Test conditions	maximum accuracy (test A)		real-time operation (test B)	
	recogn. rate	active states	recogn. rate	active states
speaker c (OOV: 3.2%)	92.5 %	5926	91.3 %	2326
speaker d (OOV: 2.2%)	87.7 %	8702	86.4 %	3097
speaker e (OOV: 4.3 %)	69.7 %	17 827	67.3 %	5820
speaker f (OOV: 3.6 %)	83.0 %	12 065	80.6 %	4131
speaker g (OOV: 3.3 %)	83.8 %	6804	81.8 %	2560
speaker h (OOV: 3.6 %)	88.4 %	6151	86.9 %	2396
weighted sum (OOV: 3.4 %)	84.1 %	9602	82.3 %	3395

Table 2: Recognition rates and number of active states for medical dictation (spontaneous speech)

The testset evaluated consists of 6 speakers. The results are shown for each speaker separately in order to exhibit the variabilities between the speakers. Recordings were done under real-world conditions in a hospital. Therefore, the recordings contain noise, hesitations, repetitions and restarts. Thus, the rate of OOV-words is shown, too.

Although the medical dictation task is more simple concerning the lower perplexity than the newspaper corpus, recognition rates are worse. This results from the different recording and speaking conditions and the high rate of OOV words (on average: 3.4%). Some physicians dictate very fast and use a high number of acronyms and short cuts. These short words are responsible for most recognition errors. Since the training corpus for the acoustical models contains read speech mainly the spontaneous speech of the medical application is modelled quite bad. Even more important is the point, that doctors dictated how they are used to, this means they spoke like dictating for a human secretary. Therefore, the number of corrections and restarts is very high, varying from speaker to speaker.

Especially interesting is the result of speaker e, for whom we got a significantly lower recognition accuracy. The reasons have to be evaluated in more detail, but two properties seem to be very important. At first, the speaker is highly uncooperative; its fraction of restarts and corrections is much higher than for the remaining speakers. Secondly, speaker e speaks very fast. This leads to a high number of deletions (11% of the total number of words) and a high number of confusions of word endings. Finally, we must note that for two speakers (e, f) real-time operation can not be granted using the parameters derived from the newspaper dictation task.

5 CONCLUSIONS AND NEXT STEPS

In extending the task to larger vocabularies, next steps for the improvement of the recognizer are additional training of both language model and acoustic models on larger databases. This includes integration of noise models and more spontaneous speech in the training databases. We are planning to train separate HMMs for female and male speakers and expect both, higher recognition performance and lower number of active states. More effort in the treatment of rapid speakers has to be made. Furthermore, we will implement a short time speaker adaptation algorithm for reduction of word error rate.

6 LITERATURE

[1] R. Scheifler, J. Gettys: The X Window System; ACM Transactions on Graphics, 5(2); pp. 79-109; ACM, April 1986

[2] T. Levergood, A. Payne, J. Gettys, G. Treese, L. Stewart: AudioFile: A Network-Transparent System for Distributed Audio Applications. Technical Report 93/8. Digital Equipment Corporation, Cambridge Research Lab, 11 June 1993.

[3] V. Steinbiss, B. Tran, H. Ney: Improvements in Beam Search; Proc ICSLP ‘94; pp. 2143-2146; Yokohama; Japan

[4] A. Hauenstein, Architecture of a 10 000 Word Real Time Speech Recognizer; Proc. Eurospeech ‘93; pp. 1829-1832; Berlin, Germany

[5] A. Hauenstein, E. Marschall; Methods for Improved Speech Recognition over Telephone Lines; Proc. ICASSP ‘95, pp. 425-428, Detroit, MI, USA.

[6] S. Ortmanns, H. Ney, A. Eiden; Language-Model Look-Ahead for Large Vocabulary Speech Recognition; Proc. ICSLP ‘96; pp. 2095-2098, Philadelphia, U.S.A.

[7] G. Neville-Neil: Current Efforts in Client/Server Audio. The X Resource, Issue Eight, O’Reilly & Associates Inc., 1993.

[8] P. Witschel; Constructing Linguistic Oriented Language Models for Large Vocabulary Speech Recognition; Proc. Eurospeech ‘93, pp. 1199-1202; Berlin, Germany.

- Command Mode: concerning the recognition task for commands like “copy”, “save”, “start dictation”, an immediate response is most important.
- Asynchronous Response: while the user still dictates a paragraph, the server starts and continues to deliver recognized text back to the client. Even when the user stops dictation, words still sticking in the servers recognition pipeline have to be transferred to the client as soon as they are completely processed.

For operating systems supporting thread programming feature extraction and emission probability calculation are encapsulated in different threads than the search task. Therefore they can be performed either in SW or using a dedicated HW for acceleration of response for dictation tasks.

4 APPLICATIONS

We present recognition results for two different dictation applications of the *SpeechDialog System*. For both applications, acoustic training was performed on 36 hours of speech (375 native german speakers) or 25 000 utterances. A fraction of 4.5 hours (78 speakers) were spontaneously spoken utterances, while the remaining training data was controlled (read) speech. The acoustical models contain 1532 different phoneme segments built up by 50 000 Gaussian density functions. Feature extraction and emission probability calculation are performed on a dedicated HW-board, designed for use in PCI-bus based PCs. The number of Gaussians was fixed in order to get predictable real-time operation by the dedicated HW-board. The application runs on Windows NT based PCs using Pentium and Pentium Pro processors.

