Centre for
Telematics and
Information
Technology

Radio and Television Information Filtering through Speech Recognition

Ir. Arjen P. de Vries

To be published in the proceedings of the European Workshop on Interactive Distributed Multimedia Systems and Services (IDMS'96) 4-6 March 1996, Berlin

CTIT Technical Report series, No. 95-20 ISSN 1381 - 3625
P.O. Box 217 - 7500 AE Enschede - The Netherlands
telephone +31-53-4893779 / fax +31-53-4893247

Radio and Television Information Filtering through Speech Recognition

ir. Arjen P. de Vries [*]

Centre for Telematics and Information Technology
University of Twente

Abstract: The problem of *information overload* can be solved by the application of information filtering to the huge amount of data. Information on radio and television can be filtered using speech recognition of the audio track. A prototype system using closed captions has been developed on top of the *INQUERY* information access system. The challange of integrating speech recognition and information retrieval into a working system is a big one. The open problems are the selection of a document representation model, the recognition and selection of indexing features for speech retrieval and dealing with the erroneous output of recognition processes.

Keywords: multimedia, multimedia representation, content-based retrieval, information filtering, automatic indexing, speech recognition, content analysis, probabilistic information retrieval.

1 Introduction

The *Indexing a Sea of Audio* project forms a foundation for multimedia information filtering projects at the *Cambridge Research Laboratory (CRL)* in Boston [dV95]. CRL is a research laboratory of *Digital Equipment Coorporation (DEC)*. One of the strong points of CRL is the audio group. Interesting applications have been developed to ease the incorporation of audio into today's working environment [LPG$^+$93]. It is believed that audio will play an important role in computing in the next decades. This particular project addresses the problem of *information overload* in multimedia environments.

2 Problem Statement

The achievements in communication systems result in a society where everybody can publish information. However, we do not have the time and capabilities to process all these data. People easily get lost in large information spaces [Les89]. A lot of information exists that people would be interested in, if they only knew that it was available.

[*] This research has been conducted at the Digital Cambridge Research Lab, Massachusetts, United States of America

Since the early sixties, people have investigated the storage and retrieval of automatically indexed documents [vR79], [Sal89]. This research has been restricted to text processing. Little research has been done to investigate the information system solution for information overload applied to multimedia data. However, a lot of television and radio channels are broadcasting thousands of programs a day. How to deal with all this multimedia information is still an open problem.

A lot of expectations have been raised by speech recognition research. Laboratories reported 95% recognition rates on 5,000 word vocabularies [RHL94] and products can be found on the market [vS95]. This puts the development of a multimedia information access system for radio and television within reach. If you could transform audio into descriptive text, it would be possible to index the text representing the audio. Most information on television is *also* captured in the audio track. Of course, somebody looking for a red car crossing the railroad in some movie would not be helped. However, somebody looking for information on cars made by *Renault* would probably find an appropriate documentary.

An audio retrieval system can be used for many purposes. Of course, secret services of all countries would like to pay a lot of money for a system that can keep track of many audio channels. We could archive all meetings of a board of a large company. The formal decision process could be tracked afterwards, to study how mistakes and good decisions are made. An application in health services was suggested by a children's psychiatrist. A system storing recorded sessions can help evaluation of applied therapy to determine optimal treatment. At present, only a logbook kept by the psychiatrist can be used.

3 Research Background

3.1 Information Filtering

An *information retrieval* or *information access* system has the function of leading the user to those documents that will best enable him to satisfy his need for information. Over the years, information retrieval systems have moved from storing relatively short abstracts towards storing large collections of documents of varying size.

The terms *information dissemination* and *information filtering* refer to the delivery of information to users who submitted a profile describing their interests. The dissemination model has become increasingly more important due to the rapid advances in wide-area information systems. The simplest form of such a system is the mailing list. Examples of more advanced information filtering systems are SIFT [YGM] and NRT [SvR91]. Both systems provide dissemination of articles in USENET news.

In [BC92], the authors recognized the fact that information access and information dissemination are basically identical processes. The major difference is that information filtering usually works on a dynamic stream of input data whereas information access deals with relatively static databases. For the retrieval and filtering processes, the same ideas can be applied to both environments. A filtering application can be built on top of a traditional information access system. A common trick is to treat the user profiles as documents and the incoming news documents as queries. The same approach is taken by SIFT and *INROUTE*, the filtering system of the *INQUERY* information retrieval engine.

