ELSEVIER

Speech Communication 37 (2002) 109–131

SPEECH COMMUNICATION

www.elsevier.com/locate/specom

Large vocabulary continuous speech recognition of Broadcast News – The Philips/RWTH approach

P. Beyerlein [a,*], X. Aubert [a], R. Haeb-Umbach [a], M. Harris [a], D. Klakow [a], A. Wendemuth [a], S. Molau [b], H. Ney [b], Michael Pitz [b], A. Sixtus [b]

[a] *Philips Research Laboratories, Weisshausstrasse 2, D-52066 Aachen, Germany*
[b] *Lehrstuhl für Informatik VI, Aachen University of Technology, D-52056 Aachen, Germany*

Abstract

Automatic speech recognition of real-live broadcast news (BN) data (Hub-4) has become a challenging research topic in recent years. This paper summarizes our key efforts to build a large vocabulary continuous speech recognition system for the heterogenous BN task without inducing undesired complexity and computational resources. These key efforts included:

- automatic segmentation of the audio signal into speech utterances;
- efficient one-pass trigram decoding using look-ahead techniques;
- optimal log-linear interpolation of a variety of acoustic and language models using discriminative model combination (DMC);
- handling short-range and weak longer-range correlations in natural speech and language by the use of phrases and of distance-language models;
- improving the acoustic modeling by a robust feature extraction, channel normalization, adaptation techniques as well as automatic script selection and verification.

The starting point of the system development was the Philips 64k-NAB word-internal triphone trigram system. On the speaker-independent but microphone-dependent NAB-task (transcription of read newspaper texts) we obtained a word error rate of about 10%. Now, at the conclusion of the system development, we have arrived at Philips at an DMC-interpolated phrase-based crossword-pentaphone 4-gram system. This system transcribes BN data with an overall word error rate of about 17%.

Zusammenfassung

Die automatische Spracherkennung von aktuellen Nachrichtensendungen (“Hub-4” Aufgabe, Broadcast-News Aufgabe) ist in den vergangenen Jahren zu einem wichtigen Forschungsthema geworden. Diese Publikation faßt die Schwerpunkte unserer Arbeit beim Aufbau eines Systems zur Erkennung kontinuierlicher Sprache mit großem Vokabular für die heterogene Broadcast-News-Aufgabe zusammen, wobei wir versucht haben, die Komplexität und den Rechenaufwand des Systems so gering wie möglich zu halten. Unter anderem haben wir uns auf folgende Ziele fokussiert:

- Automatische Segmentierung des Audio-Signals in sprachliche Äußerungen;
- Effiziente einstufige Trigramm-Suche mit Look-Ahead-Techniken;

* Corresponding author.

0167-6393/02/$ - see front matter
PII: S0167-6393(01)00062-0

- Optimale log-lineare Interpolation einer Anzahl von akustischen Modellen und Sprachmodellen mit Hilfe der Diskriminativen Modellkombination (DMC);
- Behandlung von Kurzzeit- und schwachen Langzeitkorrelationen in natürlicher Sprache durch den Einsatz von Phrasen und von Abstands-Sprachmodellen;
- Verbesserung der akustischen Modellierung durch eine robuste Merkmalsextraktion, Kanalnormierung, Adaptionstechniken, wie auch durch automatische Skriptselektion und Skriptverifikation.

Der Startpunkt unserer Systementwicklung war das Philips 64k-NAB wortinterne Triphon-Trigramm-System. Auf der sprecherunabhängigen aber mikrophonabhängigen NAB-Aufgabe (Transkription von vorgelesenen Zeitungstexten) erreichten wir eine Wortfehlerrate von ca. 10%. Die Entwicklungsarbeit wurde mit dem Aufbau eines DMC-interpolierten phrasenbasierten wortübergreifenden Pentaphon-Viergramm-Systems abgeschlossen. Dieses System transkribiert Nachrichtensendungen mit einer Gesamtfehlerrate von ca. 17%.

Résumé

La transcription automatique d'émissions parlées d'informations radio-télévisées (tâche désignée par "Hub-4") a été l'objet d'intenses travaux de recherche ces dernières années. Ce papier présente les lignes principales de nos efforts d'élaboration d'un système de reconnaissance de parole continue qui soit à même de traiter le signal hétérogène provenant d'émissions d'information sans entraîner une trop grande complexité ou le recours à des ressources de calculs excessives. L'essentiel de nos efforts a porté sur les points suivants:

- La segmentation automatique du signal audio en une suite de passages parlés;
- Le décodage rapide en une passe intégrant un modèle de trigrammes avec une technique d'anticipation;
- L'interpolation log-linéaire optimale d'une variété de modèles acoustiques et grammaticaux au moyen d'une technique de combinaison discriminative de modèles (DMC);
- La prise en compte de corrélations linguistiques à court terme et, plus faiblement, à long terme au moyen de groupements de mots (phrases) et de modèles de languages dits "à distance";
- L'amélioration de la modélisation acoustique à l'aide d'une extraction robuste du contenu du signal combinée à la normalisation des canaux, l'adaptation des modèles phonétiques ainsi que la sélection et la vérification des scripts du corpus d'entraînement.

Notre point de départ fut le système Philips "NAB-64k" fondé sur l'emploi de triphones intra-mots et de modèles de trigrammes. Pour la tâche "NAB" impliquant la transcription d'articles lus à l'aide d'un microphone connu, ce système indépendant du locuteur atteint un taux d'erreur moyen de 10% au niveau du mot. Au terme de ce travail, nous avons développé un système qui combine par DMC des modèles phonétique intra-mots et inter-mots, des pentaphones, des groupements de mots ainsi que des modèles de langage jusqu'à l'ordre 4. Ce système produit une transcription d'émissions parlées d'information avec un taux d'erreur global d'environ 17%.

Keywords: Broadcast news; Hub-4; Automatic segmentation; Speaker clustering; Time-synchronous one-pass trigram decoding; Language-model look-ahead; Discriminative model combination; Log-linear interpolation; Distance language models; Phrases; Vocal tract normalization; Script verification

1. Introduction

Past speech recognition research has focused mainly on the decoding of high quality speech in quiet environments. Recently, however, the focus has shifted to speech found in the "real world". One of the data sources of real-world speech are audio recordings from radio and television broadcast news (BN). As compared to previous work involving automatic speech recognition, the BN task imposes the following additional research problems:

- Unknown sentence boundaries.
- Diverse and rapidly changing acoustic environment. Typical degradations of the speech signal are introduced by background music, noise, interfering speakers as well as by changes between studio and telephone channels. Furthermore, regional dialects or accents of non-native speakers have to be considered.

- Real-life speaking styles (spontaneous speech) as well as unknown speaker turns. Speaking styles range from carefully read speech to free and spontaneous conversation.
- Natural language. Difficulties arise from unpredictable changes of topics of the BN as well as from spontaneous reactions in free conversations.

This paper summarizes our approach in dealing with these challenges and describes the system we developed between 1997 and 1998.

2. Overview

The system architecture of the Philips/RWTH Hub-4 system is plotted in Fig. 1. The system consists of three decoding stages: segmentation, one-pass trigram decoding and discriminative model combination (DMC). The task of the segmentation stage is to handle the problem of unknown sentence boundaries. It transforms the continuous BN audio stream in a sequence of spoken utterances (segments), which are similar to sentences. Identification of acoustic channel bandwidth, gender and speaker cluster is provided. Given this set of spoken utterances a one-pass trigram decoding is performed, aiming at compact lattices with a high linguistic coverage. The lattices are rescored using all available acoustic and language models (we used 7 models). A weighted combination of the model scores is used as the decision criterion for the final transcription. The individual weights of the models are optimized with respect to the word error rate (WER) during the training stage. In the following three sections, we describe in more detail the three stages of the recognition process:

- segmentation of the audio stream;
- one-pass trigram decoding for adaptation and lattice generation;
- integration of multiple acoustic and language models by DMC.

We continue with two further sections on building the acoustic and language models needed for one-pass decoding and for DMC.

A couple of results presented here, were produced at the Aachen university of technology (RWTH). They are labeled in this paper with "RWTH".

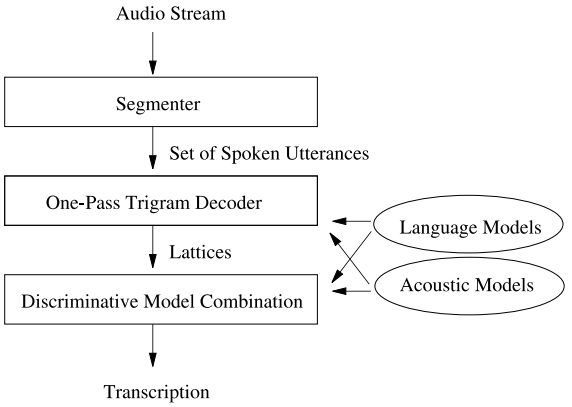

Fig. 1. Architecture of the Hub-4 system.

3. Automatic segmentation into "sentences"

In most transcription tasks boundaries of the utterances are known, and the background acoustic conditions of the utterances are fixed. Further information may also be available such as gender or channel information. Using the given information, models can easily be adapted to the conditions at hand. A BN transcription system may receive, for example, a complete 3 h input stream. In this stream, one encounters, for example, telephone speech, speech in noisy "real life" surroundings, spontaneous speech (as opposed to planned, or read speech) and non-speech (such as music, traffic noise, etc.). Different adapted models should be used in transcribing speech in these different conditions, and thus it is important to be able to divide the stream into *segments*. These segments, each contain just one speaker, speaking in uniform background conditions. Some information such as gender and channel information for each segment may also be useful in the decoding. Non-speech segments should be discarded. Finally, creation of homogeneous clusters of segments is important to adapt to specific speakers and background conditions. Two systems of segmentation are discussed in the following sections: (1) a two-pass system, consisting of a non-speech decoder and a BIC-based speaker turn detector and (2) a one-pass phoneme decoder.

