Philips J. Res. **49** *(1995) 317–352*

THE PHILIPS RESEARCH SYSTEM FOR CONTINUOUS-SPEECH RECOGNITION

by V. STEINBISS, H. NEY,[1] X. AUBERT, S. BESLING, C. DUGAST, U. ESSEN, D. GELLER, R. HAEB-UMBACH, R. KNESER, H.-G. MEIER, M. OERDER and B.-H. TRAN

Philips GmbH Forschungslaboratorien, Postfach 1980, D-52066 Aachen, Germany

Abstract

This paper gives an overview of the Philips Research system for continuous-speech recognition. The recognition architecture is based on an integrated statistical approach. The system has been successfully applied to various tasks in American English and German, ranging from small vocabulary tasks to very large vocabulary tasks and from recognition only to speech understanding. Here, we concentrate on phoneme-based continuous-speech recognition for large vocabulary recognition as used for dictation, which covers a significant part of our research work on speech recognition. We describe this task and report on experimental results. In order to allow a comparison with the performance of other systems, a section with an evaluation on the standard North American Business news (NAB[2]) task (dictation of American English newspaper text) is supplied.

Keywords: acoustic model; continuous-speech recognition; dictation; hidden Markov model (HMM); language model; large-vocabulary recognition; search.

1. Introduction

For large-vocabulary continuous-speech recognition, there are a number of operational prototype systems in research, some of them participating in the ARPA[3] research programme or its evaluations. Like the above mentioned

[1] Present address: Lehrstuhl für Informatik VI, University of Aachen (RWTH), D-52056 Aachen, Germany. E-mail: ney@informatik.rwth-aachen.de

[2] Abbreviations can be found in Table XIII.

[3] Advanced Research Projects Agency (U.S.-American organization funding, among others, speech recognition and understanding research).

systems, the prototype system described in this paper is based on techniques of statistical pattern recognition and stochastic modelling, where training data are heavily exploited and local decisions are avoided as far as possible [1,2].

The characteristic features of the approach to be presented are:

- A large-size acoustic vector capturing first- and second-order derivatives is used. There is no splitting into separate streams as in most other systems that use tied mixtures.
- The Viterbi criterion is used both in training and recognition. Continuous mixture densities are used in a way that amounts to what could be called 'statistical template matching'.
- Linear discriminant analysis improves the acoustic analysis.
- For bigram language modelling, a non-linear interpolation has been developed that gives consistently lower perplexities[4] than linear interpolation, especially for small training corpora.
- The concept of time-synchronous beam search has been extended towards a tree organization of the pronunciation lexicon, so that the search effort is significantly reduced. A phoneme look-ahead technique results in an additional improvement.

The organization of the paper is as follows. We first summarize the statistical approach to speech recognition and the experimental conditions of our dictation task. We then describe the tasks on which we develop and evaluate our system. In the system description which follows, we describe the four main entities of our recognizer: acoustic analysis, acoustic–phonetic modelling, language modelling and search; experimental results are included within the sections. To allow a comparison with the performance of other systems, a section on our North American Business news system including the November 1994 evaluation (dictation benchmark test, American English) is supplied.

2. The statistical approach to automatic speech recognition

2.1. Problems of speech recognition

We first have to understand why automatic speech recognition is a difficult task. To put it briefly, the main problem is variability. Even one and the same

[4] Defined in Section 6, 'Language modelling'.

person is unable to exactly reproduce an utterance a second time. There is a lot of variability that cannot be eliminated or effectively controlled:

- variability between speakers (e.g. different dialects, vocal tracts, ways of speaking);
- variability of sounds or words, even for the same speaker (e.g. due to context, mood of speaker);
- speaking rate;
- varying or unknown channel characteristics (e.g. different telephone lines, microphones);
- background noise (e.g. car noise, music, conversation).

Facing this situation, the right approach seems to be a statistical model of speech production. Indeed, the statistical approach to speech recognition has proven to be very successful in the last two decades.

2.2. The system architecture

The statistical approach delivers only a framework in which many choices are possible. Assume that the probabilities describing speech production were known. In this case, there is a decision procedure (Bayes' decision rule) that guarantees minimal decision error rate. However, the 'real' probabilities are unknown; instead, probability estimates have to be derived from both available data and prior knowledge. In research, we develop probabilistic models (using a comparatively small amount of prior knowledge), with free parameters that can be efficiently estimated on some limited amount of training data and that model the reality as closely as possible, i.e. that perform with a low error rate on new test material. Hidden Markov models (HMMs) and maximum-likelihood estimation are common in the speech recognition community; however, neither the model nor the estimation criterion and method are specified by the statistical approach itself [1].

Following the statistical approach, our speech recognizer can be broken up into four parts. Figure 1 presents a block diagram of the system architecture. In the pre-processing step of *acoustic analysis*, the speech signal is transformed into a sequence of acoustic vectors $x_1, \ldots, x_T$ (over time $t = 1, \ldots, T$). As the speech signal, and thus this sequence of observations, is not exactly reproducible, a statistical approach is used to model its generation. According to statistical decision theory, in order to minimize the probability of recognition

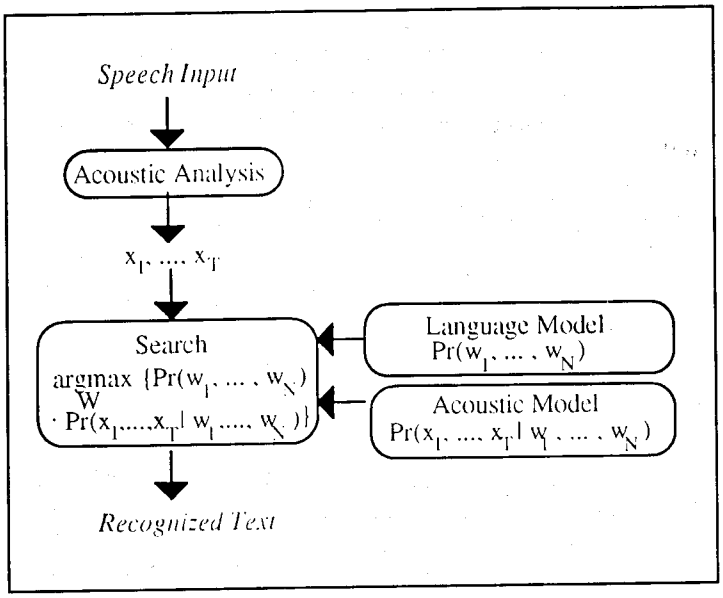

Fig. 1. System architecture.

errors, one should select the word sequence $W = w_1, \ldots, w_N$ (of unknown length N) that maximizes [3]

$$\Pr(w_1, ..., w_N | x_1, ..., x_T) = \frac{\Pr(w_1, ..., w_N) \cdot \Pr(x_1, ..., x_T | w_1, ..., w_N)}{\Pr(x_1, ..., x_T)}$$

As the denominator is constant for a given observation, this amounts to finding $w_1, ..., w_N$ that maximizes

$$\Pr(w_1, ..., w_N)\Pr(x_1, ..., x_T | w_1, ..., w_N)$$

The first term, the *a-priori* probability of word sequences $\Pr(w_1, \ldots, w_N)$, is independent of the acoustic observations and is completely specified by the *language model*. It reflects the system's knowledge of how to concatenate words of the vocabulary to form whole sentences and thus captures syntactic and semantic restriction.

The *acoustic–phonetic modelling* is reflected by the second term. $\Pr(x_1, \ldots, x_T | w_1, \ldots, w_N)$ is the conditional probability of observing the acoustic vectors $x_1, \ldots, x_T$ when the words $w_1, \ldots, w_N$ were uttered. These probabilities are estimated during the training phase of the recognition system. A large-vocabulary system typically is based on subword units like phonemes, which are concatenated according to the *pronunciation dictionary* to form the word models.

The decision on the spoken words must be taken by an optimization procedure which combines information of the language model and acoustic model, the latter being based on the phoneme models and the pronunciation dictionary. The optimization procedure is usually referred to as *search* in a state space defined by the knowledge sources.

2.3. Restrictions on the speech recognition conditions

As unrestricted, human-like speech recognition is not as yet possible today, several restrictions typically are put on using a recognition system in order to achieve an acceptable recognition performance. The most prominent are:

Speaker dependence: Speaker-dependent systems require prior training by the user, speaker-independent systems do not. A speaker-adaptive system starts speaker-independently and then adapts to the speaker, requiring less speaker-specific training material than a speaker-dependent system.

Input mode: Continuous-speech input is the way people normally talk, while isolated word recognition requires pauses introduced between the words.

Channel conditions: They range from high-quality recordings to telephone line channels and from clean recordings to recordings with, possibly strong, background noise.

Vocabulary size: From small (below 50 or 100 words) over medium to large (over 1000) vocabulary.

Other important factors are the perplexity [4] of the task (a measure of the intrinsic complexity of a task), spontaneity of speech and the consistency between training and test conditions.

