MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

The MERL SpokenQuery Information Retrieval System: A System for Retrieving Pertinent Documents from a Spoken Query

Peter Wolf and Bhiksha Raj

TR2002-57 August 2004

Abstract

This papers describes some key concepts developed and used in the design of a spoken-query based information retrieval system developed at the Mitsubishi Electric Research Labs (MERL). Innovations in the system include automatic inclusion of signature terms of documents in the recognizer vocabulary, the use of uncertainty vectors to represent spoken queries, and a method of indexing that accommodates the usage of uncertainty vectors. This paper describes these techniques and includes experimental results that demonstrate their effectiveness.

IEEE International Conference on Multimedia Expo (ICME), August 2002

IEEE International Conference on Multimedia and Expo (ICME)

The MERL SpokenQuery Information Retrieval System

A System for Retrieving Pertinent Documents from a Spoken Query

Peter Wolf and Bhiksha Raj
Mitsubishi Electric Research Labs
201 Broadway, Cambridge, MA 02139, USA

ABSTRACT

This papers describes some key concepts developed and used in the design of a spoken-query based information retrieval system developed at the Mitsubishi Electric Research Labs (MERL). Innovations in the system include automatic inclusion of signature terms of documents in the recognizer's vocabulary, the use of uncertainty vectors to represent spoken queries, and a method of indexing that accommodates the usage of uncertainty vectors. This paper describes these techniques and includes experimental results that demonstrate their effectiveness.

1. Introduction

In this paper we address some problems related to the design of Information Retrieval (IR) systems that respond to spoken queries. Such systems are extremely useful in situations where the device used for IR is too small for a keyboard, such as PDAs or cell phones; or when hands-free operation is required, such as while driving a car. The conventional approach to such tasks is to use a speech recognition system to convert the spoken utterance to a text transcription which is then passed on to a regular text-based IR search engine. The IR engine would be unaware that the query was in fact spoken and not typed.

There are three problems that can be identified with this approach:

a) Misrecognition by the speech recognition engine causes poor retrieval performance. It is well known that speech recognition systems are imperfect transcribers of speech, especially when the recording conditions for the signals are unconstrained (e.g. noise, distortion, speaker accent, speaker gender, speaker age) or the recognizer must recognize words from a very large vocabulary. Unfortunately, these conditions cannot be avoided for spoken-query based IR on devices such as hand-held computers or mobile phones. The devices are small and inexpensive, the users are not trained, and the environment in which users will use the device cannot be constrained. Also, for effective IR the recognition vocabulary must be large enough to include all possible query words. Recognition errors are therefore bound to occur, and as a result important query words may not be recognized. A spoken-query based IR system must therefore be able to account for errors made by the recognizer

b) Speech recognition engines are poor at recognizing the specialized words that identify many documents. The reason is that IR systems must index ever expanding sets of documents. Many of these documents contain new or rare words that are, in fact, the signature terms that distinguish them from other documents. These are the terms that users who wish to retrieve these documents are most likely to use in their queries. On the other hand, speech recognition systems, being pattern classifiers, are biased to favor more frequently occurring words in the language over less frequent ones. In fact, vocabularies for large-vocabulary recognition systems are usually chosen as the most frequent words in relevant corpora. This design would be counterproductive in IR systems since the signature terms for most documents, being rare, would not be in the recognizer's vocabulary and could never be recognized. An effective spoken-query based IR system must be able to actively identify signature terms of indexed documents and include them in the recognizer vocabulary.

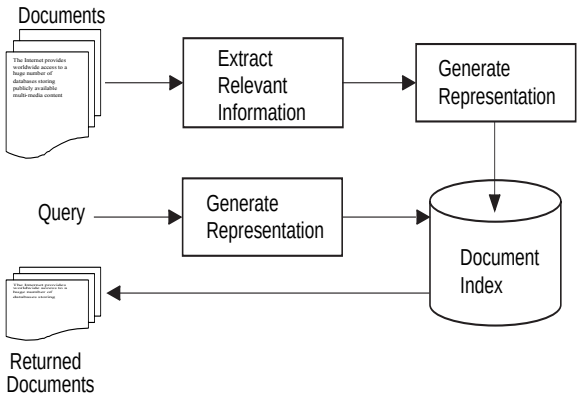

Figure 1. Schematic representation of a standard IR system.

c) Text based IR systems often do not have a document index that allows comparison between documents where the words are certain with queries where the words are uncertain. Figure 1 shows a schematic representation of a typical IR system. Information is extracted from the documents to be indexed and converted to a standard representation, which is then stored in an index. Incoming queries are also converted to a standard representation and compared against the index to locate relevant documents. The manner in which the query is represented must be compatible with the representation of documents in the index. However, in spoken-query based IR systems the representation of query may be governed by how problems a) and b) are tackled. In this case the representation of documents in the index must also be suitably designed to be compatible with the query representation.

