# Phoneme based Spoken Document Retrieval

Wessel Kraaij, Joop van Gent and
Rudie Ekkelenkamp
TNO-TPD
Stieltjesweg 1, 2600 AD Delft
The Netherlands
{kraaij,gent,ekkelen}@tpd.tno.nl

David van Leeuwen

TNO-HFRI
Kampweg 5, 2769 DE Soesterberg
The Netherlands
vanleeuwen@tm.tno.nl

## ABSTRACT

Since speech recognition technology has become more and more mature, retrieval of spoken documents has become a feasible task. We report about two cases, which aim at scalable and effective retrieval of broadcast recordings. The approach is based on a hybrid architecture, which combines the speed of off-line phoneme indexing and precision of wordspotting while maintaining a scalable architecture, which allows for frequent updates of the database where out-of-vocabulary (OOV) words are abundant. A pilot experiment has been done on a small database of recordings of a Dutch talkshow. A more extensive evaluation took place in the framework of the Spoken Document Retrieval track of TREC7 on English broadcast news.

**Keywords:** Spoken Document Retrieval, Speech Recognition, Radio broadcast databases

## 1 INTRODUCTION

This document describes ongoing experiments at TNO in the area of spoken document retrieval. This relatively new field poses new challenges to classical information retrieval because from an IR perspective the data to be indexed is highly corrupted. SDR is also quite demanding for Speech Recognizers. The source material has a high variety of speakers, low signal/noise ratio, there are many proper names (which are very important for retrieval) and the collections grow rapidly. So a SDR system should be very robust.

### 1.1 EARLY EXPERIMENTS WITH SPEECH RETRIEVAL

The idea for the approach described in this paper was inspired by the success of an earlier project called Talking Heads, which was carried out in 1995 by the independent Dutch research organisation TNO-TPD and the telecom company KPN Research.

The idea in Talking Heads was to combine a proprietary full text retrieval system called MOOI with speech recognition. MOOI was developed earlier by TNO and allowed users to retrieve information from a textual database in two steps, in the following way.

A textual query entered via the keyboard was matched with an index consisting of noun phrases, using proprietary fuzzy matching technology based on trigrams, called ISM ([12])The noun phrases were derived automatically from the database using simple forms of syntactic analysis. This first retrieval step would thus yield a list of noun phrases ranked according to “relevance” to the query, whereby relevance was defined in terms of the fuzzy matching algorithm. In a second step users could select one of the noun phrases from the retrieved list, and thus retrieve a list of all documents that contained the selected phrase.

This method worked well in the sense that the intermediate step in which the phrases were shown to the user allowed for an improvement in precision over one step approaches and a high flexibility on the query side, because the system could come back with good results in cases where the query contained typo’s, misspellings or morphological/syntactic variants of relevant

phrases in the index. The system also gave good results when applied to corrupted data such as OCR text. TNO had also experimented with a "phonetic" variation of the fuzzy matching algorithm: here *triphones* (trigrammes constructed from phonetic representations of the phrases and the queries) were used.

Talking Heads used both the two step retrieval strategy and the fuzzy matching algorithm in its "phonetic mode" to allow for retrieval of textual information using spoken queries. The front-end of the Talking Heads system consisted of the Hidden Markov Toolkit (HTK) system configured as a phone recognizer. It was tuned such that the phonetic representation of the speech data could be used for trigramme matching. Spoken queries could thus be transformed to so-called graphone strings. Graphones were phonetic representations of character strings. The graphone alphabet was developed earlier by TNO and optimised for soundex-like retrieval on Dutch texts, and later on tuned to the Talking Heads demonstrator. The (back-end) textual database was indexed with phrases, with the use of the MOOI system. These phrases were converted to a graphone representation using a rule based transfer module. The output of the HTK was matched with the graphone representation of the phrases using a combination of ISM and the Levenstein edit distance metric. For the purpose of the project a special speech model was developed for circumstances of casual use of speaker independent microphone speech.

The result was a system that translated spoken queries to graphone strings and matched them with a graphone representation of the noun phrase index. The system would thus yield a list of noun phrases on the screen, and the idea was that by using just a touch screen and a microphone the user could have full interaction with the system. In the project a mouse was used instead of a touch screen. The envisaged applications were either robust electronic information services in public buildings such as department stores, or professional electronic assistants in situations where users would need hands free information access, such as assembly or maintenance, like in a garage.

Despite of the relatively poor speech model that was used the results were quite promising. In most of the retrieval sessions arbitrary users could get access to relevant information within 6 interactions, i.e. 1 or 2 retries for query entry, and 1 or 2 retries for phrase selection. In the other cases the system just could not interpret the query.

