TNO TREC7 site report: SDR and filtering

Rudie Ekkelenkamp ,Wessel Kraaij* and*
*David van Leeuwen***

Contact Information

*) TNO-TPD (Instute for Applied Physics, department of Multimedia Technology)
Stieltjesweg 1, 2600 AD Delft
The Netherlands
**) TNO-HFRI (Human Factors Research Institute)
Kampweg 5, 2769 DE Soesterberg
The Netherlands
email: `ekkelen@tpd.tno.nl`
`kraaij@tpd.tno.nl`
`vanleeuwen@tm.tno.nl`
WWW: `http://www.tpd.tno.nl/TPD/smartsite304.html (TNO-TPD MMT)`

1. Introduction

This paper reports about experiments in the CLIR and filtering track, carried out at TNO-TPD and TNO-TM. TNO-TPD is also a member of the TwentyOne consortium and as such participated in the AdHoc task and the CLIR track. These experiments are discussed in a separate paper (cf. [Hiemstra and Kraaij98]) elsewhere in this volume.

2. SDR track

The TNO spoken document retrieval system is based on the ABBOT Large Vocabulary Continuous Speech Recognition (LVCSR) system [Renals 1998] developed by Cambridge University, Sheffield University and SoftSound, and uses word spotting, the TNO Vector Space Engine and fuzzy matching based on phoneme trigrams for indexing and retrieval. We participated in full SDR mode and experimented with several approaches

1. Fuzzy matching on a phoneme representation of the database.

2. Phone lattice based word spotting.

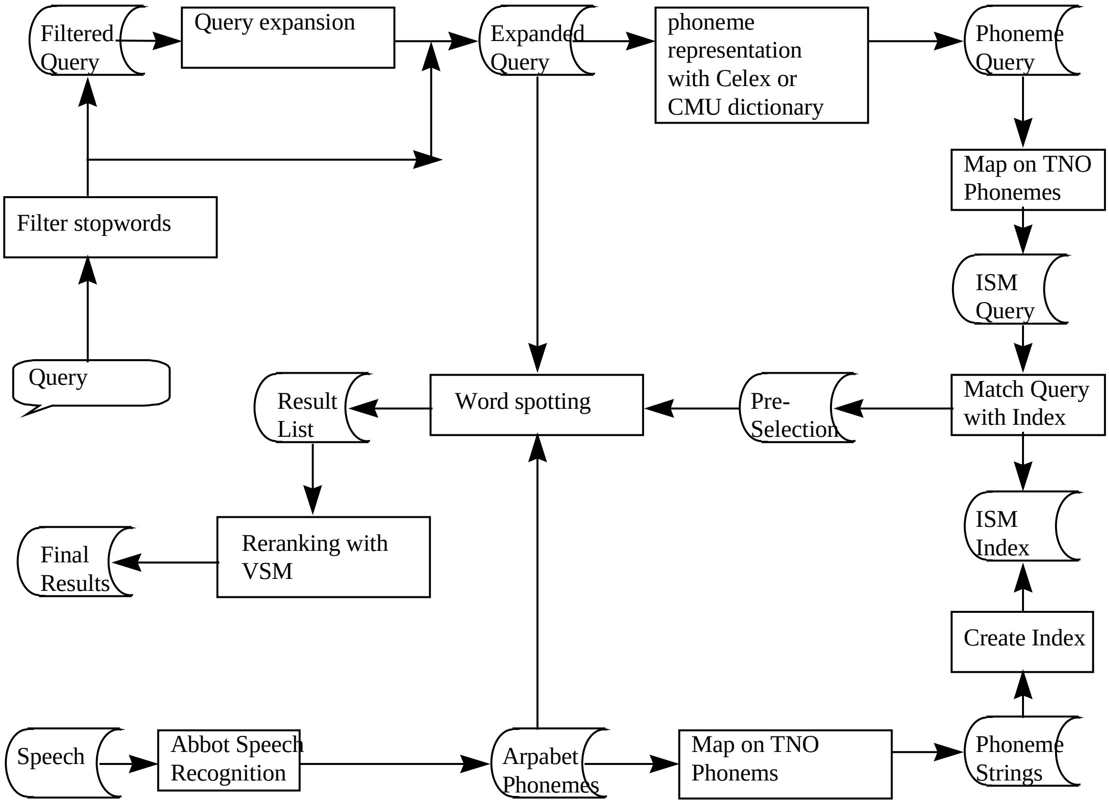
3. A hybrid approach were the fuzzy matching method acts as a first step to constrain the selection of input documents for the wordspotter. The idea is that there is a very efficient but rather imprecise first step and a relatively inefficient but precise second step.

The advantage of phoneme based approaches is that they do not restrict vocabulary. This is quite important for non English languages with a rich morphology and productive compounding. Also in a News domain, proper nouns are quite important. In this paper we only discuss the results of an approach based on methods 1 and 2, results of approaches 3 will be discussed elsewhere. Similar phone based experiments have been carried out at ETH [Wechsler1998] , University of Cambridge [James1995][Jones1996] and Dublin City University[Smeaton 1998].