3.1. Removal of non-speech

Segments containing only noise, when decoded, lead to spurious transcriptions. Such spurious transcriptions can usually be removed later using simple confidence measure techniques. It is, however, useful and more efficient, to remove such non-speech segments before decoding. In doing this, it is important not to remove true speech, as then this speech will never be decoded, resulting in unrecoverable errors.

Non-speech decoder: Different Gaussian mixture models (GMM) of speech and non-speech were used to decode the input signal. This approach has been used before (Jin et al., 1997; Hain et al., 1998). The Philips implementation is described in (Harris et al., 1999). We chose to construct models of speech, speech + background music, non-speech noise, music and of pause. Sixty minutes of each of speech, speech with music, and music was used for training. Short passages of noise and silence in the above was used to train these models. In total, around 1000 Gaussian densities were used. The five HMM models were trained using standard MFCC features. Normalization and transformation techniques, as applied in the frontend of the speech recognizer (see Section 7.1), were excluded from the feature extraction. They would normalize away the difference between the different classes and would make the speech–non-speech separation more difficult.

Recognitions were done using the above trained models and penalties for moving between models were applied in order to smooth the recognition.

Phoneme decoder: An alternative to the previous scheme, is to carry out a Viterbi decoding using regular phoneme models and a phoneme bigram model.This decoding method was used to generate the segmentation, as well as to detect the non-speech passages. Here, the HMM model set consisted of both male and female phonemes and one noise HMM model was used to model all non-speech. Segmentation was achieved by creating segments with male–female or speech–non-speech transitions as segment boundaries, and long non-speech passages were discarded.

Comparison for the removal of non-speech: There are two types of classification errors possible. Firstly, non-speech can be classified as speech. Resulting segments with mostly non-speech can be discarded later as they are decoded with a bad score. Other short non-speech passages may remain between spoken words (pauses). To evaluate the segmenter performance, we counted the amount of non-speech in the hand-made official NIST segmentation and computed the *additional amount of non-speech* left by the automatic segmenter. Secondly, speech can be classified as non-speech and discarded. The discarded words cannot be recognized, resulting in deletions – an unrecoverable error. Another error that can occur is that parts of words are lost due to misplaced non-speech passage boundaries. A word may be cut at such a boundary, resulting in part of the word lost. These error figures are given in Table 1. We found that the non-speech decoder was better at removing non-speech than the phoneme decoder, resulting in a solid starting point for the subsequent segmentation.

3.2. Segmentation and classification

Segmentation algorithms were used to determine positions of speaker and background change, and thus segment boundaries. The segmentation generated from the phoneme decoder, described in Section 3.1, was compared to the segmentation obtained by the BIC-approach (Chen and Gopalakrishnan, 1998; Harris et al., 1999). Given a passage of BN data, the BIC method is able to find the most likely position of speaker or background condition change as BIC actually looks for positions where the signal characteristics change. It

Table 1
Misclassification of speech and non-speech on Hub-4'97 evaluation set

Non-speech detector	Speech misclassified as noise (%)	Additional amount of non-speech (%)	Words cut (%)
Phoneme decoder	0.35	4.81	0.049
Non-speech decoder	0.26	0.29	0.055

Table 2
Segment boundary detection on Hub-4'97 evaluation set

Segmenter	Segment purity	Average segment length	Words cut (%)
Phoneme decoder	97.6	7.33	0.241
BIC	97.7	18.86	0.067
Official	100	15.87	0

also gives a criterion to determine whether the change at this point is significant.

Comparison of segmentations: One can judge the quality of a segmentation by the *speaker* (*cluster*) *purity* of its segments. The speaker (cluster) purity is the percentage of time that the main speaker of a particular segment (cluster) is speaking in that segment (cluster). These values for the two different segmentation are given in Table 2. The purity of the segments generated by the two methods is comparable. But the average BIC segment length is much greater than that in the phoneme decoder segmentation. Further, the BIC algorithm produces fewer word cuts than the phoneme decoder.

Classification: We classified each segment as being either male or female and as being either telephone or non-telephone. Using this information improved the subsequent clustering of the segments and could also be used in the decoding later. A segment was deemed to be a telephone segment if most of the signal energy was in the 300–3500 Hz range and non-telephone otherwise. To determine the gender of a segment, monophone male and female phoneme decoding runs were carried out.

3.3. Clustering

As mentioned previously, the aim of the clusterer is to group together "similar" segments. The distance between two segments is based on the Kullback–Leibler distance (KL2), first used in this context in (Siegler et al., 1997). This measures how "different" the segments are acoustically. Segments that are close together in time are more likely to come from the same speaker as ones that are greatly temporarily separated. For each segment S we estimate a single Gaussian probability density $G(\vec{x})$, which describes the acoustic observations $\vec{x}$ of segment S in a compact way. Now, let S_1, S_2 be two segments and let G_1, G_2 be the corresponding Gaussians. Let $\Delta T(S_1, S_2)$ be the temporal distance of both segments in terms of acoustic frames. The distance used for the clustering is

$$d(S_1, S_2) = \mathrm{KL2}(G_1, G_2) + \beta \cdot \Delta T(S_1, S_2). \quad (1)$$

The segments resulting from the BIC segmentation were clustered using a nearest-neighbor algorithm: at each stage, the two closest clusters (according to d) were merged together. For a contrast, the segments from phoneme decoder segmentation were clustered using a greedy criterion. At each stage, a segment was chosen, and clustered together with the first segment found within a certain distance (measured according to d) of it. We measured the effectiveness of the clustering algorithm by the cluster purity. The results of this are given in Table 3. It is advantageous to perform clustering after gender classification of the individual segments.

Table 3
Cluster purity, framewise gender accuracy and word error rate (WER) on the Hub-4'97 evaluation set

System	Cluster purity	Gender accuracy	WER (%) no adaptation	WER (%) after adaptation
Phoneme decoder + greedy clustering	73.6	97.49	23.7	22.6
BIC + nearest neighbor clustering (clustering after gender classification)	89.2	97.87	23.4	21.0
BIC + nearest neighbor clustering (clustering before gender classification)	89.1	96.02	23.4	21.3
Official	100	100	21.8	20.0

This improves the cluster purity and the framewise gender accuracy (see Table 3).

3.4. Effect on speaker normalization and adaptation

A gender-dependent one-pass trigram decoding with word internal triphones was carried out to show the influence of the segment/cluster purity on the recognition performance. A first recognition was done without adaption and, subsequently, a recognition using vocal tract normalization (VTN) and MLLR adapted models (Section 7.1). We see that a high cluster purity (given in Table 3) is an important factor in maximizing gains from adaptation techniques. Adaptation brought a 10.3% improvement in the BIC system, compared to a 4.6% improvement in the phoneme decoder system. Adaptation using the official (hand-made) clustering and segmentation reduced the error from 21.8% to 20.0% (an 8.3% relative improvement). The baseline error rate before adaption using the official segmentation was, however, lower. We also saw the importance of optimal gender classification.

3.5. Summary

Two automatic segmentation approaches – (1) a phoneme decoder and (2) a GMM–BIC segmenter – were compared. The GMM–BIC segmenter provides better results (Harris et al., 1999). The loss of word accuracy by automatic segmentation compared to manual segmentation is about 5% relative.

4. Efficient one-pass trigram decoding

Like most other Hub-4 systems, a 64k word trigram recognition coupled with the use of triphone models is applied in the early decoding stages to the speech utterances, obtained from the segmenter (Section 3). Longer linguistic and acoustic contexts can also be handled, though, in later stages when the search is restricted to a word lattice (Section 5). The prime decoding task thus consists in performing a first "robust" search that fulfills the requirements of a trigram language model and produces both the best sentence hypothesis for the sake of acoustic adaptation as well as a word lattice. This lattice should include the most likely alternatives for further rescoring with more complex knowledge sources in the DMC stage (see Section 5).

4.1. Representation and organization of the search space

The one-pass decoder is still based on a time-synchronous left-to-right beam search technique with a prefix tree structure of the lexicon (Ney et al., 1992). It proceeds by integrating the knowledge sources in a "breadth first" search strategy, similar to the decoder described in (Odell et al., 1994). Let N_W be the size of the vocabulary denoted as $\mathscr{W}$. In the framework of large vocabulary m-gram decoding, the search proceeds in a *four*-dimensional space, the coordinates representing, respectively, the time index, the language model node, the phoneme arc and the acoustic HMM state. The last two coordinates specify the position in the actual word model with respect to the lexical tree organization while the second axis determines the predecessor word history as taken into account by the language model. In a time-synchronous decoder, the time index is the independent variable and, based on the principle of dynamic programming, the whole search process is formulated as a recurrence from $t-1$ to the "current" time t. The active parts of the search space that are explored during decoding are thus described in terms of the last three coordinates with the time index being implicit. In the present implementation, partial hypotheses expanded by the decoder are recorded in lists organized in a three-level hierarchy, namely, language model nodes, phoneme arcs and HMM states (Ney et al., 1992). This means that all active paths that share the same predecessor word history are grouped together and are further expanded according to the lexical tree structure until they reach a next word ending. This algorithmic organization has been sometimes referred to as *using word-conditioned copies of the lexical tree*. However, this is just a mental view, the lexical tree structure being stored only once and for all. Only the dynamic programming quantities (score, backpointer, arc index, etc.) that are relevant to the still

active paths are recorded separately. The propagation of the search is controlled by a standard beam pruning strategy and peaks in terms of a large number of active states are handled by the so-called histogram pruning method (Steinbiss et al., 1994). This technique provides a very efficient way of selecting the top-N most promising paths, making it possible to work with lists of fixed size of $O(N)$.