The main problem within Talking Heads was the relatively poor speech model, that was based on insufficient training, particularly with respect to its speaker-independence. The project time and budget did not allow for a more profound training though. The results lead the research team to the conclusion that it would be worth trying a single speaker approach which would involve significantly less training. This approach would be uninteresting for the desired applications, but useful for retrieval on a database with spoken information uttered by one or just a couple of speakers, like for instance radio news items. A concurrent advantage would perhaps be the better (studio) quality of the speech signal. Within talking Heads this idea of using *triphone matching* for retrieval on speaker-dependent speech data could not be worked out anymore, but it made its way into the follow-up projects DAS+ and DRUID which provided the framework for the experiments which will be presented in this paper. DAS+ is a project for the Dutch broadcasting company TROS. Together with systems house VDA, TNO has built a prototype retrieval system for their Radio broadcast archive. DRUID is a multimedia research project funded by the Dutch government and industry via the Telematics Institute. One of the topics in DRUID is speech recognition. In order to test the DAS+ prototype on a larger scale, we participated in the SDR track of TREC7. The paper is organised as follows, this section continues with a description of our approach. Section 2 describes the pilot experiments in the DAS+ framework and section 3 presents the TREC7 SDR experiments. Section 4 gives conclusions and ideas for further work.

## 1.2 SPOKEN DOCUMENT RETRIEVAL (SDR)

With the recent rapid improvements in Speech Recognition technology, retrieval of Spoken Documents has become feasible. The SDR task is similar to the Text Retrieval task, but the document collection consists of Audio recordings containing spoken material. A typical SDR system applies large vocabulary continuous speech recognition (LVCSR) as a preprocessing step in order to use standard text retrieval techniques. A second option is to convert the audio signal into a phoneme sequence, which has certain advantages like a flexible vocabulary. The latter approach was also chosen used for the experiments, which are reported in this paper. Both conversion approaches have one common feature i.e. recognition errors which are quite frequent even with broadcast quality speech. The error rate for phoneme transcripts is higher however, because the models take less context into account.

The spoken material in SDR applications ranges from Radio News bulletins, radio talkshows to sound clippings from video broadcasts each posing their particular difficulty to SR systems. Unlike traditional SR applications where precision is of primary importance like *command & control* or *dictation*, recognition errors do not necessarily invalidate spoken document retrieval, because the goal of SDR is not to retrieve transcripts, but to retrieve and play audio segments that are relevant to the users query. If we manage to locate relevant fragments, we can just play the original recording instead of displaying the corrupt transcript. There is an analogy to the retrieval of OCR text. The latter case is likely to be less hampered by corruption because of the quite high redundancy in written text. In Radio news bulletins the redundancy is probably less high, so the need for error-tolerant search techniques is more urgent.

Depending on the application type, Recall might be more or less important, in most cases Precision is important. A final requirement for SDR systems is scalability, off-line recognition and indexing time should be at least one order of magnitude faster than real time. Retrieval response time should be low, which necessitates an architecture based on indexing instead of linear search.

The majority of SDR applications consist of a combination of an LVCSR system with a classical IR system. This means that the audio material is simply converted to a textual transcript that is input for the IR system. Successful prototype systems exist like Informedia [10] with spoken input. However, although the vocabulary of these systems is quite large, the vocabulary is fixed. Secondly, the majority of these recognizers have been trained for American English only. One approach to building an SDR application for Dutch could be to train an LVCSR system for Dutch. This is quite an effort, however, because large annotated corpora are required to train the acoustical models. We think, however, that we would encounter two more fundamental problems with a LVCSR-only based SDR system for Dutch. Firstly there's the out-of-vocabulary (OOV) problem which is quite prominent in the news domain (proper names). Secondly, the morphology of Dutch is more complex than English, requiring a much larger vocabulary for the same coverage. For the majority of Germanic languages (English being an exception) compounds are written as single orthographic units, which means that in order to be recognized, they have to be included in the lexicon of the recognizer. As compounding is a highly productive process, the OOV problem is more severe.

## 1.3 PHONEME BASED APPROACH

For DAS+ we have chosen to experiment with phone based retrieval. This choice was mainly motivated by the following arguments

- No language model required
- Off-line recognition time is much faster than LVCSR (simpler search algorithm)
- Less sensitive to the OOV problem
- Reuse of robust indexing strategies for OCR text (triphones instead of trigrams)

Pilot experiments showed that the retrieval results with triphone matching produced results with a

rather poor precision. Because experiments with a wordspotter configuration using Abbot had shown quite impressive precision, we decided to add a word-spotting step as a refinement step on the result set of the triphone search. The 2-stage search strategy has the following advantages:

- Retrieval based on triphone matching is fast, but not very precise because of the high phone error rate.
- A word spotter based on on-line phone lattice spotting is much more precise, but also slower due to the linear search process.