3. Evaluation of the prototype system

Before focusing on large-vocabulary tasks, let us briefly describe the other conditions under which our speech recognition system is used. While it remains essentially the same system, several obvious modifications reflect the varying needs of these tasks. Giving two obvious examples, we use ('soft') m-gram language models for dictation, but ('hard') finite-state grammars for voice command; and the system is based on phonemes for large vocabulary, while it is based on whole-word models for small vocabulary.

For *small vocabulary* (several to ten words), the system is used for speaker-dependent and speaker-independent voice command or digit recognition over

the telephone or in a car. A commercial system for voice dialling in a mobile telephone has been developed at Philips Communication Systems [5,6]. The application is characterized by strong background noise during recognition that is absent in the training phase. In connection with our small-vocabulary activities, we reported benchmark experiments on the TI[5] digit string database [7]. Research work within the small-vocabulary framework is described in reference [8].

The large-vocabulary (over 1000 words) version of the system is used for the automatic transcription of dictations (described below) and for an *automatic inquiry system*. This research prototype of a train inquiry system adds speech understanding and dialogue capabilities to the speech recognition part. It accepts speaker-independent input over the telephone. Its speech understanding component performs a database query, and the dialog component generates responses that are transformed into spoken output. The system does not assume a standard form of dialogue and asks back for missing information. Our automatic information system is described in references [9–12].

Large-vocabulary continuous-speech recognition for dictation is the main focus of this paper, as it is prototypical for research done in the field of speech recognition. In each of the topics acoustic modelling, language modelling, and search, improvements have to be achieved to gradually increase the overall system performance.

The large-vocabulary recognizer is run under two different experimental conditions: a Philips internal dictation task, connected to commercial requirements of the professional dictation system SP6000 of Philips Dictation Systems, and the official evaluation conditions as standardized by the US ARPA (Advanced Research Projects Agency). These conditions are different and serve different purposes.

3.1. The Philips SP6000 dictation conditions

Under the Philips dictation conditions, our system is speaker dependent, which is motivated by the fact that a high performance is necessary for the everyday use of the system in a real environment. In addition, in a professional environment there are sufficient data available to train the system. The language is German, which made the data collection and the field tests easier for us. The SP6000 is installed in several Austrian and German hospitals.

[5] Texas Instruments.

In data supplied to the scientific community, one typically tries to separate the different effects that make speech recognition difficult. In contrast, the off-line experiments of our dictation system were performed with data coming from real applications which are characterized by an accumulation of difficulties like the following:

- We observe something like spontaneous speech (with hesitations and strongly varying speaking rates) — less difficult than spontaneous speech in a human dialogue, but still different from read speech.
- Speakers sometimes strongly vary the position of the microphone (varying input channel).
- We partly used analogue tape recordings. Due to the fact that the angular velocity of one tape winding is constant for the minicassette, the bandwidth varies with the recording's position on the tape.
- Available text material for language modelling is limited for real applications. Typical values for speaker- or site-specific text corpora are 100 k to 1 M words. In addition, there is often a mismatch between uttered words and writing in real applications, such that exact transcriptions of the utterances would be preferable.
- Open vocabulary: spoken words not included in the recognition vocabulary produce recognition errors (typically 1.5 to 2 errors per missing word). We do not apply any word rejection methods for dictation, because the missing word has to be manually inserted anyway.
- Other robustness issues: in real life, it is unfortunately *not* guaranteed that the training script is consistent with the recording.

We give a very brief look on experiments conducted in connection with our speaker-dependent dictation task: the data in these experiments are real-life field data from professional text producers. Speakers M-60 and M-61 are lawyers, M-72 and M-73 are radiologists. All speakers are male and work in Vienna, Austria. The speakers were asked to dictate as usual; this includes verbalized punctuation. The dictations were recorded with hand-held microphones on analog desktop dictation equipment. (Later experiments with digital recordings showed roughly the same performance.) We processed exactly the same recordings that were also given to the secretaries for transcription. Although all speakers are very experienced with dictation, we found that recognition was harder on this material than on read texts.

Before we give experimental results for specific aspects of the system,

TABLE I

Baseline system on 4 field test speakers. Bigram LM (language model), look-ahead, no linear discriminant analysis (LDA), 9 h training, 16 000 mixture components

Speaker	Vocabulary	Test-set perplexity	Active states /centisecond	Word-error rate in (%) Del.	Ins.	Total
M-60	12 073	113	8700	3.1	1.0	10.2
M-61	15 188	176	9300	1.9	1.5	12.1
M-72	13 095	267	11 600	2.4	1.9	11.2
M-73	13 095	42	14 000	0.6	1.7	5.7

Table I can serve as a reference on the system's performance under conditions to be explained later: a relatively high acoustic resolution (16 000 mixture components) and about 9 hours of training material, but without Linear Discriminant Analysis (LDA). The test material comprised 2000 to 3000 spoken words per speaker. The number of active states per centisecond before pruning is a measure of the computational effort required for search.

A second experimental set-up is defined by standardized publicly available databases. The coming sections are devoted to this topic.

3.2. Why and when benchmarking?

To some extent, comparison between speech recognizers is part of the scientific competition: the quality of our work is largely reflected in the ability of the acoustic and language models to model reality — which is typically measured in terms of word error rate, given fixed experimental conditions. More importantly, reproducible benchmark tests allow us to validate the importance and significance of improvements across research sites.

The boundary conditions for the development of a pure research system differ somewhat from the development of a dictation system for real use. There are a lot of data available that represent the task well. As the only optimization criterion is performance in terms of error rate, we take a much finer acoustic resolution, for memory demands and processing time play a minor role here.

So far, we have benchmarked our system on

- the TI digit-string database [7];
- the DARPA RM (resource management) task [1,13–15] and participated in the last official evaluation [15];

- the Nov. '92 and the Nov. '93 evaluations (official participation for Nov. '93) of the Wall Street Journal task [16];
- the North American Business (NAB) news task (official participation in the Nov. '94 evaluation).

The latter is described in some detail in the subsequent sections.

3.3. Description of the North American Business news task

Since 1986, ARPA has organized periodic formal evaluations of Continuous-Speech Recognition (CSR) technology. Those evaluations were highly regarded for the competitive stimulus they produced, resulting in the rapid assimilation of new techniques across the CSR community worldwide.

In 1992/93, the ARPA-charted CSR Corpus Coordinating Committee (CCCC) defined a corpus and specified an evaluation scheme, the Hub and Spoke evaluation paradigm. It was conceived to accommodate the research requirements of this diverse community and to generate convincing demonstrations of technological capability. Tests were defined, to exercise the primary interests of all participants, and to include important comparisons needed to make informed decisions about the efficacy of a particular algorithm or general approach. At the same time, the evaluation preserved the important controlled baseline test, characteristic of past ARPA-sponsored evaluations, that permitted direct comparison of CSR technology across different systems.

Starting with articles from the Wall Street Journal (WSJ), the corpus was extended to five American newspapers, the so-called North American Business (NAB) news (Washington Post, New York Times, Los Angeles Times, Dow Jones Information Services and Reuters North American Business Report). While the WSJ task was artificially limited to a 64 k-word vocabulary, the NAB vocabulary for recognition is unlimited.

3.3.1. The Hub and Spoke evaluation paradigm

The Hub and Spoke evaluation paradigm [17] implies an array of fairly independent tests (the Spokes) coupled to a central test (the Hub) in some informative fashion. The Hub test is further distinguished by being an abstract representation of a fundamentally important problem in CSR and by being the only test required of all participants in the evaluation. It forms the basis for all informative *inter*-system comparisons.

The Spoke tests, on the other hand, are abstractions of problems of

somewhat less central importance in CSR and evaluation on them is optional. The Spoke tests can be informatively compared to the Hub test to calibrate the difficulty of the problem, but they are otherwise independent. The Spoke tests are specifically designed to permit *intra*-system comparisons of algorithms and methods for problems involving mismatches between training and test data.

Each Hub or Spoke test consists of a primary condition (designated the P0 condition) and several contrastive conditions (designated C1, C2, ...). In general, the primary tests are unconstrained with respect to the lexicon and acoustic or language model (LM) training allowed. The purpose of the primary condition in each test is to showcase an algorithmic or procedural solution to a problem in CSR.

In giving very strong constraints for the different test conditions, the Hub and Spoke paradigm allows a *glass box* comparison of methods. The result is a rich array of comparative and contrastive results on several important problems in large-vocabulary CSR, all calibrated to the current state-of-the-art performance levels. A complete listing of the numerical results for 1993 can be found in reference [18]. For interpretive results, the interested reader should consult the current papers of the participating sites.

It is important to remember that the only tests for which fair and informative comparisons can be made across systems (and sites) are the controlled C1 contrasts for either of the two Hub tests. All other tests are designed to produce informative comparisons only within a given system run in two contrastive modes. So in general, only within-system comparisons should be made on the Spoke tests.

The Hub and Spoke evaluation paradigm appears to have met the competing requirements of supporting the variety of important research interests within the ARPA CSR community, while providing a mechanism to focus that work into well-defined and competitively charged evaluations of enabling technology.