The acoustical models needed to carry out phone recognition and word spotting were kindly provided by Tony Robinson from the University of Cambridge.

2.1 System

The following figure shows the architecture of the TNO SDR with the different approaches as described in the following sections



2.1.1 Fuzzy matching on phoneme transcripts

Abbot is configured as a phone recogniser (instead of a continuous word recogniser), 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 tirgram 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]. Out Of Vocabulary (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.

2.1.2 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 phoneset, of each 16 ms time frame. These lattices can be used to do both phone recognition (see 2.1.1) 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 simple phones in a finite state grammar, and the query words are spotted in the phone lattices using the finite state grammar decoder of Abbot. This is effectively a linear search.

After word spotting all documents will be matched with a vector space model and ranked by similarities of the match. This approach has been used for submitting the S1 task.

2.2 Results

2.2.1 Official runs

We used a single strategy for the R1, B1 and B2 tasks. A vector space index has been built on the documents and the topics have been matched with index. The weighting scheme used is okapi9, as used in the PRISE engine from NIST.

okapi9 defines the *tf* component as: $tf/(tf + {}^{2}log(1.0 + (doclen / avg\_doclen)))$

This resulted in the following average precision values for the tasks R1, B1 and B2.

Run Type	***AVP***
R1: Reference Retrieval using human-generated "perfect" transcripts	0.3970
B1: Baseline Retrieval using medium error (35% WER) recognizer transcripts	0.3533
B2: Baseline Retrieval using high error (50%WER) recognizer transcripts	0.2833

For the S1 run we submitted a run based on the method described in 2.1.2.

S1: Full SDR based on wordspotting	0.0436

2.2.2 Unofficial run

After receiving the relevance judgements some unofficial runs have been 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. 2.3) and the best run for the S1 task using the word spotting approach has an average precision of 0.1219. This run is based on the new Twente term weighting scheme (cf. [Hiemstra and Kraaij98] this volume and [Hiemstra98]).

2.3 Discussion

The baseline runs show that the average precision decreases steadily with increased word error rate. Still with a 50% WER 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. To be able to evaluate the runs the results for the R1 run were used as relevance judgements. It turned out to be very hard to tune the system with these judgements. Two other problems were related to document length normalisation and inactive term weighting.

Post-hoc analysis of our S1 run revealed some problems:

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 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 small words

Another big problem is the word spotting for small words since many false alarms have been generated. 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.

OOV query terms

In the Dutch version of the word spotter a rule base text to phoneme converter is used to transform queries into their phonetic representation. Unfortunately no text to phoneme converter was available for English, so the CMU dictionary (0.4 version) has been used. Unfortunately some topic words haven’t been found in the dictionary among which some very relevant words like: paparazzi, Montserrat and US. Since these words haven’t been spotted they will never be retrieved very well.

There is an important difference in the consequences of OOV’s 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.

A more elaborate analysis of the SDR experimental result can be found in [Kraaij 1998].

3. Filtering track: Adaptive Filtering

Because we did not participate in the filtering task in previous editions of TREC , we decided to use proven techniques. We chose the Adaptive Filtering subtask because we considered it a realistic task, close to real-world applications. Because the literature on these kind of applications is very scarce, the task to build a system based on Rocchio with a dynamic training procedure turned out to be very challenging

We built an adaptive filtering system which is initially based on Rocchio relevance feedback, we intend to migrate to rule based classifiers at a later stage. For every topic a profile (binary classifier) is built consisting of a weighted term vector, threshold function and similarities of the last N relevant and non-relevant documents that have been positively classified by the profile. Initially a profile is filled with a term frequency vector that will be weighted using a *tfidf* scheme. Collection frequencies have been intialized by taking statistics from the LA Times corpus. During the filtering process every incoming document is transformed into a weighted term frequency document vector. Every profile vector is matched with the document vector and will result in a similarity using the cosine measure. If the similarity is larger than the threshold of the profile, the document will be assigned to the profile. If the document is relevant, the profile vector is adapted using Rocchio relevance feedback [Rocchio 1971]. For all documents that have been assigned to a profile, the similarities of the match are stored in the profile. If a relevant document is assigned to a profile, the threshold of the profile is adapted using the midpoint of the averages of non-relevant similarities and relevant similarities.

3.1 Overview of the system

3.1.1 Initializing the adaptive filtering system

To determine statistics about document frequencies the LA Times 1988 Corpus has been used. The frequencies are used in weighing the profile vectors and document vectors. All terms from the corpus have been stemmed and stop words have been removed.

3.1.2 Creating the profiles

For every topic a profile is created as follows:

Stop words are removed from the topic text and the terms from the topic are stemmed using Porter. The resulting text is converted into a term frequency vector that is weighed using the following *tfidf* variant:

$^{2}\log(tf\_ij+1.0) * idf / \; ^{2}\log(doclength)$ (1)

The threshold of the profile is initially set at 0.4 and the Rocchio parameters are initialized at 2 and 4.