A more detailed description of the system will be given in section 2.

## 1.4 EVALUATION METHODS

In order to assess the retrieval quality of SR components and SDR systems as a whole, different methodologies can be applied. All methods presuppose a 'test corpus'. A classical IR test corpus consists of a collection of documents, a collection of queries and a set of relevance judgements. For a good evaluation one would like to test on several test collections, and preferably on test collections of considerable size. Such a test would produce results on 'average precision' and 'precision at cut-off levels'. Unfortunately these test collections do not yet exist in the SDR domain (an exception is the test corpus of the SDR track of TREC7 which has been constructed this year). A simpler poor mans solution to evaluation (which was used at TREC6), is to perform *Known Item Retrieval*. This procedure works as follows. First a set of unique documents is selected (this is in fact a non-trivial task). Queries are constructed from these documents. A perfect SDR system will return the document, which was the source for the query at first rank in the result list. Three measures can be derived from this evaluation method: (a) mean reciprocal rank (b) percentage of queries which retrieved the the known item in first position (c) cumulative percentage of queries that retrieve the known item by rank. A disadvantage of the method is that it does not say a lot about Precision and Recall.

When comparing results of different groups it's quite important to compare the characteristics of the test collection. Does it contain read speech or spontaneous speech, one speaker or multiple speakers? What methodology or rationale has been used to segment the audio files into separate 'audio documents'? Segmentation could for example be based on fixed time frames, on speaker pauses, on lexical or visual clues determining story boundaries in news shows (CNN). The segmentation methods have a profound effect on the characteristics of a collection, and of course, also on the usability of a system.

## 1.5 RELATED WORK

There are two European groups that have been working on phoneme based SDR for several years and which inspired our work. In 1995 Wechsler and Schauble [14] started experiments at ETH-Zürich with a HTK based phone recognizer. They performed tests on a German corpus consisting of 4 hours of broadcast news, segmented into overlapping windows of 20 seconds. Queries are titles from news stories taken from the similar period. Experiments included a comparison between bi-, tri- and tetra grams of phones. Trigrams performed 225% better (average precision) than bigrams. Tetragrams were slightly worse. They also experimented with a probabilistic method to cope with recognition errors. Phone transcripts are scanned for sequences similar to the query via a fuzzy matching procedure, bounded by a maximal edit distance. The probability that these near matches are in fact correct hits is estimated on the basis of, among others, the confusion matrix of the recognizer. The method yielded an improvement of 32 % with respect to the trigram baseline. A variant where the fuzzy matching is applied on triphones has been tested on the TREC6 SDR corpus[9]. Results are impressive, the mean reciprocal rank is improved with 63% (0.20→0.43), given a phone error rate of 55%. The method can be simplified as: select candidate hits (slots) by dynamic generation of variants of partly matching triphones within a certain maximum edit distance. Subsequently estimate

the probability of each hit, rank them and select the top 100 hits, assuming that the rest are false alarms. The approach is essentially a word-spotting architecture, based on linear search. The dynamic programming and probability re-estimation technique defines a hypothesis space which is analogous to the phone lattice structures which can be produced with Abbot. However the ETH method is probably significantly slower than the phone lattice based word spotting approach in our experiments. The phone lattice files (produced off-line) are simply searched by a finite state automaton. No dynamic programming is needed to find candidate hits.

At Cambridge University, James [4] evaluated a number of SDR configurations on a small corpus (2h27') of broadcast news. The test collection further consisted of 40 queries and relevance judgements. His best results are produced with a hybrid system: a word recognizer in combination with a (phone lattice based) wordspotter. The approach has been extended and tested in the Video Mail Retrieval project [5][15] resulting in an average precision of 85% of the baseline (retrieval on manual transcripts).

An operational Broadcast news archive has been built at Dublin City University[8]. The SR system is based on the HTK, which is trained on triphones. This significantly reduced the error rate. The computational complexity of the training process was reduced by a smart conflation of similar phones. Results in TREC6 were good.

The SDR track of TREC6 [11] showed a diverse spectrum of hybrid and fault tolerant approaches based on exploiting N-best SR output or generating confusion variants on the fly. CMU's run with an N-best recognizer was promising, Clarit's query expansion with confusion variants was not so successful, maybe due to lack of adequate term re-weighting, ETH's system performed disappointing due to high phone error rate. IBM presented an approach, which has inspired us: the LVCSR based architecture is complemented by a word spotter fall back strategy to find OOV words. Because the spotter is relatively slow, it is only started on a pre-selection of the collection as determined by an n-gram retrieval run on a phone representation. The University of Sheffield did tests with the Abbot system, configured as word recognizer. OOV words were spotted, with limited effect because the Abbot pronunciation dictionary was missing only one word.