3.2 Automatic Analysis

Manually indexing will not be possible for all information sources that we want to follow. We definitely need automatic analysis of multimedia data. This section discusses currently available techniques that can be applied in real-time. The collection of techniques implemented in the system limits the capabilities of retrieval by content. An agent-based approach seems suitable to easily extend retrieval systems with emerging analysis techniques [Mae94]. The level up to which analysis has been achieved determines the system characteristics to a large account.

Segmentation or partitioning is necessary to split the incoming continuous stream of multimedia data in a sequence of *documents*. A document is a collection of data that belongs together in a higher semantical level. A news report on CNN and a commercial selling coke are two examples of such documents. A document can contain different types of media. Content analysis focuses on the analysis of information content of these documents. It should be noticed that segmentation information can be viewed as a special type of content information. If segmentation of an audio track identifies a three minute fragment as a song, this fact reveals more than just the borders.

In [Hea94], the text segmentation algorithm *TextTiling* is described. Term repetition is used as a feature to find topic changes. The algorithm is found to produce segmentation that corresponds well to human judgment of the major subtopic boundaries. For the segmentation of audio, several information sources can be used. People have developed speaker change detection and speech emphasis detection [Aro94], [CW92]. In [dV95], a segmentation algorithm in silence, speech and music is described. For the segmentation of video, it is also possible to use the sizes of the compressed frames. In [DLM$^+$94], a real-time algorithm is described that can be applied to produce *storyboards* of a video [LAF$^+$93].

An even harder problem is to index the content information of the parts in the partitioned stream. In case of a text-based environment, full-text indexing can be applied. If a perfect speech recognizer were available, the same would hold for speech data. Algorithms for content analysis of audio are speaker identification [RS78], word spotting [WB91] and speech recognition [RHL94]. Most image analysis algorithms are not real-time. For example, it is almost impossible to recognize a car in arbitrary images. For video, this implies that the only content information we will have available are closed captions. Speech recognition of the audio track seems the solution to get content information when closed captions are not available.

3.3 Filtering Speech

One of the best speech recognition systems is the *SPHINX-II* system, developed at Carnegie Mellon University. This large vocabulary continuous speech recognition system has achieved a 95% success rate on generalized tests for a 5000 word general dictation task. Speech recognition is viewed as a pattern recognition process. Phonetic units are modeled during training and used during recognition. *Hidden Markov Models (HMM)* are the most common approach to model phonetic units [SBG94], [Cox90]. For speech recognition, this tool seems to work well. However, a speech recognizer built with this technology has a restricted *preknown* vocabulary. If the word model is not known by the recognizer, the model of another word will be chosen as the best fit.

Unfortunately, with a restricted vocabulary, speech recognition is not very interesting for the purpose of automatic indexing. Typical queries we would want to perform on radio and television data refer to names and places. We can never anticipate the vocabulary for all possible search terms. Therefore, we have to use another approach to indexing speech.

Speech recognition based on phoneme models alone would not have a restricted vocabulary. The influence of the context on some of the phonemes makes recognition of phonemes as indexing features an impossible task though. In [GS92], an information retrieval model of speech documents is introduced that is in principle vocabulary independent. The key idea is that a small number of *indexing features* consisting of phoneme sequences was identified. These indexing features can be identified in text files as well. Only two different classes of phonemes are used: vowels (V) and consonants (C). The indexing features they proposed are V^+, V^+C^+, C^+V^+ and $C^+V^+C^+$[2]. The letters A, E, I, O, U and Y were denoted as vowels, the other letters are consonants.

The reason why this idea can be used to index speech data with an unlimited vocabulary is that it is not the purpose to recognize the words correctly (which would not be possible yet). The speech recognizer only has to identify the features that occur in the text. These features are only parts of the words. The idea is similar to text retrieval experiments with digrams and trigrams [Wil79]. However, in speech data it is not possible to detect word boundaries. It has been shown that this approach to indexing speech data is indeed feasible on a relatively small data set [SW].

4 INQUERY

INQUERY is a probabilistic information retrieval system [TC91], [CCH92]. It is based on inference networks [Pea89] and capable of dealing with Gigabyte collections. It has shown to be among the best information retrieval systems in the world [CC93]. Information retrieval is viewed as an inference process in which we estimate $\mathcal{P}(\mathcal{I} \mid \text{document})$, the probability that a user's information

[2] V^+ means one or more vowels

need is met given a document as evidence. Of course, uncertainness has to be taken into account.

