ELECTRONIC WORKSHOPS IN COMPUTING

Series edited by Professor C.J. van Rijsbergen

Ian Ruthven (Ed)

Miro'95

Proceedings of the Final Workshop on Multimedia Information Retrieval
(Miro '95)
Glasgow, Scotland
18-20 September 1995

Paper:

Speech Retrieval Based on Automatic Indexing

Martin Wechsler and Peter Schäuble

Published in collaboration with the
British Computer Society

©Copyright in this paper belongs to the author(s)

Speech Retrieval Based on Automatic Indexing

Martin Wechsler, Peter Schäuble
Swiss Federal Institute of Technology (ETH)
Zürich, Switzerland

Abstract

We present a system that retrieves audio recordings containing spoken text in response to a given textual query. In particular, we describe indexing methods that automatically describe the content of the recordings. The indexing methods, which are based on phoneme recognition output, take account of speech recognition errors. Additionally, the indexing methods we present are suitable for a language where many word inflections and compounds may occur. To compare different indexing methods, we have evaluated the retrieval effectiveness on a test collection of 1289 documents and 26 queries. The results show that better effectiveness can be achieved when taking into account the characteristics of the underlying speech recognition system.

1 Introduction

A *speech retrieval system* accepts vague queries and it performs best-match searches to find speech recordings that are likely to be relevant to the queries. Efficient best-match searches require that the speech recordings are indexed in a previous step. We focus on effective *automatic* indexing methods that are based on automatic speech recognition.

Automatic indexing of speech recordings is a difficult task for several reasons. One main reason is the limited size of vocabularies of speech recognition systems, which are at least one order of magnitude smaller than the indexing vocabularies of text retrieval systems. Another main problem is the deterioration of the retrieval effectiveness due to speech recognition errors that invariably occur when speech recordings are converted into sequences of language units (e.g. words or phonemes).

We present the design and evaluation of several automatic indexing methods for a prototype speech retrieval system for German speech recordings. Recognition of German speech is challenging because very little training data is available. The German language also encompasses a large number of different word inflections and compounds, which makes the retrieval task more difficult.

Other approaches to speech retrieval are focusing on *English* speech recordings. English speech retrieval is less difficult than German speech retrieval because there are fewer word inflections and in particular there exists a large amount of training data [Lamel et al., 1986],[Garofolo et al., 1990],[LDC, 1992]. At Cambridge University, a video mail retrieval system is being developed that currently accepts 35 query words [Sparck-Jones et al., 1995]. Similar work was also done by David James [1995] who experimented on retrieval of English news broadcasts. At Carnegie Mellon University, a retrieval system is being developed for a digital video library [Hauptmann et al., 1995], [Informedia, 1996].

As our main contribution, we present a family of speech retrieval methods suitable for a language with many different word inflections and for cases where limited training data is available.

This paper is structured as follows. In Section 2, we describe methods to index and retrieve speech documents. An overview of the current prototype retrieval system and the speech recognition component is given in Section 3. In Section 4, we report on experiments made to determine the retrieval effectiveness and to compare the indexing methods. Conclusions are drawn in Section 5.

2 Indexing Methods for Speech Documents

Indexing is the process of generating document descriptions that contain clues about the content of the documents. A document description consists of (possibly weighted) indexing features that have been identified in the document. In the case of text retrieval, words and phrases are the most common indexing features.

2.1 Speech Recognition Component

Indexing speech documents automatically requires speech recognition technology. Ideally, a word recognition system would transcribe the spoken document into text, such that text retrieval methods could be applied. However, this is not feasible for the following reasons:

- There exist too many different words for recognition. For instance, the TREC document collection contains more than 500,000 different English word stems [Harman, 1994], whereas the vocabulary size of current state-of-the-art recognition systems is about 65,000 word forms [Woodland et al., 1995]. In the case of the German language the situation is even worse because of the vast number of different word inflections.
- Infrequent words like person or company names should be in the recognition vocabulary because they are excellent indexing features. However, they are not known at indexing time.
- In the case of German, we lack sufficient data to train the acoustic word models used for recognition.

Therefore we use *phonemes* as the basic units in recognition. Using the Hidden Markov Toolkit developed by Young et al. [1993], we have built a phoneme recogniser, which produces phonemic transcriptions of the speech documents (Figure 1). This intermediate representation of the speech is a suitable basis for indexing, since it is independent of the

<i>Text</i>	In Algerien	hat die Islamische	Heilsfront	FIS	die Bevoelkerung ...
<i>correct</i>	i n a l g e r i @ n h a t d i i s l a m i s c h @ h e i l s f r O n t s i l E f i E s d i b e f o e l k E r u N ...				
<i>recognized</i>	i m a E d i E i @ n h a t s t i i s l a m i s c h e h e i l c h s v O n t s i l E f i E s d e b E s c h e i r E k r u N ...				