Conclusion: Phoneme based approaches are feasible, though extra care for term weighting schemes is required. In an environment with only a few OOV words, word based recognizers are much more effective. In operational systems, hybrid approaches are probably the best option.

## 2 SYSTEM ARCHITECTURE

The TNO spoken document retrieval system is based on the ABBOT Large Vocabulary Continuous Speech Recognition (LVCSR) system [7]developed by Cambridge University, Sheffield University and SoftSound. Abbot is complemented by a set of Indexing and Fuzzy matching modules, developed at TNO-TPD.

The following figure shows the architecture of the TNO system as used in the TREC7 experiments. For the pilot experiment, which is described in section 3 a more simplistic term weighting strategy was used.

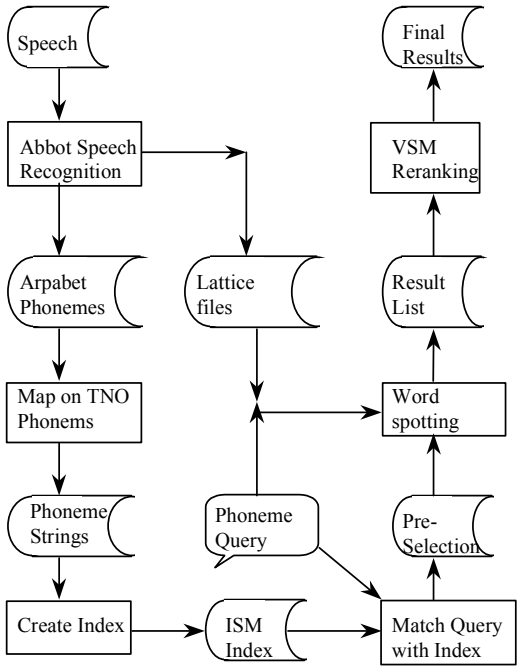


Figure 1: System architecture

In the left branch, audio files are converted to phone strings and phone lattice files. The phoneme transcripts are segmented and indexed by triphone sequences. There are two posibble retrieval strategies:

- A query is mapped to a phoneme representation based on the CMU dictionary. This phoneme representation is matched on the triphone index using the fuzzy matching tool ISM.
- The phoneme representation of the query is input for a word spotter, that searches the phone lattice representation of the collection.

These techniques will be discussed in some more detail in the following sections.

## 2.1 FUZZY MATCHING ON PHONEME TRANSCRIPTS

Abbot is configured as a phone recognizer (instead of a continuous word recognizer), in order to generate phone [1] transcripts of the speech documents. These are in turn converted to phoneme strings by segmenting the phones on pause symbols and mapping the phone symbols onto the characters a-z and A-Z. The phoneme strings are input for a fuzzy index based on phoneme trigrams (ISM index). For retrieval a fuzzy match is carried out between a phoneme representation of the query and the phoneme trigram index resulting in the top N documents which contain phrases similar to the query. The phonetic representation of the topic is determined by using the Carnegie Mellon Pronouncing Dictionary [CMU, 1995]. OOV words have been ignored.

[1] A *phone* is an acoustical realisation of a sound. A *phoneme* is a conceptual representation of a sound. There can be several phone realizations for a single phoneme in a language, for instance the `t' in `top' is aspirated, while the one in `stop' is not.

## 2.2 WORD SPOTTING WITH *TF.IDF* TERM WEIGHTING

**Off-line processing**
First, Abbot generates phone lattices, by reducing the acoustical input to posterior probability vectors of all phones in the phone set, of each 16 ms time frame. These lattices can be used to do both phone recognition (see 2.1) and on-line word spotting.

**On-line processing**
For word spotting, a phonemic representation of all query words is made. The words are mixed with single phones in a finite state grammar, and the query terms are spotted in the phone lattices using the finite state grammar decoder of Abbot. This is effectively a linear search. Finally all documents with query term hits will be scored and ranked via a standard *tf.idf* term weighting strategy.

# 3 PILOT EXPERIMENT

A first series of experiments with phoneme based SDR was performed within the DAS+ project. The test collection consisted of 1380 seconds of "Kamerbreed", a talkshow about news topics. This collection is quite different from the typical SDR test collections, which are typically dominated by, read speech and well-directed interviews. Our test collection was characterized by spontaneous speech, with interruptions and discussions. Part of the material was transcribed manually and segmented into utterances, separating different speakers and not splitting semantical units.

