Information Retrieval, 3, 173–188, 2000

New Approaches to Spoken Document Retrieval

MARTIN WECHSLER
McKinsey & Company, Switzerland

EUGEN MUNTEANU AND PETER SCHÄUBLE
Eurospider Information Technology A6, Zurich, Switzerland

Received August 30, 1999; Revised August 30, 1999; Accepted June 12, 2000

Abstract. This paper presents four novel techniques for open-vocabulary spoken document retrieval: a method to detect slots that possibly contain a query feature; a method to estimate occurrence probabilities; a technique that we call collection-wide probability re-estimation and a weighting scheme which takes advantage of the fact that long query features are detected more reliably. These four techniques have been evaluated using the TREC-6 spoken document retrieval test collection to determine the improvements in retrieval effectiveness with respect to a baseline retrieval method. Results show that the retrieval effectiveness can be improved considerably despite the large number of speech recognition errors.

Keywords: spoken document retrieval, speech recognition, retrieval effectiveness

1. Introduction

Finding relevant information in *spoken documents* is a challenging task for modern multimedia information systems (Schäuble 1997). In the last decade there has been increasing interest in the development of systems that provide content-based access to spoken information such as radio, TV or video material (Glavitsch and Schäuble 1992). This interest has even increased since the initial *spoken document retrieval* (SDR) track within the TREC-6 conference (Voorhees et al. 1998).

Speech recognition and *information retrieval* techniques enable automatic content-based indexing and efficient retrieval of spoken documents that are relevant to a user's query. To approximate the quality of state-of-the-art text retrieval systems when dealing with information in spoken form, we have to address mainly the following problem.

The main problem when applying speech recognition for SDR is the accuracy of the recognition output. Automatic speech recognition is a difficult task and accordingly, its output often contains a considerable number of *recognition errors*. The recognition accuracy is mainly dependent on (1) the amount and quality of acoustic *training data*, (2) the number and gender of different speakers, (3) the number of units to recognize, and (4) the recording environment of the speech documents. Moreover, there are no acoustic pauses between words in continuous speech as opposed to blanks in texts.

Recognition errors usually degrade the effectiveness of a SDR system. Strategies against this problem are (1) to improve the speech recognition accuracy, which requires a huge

amount of training data and time, and/or (2) to develop retrieval methods that are more error-tolerant. In this paper we focus on the second strategy.

Researchers have addressed the problem of SDR in mainly two different ways. One way is to utilize a (*large*) *vocabulary speech recognizer* to convert the speech into text, to which well-established text retrieval methods can be applied. This approach is taken by several groups (Jones et al. 1995, Wactlar et al. 1996, Allan et al. 1998, Abberley et al. 1998). Note that in the TREC-6 SDR track *word-level* transcripts of the spoken documents were provided to enable a participation without having to cope with speech recognition. A considerable drawback of this approach is the fact that the query vocabulary, which is implicitly defined by the recognition vocabulary, is (a) *limited in size* and (b) has to be *specified and trained in advance*.

An alternative approach is to perform retrieval on *phoneme-level* transcriptions provided by a phoneme recognizer. Phoneme-recognition based SDR has the advantages that (a) the recognizer is less expensive with respect to the training effort, and (b) *open-vocabulary* retrieval is possible, because the recognition component is not bound to any vocabulary. Indexing the transcriptions may be accomplished e.g. by extracting phoneme N-grams (Wechsler and Schäuble 1995, Ng and Zue 1997). Alternatively, Brown and colleagues (1996) present a *word spotting* technique that operates on phone-lattices, which are multi-hypotheses phonetic transcriptions.

Finally, both word and phoneme recognition based SDR has been investigated in combination (James 1996, Jones et al. 1996, Witbrock and Hauptmann 1997). Results indicate that combined methods outperform either single approach, however they require larger recognition effort.

Previous work done at ETH describes a phoneme-recognition based retrieval method that combines error-tolerant *word spotting* and a new probabilistic weighting technique for retrieval (Wechsler and Schäuble 1995, Sheridan et al. 1997).

The main contribution of this work is the presentation and evaluation of four novel techniques to improve the effectiveness of phoneme-based spoken document retrieval, thus enabling open-vocabulary retrieval. These techniques consist of a new method to detect occurrences of query features, a new method to estimate occurrence probabilities, a *collection-wide probability re-estimation technique*, and feature length weighting.

The paper is structured as follows. In Section 2 we describe a baseline method for spoken document retrieval and present the four techniques to improve the retrieval effectiveness. Section 3 contains contextual information about experiments we performed to evaluate the techniques. Experimental results are presented and discussed in Section 4. We conclude our findings in Section 5.

2. Retrieval methods

Our approach to SDR is based on a *phoneme recognizer* which initially transforms the spoken documents into *phoneme sequences*. On the query side, a pronunciation dictionary (CMU 1995) serves to translate written query words or phrases into phoneme sequences. We call these query elements *query features*. In the case where a query word is not in the dictionary, we use rules to generate the corresponding phoneme sequence (Wasser 1985).

These phoneme sequences are *spotted* in the document sequences, and by taking into account the distribution of the query features in the documents, a *Retrieval Status Value* (RSV) is computed for each query-document pair. The RSV is a measure for the estimated relevance of a document with respect to a query. The spoken documents are presented to the user in decreasing order of their RSVs.