4.2. m-gram Language-model constraints

The generalization to longer-span m-gram language models ($m > 2$) simply follows from a proper definition of the language nodes which depend on their $m-1$ predecessor word history, and from the use of a hash table to insure the efficiency of the recombination stage (Ortmanns et al., 1996b, 1997). Indeed, the use of a probabilistic m-gram language model has two well-known implications:

1. The search network is fully branched at the word level, each word being possibly followed by any other.
2. Word probabilities depend on their $m-1$ predecessors:

$$P(W_n|W_{n-1}, W_{n-2}, \ldots, W_0) = P(W_n|W_{n-1}, \ldots, W_{n-m+1}).$$

The dynamic programming principle requires to keep track of the individual $m-1$ word histories until the optimization step can take place at the next word ending. This leads to the definition of language model nodes for recombination that are made dependent on the last $m-1$ words. When using within-word acoustic models, each node is further connected to the whole set of words in the lexicon. This actually means re-entering the lexical prefix tree structure at its root. As an example, when decoding with a trigram language model and within-word phonetic models there is a total of $(N_W \times N_W)$ word pair nodes, each one linked to the whole lexicon. In contrast with bigram decoding where the N_W nodes can be directly associated to word end indices and stored in a static array, the extension to longer span language models asks for another solution owing to the huge number of possible language model nodes. For a 64k trigram, this number is actually close to 4.1 billion! Since only a very small fraction of all possible language model nodes are activated simultaneously in the course of the search process, hashing appears to be the ideally suited technique. Our solution consisted of associating each active node with a bijective index from which standard hashing provides an entry to the recombination table. For a trigram node $\{u, v\}$, the hash key is defined as

$$H(u, v) = M_W * u + v, \quad u, v > 0, \quad M_W > N_W, \tag{2}$$

and is subjected to a modulo operation for providing a first entry in the table. The bijective character of the hash key allows to easily retrieve the language model history by successive modulo and division operations. This method is fairly general in its principle (apart from secondary range problems) and has been successfully applied so far up to a 4-gram language model using a hash table of moderate size, typically, a few ten thousand entries for a 64k vocabulary (Aubert, 1999). This last reference also explains the generalization of the present m-gram decoding algorithm for cross-word phonetic models.

4.3. Look-ahead of language model probabilities

A well-known problem when using a prefix tree is that word identities are only known at the tree leaves. Postponing the use of the language model probabilities up to this point is disadvantageous since (1) the language model predictive capabilities are delayed and (2) the accumulated scores incur clear discontinuities at word-ends, both factors affecting the pruning efficacy. The solution consists in distributing the language model scores across the lexical tree by factorizing the word probabilities such that they can be applied incrementally at each phone arc, a process we have called "smearing" (Steinbiss et al., 1994). Smearing the exact m-gram scores appears computationally expensive due to the dependency on the $m-1$ predecessors. Therefore in (Aubert et al., 1994) a simplified approach has been followed by smearing the unigram scores that can be easily pre-processed and stored (Steinbiss et al., 1994). The topic of making early use of language model constraints has also been

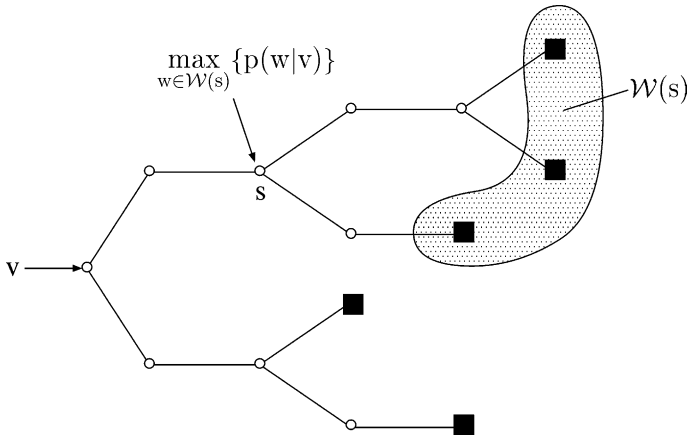

Fig. 2. Anticipated probabilities for language model look-ahead.

addressed in several other systems (Odell, 1995; Alleva et al., 1996). In this decoder, bigram scores are smeared over the lexical tree following the method in (Ortmanns et al., 1996a). This involves a compact tree configuration and a look-ahead cache in conjunction with a fast access to the language model probabilities such that any needed partial bigram scores can be made available on demand. More precisely, language model look-ahead pruning is achieved by anticipating the language probabilities as a function of the nodes of the lexical tree, each node being associated to the *maximum* language model probability over all words that can be reached from that particular node. The concept of anticipating the language probabilities for each node of the lexical tree is illustrated in Fig. 2. Using the bigram conditional probability $p(w|v)$, the anticipated language probability $\pi_v(s)$ for state s and predecessor word v is defined as

$$\pi_v(s) := \max_{w \in \mathscr{W}(s)} p(w|v), \tag{3}$$

where $\mathscr{W}(s)$ is the set of words that can be reached from the tree state s. Strictly speaking, we should use the tree nodes (or arcs) rather than the states of the hidden Markov models that are associated with each arc. However, each initial state of a phoneme arc can be identified with its associated tree node. The anticipated language model probabilities are combined with the scores of the state hypothesis $(t, s; v)$ during the search process and thus enhance the beam pruning efficacy since more state hypotheses can be discarded. The entries of the table $\pi_v(s)$ have to be computed *on demand* and are cached in a look-up table designed to keep a maximum number of look-ahead trees typically comprised between 500 and 1000. In addition, the anticipated language probabilities are only computed for the first four arc generations of the lexical tree. Informal experiments have shown that there is no significant advantage for considering more generations. More details can be found in (Ortmanns et al., 1996a). As the experiments show, when bigram scores are smeared instead of unigram scores, the number of states that have to be expanded is significantly reduced and less search errors occur (see Table 4), although this is done at the expense of a significant memory overhead.

4.4. Trigram decoding experiments on the Eval'97 test set

A number of trigram decoding experiments were carried out to evaluate the benefit of the new search algorithms. These experiments were run on the so-called partitioned evaluation set of Nov'97, using the correct segmentation provided by NIST. This represented 2.9 h of recordings pre-processed as a set of 657 segments with a total of 32,832 spoken words including hesitations. The acoustic modeling was provided by gender-dependent mixture densities of within-word triphones trained on 96 h of BN data (see Section 7.3, BN-96 h

Table 4
Bigram versus unigram language model look-ahead in one-pass 64k trigram decoding using the RWTH Hub-4 system on the Hub-4'97 PE evaluation set

Speech quality	Look-ahead	States	Arcs	Trees	WER (%)	Decoding time
A: all focus conditions	Unigram	33,675	9528	63	24.4	100 (ref.)
		58,429	16,627	115	24.2	135
		243,555	70,896	215	23.4	254
	Bigram	26,770	7464	64	24.0	94
		34,367	9516	71	23.8	110
		68,207	18,075	89	23.6	132
B: only F0 condition	Unigram	25595	7216	44	15.5	100 (ref.)
		53,960	15,194	87	15.2	142
		206,241	59,992	155	14.7	256
	Bigram	18,929	5275	42	15.2	90
		25,880	7143	48	15.1	108
		40,414	10,792	55	14.9	118

corpus) without any further acoustic adaptation. The first contrastive test concerned the comparison of one-pass trigram decoding with the former two-pass strategy. Note, however, that the reported experiments were all run with the new decoder including bigram look-ahead pruning. The bigram decoder used in the Hub-4'97 evaluation was slower by a factor of at least two. The word lattice produced with the bigram decoding was generated with conservative pruning beams, insuring a large density. The bigram lattice achieves a graph error rate of about 7%. As can be seen from Table 5, compared to the two-pass strategy the one-pass trigram decoding achieved a slightly lower error rate while being about 10% slower in terms of the overall decoding time (117 time units versus 107). Although not significant, the slight gain in accuracy observed in the one-pass results might be attributed to (1) the word-pair approximation made in the lattice rescoring stage (Aubert and Ney, 1995) and (2) a reduced number of search errors when applying the trigram from the beginning. The second contrast compared a one-pass trigram decoding for two different setups with a vocabulary of 20k and 64k words, respectively. The main factor concerned the influence of the out-of-vocabulary (OOV) words on the error rate together with the relative increase of the search cost when the lexicon is increased by a factor of 3, from 24,780 to 72,965 entries. As can be seen in Table 6, the 64k setup provided about 10% relative error reduction with respect to the 20k setup, the

Table 5
One-pass versus two-pass 64k trigram decoding on Hub-4'97 PE evaluation set

	2G Decoding	3G Rescoring	3G 1-Pass decoding
Word error rate (%)	24.2	21.7	21.1
2G → 3G Improvement	Reference	−10.5%	−12.8%
2P → 1P Improvement	–	Reference	−2.6%
Relative decoding time	100	7	117

Table 6
One-pass trigram decoding for 20k and 64k Vocabulary on Hub-4'97 PE evaluation set

3G 1-Pass decoding	20k Setup	64k Setup	Difference
Out-of-vocabulary rate	2.4%	0.6%	−1.8% Absolute
Word error rate (%)	23.6	21.1	−10.6% Relative
Relative decoding time	100	125	+25% Relative

decoding cost being increased by one-quarter. It is interesting to note that the absolute reduction of the error rate (2.5%) represented about 1.4 times the reduction of OOV words. Note also that the increase of the 64k decoding costs was partly due to the larger language model. The third contrastive experiment concerned the language model look-ahead pruning technique when the bigram probabilities are smeared instead of just the unigram ones. When using bigram instead of unigram probabilities, the look-ahead memory costs are dramatically increased since the unigram probabilities do not depend on word history and can simply be stored in a static array. As the experiments show, however, unigram look-ahead is significantly outperformed by bigram look-ahead in terms of number of active states that have to be expanded for obtaining the same accuracy. With respect to the former experiments, the results presented in Table 4 have been obtained with a smaller set of acoustic models which explains the higher error rates (24% versus 21%).