3.3.2. 1994 Hub and Spoke test descriptions

The abstract problem represented by all the tests in the 1994 evaluation was the dictation of news stories, with an emphasis on financial news. Most of the tests in the 1994 evaluation used speech data from subjects reading diverse articles from the 5 different NAB newspapers mentioned above. Typical tests used 20 subjects reading 15 to 20 sentences each. Each test had equal numbers of male and female subjects. The primary microphone was the Sennheiser HMD-410.

3.3.3. The 1994 Hub: H1 — unlimited vocabulary read NAB news baseline

The Hub for 1994 was split into two tests differing in recording conditions [office quality (hub H1) and telephone quality (hub H2) recordings]. It was designed to measure state-of-the-art performance on an unlimited-vocabulary speaker-independent test, using clean test data that were well-matched to the training data.

The primary H1 test (H1–P0) allowed any language model (LM) or acoustic training data to be used. In addition, the temporal order of the utterances and the location of subject-session boundaries in the utterance sequence was given to encourage the use of unsupervised incremental adaptation techniques. To permit direct comparisons of acoustic modelling technology between different systems, a required contrastive test (H1-C1) controlled the amount of training data and specified the LM statistics. This contrast was run as a static speaker-independent test, so utterance order and session boundaries were not given to the system.

For H1-C1, the acoustic training data were limited to 62 hours of speech drawn from one of two segments of the combined WSJ0 and WSJ1 corpora. One segment was made up of speech data from 284 subjects (the 'short-term speakers') who produced 100 to 150 utterances each. The other segment had 37 subjects ('long-term speakers') who produced either 600 or 1200 utterances each. Participants were free to choose which acoustic training corpus to use.

The common required LM specified for the H1-C1 test was produced by Rosenberg at Carnegie Mellon University. It was a 3-gram back-off LM estimated from 247 M words of text: a 3-year WSJ0 text corpus (1987–1989) of 121 M words, 115 M words from Agency Press and 11.6 M words from the San Jose Mercury. Its lexicon was defined as the 20 k most frequent words in the corpus, hence, the test contained some words outside the vocabulary.

Training data as described above, both speech and text, were made available to all partners who wished to participate at the benchmarking.

An optional contrast test, H1-C2, was specified as an extension of the H1-P0 where supervised adaptation was allowed.

3.3.4. The 1994 Spokes

There were 7 Spoke tests in the 1994 evaluation that were designed to support the major interests of the participating sites at the time.

Spoke S0 as a 5 k word test is intended for calibration of systems used in other 5 k Spokes (S3, S4, S5 and S10). Spoke S2 supported problems in LM adaptation primarily. Non-business news (e.g. on AIDS) is to be recognized. A small corpus of 10 k words on the same subject is given to adapt the LM.

Spokes S3 and S4 were targeted at speaker adaptation methods, S3 being particularized to non-native speaker adaptation. Adaptation to different microphones was the focus of Spoke S5. Noise recorded in a car travelling at 55 mph with closed windows and air-conditioning turned on has been digitally added to read speech to allow noise reduction in Spoke S10. Spoke S9 looked at data from a potential application for large-vocabulary CSR — spontaneous dictation of news stories from print-media journalists.

All spokes except S2 and S9 used read speech from the WSJ0 5 k-word prompting texts. All spokes except S5 used data from the Sennheiser microphone.

4. Feature extraction

After the description of the evaluation conditions, let us describe the speech recognition system, beginning with the feature extraction module.

4.1. Spectral analysis

The acoustic signal is low-pass-filtered and digitized with a sampling frequency of 16 kHz. The following steps are performed for every frame, i.e. every 10 ms:

- Application of a Hamming window to a 25-ms segment.
- 512-point FFT after padding with zero-valued samples.
- Cepstral smoothing of the logarithmic FFT intensities using a $\sin(x)/x$ kernel function.
- In the range from 200 Hz to 6400 Hz, sampling at 30 frequency points that roughly correspond to a Mel-frequency scale.
- Normalization of the 30 spectral intensities with respect to their mean value. Together with this 'energy' value, they form the 31-dimensional acoustic vector $y(t)$.

To account for varying recording conditions in the dictation task, each acoustic vector is normalized with respect to the long-term spectrum as obtained by averaging over a part of the sentence (similar to cepstral subtraction).

For the recognition of smaller vocabularies, the sampling method is applied with modified parameters: typical values are 8 kHz sampling rate (with the obvious modifications of the frequency band definitions) and 10 to 16 ms frame width.

In order to capture the temporal structure of the speech signal, each acoustic vector $y(t)$ is then augmented by slope and curvature information over the time axis. Thus, the original sequence $y(t)$ of acoustic vectors is replaced by

$$x(t) := \begin{bmatrix} y(t) \\ y'(t) \\ y''(t) \end{bmatrix} = \begin{bmatrix} y(t) \\ y(t) - y(t - \Delta t) \\ y(t + \Delta t) - 2y(t) + y(t - \Delta t) \end{bmatrix}, \quad (1)$$

where the first- and second-order differences were chosen to cover the time intervals $[t - \Delta t, t]$ and $[t - \Delta t, t + \Delta t]$, respectively. The time delay Δt is typically 30 ms. The new sequence of acoustic vectors $x_1, \ldots, x_T$ in a higher-dimensional vector space serves as input to the subsequent processing steps. For the first and second differences of the 30 spectral intensities, pairs of adjacent spectral intensities are averaged so that the final vector consists of 63 components: 30 spectral intensities, 15 first- and 15 second-order differences, and 3 components representing energy and its differences.

4.2. Linear discriminant analysis

Linear discriminant analysis (LDA) is a well-known technique in statistical pattern classification for improving the discrimination between classes in a high-dimensional vector space [19]. The basic idea is to find a linear transformation such that a suitable criterion of class separability is maximized. The transformation is obtained as the eigenvector decomposition of the product of two scatter or covariance matrices, the total-scatter matrix and the inverse of the average within-class scatter matrix. Recently, this technique has been successfully applied to speech recognition, for both small- [7,20] and large-vocabulary tasks [21].

When applying LDA to speech recognition, the choice of the proper classes to be discriminated is not obvious — are they whole phonemes, phoneme states or the mixture components of a state? Our experiments indicated that the states are a good choice. The computation of the LDA transform is further complicated by the time alignment problem. Therefore, we use a three-step training. With our standard iterative training we obtain a segmentation of the training data, which provides the class labels for the subsequent estimation of the LDA transform. The third step is a new iterative training using LDA-transformed acoustic vectors.

Table II shows the improvement by LDA. Note that since a *single* class-*in*dependent transformation matrix is used, the matrix multiplication is done in the acoustic front end once per frame rather than for each log-likelihood calculation. Experiments on other databases [15] showed that even for

TABLE II
Effect of linear discriminant analysis (LDA) on the word-error rate (in %). About 3 h of training material, 4000 densities

Speaker	No LDA	LDA
M-60	12.3	10.4
M-61	15.0	12.3

speaker-independent recognition, one single transformation gives satisfactory results.

5. Acoustic–phonetic modelling

5.1. Mixture densities

The acoustic conditional probabilities $\Pr(x_1,\ldots,x_T/w_1,\ldots,w_N)$ are obtained by concatenating the corresponding word models, which again are obtained by concatenating phoneme models according to the pronunciation lexicon. We use inventories of 40 to 50 phoneme symbols including symbols for silence and maybe glottal stop. (For the English language, we use triphones as basic units, Section 5.3.) As in many other systems, these subword units are modelled by stochastic finite-state automata, the so-called Hidden Markov Models (HMMs) [3,22,23].

For each state s of the HMM, there is an emission probability density $q(x_t|s)$ of generating the vector x_t. The phoneme unit shown in Fig. 2 has a tripartite structure in order to take account of left and right acoustic dependences. Each of the three parts consists of two states with identical (or tied) emission distributions. The transition probabilities, which allow loop, jump and skip, are tied over all states. Unlike most other HMM structures, this

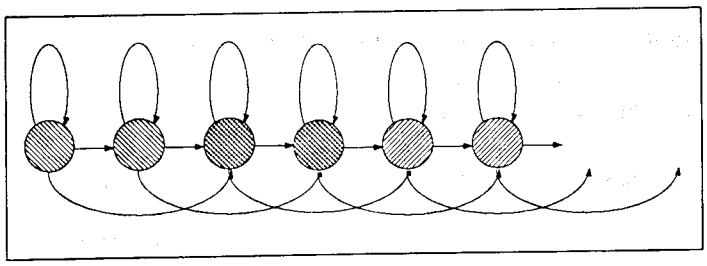

Fig. 2. Topology of phoneme Hidden Markov Model (HMM).

structure has a simple duration model whose most likely duration of 60 ms is close to the average phoneme duration.

