Speech-Based Retrieval Using Semantic Co-Occurrence Filtering

Julian Kupiec, Don Kimber and Vijay Balasubramanian

Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304

ABSTRACT

In this paper we demonstrate that speech recognition can be effectively applied to information retrieval (IR) applications. Our system exploits the fact that the intended words of a spoken query tend to co-occur in text documents in close proximity whereas word combinations that are the result of recognition errors are usually not semantically correlated and thus do not appear together. Termed "Semantic Co-occurrence Filtering" this enables the system to simultaneously disambiguate word hypotheses and find relevant text for retrieval. The system is built by integrating standard IR and speech recognition techniques. An evaluation of the system is presented and we discuss several refinements to the functionality.

1. Introduction

In applying speech recognition techniques to retrieve information from large unrestricted text corpora, several issues immediately arise. The recognition vocabulary is very large (being the same size as the corpus vocabulary). Each new corpus may cover a different domain, requiring new specialized vocabulary. Furthermore the constraint afforded by domain-dependent language models may be precluded due to the expense involved in constructing them.

One approach to these problems obviates the need for any word vocabulary to be defined [1, 2]. This is done by defining a phonetic inventory based on phonetically stable sub-word units which have corresponding orthographic counterparts. This scheme has the potential advantage that both speech and text can be indexed in terms of the same units and thus speech might be used to access text and vice-versa. Subword units are considered to be independent and matching is performed using vector-space similarity measures.

Our concern in this paper is to provide speech access to text and our approach differs from the former in that whole words are used to constrain matching; we believe this to be more effective than splitting words into smaller independent units. We use boolean retrieval with proximity constraints rather than vector-space measures. Our approach also accommodates standard phonetic alphabets (we employ a set of 39 phones in contrast to the former technique which uses about 1000 phonetic units).

To demonstrate the feasibility of our approach we have implemented a prototype. The user speaks each word of a query separately and is presented with the most relevant titles, each accompanied by the relevant word hypotheses. The combination of speech processing and retrieval currently takes about 20-30 seconds. Figure 1 shows all titles produced for the query "Maltese Falcon".

The IR system acts as a novel kind of language model. The text corpus is used directly; it is not necessary to pre-compute statistical estimates, only to index the text as appropriate for the retrieval system.

1. Maltese Falcon, The	(maltese falcon)
2. Astor, Mary	(maltese falcon)
3. film noir	(maltese falcon)
4. Bogart, Humphrey	(maltese falcon)
5. Huston, John	(maltese falcon)
6. Hammett, Dashiell	(maltese falcon)
7. Louis XV, King of France	(marquise faction)
8. rum	(indies factor)
9. drama	(please fashion)

Figure 1: Presentation of Search Results

2. System Components

The overall architecture of the system is shown in Figure 2. We will first describe the IR and speech systems and then the ancillary components that integrate them.

2.1. Retrieval System

We use the Text Database [3] for indexing and for boolean search with proximity constraints. We have experimented with Grolier's encyclopedia [4] which is a corpus of modest size (8M words) spanning diverse topics. There are 27,000 articles in the encyclopedia and an uninflected word dictionary for it contains 100,000 entries. We use a stop list [1] containing approximately 100 words. The fact that short common words are included in the stop list is fortuitous for our speech-based retrieval because they are difficult to recognize.

2.2. Phonetic Recognizer

The phonetic recognition component of the system uses standard hidden Markov model (HMM) based speech recognition methods. The system currently operates in a speaker-dependent, isolated-word mode as this was the simplest to integrate and known to be more robust operationally. Input

[1] A stop list contains common words that are not indexed because they are not useful as query terms.