Part A of Table 4 shows that the error rate averaged over all focus-conditions does not vary dramatically from one experiment to the other. Nevertheless, for reaching a desired error rate the figures indicate that bigram look-ahead is able to reduce the number of hypotheses by an average factor of two or more with respect to unigram look-ahead. Looking at the specific conditions, it appears that bigram look-ahead does not perform equally well for each "focus". Part B of Table 4 relates only to the F0 condition, i.e. clean and planned broadcast speech representing 42% of all Nov'97 data. Here the benefit of bigram look-ahead with respect to unigram look-ahead appears clearer in terms of number of active hypotheses after pruning. This corresponds to our results obtained on the NAB'94 read speech corpus (Ortmanns et al., 1996a). On the other hand, when looking at more difficult conditions, for example F3 (speech in the presence of background music) or F4 (speech under degraded acoustical conditions), bigram look-ahead does not seem to have any real advantage over unigram look-ahead. This might be an indication that the language model constraints are not powerful enough for these difficult decoding conditions.

4.5. Summary

Summarizing the results of the experiments that have been reported above, three main conclusions can be drawn concerning the "prime" trigram decoding stage when applied to the transcription of BN data:

1. One-pass trigram decoding compares favorably with a two-pass strategy, the overall computational costs – including memory and CPU time – being only moderately increased.
2. Due to the prefix tree organization of the lexicon, decoding costs show a relatively slow increase when the vocabulary is enlarged from 20k to 64k.
3. Compared with unigram look-ahead, bigram look-ahead is able to reduce significantly the search cost for most focus conditions, even though the additional memory costs cannot be neglected.

5. Discriminative model combination

During the course of an evaluation, state-of-the-art speech recognition systems use multiple acoustic and language model sets with increasing complexity to obtain the best of all possible WERs. Applying a multi-pass decoding strategy is typically the way to incorporate multiple model sets into the decoder. The Hub-4 sites used five or more decoding passes in their evaluation systems. In a multi-pass decoding setup various model sets are applied in a predefined order for successive improvement of the decoder output. This order is typically optimized on development data. Philips used this approach for the 1997 Hub-4 evaluation, where the recognition started with word-internal tripone bigram decoding. Next it was extended in five passes on a lattice to a VTN/MLLR adapted crossword triphone trigram decoding, a process that is described in (Beyerlein et al., 1998).

A simpler and still optimal alternative to a sophisticated multiple-pass decoding strategy is DMC (Beyerlein, 1997), which integrates all obtainable models into one decoding pass. Thus we acquire a decoder containing information combined directly from all model sets.

The goal of DMC is to optimally integrate all given (acoustic and language) models into one log-linear posterior probability distribution. Assuming we are given M different acoustic and language models, which are identified by numbers $j = 1, \ldots, M$. From model j we can compute the posterior probability $p_j(k|x)$, $p_j(k'|x)$ of hypothesized classes k, k' given an observation x. The models are now log-linearly combined into a distribution of the exponential family

$$p_\Lambda(k|x) = \exp\left\{ -\log Z_\Lambda(x) + \sum_{j=1}^{M} \lambda_j \log p_j(k|x) \right\}. \tag{4}$$

The coefficients $\Lambda = (\lambda_1, \ldots, \lambda_M)^{\mathrm{T}}$ can be interpreted as weights of the models j within the model combination (4). The value $Z_\Lambda(x)$ is a normalization constant. As opposed to the maximum entropy approach, which leads to a distribution of the same functional form, the coefficients Λ are optimized with respect to the WER of the discriminant function (5),

$$\log \frac{p_\Lambda(k|x)}{p_\Lambda(k'|x)} = \sum_{j=1}^{M} \lambda_j \log \frac{p_j(k|x)}{p_j(k'|x)}. \tag{5}$$

DMC will optimize the so-called language weight (or language model factor), if only one acoustic and one language model are combined. Now, since the weight λ_j of the model j within the combination depends on its ability to provide information for correct classification, DMC allows for the optimal integration of any set of models into one decoder.

We are given a set of sentences $n = 1, \ldots, N$ for DMC training. For every training sentence we observe x_n (spoken utterance) and we know the correct class assignment k_n (spoken word sequence). We can define the set of rival classes $k \neq k_n$ using a preliminary decoding (if appropriate), and the number of word errors of the rival class k can be computed with the help of the Levenstein distance $\mathscr{L}(k_n, k)$. The model combination should then minimize the word error count $E(\Lambda)$,

$$E(\Lambda) = \sum_{n=1}^{N} \mathscr{L}\left(k_n, \arg\max_{k \neq k_n} \left(\log \frac{p_\Lambda(k|x_n)}{p_\Lambda(k_n|x_n)} \right)\right), \tag{6}$$

on representative training data to assure optimality on an independent test set. As this optimization criterion is not differentiable, we approximate it by a smoothed word error count,

$$E_{\mathrm{MWE}}(\Lambda) = \sum_{n=1}^{N} \sum_{k \neq k_n} \mathscr{L}(k, k_n) S(k, n, \Lambda), \tag{7}$$

where $S(k, n, \Lambda)$ is a smoothed indicator function. If the classifier (5) will select hypothesis k, $S(k, n, \Lambda)$ should be close to one, and if the classifier (5) will reject hypothesis k, it should be close to zero. A possible indicator function with these properties is

$$S(k, n, \Lambda) = \frac{p_\Lambda(k|x_n)^\eta}{\sum_{k'} p_\Lambda(k'|x_n)^\eta}, \tag{8}$$

where η is a suitable constant. A similar indicator function was already used in (Ney, 1995) to smooth the output of a classifier. An iterative gradient descent scheme is obtained from the optimization of $E_{\mathrm{MWE}}(\Lambda)$ with respect to Λ. The following second degree function is another possible indicator function with similar properties:

$$S(k, n, \Lambda) = \begin{cases} \left(\frac{g+B}{A+B}\right)^2, & -B < g < A, \\ 0, & g > A, \\ 0, & g < -B, \end{cases} \tag{9}$$

with

$$g = \log \frac{p_\Lambda(k|x_n)}{p_\Lambda(k_n|x_n)},$$

which gives a closed form matrix solution for Λ, where α is a Lagrangian multiplier,

$$(\alpha, \Lambda^{\mathrm{T}})^{\mathrm{T}} = BQ^{-1}P, \tag{10}$$

with

$$Q_{0,0} = 0, \quad Q_{0,j} = 1, \quad Q_{i,0} = \frac{1}{2}(A+B)^2,$$

$$Q_{i,j} = \sum_{n=1}^{N} \sum_{k \neq k_n} \mathscr{L}(k, k_n) \left\{ \log \frac{p_i(k_n|x_n)}{p_i(k|x_n)} \right\} \left\{ \log \frac{p_j(k_n|x_n)}{p_j(k|x_n)} \right\},$$

$$P_0 = \frac{1}{B},$$

$$P_i = \sum_{n=1}^{N} \sum_{k \neq k_n} \mathscr{L}(k, k_n) \left\{ \log \frac{p_i(k_n|x_n)}{p_i(k|x_n)} \right\}$$

$$(i, j = 1, \ldots, M).$$

The form of the second degree function is determined by the values A, B and by the set of hypotheses used for the training. Both indicator functions result in similar and reasonable DMC coefficients λ_j. The fact that the smoothed word error count (7) equals the empirical word error count (6) if η in (8) approaches infinity or if A, B in (9) approach zero explains this result.

Hub-4 development data were used to carry out the training of the DMC coefficients. The lattices, obtained by a one-pass trigram decoding (see Section 4), were expanded and rescored using the following phrase-based acoustic and language models (Sections 6 and 7): wordinternal triphones (ww); VTN/MLLR adapted wordinternal triphones (wwad); VTN/MLLR adapted cross-word triphones (xwad); VTN/MLLR adapted wordinternal pentaphones (5wwad); unigram, bigram, trigram, d1 bigram (tgset); unigram, bigram, trigram, d1 bigram, d2 Bigram (fgset). The obtained scores were interpolated using DMC resulting in the final system output. Table 7 gives an overview over several decodings. During a first decoding iteration a system, capturing a phrase-based cross-word pentaphone context and a trigram language model context was built (wwad + xwad + 5wwad + tgset). Compared to the baseline error rate of 20.7%, this system shows a WER of 18.9%. The adaptation of the acoustic and language models was repeated based on the output of the wwad + xwad + 5wwad + tgset system in a second decoding iteration (∗). By adding the d2-bigram language model to the combined set of models, the system was extended to a 4-gram context. It should be noted that the weights of the log-linear language model interpolation described in Section 6 are similar to the weights obtained from DMC! The wwad + xwad + 5wwad + fgset* system exhibited a word error rate of 17.9% on the Hub-4'97 evaluation data.

When compared to the simple voting at the level of the recognized word sequence as is done with ROVER (Fiscus, 1997), the log-linear interpolation (LLI) of context-dependent acoustic and distance-language models via DMC shows to be more powerful (Beyerlein et al., 1999). The basic difference between ROVER and DMC is that DMC provides a consistent framework for discriminative training and decoding of multi-model combinations. In addition DMC interpolates the model scores on lattices before decoding the output word sequence, whereas ROVER combines the decoder output of multiple systems. The use of confidence measures to weight the ROVER combination does not change the conceptual difference of both approaches, since confidence measures could be applied as additional models in the DMC framework. Table 8 shows an experimental comparism of both approaches. For the tests the NIST SCTK-1.2 ROVER software was used.