In this paper we address all three of these problems. Our solutions include key term spotting based vocabulary update, certainty-based spoken query representation, and projection-based indexing. In the first, we automatically detect new key terms in the indexed documents and use them to augment the vocabulary of the recognizer. In the second, instead of using the best choice transcript output of the speech recognizer to determine query words, we use its search space of possible hypotheses to generate certainty-based

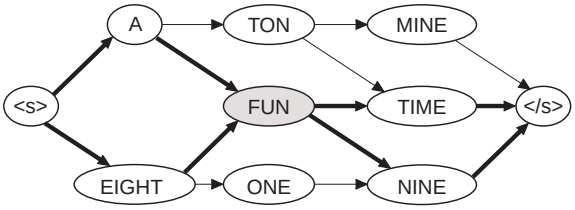

Figure 2. Example of a simple lattice. The thick lines represent all the paths through the lattice that go through the word "FUN". The ratio of the total likelihood of these paths to the total likelihood of the lattice gives us the *a posteriori* probability of "FUN".

query vectors. In the third we represent the document index with low dimensional projections of word count vectors that can be directly compared against the query vectors. Using these solutions, we achieve superior results as compared to those obtained when the recognizer is blindly used as a speech-to-text convertor. In Sections 2, 3 and 4 we present each of these solutions. In Section 5 we describe an integrated implementation of a spoken-query based IR system that uses these solutions. Experimental results and conclusions are presented in Sections 6 and 7, respectively.

2. Certainty Based Query Representation

Speech recognition systems consider many possible hypotheses when attempting to recognize an utterance. These various alternate hypotheses are represented as a graph that is commonly known as a lattice. Figure 2 shows an example of a lattice. The best choice transcript generated as a result by the recognizer is the most likely path through this lattice (i.e. the path with the best score). However, the words that were actually spoken are often found in the lattice, even though they may not be in the most likely path. Every word in the lattice can be ascribed a measure of certainty that it was indeed spoken, regardless of whether or not it was on the best path. Certainty-based query representation is based on the measurement of the certainties of all words in the lattice.

We measure the certainty of any word in the lattice as its *a posteriori* probability. The *a posteriori* probability of any word in the word lattice is the ratio of the total likelihood scores of all paths through the lattice that pass through the node representing that word, to the total likelihood score of all paths through the lattice. Path scores are computed using the acoustic likelihoods of the nodes [1]. The acoustic likelihood of any node in the lattice represents the logarithm of the probability of that node computed by the recognizer from the acoustic signal and its internal statistical models. The total probability of any path through the lattice is given by

$$P(n_1, n_2, \ldots, n_W) = \exp(L(n_1) + L(n_2) + \ldots + L(n_w)) \quad (1)$$

where n_i represents the i^{th} node in the path and $L(n_i)$ represents its likelihood. The total probability of all paths that pass through a node, as well as the total probability of all paths through the lattice, can be computed using the forward backward algorithm. Let $P_{total}(n_i)$ represent the total probability of all paths that pass through the node n_i. Let P_{total} represent the total probability of all paths through the lattice. The *a posteriori* probability of the node n_i is given by

$$P_{aposteriori}(n_i) = \frac{P_{total}(n_i)}{P_{total}} \quad (2)$$

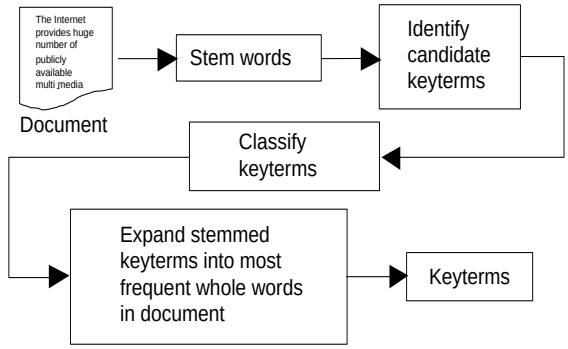

Figure 3. Algorithm for detecting key terms.in a document.