A phoneme sequence in a spoken document is comparable to a sequence of characters in a text document. However, there are two main differences to the text case. First, phoneme sequences do not contain word boundaries since we often do not pause between words in fluent speech. Thus our retrieval method must be able to *locate* individual occurrences of query features. We call those occurrences *slots*. Second, the sequences are corrupted due to recognition errors. Thus, an effective retrieval method has to take these errors into account.

Our retrieval method consists of three components: (1) a *slot detection* method, which detects possible occurences of query features in the documents, (2) a *probability estimation* method, which estimates the probability that a slot is an utterance of a query feature, and (3) a *weighting and retrieval function* which estimates how well the content of a document fits to the query content. In the following subsection, we describe our baseline retrieval method and present our novel techniques.

2.1. Baseline retrieval method

Let $d_j \in D$ be a spoken document in a collection D and let $\varphi_i \in q$ be an indexing feature of a query q. The documents and query features are assumed to be phoneme sequences. We write

$$d_j = \langle d_j[0], \ldots, d_j[l_{d_j} - 1]\rangle \tag{1}$$

$$\varphi_i = \langle \varphi_i[0], \ldots, \varphi_i[l_{\varphi_i} - 1]\rangle, \tag{2}$$

where $d_j[k]$ and $\varphi_i[k]$ denote the $(k+1)$-th phoneme within the sequence. Furthermore, l_{d_j} and l_{φ_i} denote the length of the document or query feature, respectively. The lengths are expressed in number of phonemes. For each query feature we first have to detect possible slots within the documents of the collection. On the phoneme level, a slot

$$s = \langle d_j[b], \ldots, d_j[b + l - 1]\rangle \tag{3}$$

is a phoneme subsequence within a document d_j where b denotes the start position and l the length of the slot. All slots detected to a query feature φ_i in a document d_j yield a *slot set* $S(\varphi_i, d_j)$.

The baseline retrieval method employs a trivial slot detection technique. To a given phoneme sequence of a query feature the technique detects all slots that are *identical* matches. Clearly, this method is not robust against phoneme recognition errors, since every error may cause a query feature to be missed. However, Mittendorf (1998) shows that this simple baseline method is surprisingly effective under certain circumstances, for example if the documents are sufficiently long.

In the baseline method we assume that each slot detected is a query feature occurrence, whereas in our novel techniques an occurrence probability is estimated for each slot. In other words, all slot probabilities are set to one in the baseline method.

Document and query weights for retrieval are determined as follows. We employ the inner vector product as our basic retrieval function:

$$\mathrm{RSV}(q, d_j) := \sum_{\varphi_i \in q} a_{i,j} b_i, \tag{4}$$

where $a_{i,j}$ denotes the document weight of φ_i in d_j and b_i denotes the query weight of φ_i in q.

For weighting and retrieval we adapted the *lnu.ltn* retrieval method (Buckley et al. 1994) for speech. The document weights for the baseline method are defined as

$$a_{i,j} := \frac{1}{(1-\alpha)\bar{l} + \alpha l_{d_j}} \log(1 + \mathrm{eff}(\varphi_i, d_j)) \tag{5}$$

where l_{d_j} denotes the length of d_j, $\bar{l}$ denotes the average document length in the collection, and α is the slope (Singhal et al. 1996). We used $\alpha = 0.25$ throughout our experiments. There are three modifications compared to the standard *lnu* document weights. First, we use an *expected feature frequency* (eff), which denotes the number of expected occurrences of a feature in a document. Mittendorf and colleagues (1995) show that the eff can be written as the sum of slot probabilities:

$$\mathrm{eff}(\varphi_i, d_j) := \sum_{s \in S(\varphi_i, d_j)} P(\varphi_i, s). \tag{6}$$

The idea behind expected feature frequencies is to allow for the uncertainty concerning the presence of query features in spoken document retrieval. Slots with higher probabilities correspond to more reliable hits and should thus get higher weights. Note again that in our baseline method the expected feature frequency equals the number of slots detected.

The second modification concerns the document weight $\log(1 + \mathrm{eff}(\varphi_i, d_j))$ which was originally $(1 + \log(\mathrm{ff}(\varphi_i, d_j))$. This change was necessary to avoid negative weights in the case $0 \leq \mathrm{eff} < 1$. In the third modification we adjusted pivoted document normalization (Singhal et al. 1996) by defining the length of a spoken document with the number of phonemes recognized.

The query weights b_i in (4) are defined as

$$b_i := (1 + \log(\mathrm{ff}(\varphi_i, q))) \, \mathrm{iecf}(\varphi_i) \tag{7}$$

$$\mathrm{iecf}(\varphi_i) := 1 + \log\left(\frac{C_q + 1}{\mathrm{ecf}(\varphi) + 1}\right) \tag{8}$$

$$\mathrm{ecf}(\varphi_i) := \sum_{d_j \in D} \mathrm{eff}(\varphi_i, d_j) \tag{9}$$

$$C_q := \max_{\varphi \in q}(\mathrm{ecf}(\varphi)). \tag{10}$$

The *feature frequency* ff(φ_i, q) denotes the number of occurrences of φ_i, in the query, whereas the *inverse expected collection frequency* iecf(φ_i) is a collection-wide feature weight very similar to the *inverse document frequency* (idf). The iecf is defined in (8) as a function of the *expected collection frequency* ecf (φ_i), which denotes the expected number of all occurrences of a feature in the collection (9). The value C_q (10) is a query dependent constant defined in such a way that iecf(φ) ≥ 1 for all $\varphi \in q$. The iecf emphasizes query words that occur less frequently in the collection, where for the most frequent feature of the query it holds that iecf $= 1$.