Table 7
Word error rates (%) for the LLI of acoustic and language models using DMC on the Hub-4'97 evaluation data, abbreviations are explained in Section 5

Models	# of models	WER (%)
xwad + tg (baseline)	2	20.7
wwad + xwad + tg	3	20.2
wwad + xwad + 5wwad + tg	4	19.5
wwad + xwad + 5wwad + tgset	7	18.9
wwad + xwad + 5wwad + fgset*	8	17.9

5.1. Summary

For the integration of multiple model sets into a decoder, two methods may be applied, multi-pass decoding (including ROVER) and DMC. DMC is the preferred method, since it is a simple device for combining any available model sets into one decoding pass, while at the same time directly optimizing the WER of the classifier on training data.

The following two sections describe the acoustic and language models, which were applied in the one-pass trigram decoding and in the final DMC-pass.

6. Building phrase-based distance language models

Natural language created by humans is correlated. Using a particular word not only influences the word immediately following, but up to the next 1000 words (Peters and Klakow, 1999). Thus these correlations have to be captured in the best way possible to reduce the resources needed and to minimize the number of parameters. In the course

Table 8
Comparison of ROVER and DMC on the Hub-4'97 evaluation data, abbreviations are explained in Section 5

Models	DMC (# of models)	ROVER (# of systems)
wwad + tg	21.6 (1 + 1)	−(1 · 1)
xwad + tg	20.7 (1 + 1)	−(1 · 1)
wwad + xwad + tg	20.2 (2 + 1)	22.5 (2 · 1 = 2)
wwad + xwad + 5wwad + tg	19.5 (3 + 1)	19.9 (3 · 1 = 3)
wwad + xwad + 5wwad + tgset	18.9 (3 + 4)	20.2 (3 · 4 = 12)

of the past few years, new methods have been developed for Hub-4 serving this purpose: the use of phrases, consisting of several consecutive strongly correlated words model short-range dependencies; and log-linear interpolation (LLI) of distance language models which allows for an efficient description of long-range effects. Those two methods will be described in the following two sections.

Modeling strong short-range correlations by using phrases: Over a short-range natural language is strongly correlated, a fact observed as an improvement when going from a unigram to a bigram. There are also bigrams, however, that do not contain any information and that do not contribute to the improvement. The desired solution should selectively increase the language model context. Varigrams (Kneser, 1996) were constructed for this purpose although these also have a drawback in that search algorithms must use a finite, well-defined context. A simple but efficient solution to this problem is the use of phrases (Klakow, 1997). They are made up of important pairs of words, like `in_the`, `of_the` or `a_lot_of`. The main idea is to select candidate pairs of words based on the frequency of the pairs. Those pairs, however, may overlap (like `a lot` and `lot of`) and hence cannot be joined to one phrase on the training corpus simultaneously. Therefore, the list of candidate pairs is sorted by frequency. We then start from the top of the list by selecting all other pairs that may cause ambiguities and remove those from the list (meaning that `a lot` is kept while `lot of` is removed). Now, the training corpus is processed and all pairs in the remaining list are joined to longer units (now, `a_lot` is a proper entry of the vocabulary). The second essential part of the algorithm is to check whether phrases have still a sufficiently high frequency. If not, those phrases are broken apart into the constituents. The whole procedure is iterated until there are no more candidates for the join or split operation. During iteration, longer and longer phrases are built (the second iteration generates `a_lot_of` from `a_lot of`). By choosing frequency bounds for selecting candidate pairs and for splitting, the total number of phrases can be determined. The BN training corpus consists of 140 million words of transcribed BN and the test set is taken from the 1996 development data. The vocabulary size is 64k words and 330 phrases are constructed. Perplexities are shown in Table 9. The improvement for bigrams is 8.4% and for trigrams 4.1%.

Modeling weak correlations by LLI: Maximum entropy language models are known to work well with triggers and other long-range dependencies in natural language (Rosenfeld, 1994). However, they suffer from a severe deficiency: they need large amounts of CPU-time in training. This can be circumvented as is done by Darroch and Ratcliff (1972) on generalized iterative scaling (GIS). It turns out, that the first iteration step of GIS can be carried out manually and a closed form solution is obtained,

$$p_A(w|h) = \frac{1}{Z_A(h)} \prod_i p_i(w|h)^{\lambda_i}, \tag{11}$$

where $p_i(w|h)$ correspond to the marginal distributions used as constraints of the maximum entropy

Table 9
Language model perplexities on Hub-4'97 evaluation set

	Unigram	Bigram	Trigram
64k Words	1026.4	257.1	180.0
+330 Phrases	841.2	235.4	172.7

Table 10
Comparison of ordinary backing-off (BO) models, of linear interpolation (Lin) and LLI on Hub-4'97 evaluation set, the shortcuts are explained in Section 6

Model	Perplexity
BO bigram (= d0)	216
BO trigram	150
BO 4-gram	144
Lin d0 + d1	204
LLI d0 + d1	175
Lin Tri + d0 + d1 + d2	146
LLI Tri + d0 + d1 + d2	136
Lin Tri + d0 + ··· + d5	146
LLI Tri + d0 + ··· + d5	130

models (Klakow, 1998). The λ_i are free parameters weighing the different constraint equations. The λ_i can be optimized on a separate cross-validation corpus as already known from linear interpolation. For Hub-4, distance bigram models were used as constraints. Various distance patterns have been investigated with the results being presented in Table 10. Here, for example a d1-bigram (distance-1-bigram) model is in fact a 3-gram language model $p(w|v,u)$, where the dependency on v is neglected, $p(w|v,u) = p(w|*,u)$.

In contrast, we also present the linear interpolation of the same components. It is obvious that linear interpolation is always outperformed by LLI. Moreover LLI of a trigram, a bigram, a d1-bigram and a d2-bigram (a model having effectively a 4-gram context because of the d2-bigram) is better than the 4-gram at much lower memory consumption. It is also observed that following this pattern 7-grams can be built with a small additional decrease in perplexity. The distance-bigrams used above were pure word models, which can be further improved by using class-language models trained on distance-classes (see Table 11).

Table 11
Perplexities for the LLI of distance bigram models (use of classes) on Hub-4'97 evaluation set

Model	Perplexity
d2 Bigram BN	739
d2 Bigram BN, classes	661
LLI Tri + d0 + d1 + d2 BN	136
LLI Tri + d0 + d1 + d2 BN, classes	118

6.1. Summary

Phrases combine neighboring words to one unit improving the performance of a language model by selectively increasing the context length similarly to varigrams but at much lower cost. Log-linear interpolation is most efficient when used to combine distance-bigrams with other models. It allows building effective 4-gram models (i.e. which do not use explicit 4-gram information) which outperform backing-off (BO) 4-grams in terms of memory requirements and perplexity. This result fits well in the idea of combining the various distance-language models and a set of acoustic models into one decoder during the DMC-pass (Section 5). The performance gain by the use of the distance language models is summarized in Table 7 in Section 5.

Note, that the difference between DMC and the described log-linear language model interpolation is given (1) by the training criterion of the model combination (likelihood versus WER) and (2) by the scope of the model combination. DMC allows for the integration of language model and acoustic model information (which is described in the following section).

7. Building robust phrase-based acoustic models

7.1. Feature extraction, normalization and speaker adaptation

Mel-frequency cepstral coefficients (MFCC) (Davis and Mermelstein, 1980) are probably the most popular features for speech recognition. Nevertheless, there is still active research in superior speech representations for speech recognition. A lot of effort is devoted to exploiting physiological and psychoacoustic findings about human perception. As an example, Hermansky (1990) has extended linear prediction analysis to perceptual linear prediction (PLP) by introducing concepts from psychophysics. Recently people have introduced similar psychophysical concepts into the well-known mel-frequency cepstral analysis of speech (Woodland et al., 1997) and devised a variant of MFCC called MF-PLP (see Fig. 3). We experimented with variations of MFCC and MF-

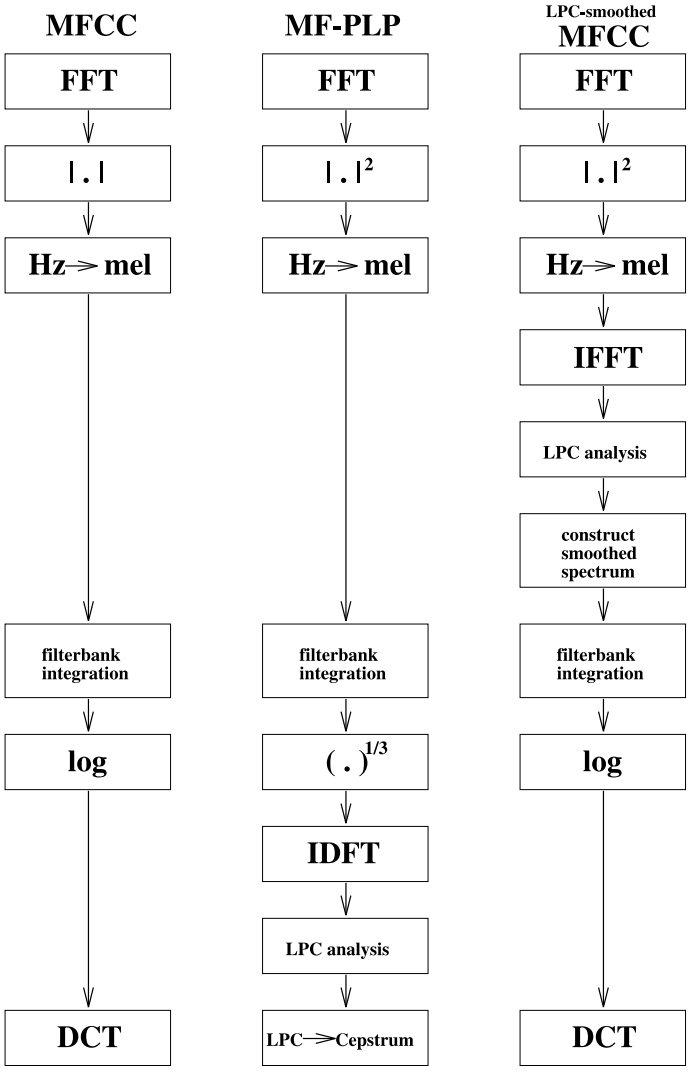

Fig. 3. Feature extractions.