All words in the lattice are stemmed and their *a posteriori* probabilities computed. Stemming removes the suffixes of words, thereby making functionally similar words identical [1]. The *a posteriori* probabilities of the words in the lattice are then used to construct a *query vector*. Each element in the query vector represents one of the words in the vocabulary of the index. The value of the component corresponding to any word is the total of the *a posteriori* probabilities of all instances of that word in the lattice. If a word does not occur in the lattice, its component in the query vector is set to 0.

3. Keyterm Spotting Based Vocabulary Update

Most documents contain signature terms that help identify the nature of their contents. These signature terms may include both key*words* and key*phrases* that are strings of two or three words. Keyphrases typically contain one or more keywords. Users may use both keywords and keyphrases when querying for a document. It is essential for the keywords to be present in the vocabulary of the speech recognition component of a spoken query-based IR system. They must therefore be identified and incorporated into it. The ability of the system to correctly recognize keywords is enhanced if keyphrases in the documents are incorporated in the recognizer's grammar as well. For this reason, keyphrases must also be identified and used for recognition, where possible. We will refer to keywords and keyphrases as keyterms in this paper.

Keyterms are frequently marked using the <meta> tag in documents encoded in markup languages. When such tags are present, we can simply utilize these to locate keyterms and incorporate them into the recognizer. When these tags are not available, however, we must identify the keyterms automatically. Our algorithm for keyterm detection is similar to many of the keyterm detection algorithms proposed in the literature [3]. It begins by stemming all the words in the document. Following this, candidate keyterms are identified. Candidate keywords are words that are present in the document but not in the current recognition vocabulary. Candidate keyphrases are all sequences of up to 3 words such that none of the words is a stop word, *i.e.* words such as "a", "and", "to" etc. whose function is purely grammatical. For each of the candidates, feature vectors that contain measurements such as the frequency of occurrence of the term in the document, the relative position of the first occurrence, the average length in characters of the unstemmed versions of the term etc., are computed. These vectors are then passed to a classifier that determines whether they are keyterms or not. The classifier used is a decision tree [4] that has been trained on a hand tagged corpus of documents. All stemmed candidates that are classified as keyterms are then returned to their most frequently

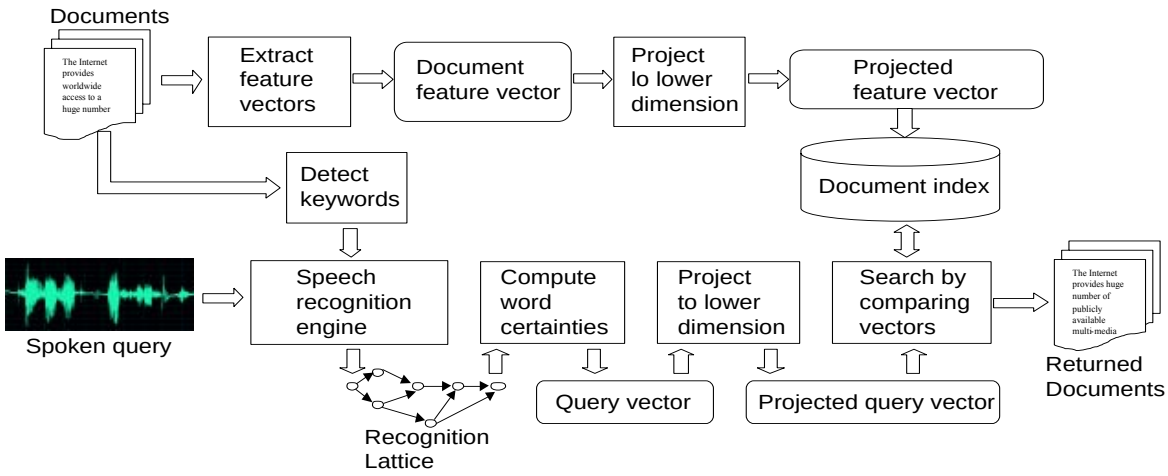

Figure 4. Schematic of the architecture of SpokenQuery

occurring unstemmed version in the document. The entire procedure is represented pictorially in Figure 3. All identified keyterms are then incorporated into the speech recognition system.