Figure 1: Sample of a radio news sentence: text, correct transcription and phoneme recognition output

indexing vocabulary. That is, the transcriptions allow the spotting of *any* query word, if its phonemic transcription is available. Furthermore, modifications in indexing methods are possible without expendable changes in the recognition system. Our phoneme recogniser is described more thoroughly in Section 3.1.

There are two major differences between speech and text retrieval, as can be noticed in Figure 1. First, word boundaries are difficult to detect, since there is no explicit word delimiter in speech. Second, *recognition errors* veil the true content of the speech documents, which makes the retrieval task more difficult and thus requires appropriate indexing methods.

2.2 Indexing and Retrieval

We first give a formal description of an indexing and retrieval method based on the vector space model. Let D be a collection of speech documents and q a user query. An *indexing method* is a function that maps a document $d_j \in D$ (query q) into a document description vector $\vec{d}_j$ (query description vector $\vec{q}$). The dimension of $\vec{d}_j$ and $\vec{q}$ is given by the number of different *indexing features* in the *indexing vocabulary* $\Phi = \{\varphi_0, \dots, \varphi_{m-1}\}$. We will introduce various indexing vocabularies later in this section. The description vectors are defined as

$$\vec{d}_j := (a_{0j}, \dots, a_{m-1j})^T \quad (1)$$

$$\vec{q} := (b_0, \dots, b_{m-1})^T \quad (2)$$

with *weights* a_{ij} (b_i) that represent the relevance of the indexing feature φ_i in the document d_j (query q).

To generate a list of documents that are ranked in decreasing order of their estimated relevance to the query, the system computes the Retrieval Status Value (RSV) between q and every document d_j using a standard cosine measure [van Rijsbergen, 1979, p.41]

$$RSV(q, d_j) = \frac{\vec{q}^T \vec{d}_j}{\sqrt{\vec{q}^T \vec{q}} \sqrt{\vec{d}_j^T \vec{d}_j}}.$$

In the following sections we describe two indexing methods for speech documents in more detail. Both indexing methods refer to the general definition of (1) and (2). The input to the indexing methods consists of phonemic transcriptions.

2.3 N-Gram Indexing

In the first indexing method, the set of indexing features consists of phoneme N-grams, for a given N . The technique of using N-grams for indexing is well known in searching tasks, e.g. letter based N-grams in text retrieval [Teufel, 1989],[Cavarnar, 1992]. The phonemic transcription of a speech document d_j is simply decomposed into overlapping phoneme N-grams. The example in Figure 1 would yield the indexing features $i_m_a, m_a_E, \dots$ ($n = 3$). The weight of such an indexing feature φ_i in the document d_j is defined as

$$a_{ij} := \text{ff}(\varphi_i, d_j) \text{idf}(\varphi_i), \quad (3)$$

where $\text{ff}(\varphi_i, d_j)$ denotes the feature frequency (the number of occurrences of φ_i in d_j), and $\text{idf}(\varphi_i)$ denotes the inverse document frequency

$$\text{idf}(\varphi_i) := \log \left(\frac{1+|D|}{1+\text{df}(\varphi_i)} \right),$$

which is function of the document frequency $\text{df}(\varphi_i)$, that is the number of documents containing φ_i .

A query—in our case entered as natural language text—is first transcribed into a phonemic transcription using a pronunciation dictionary. Thereafter, it is also decomposed into N-grams, and the query description $\bar{q}$ is derived according to (2), using the weights

$$b_i := \text{ff}(\varphi_i, q) \text{idf}(\varphi_i). \quad (4)$$

This simple method is suitable for indexing speech documents for the following reasons. First, it accounts for different word inflections and compounds that are common in German. Although the query and the document may contain different inflections of the same word, it is likely that there are matching N-grams corresponding to the common stem of the word. Second, this method is tolerant to recognition errors, if N is not too large. Any sequence of N correctly recognised phonemes yields a correct N-gram.

2.4 Word Matching and Probabilistic Weighting

The second indexing method identifies possible occurrences of query words in the phonemic transcriptions, i.e. the query words are the indexing features. The probability of occurrence serves to calculate the weights in the document description vectors (1). A similar method was applied in Mittendorf et al. [1995], where retrieval is performed on corrupted text obtained by Optical Character Recognition (OCR).