In our model of probabilistic feature occurrences it is theoretically possible to compute the idf of a feature based on the expected *document* frequency as shown in Mittendorf et al. (1995). The document frequency denotes the number of documents that contain a certain feature. However, earlier experiments have shown that the estimation of expected document frequencies is not robust.

A text query is indexed by transcribing single words and phrases as phoneme sequences. Single words are transcribed by means of a pronunciation dictionary. For out-of-vocabulary words, we adapted a rule-based phone translation system (Wasser 1985). Additionally, pronunciations of consecutive pairs of non stop words are concatenated to phrase phoneme sequences. Thus, the baseline method also uses phrase features.

2.2. Novel retrieval method

The main problem in retrieval on phonemic output is the fact that the recognition result is corrupted by a considerable amount of recognition errors. State-of-the art phoneme recognizers still operate with phoneme error rates of at least 25% (Robinson 1994). The phoneme error rate is the percentage of phoneme substitutions, insertions and deletions with respect to the reference phoneme sequence. The recognizer we used for this paper has even a phoneme error rate of 55%. This quite poor performance is related to the heterogeneous nature of the speech data (various recording environments and multiple speakers), and to the fact that we did not spend much time for training. Figure 1 shows a sample extract of the recognition output with the corresponding text. By comparing the phoneme sequence of the query word “cigarette” (which is /sigBet/ for our phoneme alphabet) to the document phoneme sequence, one can see that each occurrence is corrupted by various recognition errors. There is no entire match between the query word and the three occurrences.

In the following, we describe new error-robust methods both for slot detection and probability estimation, and we investigate the feasibility of taking the length of query features into account.

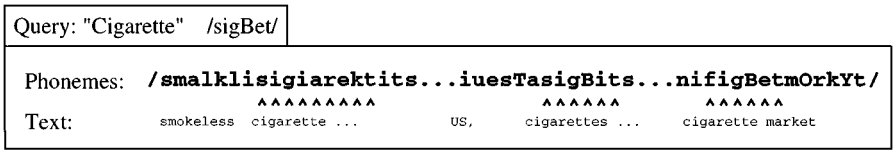

Figure 1. An extract of a document phoneme sequence, corrupted by recognition errors.

Slot detection. The new slot detection method accounts for all three types of phoneme errors (substitution, insertion, deletion). It can be divided into two steps. In a first step, each document position is scored according to its quality of being a slot beginning. In the second step each position is tested in decreasing order of its slot beginning quality. For positions satisfying certain selection criteria, a slot is established by detecting an appropriate slot end point. In what follows, we describe both steps in more detail.

Let φ be a query feature and d_j a phoneme sequence of a document. First, a *bin* is initialized for each phoneme position in the document such that $bin[k] = 0$ for $k = 0, \ldots, l_{d_j} - 1$. Subsequently, the bins are filled in such a way that $bin[k]$ contains the number of phonemes that are *identical* to the corresponding query feature phoneme, if k was a slot beginning, i.e.

$$bin[k] := |\{x | 0 \leq x \leq l_\varphi \wedge d_j[k + x] = \varphi[x]\}|. \tag{11}$$

Again, l_φ denotes the number of phonemes in φ. The value $bin[k]$ reflects a trivial slot beginning score for position k, and $bin[k] \leq l_\varphi$ holds. It is evident that any substitution error causes $bin[k]$ to decrease. In the next step, insertion and deletion errors are taken into account in the following way. For each position k another score $bs[k]$ (beginning score) is calculated by accumulating bins from a window around k. The beginning scores are defined as

$$bs[k] := \sum_{i=k-w_\varphi/2}^{k+w_\varphi/2} bin[i] \tag{12}$$

$$w_\varphi := 1 + 2 \cdot \begin{cases} 0 & l_\varphi < 5 \\ 1 & 5 \leq l_\varphi < 10 \\ 2 & l_\varphi \geq 10, \end{cases} \tag{13}$$

where w_φ denotes the window size, which we defined empirically as a function depending on the feature length (13). The value $\frac{w_\varphi}{2}$ in (12) reflects the maximum number of insertion or deletion errors taken into account. For example, if a slot starting at position k contains one insertion error, $bin[k + 1]$ should also contribute to the slot beginning score $bs[k]$. In this way, $bs[k]$ contains an approximation of the number of common phonemes between φ and a slot starting at position k.

In the second step, slots are established in a top-down manner. First, all positions within d_j are sorted in decreasing order of their *bs*-values. Then, starting with the best position, slots are established as long as their *bs*-values are greater than a threshold $\tau \cdot l_\varphi$ ($0 \leq \tau \leq 1$). The threshold is a lower bound for the number of common phonemes between a query feature and a valid slot. A good choice for the parameter τ is the ratio of correctly recognized phonemes, which may be determined on a recognition test set. This ratio is called *percent correct* (Lee 1989, p. 147). We used $\tau = 0.5$ in our experiments. To establish a slot, its end is first detected by searching for matching phonemes in a window around the expected slot end, which is given by the slot beginning and by l_φ. To avoid multiple overlapping slots, the slot is only established if there is no overlap with previously established slots (for the same query feature). If all criteria are satisfied, the slot is added to the slot set $S(\varphi, d_j)$.