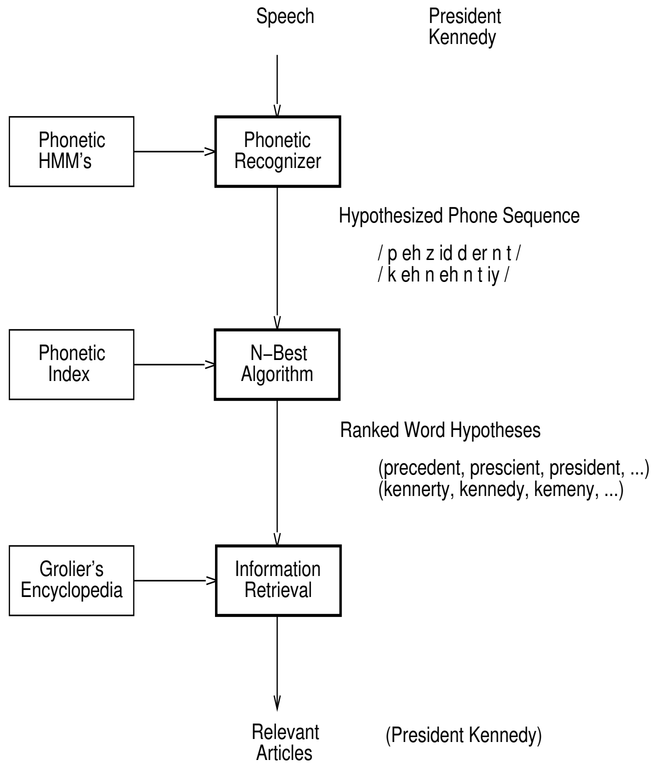

Figure 2: System Components

to the system was from a Sennheiser HMD-414 microphone, sampled at a rate of 16KHz. Feature vectors consisted of 14 Mel-scaled cepstra, their derivatives and a log energy derivative. These were computed from 20 msec frames taken at a rate of 100 per second. Training data for each speaker was taken from 1000 words spoken in isolation. Each phonetic model is a three state HMM with Gaussian output distributions having diagonal covariance matrices. The topology of the phonetic models is shown in Figure 3. Continuous training was used to avoid the need for phonetically labelling training data by hand. The models were initialized from speaker independent models trained on the TIMIT speech database [5]. For recognition, the models were placed in a network with probabilities reflecting the phonetic bigram statistics of the lexicon. For each spoken word, a hypothesized phone sequence was determined by the maximum likelihood state sequence through the network, computed using the Viterbi algorithm.

2.3. Phonetic Dictionary

To use the IR system with speech we construct a phonetic dictionary which is a table giving a basic phonetic spelling for each entry in the word dictionary. For example the phonetic spelling for the word "president" is the string of phonetic symbols "P R EH Z IH D EH N T". In our implementation we associate a single phonetic spelling with each word. More generally, phonological variants, alternative pronunciations or even translations into other languages can also be placed in the phonetic dictionary. In our arrangement the user needs to speak the uninflected word form that corresponds to the uninflected spelling that is used for indexing. (Again, we can dispense with this by including the phonetic spellings of word inflections.)

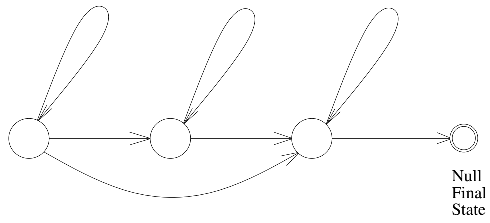

Figure 3: Topology of a Phone HMM

The question remains as to how we find phonetic spellings for all the entries in the word dictionary. We have split this problem into two parts. The first is to obtain a list of words and their phonetic spellings. We have adapted a list containing phonetic spellings for 175,000 words [6]. Of the 100,000 word types in the encyclopedia, 43,000 were covered by this list. Although this is less than half of the total vocabulary size, it nevertheless does represent the majority of actual word instances in the encyclopedia. To cover the rest of the words we propose the application of techniques for automatically producing phonetic spellings, e.g. [7, 8]. Such techniques are prevalent in text-to-speech synthesis.

3. N-Best Matching

For each spoken word, the recognizer outputs the most likely corresponding phone sequence. As a result of recognition errors, the phonetic sequence may not match any entry in the phonetic dictionary, or worse, might match an incorrect word. For example, when a speaker intended the word "president" the recognizer output "P R EH S EH D EH N T" would incorrectly match the word "precedent". We therefore employ a statistical model of the errors typically made by the recognizer and use it to determine what words were likely to have been said.

Given the phonetic sequence produced by the recognizer, and a statistical model of recognition errors, we want to efficiently determine the n most likely entries in the phonetic dictionary to have been the actual spoken word. As will become apparent, our objective is to make sure the intended word is somewhere in the list. We have investigated two methods for producing the n-best word hypotheses. The first follows a generate-and-test strategy and the second, more successful approach involves an HMM-based search through the phonetic dictionary.