## 3.1 TRAINING THE PHONE RECOGNIZER

For acoustical training of the phone recognizer, approximately 4 hoursof speech data was available. This material was collected from 24 one-hour broadcasts of the radio programme “Kamerbreed.” In order to achieve high recognition accuracy, speaker-dependent models were made. Therefore, only models for two speakers were used, these speakers have the

function of interviewer in the radio show. At a later stage we plan to make speaker-independent models. The training material was transcribed manually at the word level. Because of the spontaneous character of the material, several notation conventions were introduced. Examples are:

1. special symbols for "breath" (an audible breath) and "eh" (a spoken filler word).
2. Hesitations, repetitions, and talking errors were spelled out phonetically in case they could not be transcribed as words.

For all words transcribed in the training material, a pronunciation in terms of phones was looked up in the CELEX dictionary. For words not in this dictionary, pronunciations were produced manually. For the latter class of pronunciations, suggestions were made available automatically by a compound splitter, because for Dutch, OOV words are often compounds.

For training of the acoustical parameters of the recognition system, an earlier version of the system was used as a bootstrap model. Training was performed after forced alignment of the phone transcription with the acoustical data. Acoustical training included the updating of the weights in the recurrent neural network, which forms the heart of the Abbot recognition system. Also the phone Markov models, which model phone durations, were re-estimated. The performance of the speaker-dependent models was measured from the development test set, giving phone error rates of 34% and 39% for the two speakers. Approximately half of the errors are deletions. Subjective comparisons of the acoustic material and the reference phone transcription suggest that the deleted phones are indeed never pronounced.

## 3.2 EXPERIMENT

First, Abbot was used to produce both a phone sequence representation and a phone lattice representation of each segment. The test collection was quite small: 134 segments. 74 queries were constructed from these segments in order to do a *known item retrieval* evaluation. The queries were converted to phoneme sequences using a grapheme to phoneme converter (G2P) The G2P uses a standard phoneme dictionary (CELEX) a large list of proper names and a simple algorithm to decompose compounds. We did experiments with triphone fuzzy matching on the phone transcripts (ISM [12]), and with Abbot word-spotting using the phone lattice files. In the latter case the phoneme representation of a query term is used to construct a finite state automaton to search through the lattice. The word spotter runs did not use any form of term weighting, straight *tf* was used as document score, queries were 2.18 words long on average.

The following experiments have been performed: (a) baseline: manually transcribed data (b) triphone matching (ISM) on phonemes (c) word-spotting on phone lattices. We tested a set of minor variations of the wordspotter configuration (i) one word-spot run per query term (ii) one word-spot run per query (iii) addition of phrases. The rationale for these experiments is that the false alarm rate is lower for longer words (phrases) and that a larger vocabulary for the wordspotter increases accuracy, because overlap between spotted words is impossible.

## 3.3 RESULTS

### 3.3.1 Wordspotter versus triphone matching

Different word spotter configurations were tested. We counted the number of queries that did not retrieve the known item at first rank. As a measure for discriminatory power, we also computed the mean size of the first rank.

| test on 74 queries | # failures | # rank 1 |
|---|---|---|
| Ws1: 1 word/run | 5 | 3.27 |
| Ws2: stopword removal | 4 | 3.57 |
| Ws3: phrases in single run | 6 | 3.28 |
| Ws4: concatenated phrases | 7 | 3.27 |
| Ws5: 1 run/query | 5 | 3.18 |
| Triphones (ISM) | 25 | 1.39 |

**Table 1: Results of wordspotting and triphone**

**matching**

**Table 1** shows the tremendous difference between wordspotting and triphone matching. On a queryset of 74 queries ISM fails about 1 out of 3 times to retrieve the known item at first rank. The Wordspotte, however, fails only 4 times. The word spotting runs can probably be improved further by a proper weighting scheme, which uses the confidence information, which in principle is available from the spotter. Small modifications that improved results were stopword removal (test2) and starting the wordspotter with the full set of query terms (test5). The experiments with phrases did not yield uniform improvements. A more sophisticated weighting scheme could probably help.