Two types of errors may occur during slot detection. A *miss* denotes a slot where φ is spoken but the slot is not detected. A *false alarm* denotes the detection of a slot where φ was not spoken. Both types of errors may affect the retrieval effectiveness and there is a trade-off between them. The goal is to minimize both the number of misses and the number of false alarms.

Probability estimation. In this section we describe a technique which estimates an occurrence probability for a slot based on the comparison of phoneme sequences. In previous work we experimented with probability estimation functions that either required training utterances (Wechsler and Schäuble 1995) or were determined empirically (Sheridan et al. 1997). In the following we present an estimation method that incorporates phoneme confusion statistics from the recognizer and makes the most of collection-wide information.

Let φ be a query feature and let s be a slot in d_j. We write

$$\varphi = \langle \varphi[0], \ldots, \varphi[l_\varphi - 1] \rangle$$
$$s = \langle s[0], \ldots, s[l-1] \rangle := \langle d_j[b], \ldots, d_j[b+l-1] \rangle.$$

The new probability estimation function first derives a string similarity between the slot and the query feature. The basic structure of the similarity function is based on the dynamic programming idea (Rabiner 1993, p. 223). We write s_u for the substring of the first u phonemes in s. Similarly, φ_v denotes the first v phonemes in φ. The similarity function is defined recursively as

$$\begin{aligned}
&\mathrm{sim}(s_1, \varphi_v) := t(\varphi[v] \rightarrow s[0]) \\
&\mathrm{sim}(s_u, \varphi_1) := t(\varphi[0] \rightarrow s[u]) \\
&\mathrm{sim}(s_u, \varphi_v) := \\
&\max \begin{cases} \mathrm{sim}(s_{u-1}, \varphi_{v-1}) + t(\varphi[v] \rightarrow s[u]) \\ \mathrm{sim}(s_{u-2}, \varphi_{v-1}) + t(\varphi[v] \rightarrow s[u-1]s[u]) \\ \mathrm{sim}(s_{u-1}, \varphi_{v-2}) + t(\varphi[v-1]\varphi[v] \rightarrow s[u]). \end{cases}
\end{aligned}$$

The function $t(\ \ )$ defines *elementary similarities*: $t(\varphi[v] \rightarrow s[u])$ denotes the similarity of the phonemes $\varphi[v]$ and $s[u]$, $t(\varphi[v] \rightarrow s[u-1]s[u])$ denotes the similarity of the phoneme $\varphi[v]$ and the phoneme string $s[u-1]s[u]$, and $t(\varphi[v-1]\varphi[v] \rightarrow s[u])$ denotes the similarity of the phoneme string $\varphi[v-1]\varphi[v]$ and $s[u]$. We estimate these elementary similarities based on *phoneme substitution*, *insertion and deletion probabilities* as follows:

$$\begin{aligned}
t(\varphi[v] \rightarrow s[u]) &= \mathrm{P}_{\mathrm{Sub}_{vu}} := \mathrm{P}_{\mathrm{Sub}}(\varphi[v] \rightarrow s[u]) \\
t(\varphi[v] \rightarrow s[u-1]s[u]) &:= \mathrm{P}_{\mathrm{Ins}}(s[u-1]) * \mathrm{P}_{\mathrm{Sub}_{vu}} \\
t(\varphi[v-1]\varphi[v] \rightarrow s[u]) &:= \mathrm{P}_{\mathrm{Del}}(\varphi[v-1]) * \mathrm{P}_{\mathrm{Sub}vu}
\end{aligned}$$

$P_{\mathrm{Sub}}, P_{\mathrm{Ins}}$ and P_{Del} constitute probabilities modeling the phoneme recognition process. $P_{\mathrm{Sub}}(p \rightarrow p')$ (abbreviated $\mathrm{P}_{\mathrm{Sub}_{pp'}}$) denotes the probability that the recognizer substitutes a phoneme p with p'. Similarly, $P_{\mathrm{Ins}}(p)(P_{\mathrm{Del}}(p))$ denote the probability that p is inserted

	phonemic	$P(s, \varphi)$
feature φ	/Wlimpik/	
slots s	/nlimpik/	0.867
	/WlimpOs/	0.702
	/bliwpip/	0.601
	/alinTisk/	0.506
	/apikpii/	0.443
	/Obiiwii/	0.334

Figure 2. Some slots and estimated probabilities to the query word *Olympic*.

(deleted) during recognition. These probabilities can be derived from a *confusion matrix*. A confusion matrix can be calculated by running the phoneme recognizer over training data and by aligning the phoneme output with the reference sequences. Based on the string similarity function, the final occurrence probability of a slot is estimated as

$$P(\varphi, s) := \frac{\text{sim}(s_{ls}, \varphi_{l\varphi})}{\text{sim}(\varphi_{l\varphi}, \varphi_{l\varphi})}. \tag{14}$$

Examples of occurrence probabilities for various slots are given in figure 2.