The retrieval model supports multiple *representation schemes*, allows combination of results of different queries and query types and facilitates flexible matching between the terms or concepts used in the queries and those assigned to documents. The notion of multiple representation schemes refers to the fact that documents can be represented in different ways. Information from the abstract of a paper should be treated differently from information that is only given in the complete paper. A given query will retrieve different documents when applied to different representations.

The main tasks performed by INQUERY are building the collection, interpreting the query and using the inference networks to retrieve documents. During collection building, documents are parsed and representation terms are identified. The standard parser relies on a subset of SGML to identify the parts of the document to index like title and text. This parser can be extended or even completely replaced with another parser. The parser can be changed easily since it has been implemented with the standard Unix utilities *lex* and *yacc*.

Although the INQUERY system has been built for text representations, nothing limits the underlying mechanisms to be applied to other representation concepts. For terms that can be expressed in ASCII representation, the original system can be used without a problem. Therefore, the speech documents can be indexed with INQUERY using the features mentioned in section 3.3.

A drawback of using INQUERY for our project was the fact that the system could not deal with incremental indexing. Fortunately, the new version of INQUERY has solved this problem. Instead of using flat inverted files in a custom build system, the system was rebuilt on top of the persistent object store Mneme [BCCM94]. The new technique permits incremental indexing in a fast and efficient way [BCC94].

A more serious problem is that the performance of INQUERY on imprecise data is not known. The formula used to calculate the probability of usefulness of a document is based on term frequencies. These frequencies will have wrong values if the recognition process is not errorfree. This can result in a degradation on recall and precision. [TBCE94], [TBC94] and [CHTB92] report about information retrieval on texts that have been generated using *Optical Character Recognition (OCR)*. This way, the effects of noisy data on the retrieval process have been studied. The retrieval performance was measured on the scanned collection and compared with the results on a manually edited version of the database. [CHTB92] found that high quality OCR devices cause almost no degradation of the accuracy of retrieval, but low quality devices applied to collections of short documents can result in significant degradation of performance. The use of a speech recognizer to produce descriptive text is analogous to the use of OCR devices in these experiments. The results seem to predict that speech recognition should be of high quality to make good information retrieval possible.

Research has to be done whether speech recognition performs well enough to produce output that can be used reliably in a retrieval system. Knowledge of common mistakes by the recognition process could help improve retrieval. Experience with preprocessing the input data on scanned documents in INQUERY is reported in [TBC94]. The results were promising. The inference network model may be a good candidate to deal with these error models. Another level of representation nodes could be added to model the probability of correct recognition. The mathematical implications of this idea for the correctness and the computational complexity still have to be analyzed thoroughly.

5 The Prototype System

So far, a speech recognizer has not been used. We did have a *closed caption decoder* [Fed] though. Closed captions are like subtitles, but may contain slightly more information, eg. 'music' or 'knock knock'. Closed captions are primarily focused on hearing-impaired people. They are added to the television signal manually. These closed captions can be thought of as the output of an almost perfect speech recognizer. Some of the television companies broadcast closed captions together with their audio and video signal. Example broadcasters are the Public Broadcasting System and CNN Headline News.

The application *store-24* is build on top of the *audiofile (AF)* toolbox and was implemented using *Tcl* and *Tk* [Ous94]. Store-24 is a ringbuffer that stores the last twentyfour hours of a radio channel. Because this application is implemented on top of audiofile [LPG+93], everybody in the laboratory (and in principle in the whole world) can listen to this radio. You never miss the news because you can just reposition the audio in time.

5.1 Architecture

The abstract design of the prototype is shown in Fig. 1. It is based on a dual architecture. The incoming data is split in an information content stream and a raw data stream by the automatic segmentation and recognition unit. In the prototype, the information content stream contains the closed-captions of the television data. The storage unit dealing with the raw data is implemented using the *store-24* application. The prototype system only deals with audio.

The output of the automatic segmentation unit is stored separately from the raw audio data. A neural network, used to produce segmentation information, is described in [dV95]. The output of the content analysis unit is stored in an INQUERY data collection. The information content stream only contains the closed captions of the television signal because a speech recognizer was not available yet.

The different versions of the data are linked by {date, time, channel}-tuples. The user of the prototype system can search the data collection by content. INQUERY accepts natural language queries and returns a list of documents. The user interface of the prototype system is implemented on the *World Wide Web*.