3.1.3 Creating a document vector

When a document arrives for filtering, it is first converted into a term frequency document vector. Then the weights are determined based according to weighting function (1). The statistics about term frequencies and document frequencies are updated with the new document. The resulting weighted document vector will be matched with the profile vector.

3.1.4 Matching the document with the profiles

After a weighted document vector has been created, every filter profile is matched with the document to determine the similarity of the document to the profile.

For the profile document similarity we took the cosine measure. If the similarity is larger than the threshold of the filter profile, the document is assigned to the associated topic.

3.1.5 Updating the profile

Once a document is assigned to a topic, the relevance judgements can be used to update the profile of the topic.

First will be determined whether the assigned document was relevant to the profile. If so, the similarity will be stored in a history list of the last N similarities of documents that have been assigned to the document. For a relevant document the profile vector will also be updated using Rocchio relevance feedback:

Updating a filter profile vector V with a document vector D that is relevant using Rocchio can be done as follows:

$V_{new} = \alpha * V_{old} + \beta * D$, where α and β are to be defined.

If the weighted elements of the filter profile vector V get too large after updating they are normalized by dividing them by a constant value. This is needed to be able to use a constant threshold during the filtering process.

Finally the profile threshold is adapted using the similarities of the N previously assigned relevant and non-relevant documents. The average of the N relevant similarities and the average of the N non-relevant similarities will be calculated and the new threshold in the middle of these averages.

3.2 Results

Using the previously described system the following results have been obtained. Two runs have been submitted:

TNOAF102 and TNOAF103 (preferred run).

For all runs the Rocchio parameters have been set to 2 (for α) and 4 (for β); no negative relevance feedback has been used. For TNOAF102 the initial threshold has been set to 0.35 and for TNOAF103 the threshold has been set to 0.40. There has been no tuning for F1 or F3 so these runs have been submitted for both utilities.

After training with the AP 1988 and AP 1989 corpora it turned out that the TNOAF103 run performed best. The following table shows a summary of the evaluation, each cell lists first the number of profiles that were 'silent' (i.e. did not retrieve any document), then the number of topics above and below median respectively.

TNOAF103	AP88	AP89	AP90
F1	26/12/12	26/9/15	25/10/15
F3	26/10/14	26/6/18	25/9/16

Tabel 1: Topicset breakdown figures for F1 and F3: 'silent'/above median/below median

3.3 Discussion

Compared to the track medians the TNOAF103 run scores slightly below median, which is promising taking into account that this is our first filtering application. Biggest problem are the 'silent' profiles, most of which should have found relevant documents. Apart from these topics, there is a considerable number of profiles where the run scores better than median performance. Preliminary conclusion: the approach does work, but it has to be tuned. Tuning will involve a careful examination of a selection of characteristic topics. Because averaging the F1 utility over the topicset is pointless we mention a few observations based on a first glance at the track results:

- About 10-15 topics perform better than median (cf. table)
- Most (25) topics stay 'silent'. The starting threshold is probably too high for these topics.
- Other topics score extremely bad, the starting threshold is probably too low for these topics.
- Experiments with other threshold setting did not yield global improvement.

We can conclude that a uniform starting threshold is ineffective. We intend to do a more detailed investigation to solve this problem and to find other dominant factors.

4. Conclusions

We have succeeded in building laboratory versions of an application for Spoken Document Retrieval based on phone recognition and a system for adaptive filtering. The initial results revealed a number of errors, some of which have already been corrected, resulting in big improvements. As such, the TREC6 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. For filtering we intend to extend our work in two directions. We expect that selection of better input features, e.g. phrases, semantic labels etc. will improve the results of the system. Secondly we have planned experiments with rule based classifiers which are less sensitive to the threshold problem.

5. Acknowledgements

We thank Tony Robinson from the University of Cambridge for providing us with the acoustical models for American English. We also thank Djoerd Hiemstra from the University of Twente and Joop van Gent (TNO-TPD) for fruitful discussions on thresholding and filtering in general.

6. References

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

[Hiemstra 1998a] 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.

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

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

[Jones 1996] 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.

[Kraaij 1998] Kraaij, W., van Gent, J., Ekkelenkamp, R., van Leeuwen, D. *Phoneme based spoken document retrieval* In: D. Hiemstra, F.M.G. de Jong, K. Netter (eds.), Language Technology in Multimedia Information Retrieval. Proceedings Twente workshop on Language Technology (TWLT14), pp. 141-153, Enschede, 1998.

[NIST 1998] The ZPRISE 1.0 Home page: www-nlpir.nist.gov/~over/zp2.

[Rocchio 1971] Rocchio, J. J. *Relevance Feedback in Information Retrieval* , In G. Salton (ed.), *The SMART Retrieval System*, Englewood Cliffs, N.J, Prentice Hall, 1971.

[Renals 1998] A. J. Robinson, M. Hochberg and S. Renals, *"The use of recurrent networks in continuous speech recognition".* In C. H. Lee, K. K. Paliwal, and F. K. Soong, eds., Automatic Speech and Speaker recognition---Advanced Topics, chapter 10, pages 233--258. Kluwer Academic Publishers, 1996.

[Smeaton 1998] 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.

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