Collection-wide probability re-estimation. Running preliminary experiments with our new slot detection and probability estimation method, we noticed that a considerable portion of detected slots were false alarms. This is true particularly for short words. Figure 3 shows all slot probabilities of slots detected in a German speech collection for various query words. The slot probabilities for each word are displayed in decreasing order. For example, 9,000 slots are detected for the word (or subword) *zehn*, although its number of spoken

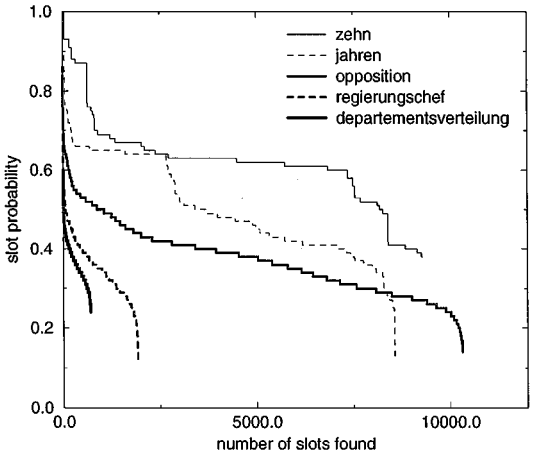

Figure 3. All slot probabilities in decreasing order for five German query words. The words were detected in a German speech collection.

occurrences in the collection is only 227, which means that 8,773 slots are false alarms. Apart from that, as can be seen in figure 3, there is a tendency that probability estimates of shorter words are higher. As a consequence for retrieval, the document weights (5) and the RSVs (4) are corrupted by many false alarm slots with high probabilities (6). In fact, we observed that the retrieval effectiveness dropped in this configuration.

To prevent this effect, it is necessary to estimate probabilities more accurately in order to emphasize hit slots and to eliminate false alarms. We propose a method which we call *collection-wide probability re-estimation*. The idea is to collect all slot probabilities detected for a single query feature in the *entire* collection (collection-wide), as illustrated in figure 3. Then, we select the top N slots and discard the rest, assuming that the rest are false alarms. In figure 3 this corresponds to a vertical line that acts as a threshold at position $x = N$. In other words, we focus on those slots that are most similar to the query feature. Subsequently, the probabilities of the top N slots are re-estimated as

$$P'(\varphi, s) := \begin{cases} \frac{P(\varphi,s)-P_N(\varphi)}{1-P_N(\varphi)} & P(\varphi, s) \geq P_N(\varphi) \\ 0 & \text{else,} \end{cases} \tag{15}$$

where $P_N(\varphi)$ denotes the N-th best probability for a feature φ from the string similarity based estimation (Section 2.2).

Note that this threshold is feature specific and thus more accurate compared to e.g. a constant threshold. In figure 3, a constant threshold corresponds to a horizontal line at a certain threshold probability. However, as can be seen easily, a constant threshold would tend to prefer shorter words from longer words. This effect would be undesired because longer words can be detected more reliably due to a larger phoneme context.

Feature length weighting. For the most part, indexing features can be detected reliably in text documents, be it short (e.g. "dog") or longer (e.g. "dependability"). This is not true for spoken documents. Longer words or phrases provide more information for the detection and recognition process and thus can be detected more reliably. Similar statements have been made by other research groups (Brown et al. 1996). This leads to the idea of incorporating the length of a particular indexing feature into the weighting and retrieval function. We propose a slightly extended definition of the query weights (7) as

$$b_i' := b_i * (l_{\varphi_i})^{\beta}, \tag{16}$$

where l_{φ_i} is the number of phonemes of the query feature and β is a tuning parameter. We will evaluate the effect of feature length weighting in Section 4.

3. Test settings

We have experimented on the methods described above using the test collection provided for the purposes of the Spoken Document Retrieval Track of the TREC-6 conference. The collection consists of 1451 documents from the 1996 Broadcast News Corpus (LDC 1996) representing approximately 50 hours of recorded material. A document contains 276 words

on average. More details about the collection can be found in (Voorhees et al. 1998). We used three different *versions* of the same collection, namely

– PRT: *p*honeme-level *r*ecognition *t*ranscripts generated by our own phoneme recognizer,
– SRT: word-level *s*peech *r*ecognition *t*ranscripts, provided by IBM's word recognition system, and
– LTT: manually entered *l*exical *t*ext *t*ranscripts, provided by NIST.

The SRT and LTT versions are both word-level transcripts. In order to evaluate our developments also on these versions, we translated these transcripts into phoneme sequences. For this purpose we adapted the Carnegie Mellon Pronouncing Dictionary (CMU 1995) to our phoneme set. Words not contained in this dictionary were transcribed using a rule-based text-to-phoneme converter (Wasser 1985). Thus, the final SRT and LTT *phoneme-level* transcripts can be interpreted as a collection with *low* (SRT) and *no* (LTT) phoneme corruption, respectively.

To come up with the PRT collection version, we built a *simple* speaker-independent phoneme recognizer based on Hidden Markov Models (Rabiner 1993) using the HTK Toolkit (Young et al. 1993). We trained acoustic models for a set of 40 monophones using the TIMIT speech corpus (Garofolo et al. 1990). Further, we built a set of context-dependent biphone models and trained this set on the SDR TREC-6 training collection (another 50 hours). For recognition, we used a stochastic phonebigram language model to avoid the output of unlikely phone sequences. The recognized phone sequences were further processed by clustering some of the acoustically most similar monophones into 30 broader classes, which we call *phonemes*. More details about our phoneme recognizer can be found in Mateev et al. (1998).