PLP on the Hub-4 data. Of those, the most promising setup was what we called "LPC-smoothed MFCC", where only the LPC smoothing of the mel-warped spectrum is added to the MFCC analysis. This configuration is similar to what BBN used in their Hub-4 system (Kubala, 1998). Table 12 presents trigram recognition results for the three cepstral parameterizations described above on the Hub-4 Eval'97 test set (partitioned evaluation). As can be seen, neither MF-PLP nor LPC-smoothed MFCC was able to consistently outperform the MFCC feature set. Since this holds also after acoustic adaptation, standard MFCC signal analysis was used for all further investigations. Note, however, that we were able to improve upon MFCC by feature set combinations (Haeb-Umbach and Loog, 1999) both at the acoustic likelihood level by DMC (Beyerlein, 1997, see Section 5) and at the recognized word level by ROVER (Fiscus, 1997).

Channel normalization: aims at improving the insensitivity of the feature vector to distortions. It

Table 12
Word error rates in % for MFCC, MF-PLP and LPC-smoothed MFCC feature vectors on Hub-4'97 evaluation set

		Focus condition						
Cepstral parameterization	Overall	F0	F1	F2	F3	F4	F5	FX
MFCC	21.6	13.1	20.1	32.2	30.9	25.6	23.9	37.2
MF-PLP	22.1	13.4	21.3	31.4	32.5	26.0	27.1	38.1
LPC-smoothed MFCC	21.9	13.4	20.5	31.9	31.4	25.2	25.6	38.8

Gender-dependent setup, BN-96 h training corpus, within-word triphone models, trigram language model.

is well known that a constant, though unknown channel transfer function affects the mean of the cepstral features. Further, it has been observed that additive-noise results, among other effects, in a mean shift and reduction of the variance of the distributions of the cepstral coefficients (Openshaw and Mason, 1994). These observations motivate the use of cepstral mean and variance normalization. The mean and variance normalized feature $y_k(t)$ is computed as follows:

$$y_k(t) = \frac{x_k(t) - \bar{x}_k(t)}{\hat{\sigma}_k(t)}, \quad k = 1, \ldots, K, \tag{12}$$

where k is the cepstral index, K being the number of (static) features. $\bar{x}_k(t)$ is an estimate of the mean and $\hat{\sigma}_k(t)$ is an estimate of the standard deviation of the input cepstral feature $x_k(t)$. Both mean and variance are computed over a block of frames, in our case over one segment, as delivered by the segmenter (Section 3). This operation is carried out on all static cepstral coefficients. On the Hub-4'96 development data, we observed on average a performance improvement of about 3% relative due to variance normalization, see Table 13.

Vocal tract normalization: performs a speaker normalization in the signal space by, typically linearly, warping the frequency axis by a speaker-specific warping factor. The intention is that after normalization the influence of differences in the vocal tract length across speakers on the computed feature vectors are removed to a great extent. Vocal tract normalization can be carried out in training and in recognition, and it can be used in a gender-dependent (GD) and in a gender-independent (GI) setup. Table 14 shows results on the Hub-4'96 development data for some test scenarios. Compared to our results on the Wallstreet Journal database (Welling et al., 1998), the error rate reduction due

Table 13
Effect of variance normalization on word error rates in % on Hub-4'96 development set (male speakers only)

		Focus condition						
Variance normalization	Overall	F0	F1	F2	F3	F4	F5	FX
No	38.6	17.5	38.2	46.0	39.4	33.6	37.8	65.8
Yes	37.3	18.4	36.5	44.4	35.9	31.8	36.3	64.3

Gender-dependent setup, 30 h training data, within-word models, bigram language model.

Table 14
Word error rates in % on Hub-4'96 development set (male speakers only) for different VTN scenarios

	VTN in			Focus condition						
Setup	Training	Recognition	Overall	F0	F1	F2	F3	F4	F5	FX
GD	No	No	36.2	16.9	36.6	43.4	32.6	30.0	37.0	61.1
GD	No	Yes	35.5	16.8	35.8	40.7	31.7	29.7	36.5	63.0
GI	No	No	36.5	17.1	36.5	45.1	33.7	29.3	36.6	61.2
GI	Yes	Yes	35.3	16.4	35.3	42.4	30.5	29.7	34.1	62.4

Gender-dependent setup, 30 h training data, within-word triphone models, bigram language model.

to VTN was considerably smaller, e.g. 3.3%, when using VTN in training and recognition in a GI setup, as opposed to 11% on WSJ. For the warping factor selection in recognition we adopted the scheme proposed in (Lee and Rose, 1996), which requires a preliminary transcription of the utterance to be recognized. This fits nicely with our decoding strategy, which is two-pass anyway due to the MLLR speaker adaptation.

MLLR unsupervised adaptation: of the mean vectors is applied to clusters of segments using the Least Mean Squares approximation (Thelen et al., 1997). The regression classes are based on phonetic knowledge and are dynamically defined using a tree organization. The amount of adaptation speech determines both the number of active regression classes and the structure of the MLLR transformation matrices. Table 15 highlights the WERs of the first and second decoding pass. It can be seen that the joint effect of VTN and MLLR sums up to a reduction of the WER by 10.3% relative from 23.4% to 21.0% on the Hub-4 Eval'97 test data.

7.2. Review of the training strategy

Speech recorded from radio or television broadcasts exhibits large variations with respect to the quality of the microphone or channel, the characteristics of the speaker, and the condition of the background. The data range from high-quality studio recordings of experienced announcers to very noisy interviews with stressed analysts at the NYSE. Thus the so-called focus-conditions (F-conditions) were introduced by the Hub-4-Society (see Table 16). Accordingly, training and test data were labeled with respect to these focus conditions. In the Hub-4 '96 evaluation most of the sites used F-condition specific models. BBN decided to train just a single model set for all F-conditions (Schwartz et al., 1997). This simplified the system enormously and rendered condition classification obsolete, while at the same time maintaining good recognition accuracy.

We decided to similarly direct our research efforts by not only using a single model set but also using channel-independent models to further simplify the process. Here we relied on the normalization and adaptation techniques described in Section 7.1. The goal was to obtain a compact system architecture. During the 1996 Hub-4 evaluation no preference for one of the applied training strategies could be found:

- training on Wall Street Journal data followed by supervised adaptation on Hub-4, possibly even on each focus condition specifically;
- training focus-specific models on the Hub-4 data or
- training one general model set for all F-conditions.

We revisited this question and investigated several alternatives. Specifically, we compared the following scenarios:

1. Training on the WSJ0 + 1, SI-284 training data and subsequent supervised adaptation on each of the Hub-4 focus conditions specifically. We

Table 16
Hub-4 focus conditions

F0	Clean planned speech, e.g. tv-news
F1	Clean spontaneous speech, e.g. tv-discussions
F2	F0 + F1 over telephone, e.g. telephone interview
F3	F0 + background music, e.g. start of tv-news
F4	F0 + background noise, e.g. applause
F5	F0 + non-native dialect, e.g. british english
FX	Any other combination of difficulties

Table 15
Word error rates in % on Hub-4'97 evaluation set before and after VTN and MLLR adaptation

		Focus condition						
	Overall	F0	F1	F2	F3	F4	F5	FX
No adaptation	23.4	14.0	21.4	33.7	32.5	27.1	24.7	45.8
+ VTN + MLLR adaptation	21.0	12.8	19.8	29.2	31.4	24.7	21.2	39.1

Gender-dependent setup, BN-96 h training corpus, within-word triphone models, trigram language model.

Table 17
Word error rates in % on Hub-4'96 PE development set (male speakers only) for different training scenarios

Training scenario	Overall	Focus condition						
		F0	F1	F2	F3	F4	F5	FX
1	41.9	18.7	41.8	50.2	43.9	38.0	40.8	67.6
2	42.4	18.4	43.1	49.7	42.7	35.7	47.4	69.6
3	38.6	17.5	38.2	46.0	39.4	33.6	37.8	65.8

Bigram language model, gender-dependent setup, within-word triphone models, no adaptation in recognition.

assumed, that we know the focus conditions of the test data.
2. Training of a separate model set on each of the Hub-4 focus conditions. Here we assumed again that we know the focus conditions of the test data.
3. Training of one model set on all available Hub-4 data. A selection of a proper focus condition is obsolete.

Note that for each scenario we trained separate model sets for male and female speakers (GD setup). The test results reported in Table 17 favored the focus-independent scenario. The clear advantage of training a single model set on all Hub-4 data, as is evident from Table 17, can be explained by the limited amount of acoustic training data for each of the F-conditions.

The optimization of the training procedure implies in addition a review of the acoustic training data, which is done in the following section.

7.3. Corpus selection and verification

For the acoustic models of the 1997 Philips Hub-4 system, (Beyerlein et al., 1998), a subset of 46 out of the 76 h of speech data released by LDC in 1996/97 was used (BN-46 h corpus). It was derived by manually verifying the complete training corpus. All erroneous speech segments were discarded and only a few obvious errors were corrected. We found two types of errors in the BN training corpus:

- incorrect transcriptions, i.e. wrongly transcribed or missing words and incorrect segment boundaries and
- false begin or end times of speech segments.

To detect these errors automatically we applied a forced Viterbi alignment with GI low resolution acoustic models (2000 tied states, 60k densities) trained on the manually cleaned Hub-4 training data. Each training utterance (segment) was classified according to the following criteria:

1. optimal path reaches the terminal HMM state;
2. size of the search space required for the alignment (beam width);
3. acoustic score of the whole segment, normalized by the number of time frames;
4. normalized acoustic word scores;
5. duration of each word in the segment and
6. segment boundary found by joining adjacent segments followed by a forced alignment.

We applied the criteria to the 1996/97 Hub-4 training corpus (15 389 segments, 76 h).