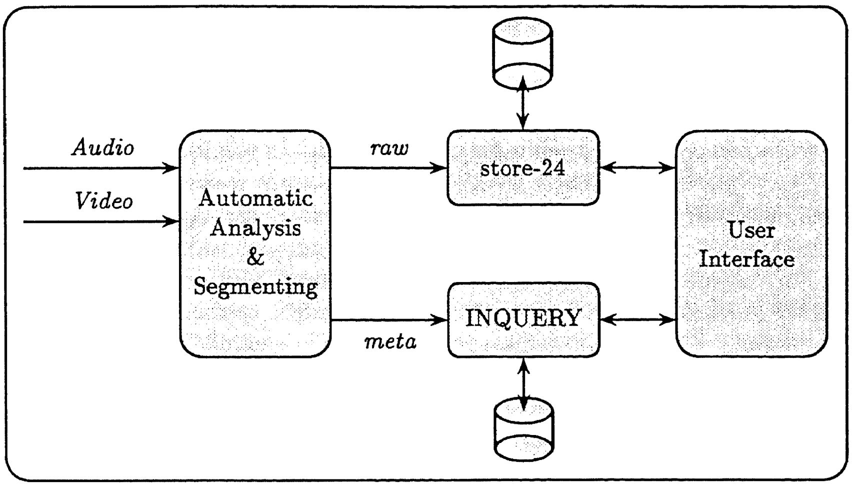

Fig. 1. The prototype framework

As result of a query, the system presents both the information content version and the original audio version of the document. It redirects the playback point of the user's active *store-24* application to the correct channel at the time the document started. A better approach than directly positioning store-24 would be to return a list of links that will position the audio after pressing. If the ringbuffer contained a longer time of audio, eg. a full month, these links could be automatically included in other documents like a user's homepage.

5.2 Implementation

The prototype has been realized on a UNIX platform. The system was implemented using C and Tcl/Tk [Ous94]. Using Tcl simplifies the handling of user input and the integration of different building blocks in one environment.

Because an audio document can be arbitrarily long and the recognition process will make mistakes, fast browsing through the retrieved audio document has to be implemented. The user does not want to listen to ten minutes of audio to find out that the speech recognizer did mix up two words. To enable jumping to retrieved representation concepts, an index of {timestamp, representation concept}-tuples has to be stored within the document. For implementation purposes I chose to store this information in a table containing {line, word, seconds} format, where seconds stores the time relative to the beginning of the document. I chose a word oriented approach of referring to locations within the document because this is the representation that is used within INQUERY. If we would use the indexing features of section 3.3, such a 'word' would be a phoneme sequence.

Of course, if we want to store real multimedia documents with the audio at the same location as the text, this representation cannot work any more.

The software accessing the closed caption decoder board does not output documents but just a long, unreadable list of {timestamp, caption}-tuples. These tuples are read by my software and segmented into documents whenever it takes longer than a treshold time for new captions to come in. On CNN Headline News, the only documents that are captioned are news reports approximately between 6 pm and 10 am and most of the commercials. An algorithm segmenting the incoming captions based on a time treshold of twelve seconds turned out to make little mistakes. Of course, this segmentation strategy would not have worked if all information were captioned. In a real application, the segmentation information from different sources mentioned in section 3.2 has to be combined. This is a very hard problem that has received very little attention at present.

The standard parser that comes with the INQUERY package deals with documents marked up in a subset of SGML. I decided to use this parser for my system as I believe that SGML and HyTime will be commonly used to describe (multimedia) documents in the near future [VB95]. An example of a document that I composed automatically from the incoming closed captions is shown in Fig. 2.

```
<DOC>
<DOCNO> CNN-04/04/95-02:00:03 <\DOCNO>
<DATE> 04/04/95 </DATE>
<TIME> 02:00:03 </TIME>
<SOURCE> CNN <\SOURCE>
<TIMES>
{1 1 2}{2 1 3}{3 1 13}{4 1 14}
{5 1 17}{6 1 19}{7 1 20}{8 1 22}
       ...
</TIMES>
<TEXT>
captions paid for by
the us department of education
live from atlanta
headline news
david goodnow reporting
texas authorities are trying
to figure out what caused
        ...
</TEXT>
</DOC>
```

Fig. 2. An example document

To enable the use of INQUERY, the parser of the indexing subsystem had to be extended and the interfacing between retrieval subsystem and user interface had to be implemented. With the help of the INQUERY people from University of Massachussetts at Amherst[3] I managed to make the INQUERY system deal with the document layout explained in the previous paragraph. I added a TIME field to the parser that contains the start time of a document. Using field indexing options from INQUERY, this enables the user to formulate queries for documents of 'this morning'.