We evaluated the error rate of our phoneme recognizer on a 7.5 hour subset of the training collection and found an error rate of 54.72%. This rather poor recognition quality indicates that our PRT collection version consists of highly corrupted data and thus serves us as a suitable base for the evaluation of error-tolerant methods.

The query set consists of 49 topics. Each topic is expressed by 12 words on average (including stop words). The topics were processed by first discarding all stop words. Remaining words were then transcribed into individual phoneme sequences using the pronunciation dictionary or the rule-based text-to-phoneme converter described above. Additionally, a phrase feature was added for each pair of subsequent non-stop words by concatenating the phoneme sequences of the participant words.

The topics contain a considerable number of rare words such as geographical names ("Wilmington", "Israeli"), proper names ("Ridge", "Goldfinger") or other terms ("Unabomber", "Valujet"). Such words are crucial for retrieval, because they act as good discriminators between relevant and non-relevant documents.

The retrieval problem investigated here (and in the TREC-6 SDR track) is *known item search*. This kind of retrieval task simulates a user seeking a particular, partially-remembered document in the collection. We employ the same evaluation measures as used in the TREC-6 SDR track, namely

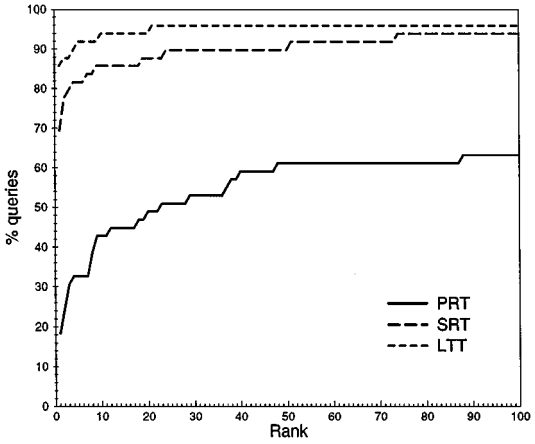

Figure 4. Cumulative percentage of topics that retrieve a known item by given rank; baseline method on PRT, SRT and LTT transcripts.

– mean reciprocal rank, which is (rank of the known item)$^{-1}$ averaged over all topics where the known item was found at all,
– %retr@1, the percentage of topics for which the known item was top ranked,
– cumulative percentage of topics that retrieve the known item by given rank (a 2D graph).

This enables the comparison of our results to submitted SDR runs of TREC-6. In the next section we present and discuss results achieved in a number of different experiments.

4. Results and discussion

In a first experiment we evaluated the retrieval effectiveness of our baseline retrieval method (Section 2.1). In figure 4 we present the results for the three different collection versions described in Section 3. The curve labelled with PRT reflects the high corruption of our phoneme recognizer output compared to the error-free collection (LTT). Table 1 shows the mean reciprocal rank and the percentage of retrieved documents at rank one for our baseline method. As expected, the baseline retrieval method does not perform well on highly corrupted phoneme output. This is due to the restrictive slot detection method, which is based on *exact* phoneme sequence matching (Section 2.1). However, the results obtained on SRT and LTT justify that our baseline weighting and retrieval functions. (4)–(10) are appropriate for documents with few or no recognition errors. Compared to the overall best runs submitted to TREC-6, we observed an increase in terms of mean reciprocal rank of 4.3% for SRT and 4.5% for LTT respectively.

In a second experiment we determined the retrieval effectiveness for our new slot detection and probability estimation methods (Sections 2.2–2.2). Table 2 shows the results of the comparison against the baseline method for the collections PRT and SRT. The parameter N denotes the number of selected slots for collection-wide probability re-estimation

(Section 2.2). All other parameters were left unchanged. The results show that our methods improve retrieval effectiveness by up to 63% in the case of highly corrupted data (PRT). An improvement was somehow expected since the new techniques are based on *error-tolerant* slot detection. However, the degree of the improvement is still noteworthy if considering that error-tolerant slot detection also detects many false alarms.

Varying the parameter N in Eq. (15), we allow more or less slots to be considered. This is equivalent with trading off recall and precision in the context of feature detection. Increasing N improves detection recall but lowers precision. As a consequence, more and more false alarms contribute to the RSVs by increased expected feature frequencies in non-relevant documents. This causes the retrieval effectiveness to drop. Moreover, our initial probability estimation method (Section 2.2) produces rather high probabilities even for false alarm slots, as can be seen e.g. for the word "zehn" in figure 3. This is especially true for short words, since there is less phoneme context that can be used for the estimation.

On the SRT collection we observe a decrease in retrieval effectiveness, which is however relatively small in the case of small N. Apparently the negative effect of additionally considered false alarms is stronger than the positive effect that word recognition errors, which are present in the SRT collection, may be compensated with phonemic matching.

In the third experiment we investigate the influence of feature length weighting (Section 2.2) on retrieval effectiveness. The length factor $(l_{\varphi_i})^{\beta}$ in Eq. (7) has the role to put more weight on longer features, since they are detected more reliably due to a larger phoneme context. Table 3 shows results when comparing this extension to our baseline method for some value of β. A small improvement was observed only for $\beta = 0.6$ on PRT. In all other cases we found a slight decrease in terms of mean reciprocal rank. However, this method was originally motivated by experiments performed on a *German* radio news collection, where we observed consistent improvements (Table 4). The explanation could lay in the fact that both the mean and the variation of the word length distribution are smaller for English compared to German.