No pronunciation variants are used in the pronunciation lexicon, such that the emission distributions have to model deviations from the standard pronunciation as well as coarticulatory effects. The best results were obtained for continuous mixture densities

$$q(x_t|s) = \sum_k c_k(s) b_k(x_t|s) \quad \text{with} \quad 0 \le c_k(s) \le 1 \quad \text{and} \quad \sum_k c_k(s) = 1, \quad (2)$$

where the so-called component densities $b_k(.|.)$ are unimodal densities such as Gaussians or (as in our system) Laplacians:

$$b_k(x_t|s) = \prod_n \left(\frac{1}{2v(n)} \right) \cdot \exp\left(- \sum_n \frac{|x_t(n) - r_{k,s}(n)|}{v(n)} \right) \quad (3)$$

n is the index of the vector components. Each density is completely specified by its location vector $r_{k,s}$. The vector v of absolute deviations is assumed to be independent of both the component densities and the states, and thus serves as an overall scaling for the acoustic vectors.

In contrast to other systems, the Viterbi criterion is used both in training and recognition. This applies even to the level of mixture components, such that the sum over the component densities in eq. (2) is replaced by their maximum [1].

Table III shows how the error-rate depends on the training-set size and the acoustic resolution. Monophones (i.e. context-independent phonemes) were used here; so far, we could only achieve slight improvements with context-dependent phonemes in German.

TABLE III

Error-rate as a function of training set size and number of densities. Speaker M-60, vocabulary size 12 073 words, test-set perplexity 113

Training material no. of densities	0.7 h	1.2 h	2.0 h	3.2 h	9.5 h
4000	16.1%	14.4%	13.1%	12.3%	11.4%
8000	—	13.4%	12.9%	11.7%	10.8%
16 000	—	—	—	11.6%	10.1%
32 000	—	—	—	—	10.0%
64 000	—	—	—	—	9.1%

While we typically develop our system on the speaker-dependent German dictation task, we have also successfully benchmarked our system on both the speaker-dependent and the speaker-independent part of the well-known American English DARPA (Defense Advanced Research Projects Agency) RM (resource management) task [1,15] and on the ARPA Wall Street Journal (WSJ) and North American Business (NAB) news tasks. The major modifications of our system and the WSJ benchmark results are described in Sections 5.2, 7.6 and 7.7.

5.2. State tying

Speaker-independent recognition in a very-large-vocabulary task like the Wall Street Journal (WSJ) and the North American Business (NAB) news tasks imposes very strong requirements on the acoustic–phonetic modelling of the recognition system. In order to both improve performance and reduce the number of densities, a clustering technique has been integrated into the acoustic–phonetic training procedure of our continuous-density hidden Markov model (HMM) speech recognizer [24]. The main idea of clustering is to concentrate what is acoustically similar. For a continuous-density HMM system, acoustic similarity can be seen at different levels: at the phoneme level (triphone), the state (or mixture) level and the density level.

Clustering at the first two levels (phoneme and state) leads to model-tying and state-tying, respectively. It answers the question 'Which triphones are acoustically similar?' and helps us to define a reduced set of models to be trained. It should give us the possibility to avoid the duplication of models, and therefore reduce the number of parameters of our system. Furthermore, it can more efficiently exploit the training material, for example, while training rarely seen states together with more robust ones. Clustering at this level is also known in the literature as tying. Having in mind the work at Carnegie-Mellon University [25] and at Cambridge University [26], we decided to concentrate on state-tying rather than triphone tying.

Our state-tying technique is very similar to reference [26]. A furthest-neighbour criterion has been applied directly to the spectral mean vectors. Whereas the furthest-neighbour does not quantify the soundness of a rarely seen model, it takes the spectral mean vectors as they have been observed. The distance measure

$$d(C_i, C_j) = \max_{m_k \in C_i, n_l \in C_j} \left[\sum_c (|m_{k,c} - n_{l,c}|) \right] \tag{4}$$

calculates the distance between two clusters C_i and C_j, where each cluster is defined by a set of mean vectors m_k and n_l. Two clusters are clustered together

if their distance lies below a certain threshold. The new cluster will be the union of the original clusters.

5.3. Experimental results

Table IV presents results for different initial numbers of states and different thresholds on the maximum diameter of a cluster ('cluster threshold') using the furthest-neighbour criterion. The training set used for these experiments is the so-called WSJ0 training set summing up to around 15 hours of speech equally balanced between male and female speakers. Our base system (WSJ0-a) is the non-tied system optimized for the 5000 word vocabulary WSJ benchmarking test from November 1993 [16]. The optimal number of triphone-states for our base system had been set to 2208 + 130 triphone and monophone states, respectively. As can be seen from the first two lines of Table IV, applying state-tying led to a reduction of the number of densities by a factor of two without changing the total word error rate over three test sets. These three test sets consist in total of 20 774 pronounced words.

State-tying allows to group together triphones which are acoustically similar but not necessarily often seen. The consequence is that more triphones can be modelled: the triphone coverage of the test set lexicon will be higher. We increased the number of triphones to be modelled and found an optimum at 1855 triphones (1855*3 = 5565 states), which defines our second system (WSJ0-b, lines 3 and 4 from Table IV). This leads to word error rate improvements of more than 6% on the same set of test sets, when compared to the WSJ0-a system.

As stated above, the optimal number of triphones modelled on the WSJ0 material was 1855. We give hereafter results while modelling all 7836 triphones seen during training. A drawback in modelling all triphones present in a training set is that there is no observation left to model backing-off monophones. During recognition, a decision has to be taken: To which trained model will be assigned the untrained but essential new triphone? Decision trees are often used at this place.

Our solution was very pragmatic: we took from our WSJ0-b system the monophone backing-off models, properly re-scaled, and added them to our all-triphone system. As can be seen from Table IV, the word error rate (WER) increased significantly with respect to our WSJ0-b system and goes back to the WER level of our WSJ0-a system. To interpret the result, it has to be observed that from the 7836 different triphones occurring in the training, 3781 occur less than 10 times. Our conclusion is that under a certain occurrence threshold (that is 35) state-tying results in a splitting of rarely seen

TABLE IV

Bigram word error rates on different Wall Street Journal (WSJ) test sets with and without state-tying. Training was done on WSJ0

No. of states before tying	Cluster threshold	No. of states after tying	No. of densities (male + female)	5 k test set *(si_ev15, si_dev5, si_dt_05)*	Test set triphone coverage
2208	0	2208	245 k	11.90%	75%
2208	16	1336	115 k	11.92%	
5565	0	5565	225 k	12.02%	90%
5565	16	2435	163 k	11.12%	
23 508 (all triphones)	15	4621	235 k	12.20%	99.7%

training material (that would otherwise be globally modelled in a monophone) and leads to less robust modelling.

The next step was to build our models on the bigger WSJ0+1 training set totalling 62 hours of speech. To augment the triphone coverage on the recognition vocabulary, we included right context diphones to our triphone list (4087 triphones and 557 diphones). The backing-off monophone models were seen on average 350 times. Table V shows that by state-tying an improvement in the WER by more than 7% for about the same amount of densities has been achieved on the evaluation set of November 1993. This is mostly due to the triphone coverage ratio increase on the test set, from 90 to 99.6%.

6. Language modelling

The language model provides, for each word sequence, an estimate of probabilities $\Pr(w_1, \ldots, w_n)$ usually expressed by m-gram models (cf. below), which have established themselves as both a good way to reliably estimate the parameters and to keep them limited so they can be stored and retrieved. In view of the sizes of available corpora, we typically use word bigram models or a category-based bigram models (bigram class models) with automatically generated classes [27]. An overview about more general techniques in language modelling can be found in reference [28].

While maximum-likelihood estimation would suggest taking relative frequencies of bigram counts, it is common knowledge that these are particularly

TABLE V

Bigram word error rates on the evaluation test set 93 with and without state-tying for the same number of densities. Training was done on WSJ0+1

Init. no. of states	5592	18276
Cluster threshold	0	16
No. of states after tying	5722	4166
No. of densities (male + female)	523 k	495 k
20 k eval_93	17.7%	16.4%
Test-set triphone coverage	90%	99.6%

bad as estimates and that smoothing is important. The smoothing method that we use is different from those used in other systems and is explained in the following section in more detail. With this method, we achieve better results than with backing-off or linear interpolation.

6.1. Stochastic bigram and trigram models

The task of providing probabilities $\Pr(w_1,\ldots,w_n)>0$ is usually reduced to the problem of estimating conditional probabilities $\Pr(w_j|w_1,\ldots,w_{j-1})$ with given history $w_1,\ldots,w_{j-1}$ which determines the joint probabilities by the product

$$\Pr(w_1,\ldots,w_n)=\prod_{j=1}^{n}\Pr(w_j|w_1,\ldots w_{j-1})$$

Because of the limited training data one has to share the same distribution for different histories, e.g. histories which coincide with the last $m-1$ positions. Depending on the amount and structure of training data we typically use only m-gram models with $m=2$ (bigram) or $m=3$ (trigram). Even for such small history lengths there are a lot of possible bi- or trigram events which have not been observed during training before. So we are faced with the problem of guessing a non-zero probability for an event which has never been observed before. To do this in a serious way we have to use further knowledge about the stochastic process we want to describe.