I added a TIMES field to the parser to store the index table of {timestamp, caption}-tuples. Although INQUERY has been developed as a general retrieval engine, it is impossible to retrieve non-indexed parts of the documents. However, the best place to keep information that belongs to a document is within the document itself. Therefore, I had to use the undocumented get_raw_doc function call that retrieves the original unparsed document and reparse the retrieved document in my own code.

I will finish this section with an example of how the system should process a query. If the user queries for 'david goodnow', the closed caption documents stored in the INQUERY database collection are searched and INQUERY returns a list of documents ordered by decreasing probability of usefulness to the user. One of the documents that would be retrieved, is the example document given before. The system then reads the index table in the TIMES field for line 5 of the text and adds the 17 seconds to the document time stamp. Segment-24 is tuned to CNN and starts playing at 2:00:20, 17 seconds after the start of the document.

6 Conclusions and Further Work

The prototype application is a nice tool to experiment with. The implementation based on small building blocks realizes extendibility of the system. New approaches to automatic segmentation and analysis of the input data can easily be added and the improvements for retrieval can be studied. Once a standard test set has been defined, precision and recall measures can be used to find good representations of multimedia data from the viewpoint of information access.

A better document model is needed to store the documents. The segmentation information should be stored within the document. The same holds for the different representations of the data. The SGML/HyTime standard seems to be a good candidate for this purpose [VB95], [Erf93].

In theory, hooking up a speech recognizer with the prototype should be a fairly easy task. However, too many factors are unknown to predict whether this approach will result into a working product. It does not seem necessary to develop a new retrieval paradigm for speech data since we can use phoneme sequences.

Speech recognizers make mistakes. Erroneous data can confuse the information retrieval process. Intuitively, I think it is possible to extend the inference

[3] I want to thank Michelle LaMar for answering the many mail messages with questions and giving me the code of the Tcl interpreter extended with the INQUERY API calls.

network model with knowledge about the probability that a word was recognized correctly and the alternatives suggested by the recognizer. However, this still has to be proven. Implications of this idea for correctness of the inference network model have to be studied.

References

[Aro94] B.M. Arons. *Interactively skimming recorded speech.* PhD thesis, Massachusetts Institute of Technology, February 1994.

[BC92] N.J. Belkin and W.B. Croft. Information filtering and information retrieval: Two sides of the same coin? *Communications of the ACM*, 35(12):29–38, 1992.

[BCC94] E.W. Brown, J.P. Callan, and W.B. Croft. Fast incremental indexing for full-text information retrieval. In *Proceedings of the 20th International Conference on Very Large Databases (VLDB)*, Santiago, Chile, 1994.

[BCCM94] E.W. Brown, J.P. Callan, W.B. Croft, and J.E.B. Moss. Supporting full-text information retrieval with a persistent object store. In *EDBT '94*, 1994.

[CC93] J.P. Callan and W.B. Croft. An evaluation of query processing strategies using the TIPSTER collection. In *Proceedings of the sixteenth annual international ACM SIGIR conference on research and development in information retrieval*, pages 347–356, 1993.

[CCH92] J.P. Callan, W.B. Croft, and S.M. Harding. The INQUERY retrieval system. In *Proceedings of the 3rd international conference on database and expert systems applications*, pages 78–83, 1992.

[CHTB92] W.B. Croft, S.M. Harding, K. Taghva, and J. Borsack. An evaluation of information retrieval accuracy with simulated OCR output. In *Symposium of Document Analysis and Information Retrieval*, 1992.

[Cox90] S.J. Cox. *Speech and language processing*, chapter Hidden Markov Models for automatic speech recognition: theory and application, pages 209–230. Chapman and Hall, 1990.

[CW92] F.R. Chen and M.M. Withgott. The use of emphasis to automatically summarize a spoken discourse. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, San Fransisco, CA, March 1992.

[DLM+94] E. Deardorff, T.D.C. Little, J.D. Marshall, D. Venkatesh, and R. Walzer. Video scene decomposition with the motion picture parser. In *IS&T/SPIE Symposium on Electronic Imaging Science and Technology*, San Jose, 1994.