Measure 1 detected major errors like whole missing sentences or incorrect segment boundaries. As shown in the histogram (Fig. 4) a significant amount of segments was corrupted by this type of transcription error.

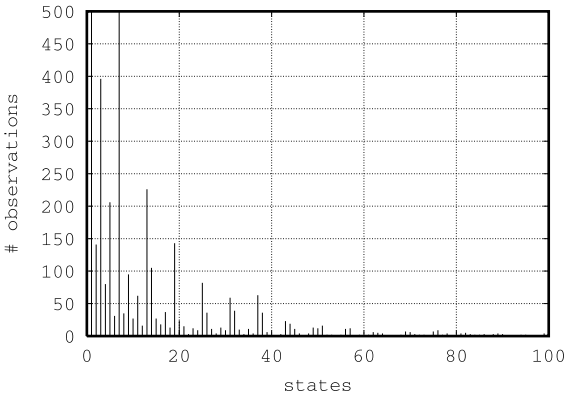

Fig. 4. Histogram over differences between the terminal HMM state in the forced alignment and the terminal state according to the training transcription.

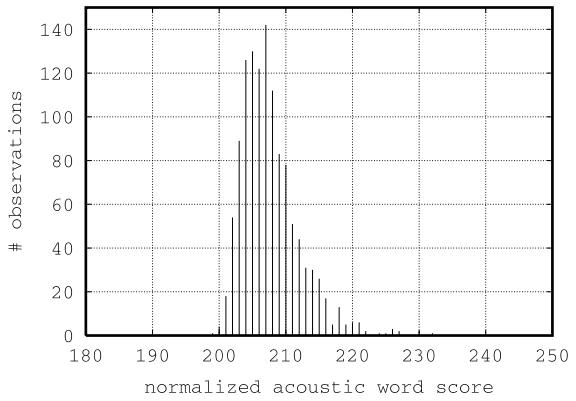

Fig. 5. Histogram over normalized acoustic word scores for the word "PRESIDENT".

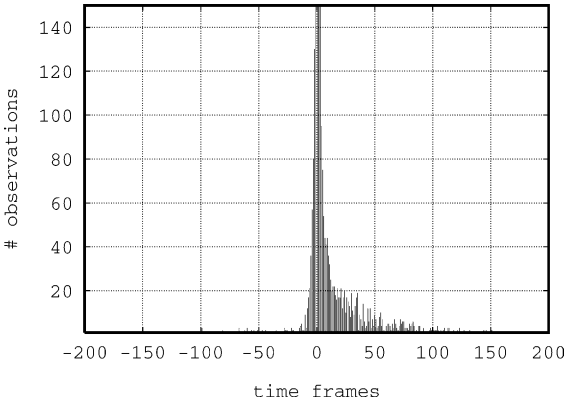

Fig. 6. Histogram over time differences between the segment boundaries of adjacent segments. Displayed are critical positive time differences at the begin of segments and negative at their ends.

Measures 2 and 3 were dependent on speaker and channel and were of little use in detecting transcription errors.

Criterion 4 correctly indicated not only missing or wrongly transcribed single words, but also utterances with strong background noise or overlapping speech. An example for the score distribution is given in Fig. 5 for the word "PRESIDENT". Large deviations from the average scores were often caused by transcription errors.

Measure 5 gave only small evidence of transcription errors as the duration of words is basically speaker and context dependent.

Finally, the across-segment criterion 6 indicated wrong segment boundaries as well as major transcription errors similar to quality measure 1. Fig. 6 shows the histogram over all boundary differences.

35% of the corpus (5429 segments, 27 h) was tagged as possibly erroneous. 72% of these segments were tagged according to criterion 4. Criteria 6 and 1 supplied 13% and 7% of the tagged segments, respectively. 8% of the segments were tagged according to two or all three criteria. The tagged segments were manually verified and corrected afterwards. 75% of the tagged segments actually contained wrong transcriptions or segment boundaries. A detailed analysis can be found in (Pitz and Molau, 1999). After correction, 75 h of training material was available (BN-75 h corpus).

The effect of improved training data was tested with acoustic models obtained from three different corpora:

- the complete 1996/97 Hub-4 training corpus which amounts to about 76 h of speech data;
- the manually checked BN-46 h corpus, in which all incorrect segments were rejected and only a few obvious errors were corrected, and
- the BN-75 h corpus, where an overall of 27 h of erroneous segments were automatically detected and manually corrected thereafter.

All recognition tests were carried out with a one-pass trigram decoder based on word-internal triphones. A closer analysis of the recognition results in Table 18 revealed that the extensive manual correction effort gave improvements for planned clean speech (F0 condition) only. The WER remained almost constant in the other focus conditions.

For the 1998 evaluation, an additional 73 h set of acoustic training data was released by LDC. In that year, the training corpus of the Philips-RWTH Hub-4 speech recognition system consisted of an 96 h subset (BN-96 h corpus) remaining after automatic verification and manual correction of the full 148 h corpus (BN-148 h). Both, the BN-96 h models and the BN-148 h models, performed very similar.

7.4. Context-dependent acoustic modeling

Phrases, defined as frequently occurring word sequences included into the vocabulary as a single word (see Section 6), are a simple means of

Table 18
Word error rates in % on Hub-4'96 evaluation set, obtained with the RWTH system for the Hub-4 task using different training corpora

Condition	46 h	76 h	75 h
F0	25.7	25.1	24.6
F1	32.4	32.2	32.2
F2	41.2	41.1	40.1
F3	33.0	36.6	35.2
F4	39.2	39.9	39.3
F5	34.3	31.1	32.5
FX	59.0	54.9	54.9
Overall	35.1	34.3	34.0

modeling long-span acoustic and language context. The pronunciation dictionary was augmented with the 330 most frequent phrases found in the BN training data. The 10 most frequent phrases found in the BN training data are: `in_the`, `of_the`, `on_the`, `to_the`, `and_the`, `you_know`, `for_the`, `to_be`, `I_think`, `that_the`. We modeled typical variations in speaking style and coarticulation of the phrases by adding pronunciation variants to the pronunciation dictionary.

In our final Hub-4 system we trained GD models on 96 h of the acoustic BN training data (see Section 7.3, BN-96 h corpus). The emission densities are modeled by mixtures of Laplacian densities L_i with a single, globally pooled deviation vector $\vec{v}$,

$$p(\vec{x}) = \sum_{i=1}^{I} w_i L_i(\vec{x}), \quad 0 \leqslant w_i \leqslant 1, \quad \sum_{i=1}^{I} w_i = 1,$$
$$L_i(\vec{x}) = \frac{1}{2^D \prod_{d=1}^{D} v_d} \exp\left\{ -\sum_{d=1}^{D} \frac{|x_d - \mu_{i,d}|}{v_d} \right\}. \tag{13}$$

The performance of a similar Gaussian mixture density system was close to that of the Laplacian mixture density system. We also applied decision tree clustering (adapted to Laplacian densities (Beyerlein et al., 1997)) for a robust within-word triphone, crossword triphone and pentaphone modeling. Table 19 contains more detailed information about the size of the acoustic models in our final system. The listed word-internal triphone models were applied in the one-pass trigram recognition, whereas all context-dependent model sets were employed in the DMC-pass.

7.5. Summary

We compared the MFC-front-end with MF-PLP and LPC-smoothed MFCC and found that MFCC performed best. Variance normalization improves the robustness of the system and the overall WER. The same holds for VTN and MLLR. Using VTN the GD and the GI systems showed a similar performance.

The review of the training strategy showed, that F-condition independent training gives the best result.

The manual transcriptions of the BN training corpora contained a number of errors affecting both individual word and segment boundaries. We

Table 19
Size of acoustic models trained on the BN-96 h training corpus

Model	# Clusters	# Densities
Word-internal triphones, male	9300	402k
Word-internal triphones, female	7800	291k
Crossword triphones, male	10,700	487k
Crossword triphones, female	8600	343k
Word-internal pentaphones, male	10,500	459k
Word-internal pentaphones, female	8200	296k

developed a method to automatically detect these errors based on a forced alignment on the training data, thus reducing the manual work. Our efforts to obtain better acoustic models by improving the training corpus had limited success only. When manually correcting transcription errors, the models will be cleaner resulting in better performance on planned clean speech (F0 condition). The recognition accuracy, however, under difficult conditions (i.e. in the presence of noise, background music, or limited bandwidth) remains almost unaffected.

8. Conclusions

A brief summary of our findings is listed below:

• *Segmenter*: Two automatic segmentation approaches were investigated for the automatic segmentation of the continuous BN audio stream: (1) a phoneme decoder and (2) a GMM–BIC segmenter. The GMM–BIC segmenter provides better results. The loss of word accuracy by automatic segmentation compared to manual segmentation is about 5% relative.

• *One-pass decoder*: One-pass trigram decoding compares favorably with a two-pass strategy, the overall computational costs – including memory and CPU time – being only moderately increased. Due to the prefix tree organization of the lexicon, decoding costs show a relatively slow increase when the vocabulary is enlarged from 20k to 64k. Compared with unigram look-ahead, bigram look-ahead is able to reduce significantly the search cost for most focus conditions, even though the additional memory costs cannot be neglected.

• *Discriminative model combination (DMC)*: For the integration of multiple model sets into a decoder, two methods may be applied, multi-pass decoding (including ROVER) and DMC. DMC is the preferred method, since it is a simple device for combining any available model sets into one decoding pass, while at the same time directly optimizing the WER of the classifier on training data. We could easily extend our system to a crossword pentaphone context by a discriminative optimization of the log-linear interpolation (LLI) of the required acoustic models.

• *Language model*: Phrases improve the performance of a language model by selectively increasing the context length similarly to varigrams but at much lower cost. Log-linear interpolation is most efficient when used to combine distance-bigrams with other models. It allows building effective 4-gram models (i.e. which do not use explicit 4-gram information) which outperform BO 4-grams in terms of memory requirements and perplexity.