Beside the well-known technique of linear interpolation, the theory for most of the commonly used estimators was established in 1953 by Good [29] who worked out an idea of Alan M. Turing; but in order to come up with practically useful 'Turing–Good' estimators one has to use some kind of smoothing.

The non-linear interpolation scheme used in our system has the advantage to do this in a way which is easy to implement. More precisely, in the case of bigram and trigram models [30], it is possible to make a first-order approximation of the Turing–Good formula which simplifies it to subtracting a constant d (typically between zero and one) from counts greater than d. Redistributing the gained probability mass to some *a-priori* distribution q leads to the concept of non-linear interpolation as introduced by reference [31].

To be more explicit, e.g. for a bigram application, let us denote the count of some bigram (v,w) in a given training corpus by $N(v,w)$. Then we may define the estimator for a bigram language model by

$$p(w|v):=\begin{cases}\dfrac{N(v,w)-d+\beta_v\cdot q(w)}{N(v)} & \text{if } N(v,w)>d\\[2ex] \dfrac{\beta_v}{N(v)}\cdot q(w) & \text{if } N(v,w)\le d\end{cases}\qquad(5)$$

if $N(v) := \Sigma_w N(v, w)$ is assumed to be positive and β_v is chosen so as to assure the constraint $\Sigma_w p(w|v) = 1$. Here q is usually chosen to be a unigram distribution. Defining a discounting function $\delta(v, w) := \min\{d, N(v, w)\}$ we easily get $\beta_v = \Sigma_w \delta(v, w)$ as well as

$$p(w|v) = \frac{N(v, w) - \delta(v, w) + \beta_v \cdot q(w)}{N(v)} \tag{6}$$

which describes a general interpolation scheme between q and the relative frequency distribution. The name *non-linear interpolation* indicates the difference from the well-known *linear interpolation* with parameter α which appears if we choose $\delta(v, w) := \alpha N(v, w)$.

6.2. Application-specific experimental results

From the theoretical derivation it is clear that non-linear interpolation is designed to incorporate different statistical knowledge (e.g. about unigram and bigram) in a way which respects the advantage of the Turing–Good estimator of providing better estimates even with relatively small training data.

In fact, in practice there are typically only small training corpora available which reflect the application and the speaker-specific characteristics. To compare the performance of non-linear interpolation and linear interpolation, we took spoken sentences from two lawyers (M-60 and M-61) and two radiologists (M-72 and M-73), as well as a larger corpus of written radiology reports (REP; see Table VI) to calculate the different test-set perplexities. (Recall that the test-set perplexity of a given language model on a test text $w_1, \ldots, w_n$ is defined as $\left[\Pi_{i=1}^{n} P(w_i|w_1, \ldots, w_{i-1})\right]^{-1/n}$ and that the logarithm of perplexity can be viewed as the empirical entropy for the actual test set.) As seen in Table VII, in all cases non-linear interpolation yields significantly lower (i.e. better) perplexities than linear interpolation. Furthermore, the relative gain becomes smaller for larger training material.

TABLE VI
Data sizes in words for specific applications

Data	Test	LM Training	Lexicon
M-60	2781	61 130	12 073
M-61	3039	71 208	15 188
M-72	2095	50 192	13 095
M-73	2296	54 375	13 095
REP	569 767	915 858	40 630

TABLE VII
Test-set perplexity for different discounting methods

Bigram-LM	Linear	Non-linear
M-60	127.0	112.2
M-61	206.4	183.2
M-72	299.8	286.9
M-73	47.4	41.9
REP	97.4	91.9

It should be noted that the unigram distribution q used in the experiments was also calculated with a non-linear interpolation scheme using a uniform distribution as background knowledge (Table VIII).

Of course all techniques may be applied also to trigrams using a conditional bigram distribution as the general background model. Even for small corpora it is possible to have a gain in perplexity if the training material gives a good coverage of frequently used phrases in a very special application (Table VIII).

To indicate that there is a great difference between specific well-tailored training material and general application-specific data, we used the unigram and bigram models trained on written radiology reports (REP) to calculate test-set perplexities on spoken radiology reports of M-72 and M-73. To make test results comparable with Table VIII, we used the lexicon of the M-72/M-73 corpus.

Tables IX and VII show that the language models trained on a small-sized corpus of speaker-specific sentences that were transcribed as spoken ('as-it-is files') perform much better than the models trained on the larger speaker-independent written text. This seems to indicate that specific data material is more important than some general kind of knowledge. Another reason

TABLE VIII
Test-set perplexity for m-gram language models ($m = 1,2,3$)

Non-linear	Unigram	Bigram	Trigram
M-60	818.9	112.2	81.2
M-61	933.4	183.2	151.2
M-72	1065.9	286.9	264.3
M-73	531.8	41.9	30.7
REP	832.4	91.9	66.5

TABLE IX
Test-set perplexities when using only written training corpora ('REP': without data as being dictated)

Test set	Unigram	Bigram
M-72	1822.5	705.2
M-73	1599.3	365.1

for this effect might be the general difference between spoken and written language. Most obvious examples for this difference (like abbreviations and punctuation) stem from some kind of mismatch between the words in written and spoken text.

6.3. Perplexity gain for large corpora

Although the techniques just presented perform quite well with small training material, there is still a strong gain in perplexity when using larger training corpora. To see the dependence between language model performance and training size we took differently sized subcorpora of up to 39 million words from the well-known Wall Street Journal corpus.

Figure 3 shows the significant loss in performance when only small corpora are used for training: The more (application specific) data, the better. This is even more true for a trigram model.

Figures on a trigram LM on a large corpus are given in Table XI. For more information on the experimental conditions, cf. Sections 3.3.3 and 7.7.5.

7. The search procedure

Time-synchronous beam search has successfully been used in the Philips continuous-speech recognizer for several years [32]. We found that it is efficient also for 10 k or more words [33]. First, all knowledge sources are available at the same level in the integrated search. Second, all hypotheses refer to the same acoustic vector sequence in time-synchronous search. These two key points allow a drastic reduction of the actual search space by pruning less promising hypotheses. A PC based implementation [34] underlines the efficiency of this search strategy.

Recently, we increased the vocabulary size in connection with our NAB benchmark system up to 64 k words. Our positive experiences are described in the last section.

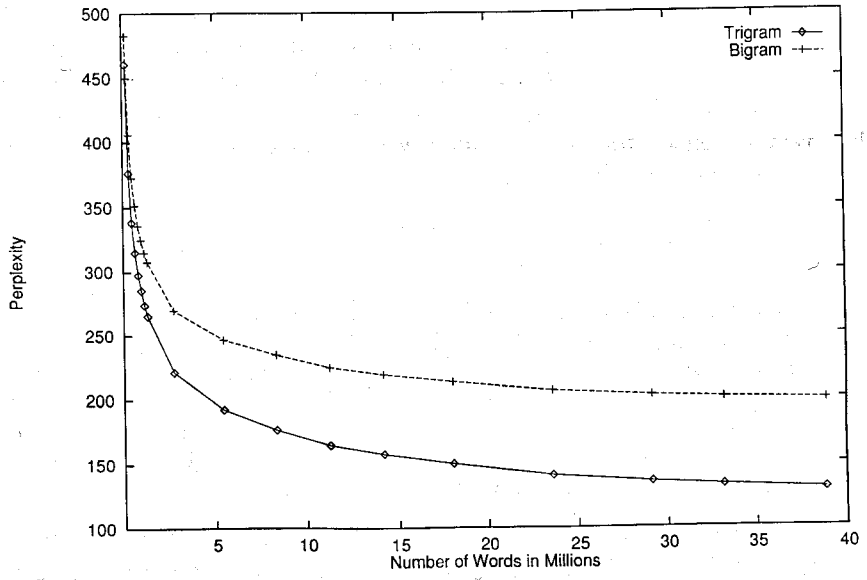

Fig. 3. Test-set perplexity for bigram and trigram language model depending on training set size.

7.1. Tree lexicon

A straightforward approach to constructing the search space is to synthetically build up word models from concatenating the appropriate phoneme models as given by the pronunciation lexicon. In this space, different copies of the same phoneme occur due to the lexical constraints. For similar reasons, the language model restrictions make it necessary to introduce several copies of the same word, representing contexts that allow for different continuations. This organization, where each state belongs to exactly one word, will be called *linear lexicon*.

When the lexicon becomes larger, e.g. from 1 k to 10 k words, it is more efficient to arrange the pronunciation lexicon as a tree of phonemes (*tree lexicon*). The compression factor for the tree lexicon as compared to the linear lexicon is even surpassed by the reduction in the number of active states, because most of the active states are located in the word beginnings (near the tree's root).

7.2. Forest search

The tree organization of the lexicon also has an undesired consequence for the organization of the search space. In contrast to a linear lexicon, the word identities are unknown at the word beginnings. Particularly for a bigram

language model, this means that separate tree copies have to be held, depending on the predecessor word. While the potential search space is blown up by a factor of the vocabulary size, e.g. 10 k, the actual search space grows much more moderately, typically by only a factor of 2. The tree organization is thus very beneficial for large-vocabulary tasks. A detailed discussion with experiments is given in reference [33].