### 3.3.2 ISM as pre-selection for the wordspotter

We also experimented with a hybrid architecture where triphone matching was used as a first step to reduce the amount of data to be searched by the word spotter. The word spotter in this test is the word spotter with stopword removal.
We experimented with several methods to make the preselection:

1. Take all documents from the first stage
2. Take the N best hits
3. Take the hits with a score higher than δ
4. Take the N best ranks

| INPUT | # failures | # rank 1 | # refs searched | (reduction factor) |
|---|---|---|---|---|
| N=all hits (1) | 6 | 2.18 | 27 | 5.0 |
| N= 20 hits (2) | 8 | 2.16 | 17.5 | 7.7 |
| N= 10 hits (2) | 11 | 1.82 | 9.6 | 13.9 |
| N= 1 hit (2) | 33 | 1.0 | 1.0 | 134 |
| δ=0.20 (3) | 21 | 1.79 | 3.46 | 38.7 |
| δ=0.40 (3) | 53 | 1.28 | 0.54 | 247.9 |
| δ= 0.50 (3) | 62 | 1.41 | 0.32 | 413 |
| Best 3 ranks (4) | 8 | 1.6 | 9.7 | 13.7 |
| Best 2 ranks (4) | 14 | 1.4 | 4.4 | 30.2 |
| Best rank (4) | 27 | 1.1 | 1.56 | 85.5 |

**Table 2: ISM + wordspotter hybrid system**

The third column in **Table 2** gives the average number of segments per query which is presented to the wordspotter, the next column gives the corresponding reduction factor. The table shows that the 2-stage architecture has a perfect means to trade time for the percentage of known items retrieved by the wordspotter. Figure 2 shows that the %found@1st_rank measure increases sharply in the initial segment of the plot. A threshold of e.g. N=7 seems quite reasonable for this (small) collection, limiting the search time for the spotter with a factor 10.

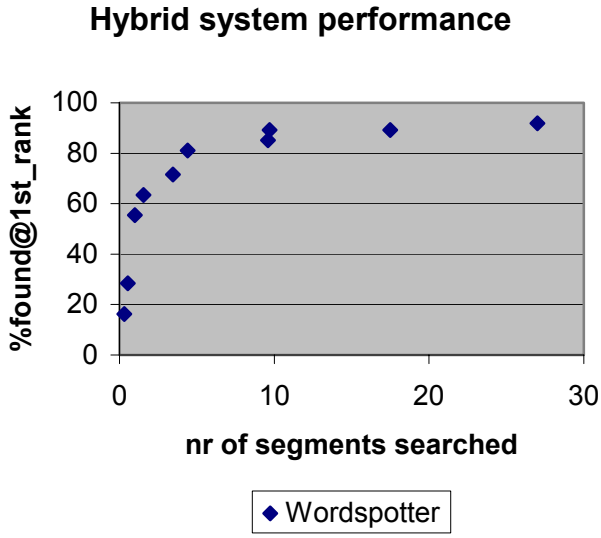


Figure 2: Trading speed for Recall

## 3.4 CONCLUSIONS

The various wordspot architecture variants do not differ much when we look at the %found@1st_rank measure. A more precise comparison can only be made when we include the confidence values of the wordspotter into the ranking algorithm. There is already evidence, however, that removing stopwords is quite useful, and that treating proper names (e.g 'Wim Kok' the Dutch prime minister) as a single acoustic unit probably improves precision. When we compare the wordspotter results with fuzzy matching on triphones, the difference is much more marked. The experiments with a hybrid approach show that ISM can be used as a coarse first step to select a list of candidate hits which in turn is

searched with the high quality word spotter. The hybrid system is much more precise than the 1st step alone, and faster than wordspotting on the full database. If a user wants more Recall, he can simply start the 2nd reranking step on a bigger fraction of the results of the 1st step. The wordspotter works in approximately 40 times real-time on a Sun ultrasparc 300Mhz.

# 4 TREC 7 SPOKEN DOCUMENT RETRIEVAL TRACK

This section reports about experiments in the SDR track of TREC7, carried out at TNO-TPD and TNO-HFRI. The corpus for this track consisted of 100 hours of American Broadcast news for training and 100 hours for evaluation. We only used the latter part, which consists of 2866 documents with an average of 268 words. The segmentation of the corpus has been done manually, in order to produce separate stories. There were 23 queries.

Again we used the Abbot SR system[7], configured as a phone recogniser. Because we lacked time, we could not train Abbot for American English. Tony Robinson from the University of Cambridge kindly provided the acoustical models (the weights of the recurrent neural network) needed to carry out phone recognition and word spotting. For retrieval we experimented with several approaches: (a) Fuzzy matching on a phoneme representation of the database and (b) Phone lattice based word spotting, with a quite standard *tfidf* term weighting strategy.

## 4.1 TRAINING

The phone models were trained at Cambridge University on the 100 hours SDR training set (the 1996 Broadcast News speech corpus , CSR-V: Hub 4). Training was done on two different subsets:

(1) All training data stripped from commercials (about 70 hours)
(2) Only studio quality material (for F0 and F1 conditions) each producing their own model.

Training was performed forwards and backwards in time, effectively resulting in 4 models. At recognition time the log probabilities of the 4 models were averaged.