Table 1. Retrieval effectiveness using baseline method on PRT, SRT and LTT.

Collection	PRT	SRT	LTT
Mean reciprocal rank	0.2617	0.7545	0.8797
%retr@1	18.36%	69.38%	85.71%

Table 2. Mean reciprocal rank for new slot detection, probability estimation and re-estimation.

Method	N	PRT	SRT
Baseline	–	0.2617	0.7545
New technique	100	0.4268 (+63%)	0.6940 (−8%)
	200	0.3985 (+52%)	0.6623 (−12%)
	400	0.3816 (+46%)	0.6562 (−13%)
	800	0.3742 (+43%)	0.6059 (−20%)
No re-estimation	∞	0.2025 (−23%)	0.3106 (−59%)

Table 3. Effect of feature length weighting for English documents in terms of mean reciprocal rank.

Method	β	PRT	SRT
Baseline	0	0.2617	0.7545
Feature length	0.6	0.2663 (+1.7%)	0.7500 (−0.6%)
Weighting	0.8	0.2531 (−3.3%)	0.7500 (−0.6%)
	1	0.2530 (−3.3%)	0.7331 (−2.8%)
	1.2	0.2547 (−2.7%)	0.7298 (−3.3%)
	1.4	0.2424 (−7.4%)	0.7283 (−3.5%)

Table 4. Effect of feature length weighting on a German collection in terms of mean reciprocal rank.

Method	β	Speech
Baseline	0	0.4068
Feature	0.4	0.4308 (+5.9%)
Length	0.6	0.4438 (+9.1%)
Weighting	0.8	0.4421 (+8.7%)
	1.0	0.4276 (+5.1%)
	1.2	0.4114 (+1.1%)

Table 5. Effectiveness of new techniques versus baseline method on PRT collection.

Method	Mean reciprocal rank	%retr@1
Baseline	0.2617	18.36%
New techniques	0.4268	38.78%
Gain	+63%	+111.22%

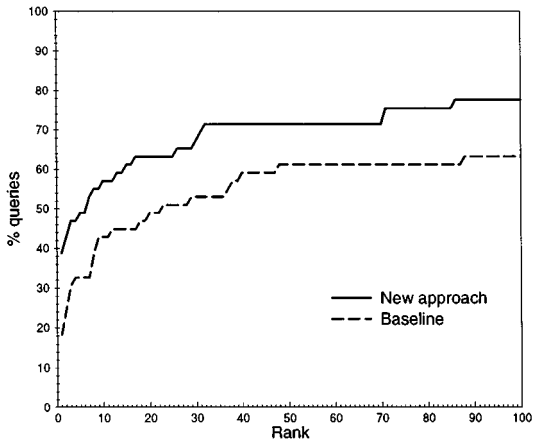

Figure 5. Effectiveness of new techniques versus baseline method on PRT collection.

Our goal was to improve the retrieval effectiveness for a retrieval method that operates on phoneme recognition output, since this approach has the main advantage of open-vocabulary querying. Thus, we finally compare the baseline method with the results achieved with the new techniques (without feature length weighting) in figure 5 and Table 5. Since the costs to inspect spoken documents are higher compared to text documents, we focus on the retrieval precision. Table 5 shows that the number of documents retrieved at rank one (%retr@1) increases by over 110% when applying our new retrieval method. This indicates that the new retrieval method is suitable for high-precision retrieval on corrupted documents.

5. Conclusions

In this paper we have presented new methods for *open-vocabulary* indexing and retrieval in spoken documents. Our techniques are based on spotting query features in phoneme sequences produced by a phoneme recognizer. Extensions include (1) a new slot detection method for highly corrupted phoneme sequences, (2) a probability estimation technique, (3) collection-wide probability re-estimation, and (4) feature length weighting.

Experiments on the TREC SDR Track collection show that the retrieval effectiveness can be improved considerably in the case of highly corrupted recognition output (55% phoneme error rate). This result has been verified with similar experiments on a German collection, where the method showed to be significantly more effective than phoneme-based N-gram retrieval (Wechsler 1998). Further, a simple variant of the method was shown to yield excellent results on phoneme-transcribed text with little or no corruption. Incorporating the length of query features into the weighting scheme seems to be beneficial for documents spoken in German, though not in English.

A particular issue is the robustness of the new method with respect to morphological variants of query features. Note that a morphological variant is easily detectable if the phonemic transcription of the query feature is a subsequence of the variant, which is true e.g. for the query feature “conflate” and the variant “conflating”. But shorter variants may also be detected as long as they are phonemically similar compared to the query feature. For example, “phonemes” may also yield slots where “phoneme” was spoken, because the error-tolerant slot detection may interpret the missing “s” as a phoneme deletion or substitution error.

We believe this work is a significant contribution to the retrieval of documents that are spoken in languages for which little training data is available. Our method is not only applicable to spoken documents but also to corrupted OCR output obtained from scanned text images. In the case of spoken documents, these techniques can be viewed as a valuable add-on to the out-of-vocabulary problem in word-recognition based retrieval.