4.1 Newspaper dictation (controlled speech)

A German newspaper corpus is used as evaluation set for dictation. The class based bigram language model comprises 10 055 phonetically different words. Due to properties of the language model this expands to 24 511 linguistically different words with different word transition probabilities. That results from the fact that many words are entered in multiple classes since they can have multiple linguistic properties (e.g. case, gender, number) [8]. The language model has a perplexity of 288.

Recognition rates and the average number of active states per frame (before pruning) are shown in table 1. Recognition rates are computed as usual (subtraction of substitutions, deletions, insertions). We tested under two different conditions.

- Test condition A: maximum recognition accuracy (no real-time operation guaranteed) for offline evaluation of the recognizer.
- Test condition B: real-time operation of search task on a 120-MHz-Pentium-PC (online mode). This means, full use of all accelerating search methods[1] presented in chapter 2.2 is made. The use of histogram pruning is of particular importance since this helps to restrict the computing power required to a maximum upper boundary so that real-time operation can be granted.

The testset comprises 6 speakers (male and female), which were divided in two subsets. The speakers read the test corpus, i.e. controlled speech had to be recognized. The rate of out-of-vocabulary (OOV) words is below 0.5%.

Test conditions	maximum accuracy (test A)		real-time operation (test B)	
	recogn. rate	active states	recogn. rate	active states
speaker set 1	92.6 %	5073	92.3 %	1815
speaker set 2	90.6 %	5892	87.2 %	1960

Table 1: Recognition rates and number of active states for newspaper dictation

As can be seen, recognition rates of more than 90% can be achieved. When adjusting the search parameters to real-time operation (test B) the number of active states are reduced to approximately 33-36% (compared to test A), while some recognition accuracy is lost. This loss is mainly due to use of phoneme fast look ahead and selection of a strict pruning threshold. Especially for speaker set 2, where recognition accuracy in test A is worse than for set 1, real-time operation leads to a larger loss in accuracy. This arises from the fact, that the search space has to be reduced more than for set 1. (Speakers with a higher recognition accuracy nearly always cause a smaller search space).

4.2 Medical dictation (spontaneous speech)

The medical corpus is used for dictation of reports in radiology departments of hospitals. The class based bigram language model consists of 10 194 words (linguistically expanded: 23 754 words) and has a perplexity of 54.

Recognition results are shown in table 2. Parameters for pruning, language model spreading and histogram pruning were taken from the previous experiments, i.e. no adaptation to the new task was done.

---

1. Phoneme fast look-ahead, histogram pruning, approximate language model spreading, and strict overall pruning

2.2 Search Strategy and Language Modelling

A forward beam search algorithm dealing with class based bigram language models [8] produces word hypotheses. The renouncement of using a multi-pass search strategy allows a direct output of the recognized strings without need of pauses between words or sentences. Word hypotheses are delivered as soon as all trace back lists contain the same history. This leads to a delay of only two or three words on average between dictation and display of the word by the application.

The pronunciation lexicon is organized internally as a tree-based lexicon with tree-copies for different predecessor words [3]. An optional phoneme fast look ahead algorithm based on phoneme units can be applied.

Furthermore, an optional histogram pruning may be used. This limits the number of active state hypotheses of the search space to a fixed number and reduces the search space up to 50% without loss of accuracy [3]. Histogram pruning is done on a "on-demand" basis. This means no additional computation is needed when the number of state hypotheses falls below the histogram pruning threshold ($maxHisto$). Furthermore, during each time step all hypotheses are expanded even if $maxHisto$ is exceeded. Thus, pruning is performed during the next time frame by re-adjusting the "standard" pruning threshold so that exactly $maxHisto$ hypotheses are expanded. Therefore, computationally intensive sorting of active state lists can be avoided.

Additionally, a simple "*approximate language model spreading*" algorithm is implemented (compare to [6]). This means that a small default language model penalty P_{def} is applied every d_{spread} frames. d_{spread} may vary between 4 and 16 frames; values corresponding to the length of a phoneme showed best performance. The probability for state i in frame t applying language model spreading is computed as:

$$p_i(t) = \max_{\sigma}\{q_i(x_t|\sigma) \times p_{\sigma}(t-1)\} \times P_{def}, \qquad (1)$$

where $q_i(x_t|\sigma)$ is the product of transition and emission probability of the underlying HMM (σ represents all predecessor states of state i). At word ends (current word: w, predecessor: v) using a bigram language model $p_i(t)$ is computed as follows:

$$p_i(t) = \max_{\sigma}\{q_i(x_t|\sigma) \times p_{\sigma}(t-1)\} \times \frac{P_{LM}(w|v)}{(P_{def})^{N_{spread}}}, \qquad (2)$$

where N_{spread} denotes for the number of applications of the default language model score P_{def}. It is computed to:

$$N_{spread} = \begin{cases} l_w / d_{spread} & \text{if} \quad l_w < N_{spread,\,max} \\ N_{spread,\,max} & \text{else} \end{cases} \qquad (3)$$

where l_w is the word length (in frames) of word w. This means language model spreading is applied at most $N_{spread,\,max}$ times per word in order to prevent pruning errors at word ends due to division by very small numbers in eq. (2) (Note that $P_{def} < 1$!).