In the remainder of this section we will discuss the characterization and estimation of error statistics, and then describe the n-best algorithms.

3.1. Characterizing Recognizer Errors

Errors made by the recognizer are described by matrices containing probabilities of various substitution, deletion and insertion errors. We produce the error matrices by using an alignment program that compares phonetic recognizer output for a set of spoken words with the correct phonetic transcriptions for those words. (This set comprises 1000 words). Speaker characteristics are also modelled in the error matrices, as are systematic pronunciation differences between the phonetic dictionary and the speaker. For words that are generated automatically (as mentioned in Section 2.3) we would expect a separate distribution to be helpful because the characteristics of an automatic system are likely to have errors distributed in a different way.

The results described in this paper are based on a context independent error model. However, recognition errors are strongly correlated with context, and an improved model would use context dependent statistics. Both of the n-best methods described below are easily adapted to the use of context dependent statistics.

Given the relatively small amount of training data used for estimating error statistics, some form of smoothing is desirable. We employ a Laplacian estimator – if phone i occurs a total of N_i times and is recognized as phone j a total of N_{ij} times, the estimated probability of such a substitution is

$$P_{SUB}(j|i) = \frac{N_{ij}+1}{N_i+M}$$

where M is the size of the phonetic alphabet ($M = 39$ for our phone set.)

3.2. Generate and Test

Our initial method for determining the n-best word hypotheses employed a best-first approach. The most likely phone substitutions/insertions/deletions are applied in a best-first order to the phone string produced by the recognizer. After each such modification if the resulting phone string is present in the phonetic index, it is added to the n-best list with its associated probability (being the product of the probabilities of the modifications applied to the original string in order to obtain it). Finite-state recognizers are used to determine whether a phone string is present in the index. This search method has the potential advantage that it does not require matching against every entry in the phonetic index to produce the n-best hypotheses.

3.3. HMM Search

This involves matching the recognizer output against a special HMM network for each phonetic entry in the index (n.b. these HMM's are quite separate from those used by the phonetic recognizer).

Let $p(w|y_1, y_2, \ldots, y_n)$ be the probability that word w was spoken given that the phonetic output produced by the recognizer is $y_1, y_2, \ldots, y_n$. It is necessary to find the n words for which $p(w|y_1, y_2, \ldots, y_n)$ is greatest. By Bayes law:

$$p(w|y_1, y_2, \ldots, y_n) = \frac{p(y_1, y_2, \ldots, y_n|w)P(w)}{p(y_1, y_2, \ldots, y_n)}$$

The prior probabilities $P(w)$ are assumed uniform here and $p(y_1, y_2, \ldots, y_n)$ is independent of w, so the problem is to find the n words for which $p(y_1, y_2, \ldots, y_n|w)$ is maximum.

If the phonetic dictionary entry for word w is $x_1, x_2, \ldots, x_m$, then given the error statistics, the probability

$$p(y_1, y_2, \ldots, y_n|w) = p(y_1, y_2, \ldots, y_n|x_1, x_2, \ldots, x_m)$$

can be computed by adding the probability of every sequence of substitutions, deletions and insertions, which when applied to $x_1, x_2, \ldots, x_m$ results in the sequence $y_1, y_2, \ldots, y_n$. Assuming that these types of errors are statistically independent, the calculation can be performed efficiently using dynamic programming. By defining a discrete HMM for w, in which the output symbols are phones, the calculation reduces to the computation of the probability that $y_1, y_2, \ldots, y_n$ would be produced by the HMM (i.e. the "forward" probability).

For example, the structure of the HMM for the word "go" consisting of phonemes /g/ and /ow/ is shown in Figure 4. The large states represent the phones of the word, and have output probabilities determined from the substitution probabilities. The remaining output states (smaller and gray in the figure) model possible insertions. The output probabilities for these states are the estimated insertion probabilities, conditioned on the event that an insertion occurs. Self loops on these states allow for the possibility of multiple insertions. The null state underneath each large phone state models the deletion of that phone, and the null states underneath insertion states allow for the possibility that no insertion occurs at that position. The transition probabilities are determined from estimated insertion and deletion probabilities.