Let $w = (w[1] \dots w[I])$ be a phonemic transcription of a query word φ_i and $t = (t[1] \dots t[R])$ a phonemic transcription of a document d_j . The indexing system first identifies a set of subsequences $\{s_0, \dots, s_{r-1}\}$ of t that are similar to the query transcription w . Let p_k denote the probability that a phonemic sequence s_k corresponds to the query word φ_i (We explain the estimation of p_k later in this section). Then, we obtain the expected feature frequency of φ_i in d_j by

$$\text{eff}(\varphi_i, d_j) = \sum_{k=0}^{r-1} p_k. \quad (5)$$

To determine the set of subsequences $\{s_0, \dots, s_{r-1}\}$, we consider all subsequences in t that have length $J \in [I - \Delta, I + \Delta]$. Setting $J := I$ would be too restrictive because of phoneme insertion and deletion errors. From the candidates, we select those r subsequences with maximal occurrence probability, such that the subsequences do not overlap. We are aware that considering all subsequences, which is of time complexity $O(R \Delta I)$, may not be feasible for large collections. We are currently investigating various data structures and methods for faster approximate matching.

The estimation of an occurrence probability is mainly based on an *edit distance* computed between the phonemic transcription w and a subsequence s . The edit distance

$$d(w, s) := \delta(I, J)$$

is a measure for the dissimilarity of w and s . We define $\delta(i, j)$ recursively as follows.

$$\delta(0, 0) := 0$$

$$\delta(i, j) := \begin{cases} \infty & (i < 0) \vee (j < 0) \\ \min \begin{cases} \delta(i-1, j) + C_{\text{Del}}(w[i]) \\ \delta(i, j-1) + C_{\text{Ins}}(s[j]) \\ \delta(i-1, j-1) + C_{\text{Sub}}(w[i], s[j]) \end{cases} & (i > 0) \wedge (j \geq 0) \vee (i = 0) \wedge (j > 0) \end{cases} \quad (6)$$

The different cases in (6) refer to the different types of errors occurring in phoneme recognition. $C_{\text{Del}}(ph)$, $C_{\text{Ins}}(ph)$ and $C_{\text{Sub}}(ph, ph')$ are cost functions representing penalties in the case of phoneme insertion, deletion or substitution. We defined those cost functions by analysing the errors of our phoneme recogniser on the training speech.

To achieve good estimates for the occurrence probabilities, we trained a probability estimation function that is based on several parameters. The probability p , that a given phonemic transcription s corresponds to w is estimated according to the formula

$$\log \left(\frac{p}{1-p} \right) = \alpha_0 + \alpha_1 I + \alpha_2 e + \alpha_3 d(w, s), \quad (7)$$

where I is the length of w , e is the number of equal phonemes in w and s , and $d(w, s)$ is the edit distance. We computed I , e and $d(w, s)$ for a set of 355 words and several substrings s on a small training set where p was set to either zero or one by manual checking. Having a set of approximately 11'000 data points, of which 358 (3%) are positive examples, we then applied logistic regression to estimate the coefficients $\alpha_0, \dots, \alpha_3$ [Hosmer & Lemeshow, 1989].

2.4.1 Weighting

Given a query q , the stop words were first removed. For the remaining words, the expected feature frequencies were calculated according to (5). Finally, the following weights were used in (1) and (2):

$$a_{ij} := \text{eff}(\varphi_i, d_j) \text{idf}(\varphi_i) \quad (8)$$

$$b_i := \text{ff}(\varphi_i, q) \text{idf}(\varphi_i). \quad (9)$$

Note that it is not possible to calculate the idf-values exactly in this approach, since the document frequency is not known. We used approximative idf-values that were derived from a text collection of a similar domain (articles of the Swiss News Agency SDA).

3 Prototype System Description

Our prototype speech retrieval system provides access to four hours of Swiss radio news, spoken by a single speaker. The language is high German with a Swiss flavour. Each news unit has a time duration of approximately 5 minutes, including various topics such as special events, politics, sports, business information and weather. Because topic boundaries are difficult to detect, the units have been segmented into fixed length passages (speech documents) of 20 seconds with an overlapping distance of 10 seconds. This segmentation yields a collection size of 1289 speech documents. Every speech document is an audio recording sampled at 16kHz with 16 bit resolution.

Queries are entered as natural language text. The system maps the input text into a phonemic transcription using a pronunciation dictionary with more than 350'000 entries [Celex, 1993]. Finally, the query is indexed as described in Section 2.

Figure 2 shows a screen dump of the current prototype. After query entering, the system returns a list of documents ranked in decreasing order of their RSV. For each document, date, time, and a passage number are shown. The user may select a rank to listen to a specific document.