Storing only the most frequent unstemmed forms of keywords in the recognition vocabulary does not affect the performance of the system adversely. This is because the stored form of any word usually occurs in the recognition lattice even when a different form of the word is spoken. This is sufficient to identify the desired document using certainty-based query representations.

4. Projection Based Indexing

Document representations proposed for IR systems. include those that treat documents as collections of words, *e.g.* the bag-of-words representation [5] and the vector space representations [6], and those that retain word sequence information, *e.g.* N-gram representations [7]. Of these, the vector space representation is most suitable for a spoken query based IR system that uses the certainty-based query representation described earlier.

In the vector space model documents are represented as vectors, where each element in the vector represents a word, and the value of that element represents the frequency with which that word occurs in the document. Documents are first stripped of stop words and the remaining word are stemmed before they are converted to the vectors. The vectors are then projected to a lower dimensional space using a linear transform derived from Singular Value Decomposition (SVD) [6] of the complete set of documents.

SVD begins by representing the set of documents being indexed as a matrix D. Representing the n^{th} document in the set as d_n, the construction of D can be represented as $D = [d_1, d_2, \ldots]$. If the the number of elements in the index vocabulary is M and the number of documents to be indexed is N, D is an $M \times N$ matrix. SVD decomposes this matrix as

$$D = U\Sigma V^T \tag{3}$$

where U is an $M \times N$ matrix, Σ is an $N \times N$ diagonal matrix and V is an $N \times N$ matrix. The diagonal entries of Σ are known as the *singular values* of D and are arranged in decreasing order of value. In order to project the document vectors down to a K dimensions, a *projection matrix* P is constructed of the first K columns of U. Any M dimensional document vector d is now projected down to a K dimensional vector d' as

$$d' = P^T d \tag{4}$$

The projected document vectors and the projection matrix P must all be stored for purposes of indexing. During retrieval, a query vector Q is also projected to a lower dimensional vector Q' using P as $Q' = P^T Q$. Q' is then compared against the document vectors in the index and the documents that are closest to it are returned. The distance between the query vector and a document d' is measured using the cosine distance metric which is given by

$$Dist(Q', d') = \frac{Q' \cdot d'}{|Q'||d'|} \tag{5}$$

If documents are added or removed from the index, changes must be made to the document matrix D. Consequently, the projection matrix P and the projected document vectors d' must all be recomputed. This task can however be performed incrementally using method such as [8], without requiring access to the entire set of documents.

5. Implementation of SpokenQuery

Figure 4 shows the overall implementation of the MERL SpokenQuery system. The initial set of documents is converted to the vector space representation and projected down to 200 dimensions using SVD. When additional documents are added to the index, both the transformation and the transformed feature sets are recomputed. The SpokenQuery server stores the projected document vectors and the SVD transformations.

The SpokenQuery system also produces and stores two versions of the vocabulary: one for the recognition engine, and one for indexing. The speech recognition engine vocabulary contains whole words. The other vocabulary is stemmed and is used to identify the components of the document and query vectors.

A posteriori probability based query vectors are computed from recognition lattices. The query vectors are projected down to 200 dimensions using the stored SVD transformation. The projected query vectors are compared against the projected document vectors in the index. Comparison is performed using the cosine measure. The top few highest scoring documents are returned to the user in decreasing order of score.

Transcript of spoken query: Volume Rendering
Best recognizer hypothesis: All You Entering
Titles of retrieved documents:

1. Architectures for Real-Time Volume Rendering
2. Bayesian Method for Recovering Surface..
3. Calculating the Distance Map for Binary Surface..
4. EWA Volume Splatting
5. Beyond Volume Rendering: Visualization,..

Table 1: Example of documents retrieved by SpokenQuery.

6. Experiments

The performance of SpokenQuery was evaluated on a corpus of 262 technical reports. The CMU Sphinx-3 speech recognition system was used for the speech recognition component of the system. The recognizer was trained with 60 hours of broadcast news data that are acoustically very dissimilar to the SpokenQuery test data. Experiments were conducted using two different language models. The first, built from broadcast news text, performed poorly on recognizing utterances associated with technical reports. The second language model was created from the text of the technical reports and performed extremely well.