[dV95] A.P. de Vries. Multimedia information access. Master's thesis, University of Twente, August 1995.

[Erf93] R. Erfle. Specification of temporal constraints in multimedia documents using HyTime. *Electronic publishing*, 6(4):397–411, 1993.

[Fed] Federal Communications Commission. *15.119 Closed caption decoder requirements for television receivers.*

[GS92] U. Glavitsch and P. Schäuble. A system for retrieving speech documents. In *Proceedings of the 15th annual international SIGIR*, pages 168–176, Denmark, 6 1992.

[Hea94] M.A. Hearst. Multi-paragraph segmentation of expository text. In *ACL '94*, Las Cruces, 1994.

[LAF+93] T.D.C. Little, G. Ahanger, R.J. Folz, J.F. Gibbon, F.W. Reeve, D.H. Schelleng, and D. Venkatesh. A digital on-demand video service supporting content- based queries. In *Proceedings of the first ACM international conference on multimedia*, pages 427–436, Anaheim California, 1993.

[Les89] M. Lesk. What to do when there's too much information. In *Hypertext '89 Proceedings*, pages 305-318, New York, 1989. ACM.

[LPG+93] Levergood, Payne, Gettys, Treese, and Stewart. AudioFile: a network-transparent system for distributed audio applications. In *USENIX Summer Conference*, June 1993.

[Mae94] P. Maes. Agents that reduce work and information overload. *Communications of the ACM*, 37(7):31–42, July 1994.

[Ous94] J.K. Ousterhout. *Tcl and the Tk toolkit.* Addison-Wesley Publishing, 1994.

[Pea89] J. Pearl. *Probabilistic reasoning in intelligent systems.* Morgan Kaufmann, California, 1989.

[RHL94] Rudnicky, Hauptmann, and Lee. Survey of current speech technology. *Communications of the ACM*, 37(3):52–57, 1994.

[RS78] L.R. Rabiner and R.W. Schafer. *Digital processing of speech.* Prentice-Hall, New-Jersey, 1978.

[Sal89] G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer.* Addison Wesley Publishing, 1989.

[SBG94] A. Syrdal, R. Bennett, and S. Greenspan. *Applied speech technology.* CRC Press, Inc., Florida, 1994.

[SvR91] M. Sanderson and C.J. van Rijsbergen. NRT: news retrieval tool. *Electronic Publishing*, 4(4):205–217, 1991.

[SW] P. Schäuble and M. Wechsler. First experiences with a system for content based retrieval of information from speech recordings. http://www-ir.inf.ethz.ch/.

[TBC94] K. Taghva, J. Borsack, and A. Condit. Results of applying probabilistic IR to OCR text. In *Proceedings of the seventeenth annual international ACM SIGIR Conference on research and development in information retrieval*, Dublin, Ireland, 1994.

[TBCE94] K. Taghva, J. Borsack, A. Condit, and S. Erva. The effects of noisy data on text retrieval. *Journal of the American Society for Information Science*, 45(1):50–58, 1994.

[TC91] H. Turtle and W.B. Croft. Evaluation of an inference network-based retrieval model. *ACM Transactions of information systems*, 9(3), 1991.

[VB95] P.A.C. Verkoulen and H.M. Blanken. SGML/HyTime for supporting cooperative authoring of multimedia applications. In *Advanced Course: Multimedia Databases in Perspective*, pages 179–212. Center for Telematics and Information Technology of the University of Twente, 1995.

[vR79] C.J. van Rijsbergen. *Information retrieval.* Butterworths, London, 2nd edition, 1979.

[vS95] Hein van Steenis. Spraakherkenning levert eindelijk produkten op. *Automatiseringsgids*, May 26 1995.

[WB91] L.D. Wilcox and M.A. Bush. HMM-based wordspotting for voice editing and indexing. In *Proceedings of the Second European Conference on Speech Communication and Technology*, Genova, Italy, September 1991.

[Wil79] P. Willet. Document retrieval experiments using indexing vocabularies of varying size. II. Hashing, truncation, digram and trigram encoding of indexing terms. *Journal of Documentation*, 35(4):296–305, 1979.

[YGM] T.W. Yan and H. Garcia-Molina. SIFT - a tool for wide-area information dissemination. http://sift.stanford.edu/.

This article was processed using the LaTeX macro package with LLNCS style