3.1 Speech Recognition

We developed a speaker dependent phone recogniser using the HTK Toolkit [Young et al., 1993], which is based on continuous density Hidden Markov Models. We trained 52 different acoustic monophone models including a silence model.

We had to generate training material for the recogniser, since we lacked a training set that is in German, speaker specific and phonetically labelled. We collected 30 minutes of radio news together with the textual and phonetic transcripts. Fortunately, we were able to use the *English* TIMIT speech corpus to initialise the acoustic models for our

Figure 2: Speech Retrieval System Prototype. The query means: *Negotiations with the European Community*. The system is playing the top ranked document. In this example, all the important query words are uttered.

German speaker. Using these models, we performed Viterbi alignment on the data to determine the phone boundaries [Young et al., 1993], which are necessary to train the acoustic models.

For recognition, we applied Viterbi decoding, which generates the most likely phone sequence to given input speech. Additionally, a stochastic phone bigram language model was incorporated to avoid the output of unlikely phone sequences. The sequences were further reduced by clustering some of the most similar monophones into 32 phoneme classes.

A preliminary evaluation showed that 72% of the phonemes are recognised correctly. This is only an upper bound for the performance, since evaluation was done on the training set. To build an evaluation set, additional speech would have to be collected, transcribed and labelled.

4 Experiments

In this Section we report on retrieval experiments to evaluate the system's effectiveness and to compare different indexing methods. We set up an IR test collection, consisting of 1289 speech documents (see Section 3) and 26 short text queries that are message titles gathered from the Swiss News Agency. Each query contains eight words on average. An example of such a query is *Kroaten stimmten über Unabhängigkeit ab*¹. To obtain relevance assessments, we let a student read the queries and listen to all the documents.

In the first retrieval experiment we compared the N-gram indexing method described in Section 2.3 for various N . Figure 3 shows the recall-precision curves for different N-gram methods. Evidently, the trigram and tetragram method perform much better compared to bigrams. Using the trigram method, a 225% improvement of average precision can be observed. Bigrams are too short and too common units for indexing purposes. They are not able to discriminate the documents sufficiently. On the other hand, tetragrams would contain much evidence about the underlying text. However, a 3% degradation of average precision compared to trigrams indicates that in the context of speech retrieval, too long units are not useful as well. This is due to phoneme recognition errors that prevent a document feature from being matched to a query feature. Apparently, trigrams seem to be a suitable compromise between indexing power and recognition errors.

The second retrieval experiment contains a comparison between the word matching and the trigram method. For

¹The Croats voted for independence.

Figure 3: Recall-precision curves for N-gram indexing methods

the word matching method, we set $r = 3$, i.e. we assume that a query word does not occur more than three times in a document. This is a reasonable assumption for documents of 20 seconds. The graph on the left of Figure 4 shows the

Figure 4: Recall-precision curves for trigram and word matching method for different weighting schemes

results using the weighting scheme defined in Section 2.3. The word matching method achieves an average precision of 0.2008, that is a 32% improvement over the trigram method.

In the other graph of Figure 4, the experiment was repeated without using idf-weights in (3),(4),(8) and (9), respectively. Here, the performance differences are even more significant. Using word matching resulted in an 42% improvement of average precision over trigram indexing. The comparison of both graphs confirms once more that the idf-weights help to improve the retrieval effectiveness. The improvements in average precision are 7% (word matching) and 15% (trigram), respectively. The more significant performance gain in the trigram method can be explained by the presence of stopwords. In N-gram indexing, we did not remove stopwords with the intention that phrases (e.g. *in Italien*) might produce additional matches. N-grams occurring in stopwords tend to have a low idf-value. Thus, incorporating idf into weighting naturally degrades the relevance of stopword N-grams.

A clear advantage of the word matching method is that it considers more of the context in the phonemic transcriptions when matching a word. A trigram bears much less context, and no information about neighbouring trigrams is used,

because the retrieval is based on the vector space model where the indexing features are assumed to be independent of each other.

On the other hand, the word matching method is computationally more expensive due to the calculation of the expected feature frequencies. However, a phoneme trigram index could be used to restrict the search space when looking for a set of possible subsequences.

5 Conclusions

In this paper we have described different indexing methods for a system that retrieves speech documents in response to given textual queries. As a preprocessing step in indexing the speech documents, we use a phoneme recognition system which produces phonemic transcriptions of the speech. Unlike other approaches that work with a fixed recognition vocabulary, we gain independence between the recognition and the indexing process. Particularly, the recognition system does not impose any limitations to the query vocabulary.

