of dialog, albeit a very primitive one where the user has to make all the moves: ask the question, assess results, indicate how to reformulate the query. Because the dialog is so one-sided, the entire process is an inefficient trial-and-error mechanism. However, relevance feedback is a powerful mechanism for obtaining high-precision and high-recall results in IR.

In Q&A the user expectations of accuracy and quality are understandably higher than in document retrieval, thus there is little tolerance for answers that are off-target, let alone irrelevant. Furthermore, due to the errors generated in the ASR, the meaning of the question might 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 output: 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 output: 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 hopefully the system generates a clarification question, thus allowing for mixed initiative. To respond to this question, the user might first state *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 provides a categorization of *ukulele*, defining it as a musical instrument. Fortunately, the keyword *musical* is recognized correctly, and it can be used to retrieve the paragraph containing the correct answer, even if *ukulele* was still not recognized.

References

[Bahl et al.1989] 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.

[Chelba and Jelinek 1998] 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.

[Jardino 1996] 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.

[Kupiek 1989] J. Kupiec. Probabilistic models for short and long distnace word dependencies in running text. In *Proceedings of the ARPA Workshop on Speech and Natural Language*, pages 290–295, Philadelphia, PA, 1989.

[Kuhn and de Mori 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.

[Lafferty et al.1992] 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.

[Lau et al.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.