7.3. Phoneme look-ahead

The phoneme look-ahead additionally reduces the number of active states by estimating whether a started phoneme will or will not survive the next few time frames (in our system typically 60 ms). In a first step, the likelihood of each phoneme ahead of the current time frame is estimated by carrying out a time-alignment. Then, each time a state hypothesis crosses a phoneme boundary, these figures are utilized for probability estimates for the best path extensions both of this and of any other state, which in turn are used to perform an additional pruning [35].

For the phoneme look-ahead, the original phoneme models are used without any simplification. Note that, in particular for the case of monophones, the number of generic states is much smaller than the number of state hypotheses. (Conversely, a non-modified application to a system with many triphones is not advantageous.) The likelihood scores[6] are stored for later use in the detailed match. Like the conventional search, the look-ahead is sped up by beam pruning; in addition, there is no need for book-keeping as in the detailed match. To further reduce computation, the look-ahead is carried out only every other time frame. For the omitted time frame, the look-ahead scores of the previous time frame are used.

7.4. Peaks in the search space: histogram pruning

Conventional beam pruning uses a pre-specified constant threshold to specify the beam of active hypotheses: at each time frame, exactly the hypotheses with log-probabilities close enough to the optimum at that time remain active, i.e. are considered for expansion at the next time; the others are pruned.

When the pruning threshold is chosen to be large enough to avoid search errors, i.e. when the optimal path is only rarely being lost due to pruning, large peaks in the actual number of active hypotheses can occur. We frequently observed peaks of 1 or 2 million hypotheses, and roughly 100 times larger

[6] We use the word 'score' for negative log-probabilities.

than the average number of hypotheses, especially for non-speech sounds or corrupted speech.

We thus introduced an additional pruning criterion: a pre-specified upper limit on the number of active points. We called this *histogram pruning* because we use a histogram on the hypotheses' scores in order to determine a pruning threshold (below a given value) such that the number of active hypotheses remains always below a given maximal number of active hypotheses.

Quite astonishingly, the experiments indicated that it is possible to choose relatively small maximal numbers for the hypotheses without introducing search errors. A typical value is 30 000 hypotheses maximum for our dictation research prototype. Besides the significant reduction in peak storage size needed, there is a reduction in the average search costs of about 30%. A detailed description of the experiments is given elsewhere [36].

7.5. Language-model look-ahead: smearing the expected LM probabilities over the tree

In the forest search organization for stochastic n-gram language models ($n > 1$), the potential search space consists of a large number of copies of the phonetic tree consisting of the recognition vocabulary. E.g., for a vocabulary of V words and a bigram LM, i.e. $n = 2$, there are $V^{n-1} = V$ copies of the phonetic tree. Informal experiments indicated that, due to beam pruning, the number of active hypotheses grows much smaller with n, like roughly a factor of 2 when going from unigram to bigram LM.

The word identities in the tree are only known at the word ends. Adding the LM log probabilities at the word ends leads to several effects that are disadvantageous for the search:

- As compared to linear search, the LM is employed with one word delay; but knowledge should be incorporated as early as possible.
- The scores of hypotheses change drastically when a word end is encountered. Especially, the pruning has to be larger than the largest LM score ('score' being defined in this paper as negative log probability).
- The same effect causes the examination of many useless word start hypotheses during silence after a word.

A remedy for all these pains is the incorporation of the LM scores as early as possible. For this purpose, in each search state, we introduce a new pruning criterion: instead of the usual score, we always investigate its sum with the minimum of the LM scores of all possible word continuations. A practical

implementation and experimental results are described in reference [36]. We achieved reductions in search space by factors of 3–5 with this method.

7.6. Lattice rescoring for longer span language models

In this and the following section, two different kinds of word lattices or word graphs are introduced, which we call *word lattice* (Section 7.6) and *word graph* (Section 7.7), respectively. Please note that the nomenclature is not standardized in the literature.

7.6.1. Basic concept

During our first tests with forest search, we made informal experiments indicating that forest search works not only with a bigram LM but also with a trigram LM, with only moderate increase of the active search space by an additional factor of roughly 2. However, for the recognition with a trigram LM, we decided to choose a different approach with a search effort about the same as for a bigram LM. In this two-step approach, a word lattice is first generated with a bigram LM and subsequently rescored with a trigram LM. The approach is open to employ more complex LMs in this post-processing step.

7.6.2. Generation of the word lattice

In the recent past, the use of word lattices or word graphs has become quite popular among the various search techniques applied to large-vocabulary continuous speech recognition [16,37–39]. The main idea about word graphs is to come up with word alternatives in regions of the speech signal where the ambiguity of the recognition is high and to apply subsequently more elaborate knowledge sources within this narrowed-down search space.

A word lattice can be efficiently generated with only minor modifications of our time-synchronous beam search algorithm based on a tree lexicon. It essentially amounts to collecting the information about word-endings as they occur in the course of the left to right decoding process. This first pass simultaneously provides the best bigram-scored sentence hypothesis, the lattice overhead being virtually negligible in terms of CPU time.

As opposed to the word-graph generation technique presented in reference [38], here we take full advantage of the bigram LM to constrain the lattice, without requiring any further optimization stage. More precisely, our analysis relies on the assumption that the position of a word boundary depends only on the word pair under consideration and not on further predecessor words. This simplification has been successfully used by BBN in their word-dependent

N-Best algorithm [40] and is also known as the 'word-pair approximation' [41].

Therefore, in the present study the lattice is defined as a time-structured list of word hypotheses consisting of word identity, start and end time, acoustic score and predecessor word identity. It has to be stressed that the collection of word-ending information is done before the bigram LM recombination takes place, to preserve as much as possible different word sequences for subsequent use with a higher-order LM.

The computational complexity of this first pass is nearly identical to that of our bigram beam search, the efficiency of which has been further improved by the new handling of the LM probabilities (see Section 7.5).

7.6.3. Trigram rescoring of the lattice

In this second pass, the trigram language model is applied to the lattice at the phrase level. More precisely, the acoustic probabilities of the word hypotheses are combined with the trigram probabilities taking account of the predecessor word as computed in the first pass. Searching for the optimal rescoring still proceeds time-synchronously and uses a Dynamic Programming (DP) recursion taking account of all time and predecessor constraints contained in the lattice [41]. The final output is the best trigram-scored sentence hypothesis under the lattice restrictions.

The optimality of this procedure (in the Viterbi sense) is preserved only under the following two conditions: the word-pair approximation for the position of a word boundary has to be valid and next, the beam used for generating the lattice must be wide enough to keep enough phrase hypotheses for subsequent trigram rescoring.

In practice, this algorithm appears to work well with relatively modest lattice densities. The computational costs are quite small since this second pass does not require any further acoustic scoring at the state level. This follows from the word-pair assumption which implies that the word boundaries have already been optimized in the first pass.

Moreover, a careful list organization allows the achievement of great efficiency (without requiring the cashing of the LM scores) to such an extent that the trigram rescoring represents only a few percent of the main bigram decoding CPU time.

7.7. Word graph search

In the method described in the previous section, the segmentation points optimized by the bigram decoding together with the acoustic scores of

respective word hypotheses are used *as such* during the trigram search, without having to compute a complete scoring at the 10-ms level again. Hence, a good decoupling is achieved between acoustic and syntactic levels since the long-span LM search is performed at the phrase level in a post-processing step.

However, the exact influence of the underlying word pair approximation is unknown and also, the use of different acoustic models implies that the word boundaries and scores have to be re-evaluated anyway. This concerns, for example, the integration of cross-word acoustic models or the use of an unsupervised speaker-adaptation scheme together with the best language model available. Therefore, we have implemented a general graph-search procedure either to perform 'full' trigram decoding or to efficiently implement more detailed acoustic models.

7.7.1. Construction of the word graph

Our starting point consists of the bigram lattice described in the previous section (also [2]). This is nothing but a time-structured list of word hypotheses consisting of word identity, start- and end-time, acoustic score and predecessor word identity. To represent all these word sequences by a graph data structure, the definition of a node has to be specified, each arc being a word hypothesis. Two cases have been considered.

In the general case, a node is simply a time-mark. This means that all word hypotheses ending at time t are pointing to the same node and might be followed by any word starting at $t+1$ in the word graph.

However, the success of the search algorithm [42] suggests that the predecessor word identity provided by the bigram decoding might be used to constrain the word graph without impairing the next pass. When this predecessor word dependence is to be kept, a node is defined as a pair [time, predecessor-ID]. In this 'bigram-constrained' word graph, the predecessor information is thus used to restrict the connections between succeeding words. Application of this constraint is supported by the observation that if a particular word pair has a very small (bigram) LM probability, any m-gram ($m>2$) including this word pair is likely to be also of very small probability.

On the other hand, we are no longer interested in the time and score information as we now intend to perform a full decoding at the 10-ms level, possibly using different acoustic models. Instead, we want to get rid of all copies of words occurring in the same contexts at consecutive time frames since they do not bring anything new in terms of syntactic richness and they will only burden the graph search process. To eliminate these copies of words appearing

in the same contexts, nodes that are closely spaced in time are merged using several reduction rules. This provides a very significant 'compaction' of the word graph.