Based on phonemic transcriptions, we have described two different indexing methods. A N-gram method and a probabilistic word matching method. Both methods account for the high number of different word inflections and compounds in the German language. For example, the query word *Europa*² matches to documents containing *europäischen*³. Similarly, for a compound like *Fussballweltmeisterschaft*⁴, the system can find matches to the base parts *Fussball*, *Welt* and *Meisterschaft*.

Although less than 72% of the phonemes are recognised correctly, it is still possible to find useful information, because the indexing methods account for recognition errors to a certain extent. The N-gram method does this by regarding a limited context, whereas the word matching method estimates occurrence probabilities by using explicit information about common recognition errors.

We have evaluated the speech retrieval system and both indexing methods on a IR test collection. Using the best indexing method, 24% of the first five documents are relevant on average. The probabilistic word matching method performs better than the N-gram method, because it uses more phoneme context while searching.

Acknowledgement

We would like to thank Thomas Nauer for his work in setting up an speech test collection and a corresponding evaluation environment.

References

- [Cavnar, 1992] Cavnar, W. (1992). N-Gram-Based Text Filtering for TREC-2. In *TREC-2 Proceedings*.
- [Celex, 1993] Celex (1993). *The Celex Lexical Database (CD-ROM)*. Linguistic Data Consortium, ldc@unagi.cis.upenn.edu, Williams Hall, University of Pennsylvania, Philadelphia, USA.
- [Garofolo et al., 1990] Garofolo, J. S., Lamel, L., & Fisher, W. (1990). *DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus CD-ROM*. U.S. Department of Commerce, Gaithersburg, MD 20899.
- [Harman, 1994] Harman, D. (1994). Overview of the Second Text REtrieval Conference (TREC-2). In *TREC-2 Proceedings*, pp. 1–20.
- [Hauptmann et al., 1995] Hauptmann, A., Witbrock, M., & Christel, M. (1995). News-on-Demand: An Application of Informedia Technology. September. <http://www.cnri.reston.va.us/home/dlib.html>.
- [Hosmer & Lemeshow, 1989] Hosmer, D. W., & Lemeshow, S. (1989). *Applied Logistic Regression*. John Wiley & Sons, New York.
- [Informedia, 1996] Informedia (1996). <http://www.informedia.cs.cmu.edu/>.

²Europe

³European

⁴football world cup

- [James, 1995] James, D. (1995). *The Application of Classical Information Retrieval to Techniques to Spoken Documents*. PhD thesis, University of Cambridge, UK.
- [Lamel et al., 1986] Lamel, L. F., Kassel, R. H., & Seneff, S. (February, 1986). Speech Database Development: Design and Analysis of the Acoustic-Phonetic Corpus. In *DARPA Speech Recognition Workshop*, pp. 100–109.
- [LDC, 1992] LDC (1992). Linguistic Data Consortium, Williams Hall, University of Pennsylvania, Philadelphia, USA, ldc@unagi.cis.upenn.edu.
- [Mittendorf et al., 1995] Mittendorf, E., Schäuble, P., & Sheridan, P. (1995). Applying Probabilistic Term Weighting to OCR Text in the Case of a Large Alphabetic Library Catalogue. In *ACM SIGIR Conference on R&D in Information Retrieval*, pp. 328–335.
- [Sparck-Jones et al., 1995] Sparck-Jones, K., Foote, J., Jones, G., & Young, S. (1995). Spoken Document Retrieval - A Multimedia Tool. In *Symposium on Document Analysis and Information Retrieval*, pp. 1–10, Las Vegas.
- [Teufel, 1989] Teufel, B. (1989). *Informationsspuren zum numerischen und graphischen Vergleich von reduzierten natürlichsprachlichen Texten*. PhD thesis, Swiss Federal Institute of Technology. VdF-Verlag, Zürich.
- [van Rijsbergen, 1979] van Rijsbergen, C. J. (1979). *Information Retrieval*. Butterworths, London, second edition.
- [Woodland et al., 1995] Woodland, P., Leggetter, C., Odell, J., Valtchev, V., & Young, S. (1995). The 1994 HTK Large Vocabulary Speech Recognition System. In *International Conference on Acoustics, Speech, and Signal Processing*, volume I, pp. 73–76.
- [Young et al., 1993] Young, S., Woodland, P., & Byrne, W. (1993). *HTK Version 1.5: User, Reference & Programmer Manual*. Entropic Cambridge Research Laboratory, Sheraton House, Castle Park, Cambridge CB3 0AX, England.