In contrast to previous algorithms [6], no consideration of the actual language model penalties to be applied at the word-end is taken. Therefore, memory demanding language model look ahead scores for all branches of the search tree need not to be stored. Instead, the exact language model penalty is applied at word ends under consideration of the default look-ahead penalties applied before.

Approximate language model spreading helps to reduce error rates when introducing strict pruning. This results from the fact that the pruning threshold can be adjusted to a more tight pruning at word ends when the language model is applied. On average we reached a reduction of the search space of 28% without loss of accuracy for 10 kWord recognition tasks.

3 CLIENT/SERVER ARCHITECTURE

The authors developed an extended version of the AudioFile System (AF) [2],[7], which provides clients access to audio input and output in a machine-independent and network-transparent way. In the process of development, our new, *SpeechDialog System* evolved supporting various speech recognition functionalities.

Three application modes are realized:

- Online: speech is recognized in real-time and text is immediately available at the client (e.g. word processor).
- Offline: speech is sampled and saved as audio data, translated to text as a batch job and entered into the application as a whole when the complete job is done.
- Remote-Online: speech is transferred from the clients computer to the speech recognition server, is recognized and transferred back as text to the client in real-time over LAN.

Having a closer look at how the user will interact with a dictation application reveals requirements related to the recognition tasks and the communication between client and server:

- Dictation Mode: the recognition task has to cope with spoken words from one, typically very large application.

A PC-BASED REAL-TIME LARGE VOCABULARY CONTINUOUS SPEECH RECOGNIZER FOR GERMAN

Meinrad Niemöller, Alfred Hauenstein, Erwin Marschall, Petra Witschel, Ulrike Harke

Siemens AG, Dept. ZT IK 5, Otto-Hahn-Ring 6, 81730 München, Germany

e-mail: Alfred.Hauenstein@mchp.siemens.de

ABSTRACT

A large vocabulary speech recognizer for German is presented. The main properties of the recognizer are speaker independence, continuous speech input and real-time operation. It is integrated into a client/server framework, which allows for simple porting between different hard- and software platforms.

Methods like simplified language model spreading in beam search and specialized word-begin and -end modelling are introduced in order to achieve real-time operation on a Pentium-based PC. Recognition tests for two different dictation applications (controlled speech newspaper dictation and spontaneous speech medical dictation) are presented showing the importance of adding efforts in the modelling of spontaneous speech.

1 INTRODUCTION

In the recent years continuous speech recognition emerged as an enabling technology for different kinds of applications.

To fill the gap between powerful recognition algorithms on the one side, and various kinds of applications on the other side, a software architecture is needed which supports integration in a flexible and extensible way. Based on the experience with client/server architectures for graphical and audio applications [1], [2], the authors implemented a real-time large vocabulary speech recognizer. The system is characterized by following design features: The recognizer core in the server allows multiple clients, it supports a variety of underlying hardware, and permits transparent access through a computer network (LAN). This framework accesses a speech recognition core using state-of-the-art recognition technology. This includes acoustical modelling using Continuous Density HMMs (CD-HMMs), linear discriminant analysis, and tree-based search algorithms.

2 SPEECH RECOGNITION TECHNOLOGY

2.1 Acoustic-Phonetic Modelling

Sampling is done at 16 kHz with 14 bit resolution using head-set microphones. For acoustical modelling we use CD-HMMs applying Gaussian density functions.

For each time frame (10 ms spaced) a 63 dimensional feature vector is extracted (30 mel-scaled cepstral, 15 Δcepstral, 15 $\Delta\Delta$cepstral, 1 energy, 1 Δenergy and 1 $\Delta\Delta$energy component). In order to minimize the effect of channel variations due to use of different microphone types, a maximum likelihood channel compensation like in [5] is applied using an estimation window of 256 frames.

Additional context is employed by building a “super-vector” consisting of 3 adjacent frames. After applying a Linear Discriminant Analysis (LDA) [5] we end up with a 47 dimensional feature vector as input for emission probability calculation.

The recognizer has a base inventory of 38 different phonemes. Each phoneme consists of three segments. The segments are context dependent diphone models. In order to handle segments which are seen rarely in the training material, a statistically based tying strategy for segments is applied. This means, segments whose number falls below a certain threshold are tied with other segments having the same left or right context.

In order to simplify and speed up the search step specialized word-begin and word-end segments are introduced instead of using crossword phoneme segments. This allows to model coarticulation effects at word boundaries without additional effort during beam search. Furthermore, if recognition accuracy is a concern, it is advantageous to use phoneme segment models that model explicit word-begins and -ends. It is superior over using segments that are tied over all different representations of word boundaries (with the exception of “long silences”) as done using “standard” crossword phoneme segments.