This result fits well in the idea of combining the various distance-language models and a set of acoustic models into one decoder during the DMC-pass.

• *Acoustic model*: We built a phrase-based crossword pentaphone Laplacian mixture density system. A focus condition independent training gave the best result. We compared the MFC-front-end with MF-PLP and LPC-smoothed MFCC and found that MFCC performed best. Variance normalization, VTN and MLLR improve the robustness of the system. The manual transcriptions of the BN training corpora contained a number of errors affecting both individual word and segment boundaries. We developed a method to automatically detect these errors. Our efforts to obtain better acoustic models by improving the training corpus had limited success only.

Summary of symbols

Symbols	*Explanation*
$p_\Lambda(w\|h)$	probability of word w given history h and parameter Λ
$Z_\Lambda(h)$	normalization term
p_i	probability model i
λ_i	weight of model i in a model combination
N_W	size of the vocabulary denoted as $\mathscr{W}$
W_j	word j in word history
$P(w\|u,v)$	probability of word w given predecessor words u, v
$H(u,v)$	hash index for word pair u, v
M_W	hash constant
$\pi_v(s)$	anticipated language model probability for state s and predecessor word v
$p_\Lambda(k\|x)$	posterior probability of class k given background information x

$E(\Lambda)$ word error count
x_n spoken utterance, training sample n
k_n correct word sequence corresponding to x_n
$S(k, n, \Lambda)$ indicator function
$\mathscr{L}(k, k_n)$ Levenshtein distance, word error count
η smoothing constant

References

Alleva, F., Huang, X., Hwang, M.-Y., 1996. Improvements on the pronunciation prefix tree search organization. In: Proc. IEEE Internat. Conf. Acoust. Speech Signal Process., Atlanta, GA, pp. 133–136.

Aubert, X., 1999. One pass crossword decoding for large vocabularies based on a lexical tree search organization. In: Proc. EUROSPEECH, Budapest, Hungary, pp. 1559–1562.

Aubert, X., Ney, H., 1995. Large vocabulary continuous speech recognition using word graphs. In: Proc. IEEE Internat. Conf. Acoust. Speech Signal Process., Detroit, MI, pp. 49–52.

Aubert, X., Dugast, C., Ney, H., Steinbiss, V., 1994. Large vocabulary continuous speech recognition of Wall Street Journal corpus. In: Proc. IEEE Internat. Conf. Acoust. Speech Signal Process., Adelaide, Australia, pp. 129–132.

Beyerlein, P., 1997. Discriminative model combination. In: Proc. 1997 IEEE Workshop Automatic Speech Recognition and Understanding, Santa Barbara, pp. 238–245.

Beyerlein, P., Ullrich, M., Wilcox, P., 1997. Modelling and decoding of crossword context dependent phones in the Philips large vocabulary continuous speech recognition system. In: Proc. EUROSPEECH, Rhodes, Greece, pp. 1163–1166.

Beyerlein, P., Aubert, X., Haeb-Umbach, R., Klakow, D., Ullrich, M., Wendemuth, A., Wilcox, P., 1998. Automatic transcription of English broadcast news. In: Proc. DARPA Broadcast News Transcription and Understanding Workshop, Virginia.

Beyerlein, P., Aubert, X., Haeb-Umbach, R., Harris, M., Klakow, D., Wendemuth, A., Molau, S., Pitz, M., Sixtus, A., 1999. The Philips/RWTH system for transcription of Broadcast News. In: Proc. EUROSPEECH, Budapest, Hungary, pp. 647–650.

Chen, S., Gopalakrishnan, P., 1998. Speaker, environment and channel change detection and clustering via the Bayesian information criterion. In: Proc. DARPA Broadcast News Transcription and Understanding Workshop, Virginia.

Darroch, J.N., Ratcliff, D., 1972. Generalized iterative scaling for log linear models. Annals Math. Stat. 43, 1470–1480.

Davis, S.B., Mermelstein, P., 1980. Comparison of parametric representations for Monosyllabic word recognition in continuously spoken sentences. IEEE T-ASSP ASSP-28 (4), 357–366.

Fiscus, J.G., 1997. A post-processing system to yield reduced word error rates: recognizer output voting error reduction (ROVER). In: Proc. 1997 IEEE Workshop Automatic Speech Recognition and Understanding, Santa Barbara, pp. 347–354.

Haeb-Umbach, R., Loog, M., 1999. An investigation of cepstral parameterisations for large vocabulary speech recognition. In: Proc. EUROSPEECH, Budapest, Hungary, pp. 1323–1326.

Hain, T., Johnson, S., Tuerk, A., Woodland, P., Young, S., 1998. Segment generatation and clustering in the HTK Broadcast News transcription system. In: Proc. DARPA Broadcast News Transcription and Understanding Workshop, Virginia.

Harris, M., Aubert, X., Haeb-Umbach, R., Beyerlein, P., 1999. In: Proc. EUROSPEECH, Budapest, Hungary, pp. 1027–1030.

Hermansky, H., 1990. Perceptual linear predictive (PLP) analysis of speech. J. Acoust. Soc. Am. 87, 1738–1752.

Jin, H., Kubala, F., Schwartz, R., 1997. Automatic speaker clustering. In: Proc. DARPA Speech Recognition Workshop, Virginia.

Klakow, D., 1997. Language-model optimization by mapping of corpora. In: Proc. IEEE Internat. Conf. Acoust. Speech Signal Process., Seattle, pp. 701–704.

Klakow, D., 1998. Log-linear interpolation of language models. In: Proc. Internat. Conf. Speech Language Process., Sidney, pp. 1695–1698.

Kneser, R., 1996. Statistical language modeling using a variable context length. In: Proc. Internat. Conf. Speech Language Process., Philadelphia, pp. 494–497.

Kubala, F. et al., 1998. The 1997 BBN BYBLOS system applied to Broadcast News transcription. In: Proc. DARPA Broadcast News Transcription and Understanding Workshop, Virginia.

Lee, L., Rose, R., 1996. Speaker normalization using efficient frequency warping procedures. In: Proc. IEEE Internat. Conf. Acoust. Speech Signal Process., Atlanta, GA, Vol. 1, pp. 353–356.

Ney, H., 1995. On the probabilistic interpretation of neural network classifiers and discriminative training criteria. IEEE Trans. Pattern Anal. Machine Intell. 17 (2), 107–119.

Ney, H., Haeb-Umbach, R., Tran, B.-H., Oerder, M., 1992. Improvements in beam search for 10,000-word continuous speech recognition. In: Proc. IEEE Internat. Conf. Acoust. Speech Signal Process., San Francisco, CA, pp. 13–16.

Odell, J.J., 1995. The use of context in large vocabulary speech recognition. Ph.D. Thesis, University of Cambridge, England.

Odell, J.J., Valtchev, V., Woodland, P.C., Young, S.J., 1994. A one pass decoder design for large vocabulary recognition. In: Proc. ARPA Spoken Language Technology Workshop, Plainsboro, NJ, pp. 405–410.

Openshaw, J.P., Mason, J.S., 1994. On the limitations of Cepstral features in noise. In: Proc. IEEE Internat. Conf. Acoust. Speech Signal Process., Adelaide, Australia, pp. II49–II52.

Ortmanns, S., Ney, H., Eiden, A., 1996a. Language-model look-ahead for large vocabulary speech recognition. In: Proc. Internat. Conf. Speech Language Process. 1996, Philadelphia, pp. 2095–2098.

Ortmanns, S., Ney, H., Seide, F., Lindam, I., 1996b. A comparison of time conditioned and word conditioned search techniques for large vocabulary speech recognition. In: Proc. Internat. Conf. Speech Language Process. 1996, Philadelphia, pp. 2091–2094.

Ortmanns, S., Ney, H., Aubert, X., 1997. A word graph algorithm for large vocabulary continuous speech recognition. Comput. Speech Language 11, 43–72.

Peters, J., Klakow, D., 1999. Capturing long-range correlations using log-linear language models. In: Verbmobil: Foundations of Speech-to-speech Translation. Springer, Berlin, pp. 79–94.

Pitz, M., Molau, S., 1999. Automatic verification of Broadcast News transcriptions. In: Proc. EUROSPEECH, Budapest, Hungary, pp. 675–678.

Rosenfeld, R., 1994. Adaptive statistical language modeling: a maximum entropy approach. Ph.D. Thesis, CMU.

Schwartz, R., Jin, H., Kubala, F., Matsoukas, S., 1997. Modeling those F-conditions – or not. In: Proc. DARPA Speech Recognition Workshop, Virginia.

Siegler, M., Jain, U., Raj, B., Stern, R.M., 1997. Automatic segmentation and clustering of Broadcast News audio. In: Proc. DARPA Speech Recognition Workshop, Virginia.

Steinbiss, V., Tran, B.-H., Ney, H., 1994. Improvements in beam search. In: Proc. Internat. Conf. Spoken Language Process., Yokohama, Japan, pp. 2143–2146.

Thelen, E., Aubert, X., Beyerlein, P., 1997. Speaker adaptation in the philips system for large vocabulary continuous speech recognition. In: Proc. IEEE Internat. Conf. Acoust. Speech Signal Process., Munich, Germany, pp. 1035–1038.

Welling, L., Haeb–Umbach, R., Aubert, X., Haberland, N., Ney, H., 1998. A study on speaker normalization using vocal tract normalization and speaker adaptive training. In: Proc. IEEE Internat. Conf. Acoust. Speech Signal Process., Seattle, Wash, Vol. 2, pp. 797–800.

Woodland, P.C., Gales, M.J.F., Pye, D., Young, S.J., 1997. Broadcast News transcription using HTK. In: Proc. IEEE Internat. Conf. Acoust. Speech Signal Process., Munich, pp. 719–722.