The HMM structure shown in Figure 4 could be replaced by a structure having no null states. However the structure chosen is preferable for two reasons. First, the computation of $p(y_1, y_2, \ldots, y_n|x_1, x_2, \ldots, x_m)$ requires only $O(mn)$ operations rather than $O(m^2n)$ which would be required without null states. Second, the computation for this structure is easily implemented using the phonetic pronunciation for each word to index a table of transition and output probabilities, so that an HMM does not need to be explicitly stored for each word.

We have implemented the n-best search efficiently and a pass through 43,000 phonetic index entries takes a few seconds. Including the signal processing (also done in software) the system takes between 5-10 seconds to produce the n-best hypotheses per spoken word (running on a Sun SPARC-10, using a value of n = 30). After the HMM search is complete we have a list of the most likely matching words and their associated probabilities.

4. Semantic Co-Occurrence Filtering

Let us consider an example where the user speaks the words "president" and "kennedy" into the system. These might result in the following rank ordered lists of word hypotheses:

president: (precedent, prescient, president...)

kennedy: (kennerty, kennedy, kemeny, remedy...)

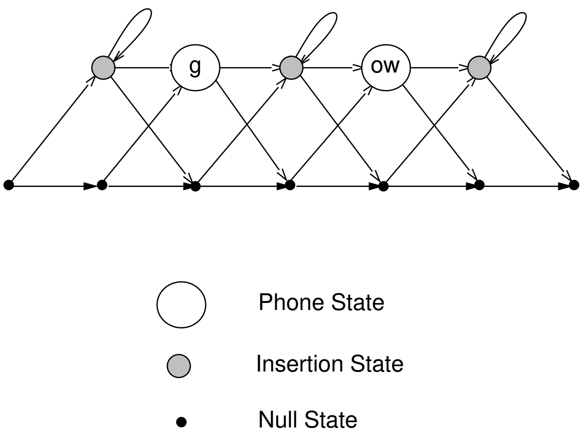

Figure 4: HMM for Word Matching

In neither case is the intended word the most likely, although both are present and near the tops of the lists. The next step effectively uses the text of the encyclopedia (accessed via the IR system) as a semantic filter. In the encyclopedia the only members of the above lists which co-occur in close proximity are "president" and "kennedy". The intended words of the query are semantically related and thus co-occur close to each other (many times in this example) but the hypotheses that are the result of recognition errors do not.

Each spoken word is represented by an OR term containing its most likely word hypotheses. These terms are combined with an AND operator having a proximity constraint on its members. For our example we might have:

(AND 15 (OR precedent, prescient, president, resident...)
(OR kennerty, kennedy, kemeny, remedy...))

This query is submitted to the IR system and segments of text that satisfy the constraints are retrieved. The word hypotheses involved in each text segment are then identified. They are used to score the segments and also to rank the best word hypotheses. Scoring includes the phonetic likelihood of the hypotheses, the total number of occurrences of specific hypothesis combinations in all retrieved text segments, and typical IR word weighting.

5. Evaluation

We created 100 queries, each composed of a few words that characterize a topic of interest (e.g. "Apollo space program"). To evaluate the benefit of semantic co-occurrence filtering directly, we verified that the words we selected had entries in the phonetic dictionary and that the encyclopedia contained at least one relevant article.

Of the 100 queries, 83 were successful (i.e. retrieved at least one relevant article in the top 25 titles). If only the top word hypotheses from the n-best component were inserted in the boolean queries, only 32 of the queries would succeed. The

Rank	N-Best	After Semantic Filter
First	111 (64%)	165 (95%)
Top 5	153 (88%)	174 (100%)
Top 10	163 (94%)	174 (100%)
Top 30	174 (100%)	174 (100%)

Table 1: Effect of filtering on word rank, for successful queries

17 unsuccessful queries all failed because the correct word was not present in the top 30 hypotheses. For each spoken word we compared the rank of the correct phonetic hypothesis output from the n-best component with that produced after semantic co-occurrence filtering. Table 1 shows that such filtering finds relevant documents while simultaneously improving recognition performance.

Some of the successful queries are shown below (practically all of the test queries comprise two or three words).