References

Abberley D, Renals S, Cook G and Robinson T (1998) The THISL spoken document retrieval system. In: Proceedings of the Sixth Text Retrieval Conference (TREC-6).

Allan J, Callan J, Croft W, Ballesteros L, Byrd D, Swan R and Xu J (1998) INQUERY does battle with TREC-6. In: Proceedings of the Sixth Text REtrieval Conference (TREC-6).

Brown M, Foote J, Jones G, Jones KS and Young S (1996) Open-vocabulary speech indexing for voice and video mail retrieval. In: ACM Multimedia Conference, Boston, MA.

Buckley C, Allan J and Salton G (1994) Automatic routing and ad-hoc retrieval using SMART: TREC 2. In: TREC-2 Proceedings, pp. 45–55.

CMU (1995) cmudict. 0.4. Carnegie Mellon University Pronouncing Dictionary, http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

Dharanipragada S, Franz M and Roukos S (1998) Audio indexing for broadcast news. In: Proceedings of the Sixth Text REtrieval Conference (TREC-6).

Garofolo JS, Lamel L and Fisher W (1990) DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus CD-ROM. U.S. Department of Commerce, Gaithersburg, MD 20899.

Glavitsch U and Schäuble P (1992) A system for retrieving speech documents. In: Belkin N, Ingwersen P and Pejtersen AM Eds., ACM SIGIR Conference on R & D in Information Retrieval, pp. 168–176.

James D (1996) A system for unrestricted topic retrieval from radio broadcasts. In: Proceedings ICASSP, Atlanta, GA, USA. pp. 279–282.

Jones G, Foote J, Jones KS and Young S (1995) Video mail retrieval using voice: An overview of the stage-2 system. In: van Rijsbergen C, Ed., Proceedings of the Final Workshop on Multimedia Information Retrieval (MIRO'95), Electronic Workshops in Computing, Glasgow. Springer.

Jones G, Foote J, Jones KS and Young S (1996) Retrieving spoken documents by combining multiple index sources. In: ACM SIGIR Conference on R & D in Information Retrieval, Zurich, pp. 30–38.

LDC (1996) DARPA continuous speech recognition corpus-IV: Radio broadcast news (CSRIV Hub-4), CD-ROM, Linguistic Data Consortium, Philadelphia, PA 19104-2608, USA, ldc@ldc.upenn.edu.

Lee KF (1989) Automatic Speech Recognition: The Development of the SPHINX System. Kluwer Academic Publishers, Boston.

Mateev B, Munteanu E, Sheridan P, Wechsler M and Schäuble P (1998) ETH TREC-6: Routing, Chinese, cross-language and spoken document retrieval. In: Proceedings of the Sixth Text REtrieval Conference (TREC-6).

Mittendorf E (1998) Data corruption and information retrieval. PhD Thesis, Swiss Federal Institute of Technology. Diss. ETH No. 12507.

Mittendorf E, Schäuble P and 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.

Ng K and Zue V (1997) Subword unit representations for spoken document retrieval. In: Proceedings of ESCA Eurospeech Conference, Rhodes, Greece.

Rabiner J (1993) Fundamentals of Speech Recognition. Prentice-Hall, Englewood Cliffs, NY.

Robinson T (1994) An application of recurrent nets to phone probability estimation. IEEE Transactions on Neural Networks, 5(3).

Schäuble P (1997) Multimedia Information Retrieval—Content-Based Information Retrieval from Large Text and Audio Databases. Kluwer Academic Publishers, Boston.

Sheridan P, Wechsler M and Schäuble P (1997) Cross-language speech retrieval: Establishing a baseline performance. In: ACM SIGIR Conference on Research & Development in Information Retrieval, Philadelphia.

Singhal A, Buckley C and Mitra M (1996) Pivoted document length normalization In: ACM SIGIR Conference on R & D in Information Retrieval, pp. 21–29.

Voorhees E, Garofolo J and Jones K (1998) The TREC-6 spoken document retrieval track. In: Proceedings of the Sixth Text REtrieval Conference (TREC-6).

Wactlar H, Hauptmann A and Witbrock M (1996) Informedia: News-on-demand experiments in speech recognition. In: Proceedings of DARPA Speech Recogition Workshop, Arden House, Harriman, NY.

Wasser JA (1985) English to phoneme translation. Public domain software, ftp://ftp.doc.ic.ac.uk/packages/unix-c/utils/phoneme.c.gz.

Wechsler M (1998) Spoken Document Retrieval Based on Phoneme Recognition. PhD Thesis, ETH Zurich. Diss. No. 12879.

Wechsler M, Munteanu E and Schäuble P (1998) New techniques for open-vocabulary spoken document retrieval. In: ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 20–27.

Wechsler M and Schäuble P (1995) Speech retrieval based on automatic indexing. In: van Rijsbergen C Ed., *(Proceedings of the Final Workshop on Multimedia Information Retrieval (MIRO '95)*, Electronic Workshops in Computing, Glasgow. Springer.

Witbrock M and Hauptmann AG (1997) Speech recognition and information retrieval: Experiments in retrieving spoken documents. In: Proceedings of the DARPA Speech Recognition Workshop, Chantilly Virginia.
Young S, Woodland P and Byrne W (1993) HTK Version 1.5: User, Reference & Programmer Manual. Entropic Cambridge Research Laboratory, Sheraton House, Castle Park, Cambridge CB3 OAX, England.