7.7.2. Cross-word triphones using word phonetic networks

For each arc in the graph, a word model has to be specified in terms of elementary acoustic units. These are typically triphones conditioned on the left and right phonemes. When cross-word co-articulation effects are *explicitly* taken into account, the triphones at the beginning and end of a word depend on the neighboring words as given by the graph structure. Therefore, multiple triphone instances are created at the initial and final positions of a word model, the number of which depends on the local graph characteristics. Note that for the bigram-constrained word graph there is only one predecessor context.

Alternative pronunciations are introduced by allowing the substitution and skip of particular phones. Cross-word-dependent assimilation rules are also used to model 'hard' pronunciation changes that occur at word juncture [43], for example when a phone is completely deleted like in '... receive(d) the ...'. As a result, a phonetic network is built up for each word hypothesis and inserted in the graph together with optional between-word silence models.

7.7.3. Decoding

Decoding proceeds from left to right using a time-synchronous search algorithm with a beam-pruning technique. However, the word graph has first to be expanded with respect to all contextual constraints introduced by either the LM or the cross-word models. For an m-gram LM, words appearing in different contexts have to be duplicated to keep track of all hypotheses differing in their final $(m-1)$ words. Consecutive word arcs are then connected with language transitions whose probabilities are given by the m-gram LM. In the case of a trigram-LM, for example, separate arc copies are made for each predecessor word and are recombined at the end of the succeeding word. This implies that if the word graph exhibits a *local* branching factor of b, with b arcs pointing to and leaving each node, b^3 language transitions are requested which leads to a prohibitive number of arcs in the region of the sentence where the ambiguity of the speech signal is high.

So far, this problem has been solved mainly by relying on the back-off property of the LM, i.e. by duplicating an arc only if the corresponding m-gram has been taken explicitly into account by the LM [37,39]. Our solution consists of two parts:

- First, the word graph is expanded dynamically on demand, that is, only when a word-end hypothesis is reached and kept active within the beam.
- Second, bigram-constrained word graphs are used that request only b^2 language transitions for a trigram LM since the predecessor dependence has already been integrated.

7.7.4. Experimental results: impact of word pair approximation and predecessor constraint

To get some measure of the accuracy loss introduced in trigram decoding either by the word pair approximation or by the predecessor constraint, several graph search strategies have been tested on the November '92 WSJ evaluation set (4 males, 4 females, 5 k and 20 k vocabularies).

We first generated word graphs of high density using our standard bigram-LM beam search to ensure that the spoken word sequences were included in the word graphs whenever possible, i.e. in the absence of Out-of-Vocabulary (OOV) words. The details of the acoustic modelling and training are described in reference [16]. Then, trigram decoding has been performed under three different search conditions, all other things being identical:

- First, we used the phrase level search algorithm relying on the word pair approximation.
- Next, we applied the 'full' graph decoding procedure with large beam widths to a general graph data structure obtained from the original bigram word graph.
- Third, the same procedure was applied to a bigram-constrained word graph that preserves the predecessor information of the original bigram decoding, however, without time and score information.

Table X summarizes the recognition results at the word level, obtained with a trigram LM for 5 k and 20 k vocabularies. For each test condition, the word-error rate is given together with both the average and maximum numbers of word arcs expanded per sentence in the course of the graph-search process.

The following conclusions can be drawn:

- The word-pair approximation introduces a relative degradation of less than 2% and we did observe that essentially short words are affected.
- Compared to general word graphs, bigram-constrained word graphs achieve the same precision.

TABLE X
Trigram results on the Nov. '92 Wall Street Journal (WSJ) test sets

Algorithm	Word-error rate	Av. no. of arcs	Max. no. of arcs
5 k Closed vocabulary			
Word pair	4.90%	—	—
General graph	4.75%	5000	108 000
Bigram graph	4.75%	1300	18 000
20 k Open vocabulary			
Word pair	11.9%	—	—
General graph	11.8%	7000	114 000
Bigram graph	11.8%	1400	14 300

- The number of arcs expanded during search is drastically reduced in the last case due to the very low branching factors of bigram-constrained word graphs.

7.7.5. Experimental results: 64 k-word trigram and speaker adaptation

Using the full graph search procedure, we can combine trigram decoding with incremental speaker adaptation. The principle of incremental unsupervised speaker adaptation amounts to updating the acoustic models after each spoken sentence by using the alignment between the speech signal and the *recognized* word sequence. The success of such a scheme depends partly upon the correctness of the recognition, hence the interest in taking the best available LM.

This technique has been applied to the North American Business (NAB) corpus which contains read articles taken from several newspapers with an unlimited vocabulary. To achieve a high coverage, a vocabulary of 64 k words has been taken. Both the development and the evaluation set include 10 male and 10 female speakers, each having uttered 15 sentences of about 25 words each.

In our system, the acoustic models are based on mixtures of continuous densities, and so far the adaptation scheme has concerned only the mean vectors [44], the mixture weights being kept fixed. Table XII summarizes the 64 k-word

TABLE XI
NAB '94 (North American Business news) coverage and perplexity for 64 k vocabulary

Set	No. of words	% OOV	Bigram perplexity	Trigram perplexity
Dev	7387	0.53	230.0	137.2
Evl	8186	0.79	231.3	137.6

recognition results. Table XI gives some information about the LM conditions such as coverage and perplexity.

The density of the word graphs is expressed in terms of the average number of word hypotheses per spoken word. Note that for the evaluation set we used much larger beam widths to minimize the risk of search errors. The average number of arcs expanded per sentence during the trigram graph-search is also given together with the real-time factor of the decoding on a DEC Alpha work-station. About 75% of the CPU time is actually devoted to the log-likelihood computations. Speaker adaptation yields a relative improvement of about 5% while a trigram reduces the errors by about 20% with respect to a bigram LM.

8. Summary

We have described the large-vocabulary continuous-speech recognition system of the Philips Research group for speech recognition. The system shows state-of-the-art performance on several different benchmark tests. One of them, the North American Business news evaluation of 1994, has been described in more detail.

In the paper, we have concentrated on the features that are distinctive to our system: (1) In acoustic modelling, non-smoothed continuous mixture densities are used. The Viterbi criterion is consistently applied both in training and

TABLE XII
NAB '94 recognition results for 64 k vocabulary (WER = word error rate)

Set	Bigram-WER	Word graph density	Trigram-WER	No. of arcs	Real-time
Dev	14.7	38	11.7	6.2 k	1.8
Evl	14.8	108	11.5	24.5 k	3.2

TABLE XIII
List of abbreviations

ARPA	Advanced Research Projects Agency
CCCC	CSR Corpus Coordinating Committee
CSR	Continuous-Speech Recognition
DARPA	Defense Advanced Research Projects Agency
FFT	Fast Fourier Transform
HMM	Hidden Markov Model
LDA	Linear Discriminant Analysis
LM	Language Model
NAB	North American Business News
OOV	Out-Of-Vocabulary words
TI	Texas Instruments
WER	Word-Error Rate
WSJ	Wall Street Journal

testing. (2) The n-gram language model uses non-linear interpolation to cope with unseen events in the training corpora. (3) Time-synchronous beam search is applied to a tree-organized lexicon in connection with a phoneme look-ahead; in an optional subsequent step, a word graph or word lattice can be rescored using additional stochastic knowledge sources, namely a finer acoustic or language model.

REFERENCES

[1] H. Ney, Modeling and search in continuous-speech recognition, Proc. Europ. Conf. on Speech Communication and Technology, Berlin, 491–500 (Sep. 1993).
[2] H. Ney and X. Aubert, A word graph algorithm for large vocabulary continuous speech recognition, Proc. ICSLP Int. Conf. on Spoken Language Processing, Yokohama, Japan, 1355–1358 (1994).
[3] F. Jelinek, Continuous speech recognition by statistical methods, Proc. of the IEEE, **64**(10), 532–556 (April 1976).

TABLE XIII
List of abbreviations

ARPA	Advanced Research Projects Agency
CCCC	CSR Corpus Coordinating Committee
CSR	Continuous-Speech Recognition
DARPA	Defense Advanced Research Projects Agency
FFT	Fast Fourier Transform
HMM	Hidden Markov Model
LDA	Linear Discriminant Analysis
LM	Language Model
NAB	North American Business News
OOV	Out-Of-Vocabulary words
TI	Texas Instruments
WER	Word-Error Rate
WSJ	Wall Street Journal

testing. (2) The *n*-gram language model uses non-linear interpolation to cope with unseen events in the training corpora. (3) Time-synchronous beam search is applied to a tree-organized lexicon in connection with a phoneme look-ahead; in an optional subsequent step, a word graph or word lattice can be rescored using additional stochastic knowledge sources, namely a finer acoustic or language model.