Example Queries:

1 first atomic bomb
2 assassinate kennedy
3 xerox corporation
4 planet jupiter
5 gecko lizard
6 chester carlson
7 solid state physic
8 discover penicillin
9 dinosaur extinct
10 mary queen scot

In constructing the test queries, sometimes only a single word immediately came to mind for some topics. In such cases we found that a useful strategy for adding another word was to use either a name, hyponym or hypernym. Thus the word "ant" was augmented by adding "insect" as a second word. Although less robust, single word queries are not precluded. Either their length may distinguish them (e.g. "savonarola" and "nitroglycerin") or the IR query can be constructed by duplicating the OR term for the single word (the constraint is then word recurrence which still has value for filtering).

6. Discussion

Our system demonstrates the feasibility of speech access to an information retrieval system in spite of the large vocabulary requirements of the task. Although the system employs a fairly basic phonetic recognizer, it is able to locate articles relevant to a multi-word query even in cases where none of the words of the query are ranked topmost. The applicability of semantic co-occurrence filtering is not limited to phonetically oriented speech recognition. The technique could be used with any recognizer that can produce rank ordered word hypotheses.

There are many opportunities for further development of the system both in terms of performance improvement and extensions to the interface and functionality.

An improvement in recognition accuracy is expected by employing context dependent phone models and error matrices. Likewise, tied Gaussian mixture output distributions generally provide better recognition accuracy than the single Gaussian distributions we are currently using [9]. We also anticipate moving from speaker dependent recognition to a speaker adaptive mode which will require far less training data for new speakers.

Concerning the interface, the necessity to speak uninflected forms is awkward. For example, a query about the film "Gone With The Wind" had to be stated as "Go Wind". As described in Section 2.3, this can obviated by including inflected phonetic spellings in the phonetic dictionary. If the recognizer were adapted to recognize a set of command words the system would gain considerable flexibility as aspects of search and presentation could be directed by the user, particularly user feedback based on the titles shown on the display screen. The small number of words involved in the display of the titles constitutes a strong constraint on their recognition.

Ideally, we would like to extend the system to handle continuous speech and identify function words. In this regard, integration with the MURAX system [10] would be an interesting development path.

7. Acknowledgments

We would like to thank our colleagues at Xerox PARC for their support and assistance, particularly Marcia Bush, Jan Pedersen and Doug Cutting. The HMM structure for word matching resulted from a conversation with David Haussler. This work was partially funded by National Science Foundation grant IRI-8719595.

References

1. Glavitsch, U. and Schäuble P., "A System for Retrieving Speech Documents", *Proceedings of the Fifteenth International ACM SIGIR Conference on Research and Development in Information Retrieval*, June 1992, pp. 168-176.
2. Glavitsch, U. and Schäuble P., "Assessing the Retrieval Effectiveness of a Speech Retrieval System by Simulating Recognition Errors", *These Proceedings*.
3. Cutting, D. R., Pedersen, J. Halvorsen P.-K., "An Object-Oriented Architecture for Text Retrieval", *Conference Proceedings of RIAO'91, Intelligent Text and Image Handling*, Barcelona, Spain, April 1991, pp. 285-298.
4. *The Academic American Encyclopedia*, Grolier Electronic Publishing, Danbury, Connecticut, 1990.
5. Lamel, L., Kassel, R. and Seneff, S., "Speech database development: Design and analysis of the acoustic-phonetic corpus.", *Proceedings, DARPA Speech Recognition Workshop*, Rpt. No. SAIC-86/1546, 1986, pp. I-61-68.
6. *Moby Pronunciator II*, Grady Ward, Arcata CA, 1993.
7. Lucassen, J. M. and Mercer, R. L., "An Information Theoretic Approach to the automatic Determination of Phonetic Baseforms", *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 1984, pp. 42.5.1-4.
8. Dedina, M. J. and Nusbaum, H. C., "PRONOUNCE: A Program for Pronunciation by Analogy", *Computer Speech and Language*, vol. 5, 1991, pp. 55-64.
9. Rabiner, L. R. and Juang, B. H., *Fundamentals of Speech Recognition*, Prentice-Hall, Englewood Cliffs, New Jersey, 1992.
10. Kupiec, J. M., "MURAX: A Robust Linguistic Approach for Question-Answering Using an On-Line Encyclopedia", *Proceedings of the Sixteenth International ACM SIGIR Conference on Research and Development in Information Retrieval*, June, 1993, pp. 181-190.