We did an initial evaluation of the quality of the phone recognizer on 2.2 hours of the TREC7 Broadcast News testset and found a phone error rate of 44% (21% substitutions, 21% deletions and 2% insertion, phone set of 55 phones)

## 4.2 OFFICIAL RUNS

We used a single strategy for the R1, B1 and B2 tasks. A vector space index was built on the transcripts and the topics were matched with this index. The weighting scheme used is okapi9, as used in the PRISE engine from NIST[6].okapi9 defines the tf component as

$$\frac{tf}{tf + \log(1 + dl/avdl)}$$

where *dl* is the document length and *avdl* represents the average document length. This resulted in the following average precision values for the tasks R1, B1 and B2.

| Run Type | AVP | %change |
|---|---|---|
| **R1**: Reference Retrieval using human-generated "perfect" transcripts | 0.3970 | 0 |
| **B1**: Baseline Retrieval using medium error (35%WER) recognizer transcripts | 0.3533 | -11 |
| **B2**: Baseline Retrieval using high error (50%WER) recognizer transcripts | 0.2833 | -29 |
| **S1**: Full SDR based on wordspotting | 0.0436 | -89 |
| **S1_fixed:** Unofficial bugfix run | 0.1219 | -69 |

Table 3: Results of the SDR TREC7 runs

For the S1 run we submitted a run based on the method described in 2.2. The only conceptual difference with the DAS+ pilot set-up on the IR

side was that we intended to do a more sophisticated term-weighting strategy. This strategy was based on the retrieval model developed within TwentyOne at University of Twente[2][3], which was used to modify the ranking based on a simple count of the number of spotted words

After receiving the relevance judgements some unofficial runs were done for the S1 task. It turned out that there were some major errors in the system. Some of these errors have been solved now (cf. 4.3) and the best run for the S1 task using the word spotting approach has an average precision of 0.1219 (runtag: S1_fixed).

## 4.3 DISCUSSION

The baseline runs show that the average precision decreases steadily with increased word error rate. But at a 50%word error rate, the performance is still quite reasonable. The S1 results were quite disappointing. We have identified a series of possible causes. First of all, due to lack of time no phoneme or phone lattice transcript of the training set was available for the S1 task. The R1 run was used as a substitute relevance judgements file. It turned out to be very hard to tune the system with these judgements. Post-hoc analysis of our S1 run revealed some problems:

- **Errors in term weighting:** termweighting was effectively inactive.
- **Document length of word spotted document**. The ranking of the documents was suboptimal because we didn't know the document length of the spoken documents. In the official S1 run we used the number of spotted words as document length. This turned out to be a bad measure for the document length. In our unofficial run we used the length of the phonetic representation of the document. This dramatically improved the performance.
- **False alarms for short words.** Another big problem for word spotting was the high false alarm rate for short words. For example: the word "gun" has been spotted 14.000 times while in the transcriptions it only occurs about 100 times. This degraded the performance dramatically. In the future the confidence value of the spotted word should be taken into account to be able to tune for small and large words. Another possibility is to test the reweighting strategies as proposed by ETH[9], or to start the wordspotter with an extra dictionary of say 1k frequent words. This will most probably decrease the false alarm rate of short words. Figure 3 gives a comparison between the number of words found by the wordspotter (grouped by word length) and the true number of occurrences as found in the manual transcripts. It can be clearly seen that especially short words generate a lot of false alarm. The alpha parameter is used by the wordspotter to limit the number of false alarms. Note that stopwords were removed from the query term set, before making this plot!
- **OOV query terms**. There is an important difference in the consequences of OOVs in the conventional word recognition based retrieval and the word spotting based retrieval. For word spotting, only phone representations of OOV query words need to be generated on-line after the query has been made. A fast word spotting search can then be performed. But unfortunately no text to phoneme converter was available for English, so the CMU dictionary (0.4 version) has been used. This means that we could not fully exploit the potential advantage of our phoneme-based approach, which in principle is not vocabulary dependent. Quite a few topic words were not found in the CMU pronunciation dictionary among which crucial terms like: paparazzi, Montserrat and US. Since the spotter skipped these words, the results for the corresponding topics were ruined.