We compared the performance of SpokenQuery against retrieval based on textual queries, and retrieval based on the recognizer's "best" hypothesis. Users were asked to query the system for documents using speech and typed input of what was spoken. The system returned the top 10 documents using the SpokenQuery, retrieval based on the best hypothesis output by the recognizer, and retrieval based on the typed input. The returned 30 documents were then tagged by the users as pertinent (2), somewhat pertinent (1) or not pertinent (0). The sum of these values was the "total pertinence" for a query result.

The performance with text-based queries does not contain any errors and therefore provides the ceiling against which the performance of the other two methods can be compared. Table 2 shows the pertinence of SpokenQuery and the "best" hypothesis, normalized by that of the text input. As expected, the performance of the naive approach using the "best" hypothesis works very well when the recognition is accurate, but degrades very quickly as the error rate increases. SpokenQuery, on the other hand, is slightly worse for very accurate recognition, but much more robust to recognition errors. Table 1 shows a typical result from these sessions using SpokenQuery. It is clear that the naive method would fail completely in this example, whereas SpokenQuery is able to retrieve all the relevant documents in our database.

LM type	Technique	Top 10	Top 5	Top 1
Matched LM	Best hyp.	0.84	0.75	0.76
	SQ	0.84	0.78	0.70
Mismatched LM	Best hyp.	0.43	0.41	0.53
	SQ	0.69	0.65	0.77

Table 2: Comparison of best hypothesis and SpokenQuery

For retrieval based on poor recognition, the ratio of the total pertinence of retrieved documents using SpokenQuery to that of textual queries was 42% better than when using the "best" hypothesis.

7. Discussion

The experiments indicate that the design of the SpokenQuery IR system is very effective. The results obtained are much better than those that can be obtained using a simple combination of a speech recognition system and a text based IR system. However, our experiments are preliminary since both the size of the index and the size of the tests were very small. More comprehensive testing using standardized databases such as the TREC database is required. These databases, however, do not come with standardized spoken query components as well, and these must be recorded. We are currently recording these spoken queries for further experimentation.

The design of SpokenQuery in the current format can also be improved. SVD-based representation of document is based on projection bases that bear no direct resemblance to query vectors. A better representation is to use non-negative matrix factorization (NMF) [9] to represent documents. NMF uses projection bases that resemble word count histograms and are inherently better suited for use with certainty-based query vectors. However, incremental updating of indices is difficult for NMF.

Another important possibility is that of deriving query vectors from phone-level recognition. Here, the recognizer would only recognize phonemes in the language and generate a lattice of phonemes. This lattice would then be used to estimate the *a posteriori* probabilities of all words in the recognition vocabulary. While this procedure is somewhat less accurate than that described in Section 2, it is considerably more flexible. The recognizer only needs to recognize a small set of phonemes and can therefore be much smaller. The recognizer could then be performed on the IR client. The phoneme lattice can be transmitted to a server that constructs query vectors from it in a post-processing step. Vocabulary and grammar update can be performed at the server without any modification of the recognizer.

REFERENCES

1. Evermann, G., and Woodland, P. C., "Large Vocabuary Recognition and Confidence Estimation using Word Posterior Probabilities", *Proc. ICASSP 2000*, Istanbul, Turkey.

2. Porter, M.F., "An algorithm for suffix stripping", *Program; automated library and information systems*, 14(3), 130-137, 1980.

3. Turney, P. D., "Learning to Extract Keyphrases from Text", *NRC Technical Report ERB-1057*, National Research Council Canada, 1999.

4. Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984), *Classification and Regression Trees*. Wadsworth, Belmont, CA

5. Monz, C., "Computational Semantics and Information Retrieval", in *Bos, J., and Kohlhase, M. (eds.), Proc. Second Workshop on Inference in Computational Semantics*, July 2000

6. Berry, M. W., Fierro, R. D., "Low-Rank Orthogonal Decompositions for Information Retrieval Applications", *Numerical Linear Algebra with Applications*, Vol 3 pp. 301-328, 1995.

7. W. Cavnar. "Using an N-gram based document representation with a vector processing retrieval model", Proc. TREC 3, 1994

8. M. Berry, "Large Scale Singular Value Computations" , *Intl. Journal of Supercomputer Applications* , Vol 6, pp. 13-49, 1992

9. Lee, D.D., and Seung, H.S, "Learning the parts of objects by non-negative matrix factorization", *Nature* 401, 788-791, 1999