REFERENCES

[1] H. Ney, Modeling and search in continuous-speech recognition, Proc. Europ. Conf. on Speech Communication and Technology, Berlin, 491–500 (Sep. 1993).
[2] H. Ney and X. Aubert, A word graph algorithm for large vocabulary continuous speech recognition, Proc. ICSLP Int. Conf. on Spoken Language Processing, Yokohama, Japan, 1355–1358 (1994).
[3] F. Jelinek, Continuous speech recognition by statistical methods, Proc. of the IEEE, **64**(10), 532–556 (April 1976).

[4] F. Jelinek, R.L. Mercer and S. Roukos, Principles of lexical language modelling for speech recognition, in Advances in Speech Signal Processing, S. Furui and M.M. Sondhi (eds.), pp. 651–699, Marcel Dekker, New York (1992).
[5] S. Dobler, D. Geller, R. Haeb-Umbach, P. Meyer, H. Ney and H.-W. Ruehl, Design and use of speech recognition algorithms for a mobile radio telephone, Speech Communication, **12**, 221–229 (1993).
[6] H.-W. Ruehl, S. Dobler, J. Weith, P. Meyer, A. Noll, H.-H. Hamer and H. Piotrowski, Speech recognition in the noisy car environment, Speech Communication, **10**, 11–22 (1991).
[7] R. Haeb-Umbach, D. Geller and H. Ney, Improvements in connected digit recognition using linear discriminant analysis and mixture densities, ICASSP (Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing), Minneapolis, MN, **11**, 239–242 (April 1993).
[8] R. Haeb-Umbach, P. Beyerlein and D. Geller, Speech recognition algorithms for voice control interfaces, Philips J. Res., **49**(4), 381–397 (1995).
[9] H. Aust, M. Oerder, F. Seide and V. Steinbiss, A spoken language inquiry system for automatic train timetable information, Philips J. Res. (1995), **49**(4), 399–418 (1995).
[10] M. Oerder and H. Aust, A realtime prototype of an automatic inquiry system, Proc. ICSLP'94 Int. Conf. on Spoken Language Processing, Yokohama, 703–706 (1994).
[11] H. Aust, M. Oerder and F. Seide, Experience with the Philips automatic train timetable information system, Proc. IVTTA'94 Second IEEE Workshop on Interactive Voice Technology for Telecommunications Applications, Kyoto, 67–72 (1994).
[12] H. Aust, M. Oerder and V. Steinbiss, Database query generation from spoken sentences, Proc. IVTTA'94 Second IEEE Workshop on Interactive Voice Technology for Telecommunications Applications, Kyoto, 141–144 (1994).
[13] H. Ney, Acoustic modelling of phoneme units for continuous speech recognition, Proc. EUSIPCO-90 Fifth Europ. Signal Processing Conf., Barcelona, 65–72 (Sept. 1990).
[14] H. Ney, V. Steinbiss, R. Haeb-Umbach, B.-H. Tran and U. Essen, An overview of the Philips research system for large-vocabulary continuous-speech recognition, Int. J. Pattern Recognition and Artificial Intelligence, **8**(1), 33–70 (1994).
[15] X. Aubert, R. Haeb-Umbach and H. Ney, Continuous mixture densities and linear discriminant analysis for improved context-dependent acoustic models, ICASSP, Minneapolis, MN, **11**, 648–651 (April 1993).
[16] X. Aubert, C. Dugast, H. Ney and V. Steinbiss, Large vocabulary continuous speech recognition of Wall Street Journal data, ICASSP'94 (Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing), Adelaide, **11**, 129–132 (1994).
[17] F. Kubala and CCCC, The hub and spoke paradigm for CSR evaluation, Proceedings of the ARPA Human Language Technology Workshop, Morgan Kaufmann (March 1994).
[18] D. Pallett, J. Fiscus, W. Fisher, J. Garofolo, B. Lund and M. Pryzbocki, 1993 benchmark tests for the ARPA spoken language program, in Proceedings of the ARPA Human Language Technology Workshop, Morgan Kaufmann (March 1994).
[19] R.O. Duda and P.E. Hart, Pattern Classification and Scene Analysis, Wiley, New York (1973).
[20] M.J. Hunt and C. Lefebvre, A comparison of several acoustic representations for speech recognition with degraded and undegraded speech, ICASSP, Glasgow, 262–265 (May 1989).
[21] R. Haeb-Umbach and H. Ney, Linear discriminant analysis for improved large vocabulary continuous speech recognition, ICASSP, San Francisco, CA, **1**, 13–16 (March 1992).
[22] J.K. Baker, Stochastic modelling for automatic speech understanding, in Speech Recognition, D.R. Reddy (ed.), pp. 512–542, Academic Press, New York (1975).
[23] S.E. Levinson, L.R. Rabiner and M.M. Sondhi, An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition, The Bell System Technical J., **62**(4), 1035–1074 (April 1983).
[24] C. Dugast, P. Beyerlein and R. Haeb-Umbach, Application of clustering techniques to mixture density modelling for continuous-speech recognition, ICASSP'95, Detroit, **1**, 524–527 (1995).
[25] M. Hwang and X. Huang, Shared-distribution hidden Markov models for speech recognition, Trans. on Speech and Audio Processing, **1**(4), 414–420 (Oct. 1993).
[26] S.J. Young and P.C. Woodland, The use of state-tying in continuous speech recognition, in Proc. EUROSPEECH '93, Vol. 3, pp. 2203–2206, Berlin, Germany (Sept. 1993).
[27] R. Kneser and H. Ney, Improved clustering techniques for class-based statistical language

modelling, Proc. Europ. Conf. on Speech Communication and Technology, Berlin, 973–976 (Sep. 1993).
[28] H. Ney, U. Essen and R. Kneser, On structuring probabalistic dependencies in stochastic language modelling, Computer Speech and Language, **8**, 1–38 (1994).
[29] I.J. Good, The population frequencies of species and the estimation of population parameters, Biometrika, **40**, 237–264 (Dec. 1953).
[30] H.-G. Meier and H. Ney, Leaving *m* samples out: generalizing the Turing–Good formula, Internal Paper (1994).
[31] H. Ney and U. Essen, On smoothing techniques for bigram-based natural language modelling, ICASSP, Toronto, 825–828 (May 1991).
[32] H. Ney, D. Mergel, A. Noll and A. Paeseler, Data driven organization of the dynamic programming beam search for continuous speech recognition, IEEE Trans. on Signal Processing, **SP-40**(2), 272–281 (Feb. 1992).
[33] H. Ney, R. Haeb-Umbach, B.-H. Tran and M. Oerder, Improvements in beam search for 10000-word continuous speech recognition, ICASSP, San Francisco, CA, **I**, 9–12 (March 1992).
[34] C. Dugast, Large-vocabulary recognition, Philips J. Res. **49**(4), 353–366 (1995).
[35] R. Haeb-Umbach and H. Ney, A look-ahead search technique for large-vocabulary continuous-speech recognition, Proc. Europ. Conf. on Speech Communication and Technology, Genova, pp. 495–498 (Sep. 1991).
[36] V. Steinbiss, B.-H. Tran and H. Ney, Improvements in beam search, Proc. ICSLP Int. Conf. on Spoken Language Processing 1994, Yokohama, Japan, 2143–2146 (1994).
[37] H. Murveit, J. Butzberger, V. Digalakis and M. Weintraub, Large-vocabulary dictation using SRI's decipher speech recognition system: progressive search techniques, Proc. ICASSP'93, Minneapolis, MN, USA, **II**, 319–322 (1993).
[38] M. Oerder and H. Ney, Word graphs: an efficient interface between continuous-speech recognition and language understanding, Proc. ICASSP'93, Minneapolis, MN, 119–122 (1993).
[39] J.L. Gauvain, L.F. Lamel, G. Adda and M. Adda-Decker, The LIMSI continuous speech dictation system: evaluation on the ARPA Wall Street Journal task, ICASSP'94, Adelaide, Australia, **1**, 557–560 (1994).
[40] R. Schwartz and S. Austin, A comparison of several approximate algorithms for finding multiple (N-BEST) sentence hypotheses, Proc. ICASSP'91, Toronto, Canada, 701–704 (1991).
[41] H. Ney, Search strategies for large-vocabulary continuous-speech recognition, Proc. of NATO Advanced Study Institute on Speech Recognition and Understanding, Bubion, Spain (1993).
[42] X. Aubert and H. Ney, Large vocabulary continuous speech recognition using word graphs, ICASSP'95, Detroit, **1**, 49–52 (1995).
[43] E.P. Giachin, A.E. Rosenberg and C.-H. Lee, Word juncture modelling using phonological rules for HMM-based continuous speech recognition, Computer Speech and Language, **5**, 155–168 (1991).
[44] X. Aubert and C. Dugast, Improved acoustic–phonetic modelling in the Philips dictation system, Proc. EUROSPEECH, Madrid (1995) submitted.
[45] S. Besling, Heuristical and statistical methods for grapheme-to-phoneme conversion, Proc. KONVENS, Springer, Vienna (1994).
[46] D. Paul and J. Baker, The design for the Wall Street Journal-based CSR corpus, in DARPA Speech and Language Workshop, Morgan Kaufmann, San Mateo, CA (1992).