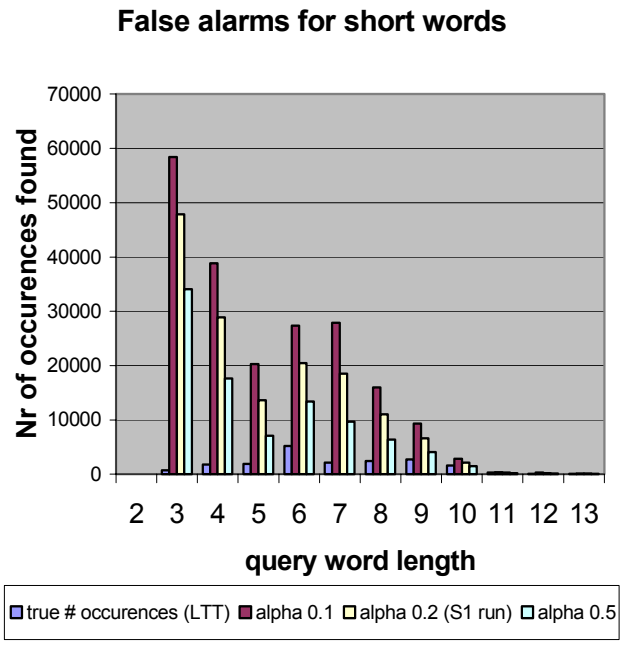


Figure 3: Rough estimate of false alarm rate

## 5 CONCLUSIONS

We have succeeded in building laboratory versions of an application for Spoken Document Retrieval based on phone recognition for Dutch and English. The pilot experiments with Dutch have shown that the 2-stage architecture is quite effective. A coarse 1st stage triphone search proved an effective means to limit the search space for the linearly operating but high quality word spotter. The initial results in the TREC7 SDR revealed a number of errors, some of which have already been corrected, resulting in big improvements. As such, the TREC7 evaluation testbed will be used to test and validate improved version of our applications. The unofficial corrected SDR runs have already shown that phone based retrieval is a feasible and scaleable approach. However for a real test of the architecture we need to do tests with a rule based grapheme to phoneme converter.

## 6 ACKNOWLEDGEMENTS

We would like to thank Tony Robinson from the University of Cambridge for providing us with the acoustical models for American English. Furthermore we would like to thank the DAS+ colleagues and especially Daan Otten and Jurgen den Hartog of TNO-TPD for their help to set up the DAS+ evaluation.

## REFERENCES

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

[2] Hiemstra, D , A Linguistically Motivated Probabilistic Model of Information Retrieval, *Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries (ECDL2)*,Crete,1998.

[3] Hiemstra, D. and W. Kraaij, TREC working notes: Twenty-One in ad-hoc and CLIR, *TREC 7 working notes*, 1998.

[4] David James, The application of Classical Information Retrieval Techniques to Spoken Documents, *Thesis*, University of Cambridge, 1995.

[5] Jones, Gareth J.F., J.T.Foote, K. Sparck-Jones and S. Young, Retrieving Spoken Documents by Combining Multiple Index Sources, *Proceedings of ACM-SIGIR 1996,* Zürich.

[6] The ZPRISE 1.0 Home page: www-nlpir.nist.gov/~over/zp2.

[7] Tony Robinson, Mike Hochberg and Steve Renals ,The use of recurrent neural networks in continuous speech recognition *http://*svrwww.*eng.cam.ac.uk/~ajr/rnn4csr94/rnn4csr94.html*

[8] Smeaton, A. F., M. Morony, G. Quinn and R. Scaife, Taiscéalái: Information Retrieval from an Archive of Spoken Radio News, *Proceedings of the Second European Conference on Research and Advanced Technology for Digital Libraries (ECDL2),* Crete, 1998.

[9] Wechsler. M., E. Munteanu and P. Schäuble, New Techniques for Open-Vocabulary Spoken Document Retrieval, Proceedings of ACM-SIGIR 1998, Melbourne.

[10] Hauptmann, Alexander G., en Witbrock, Michael J., Informedia News-On-Demand: Using Speech Recognition to Create a Digital Video Library, *CMU paper* http://informedia.cs.cmu.edu/pubs/aaaiinfo-haupt.pdf

[11] Garofolo, J., E.Voorhees, V. Stanford, TREC-6 1997 Spoken Document Retrieval Track Overview and Results, Harman, Donna (ed*.), Proceedings of the Sixth Text REtrieval Conference (TREC6*), NIST special publication 500-240, 1998.

[12] Heer, T. de, Quasi comprehension on natural language simulated by means of Information Traces, *Information Processing & Management,* 15,89-98, 1979.

[13] Jones, Gareth J.F. en James, David A., A Critical Review of State-of-the-Art Technologies for Cross-Language Speech Retrieval, *AAAI Spring symposium Cross Language Retrieval*, 1997, Stanford.

[14] Schäuble, Peter. Multimedia Information Retrieval*, Kluwer Academic Publishers*, Boston, 1997.

[15] Sparck Jones, Jones, Foote en Young, Experiments in Spoken Document retrieval*, Inofrmation Processing & Managament* **32** 399-419, 1996.