A Continuous Speech Recognition System Using Phonotactic Constraints

Bernd Plannerer
Günther Ruske

Technische Universität München

Vm **Report 123**
Mai 1996

Mai 1996

Bernd Plannerer
Günther Ruske

Forschungsgruppe Sprachverarbeitung
Lehrstuhl für Mensch-Maschine-Kommunikation
Technische Universität München
Arcisstr.21
80290 München

Tel.: (089) 2892 - 8563
e-mail: ruske@e-technik.tu-muenchen.de

Gehört zum Antragsabschnitt: TP3 Spracherkennung und Sprecheradaption

Das diesem Bericht zugrundeliegende Forschungsvorhaben wurde mit Mitteln des Bundesministeriums für Forschung und Technologie unter dem Förderkennzeichen 01 IV 102 C/6 gefördert. Die Verantwortung für den Inhalt dieser Arbeit liegt bei den Autoren.

A CONTINUOUS SPEECH RECOGNITION SYSTEM USING PHONOTACTIC CONSTRAINTS

B. Plannerer and G. Ruske

Technische Universität München
Lehrstuhl für Datenverarbeitung, Franz-Joseph-Str. 38,
D-80801 München, Germany

ABSTRACT

This paper describes a speaker-independent recognition system for continuous German speech based on semicontinuous Hidden-Markov-Models which produces a phonetic transcription of the spoken sentence. The recognition units are parts of syllables while the output is a phoneme level transcription. During recognition, the phonotactic constraints of German are taken into account by a micro syntax constrained Viterbi algorithm. A maximum likelihood training procedure based on Viterbi training together with a simple but efficient seed model generation algorithm is presented.

Keywords:
phonotactic constraints, semicontinuous HMMs, seed model generation, Viterbi training.

INTRODUCTION

Automatic recognition of large vocabularies and continuous speech necessitates introduction of some kind of subword units. One question is how to define suitable phonetically based units which usually are chosen to be phonemes. Due to coarticulation effects, phoneme-sized units have to be represented together with their phonetic context and are often realized as context-dependent units. Alternatively, the syllabic structure of speech favouarbly can be utilized since the number of possible consonant combinations between vowels is reduced drastically. These phonotactic constraints are completely independent from the actual vocabulary, they only reflect the type of language. If we use **parts of syllables** as recognition units, the phonotactic rules are fulfilled without fixing the vocabulary. In this way it is possible to test the acoustic-phonetic recognition accuracy itself, independent from the vocabulary. This enables comparison of different recognition methods or modelling approaches independently from the application. It is typical, that recognition now works in a bottom-up manner which only takes into account the phonotactic constraints. In a second stage, words have to be estimated from the recognized units. Of course, in real applications, word recognition usually is performed top-down by composing the words from the recognition units and matching the entire word pattern with the unknown input. Now all phonotactic constraints are contained in the word patterns itself. Here, integrated search techniques offer an economic solution to the overall recognition problem.
The paper deals with the phonotactic constraints of syllables which are represented by a so-called "**micro syntax**". This syntax has to be followed during application of Hidden Markov Models for recognizing the units. Now recognition rates can be evaluated which would not be possible if a fixed vocabulary would have been used.

SUBWORD UNITS

The use of syllable-based subword units has proven to be a successful approach in speech recognition.
The German language contains only about 50 initial consonant clusters (ICC), but up to 160 final consonant clusters [1,2,3]. A substantial reduction in this number is achieved by dividing the final clusters into a so-called "rudiment" (RUD) and a subsequent "suffix" (SUF), whereby suffixes contain only fricatives and plosives or combinations of both. The suffixes are achieved by cutting that part within a final consonant cluster which begins with one of the consonants /s, t, f, or ʃ/. This is possible because after suffixes no other consonants can follow. Since these 4 consonants have similar places of articulation, they can be appended to all rudiments in the same way. Now the 160 final consonant clusters are composed of 23 rudiments and 17 suffixes. The syllabic units can appear only in a fixed order which is described by a phonotactic **micro syntax** and which reduces the number of possible combinations drastically; this micro syntax has strictly to be kept during the concatenation of the Hidden Markov Models when using the Viterbi algorithm for recognition. By introducing empty initial consonant clusters, vowels in syllable initial position can be represented, too. Empty rudiments and suffixes are represented by skipping arcs. Further, there are about 20

Presented at the *Proc. EUROSPEECH-93*, Berlin, September 1993, pp. 859 - 862.

vowels in the German language, inclusive 3 diphthongs. Since consecutive vowels also may be coarticulated very strongly, they should be represented together by a common unit containing the vowel pair or a pair of vowel and diphthong. Correspondingly, these units are called "vowel clusters" (VOW). The number of vowel clusters theoretically may sum up to about 130 units; however, in applications with 1000 words only about 30 - 50 different vowel clusters are really necessary. Some problems may arise if due to the elision of a schwa-sound /∂/ some consonants from the syllable initial and the final position may come together, the combinations of which are not contained in the consonant cluster inventories. Since the schwa-sound elision mostly appears in front of /l/, /m/, /n/ and /R/, this problem can be solved by defining these consonants additionally as "syllabic consonants" (SC) /l./, /m./, /n./ and /R./ and putting them into the inventory of the **vowels**.

For instance, in the German word "haben" (/h a: b n./) now the consonant /b/ again stands in syllable **initial** position in front of the syllabic /n./ and therefore belongs to the inventory of the initial consonant clusters. Introduction of the syllabic /R./ was not necessary since most of these cases were contained in the initial consonant clusters.The final "er" in German mostly is reduced to a /a/-like vowel /R/-schwa so that the syllabic /R./ can be avoided. With these agreements, now the class VOW contains vowels, diphthongs, /R/-schwa, /m./, /n./, and /l./. By defining these inventories, recognition now always produces the fixed sequence

... ICC VOW RUD SUF ICC VOW RUD SUF ...

If a rudiment or suffix is not present, a special symbol is generated. Additionally, models for pauses are allowed to follow a rudiment or a suffix model.

During recognition, the Viterbi algorithm will find the best mapping of labels to the uttered sentence with respect to the additional constraints given by the micro syntax graph. The main advantage of this approach is that the resulting transcription will not show the well-known effects of the free running Viterbi algorithm, e.g. splitting of units into multiple repetitions of the same unit etc. These effects will not occur in our system since multiple repetitions of the same phonotactic class are prohibited by the syntax graph. Instead, the resulting transcription will always be a valid sequence of phonotactic units.

MICRO SYNTAX

Localization of the syllable nuclei can be performed explicitly by means of a segmentation algorithm or is achieved implicitly during the Viterbi recognition by means of the VOW-models. Omitting a vowel may produce severe errors in such cases where the word model demands a vowel at that position. This problem can be handled to a certain extent by alternative pronunciations with and without that vowel stored in the word pronunciation lexicon. However, a schwa-elision cannot be recognized explicitly since this sound has not been spoken at all. This task now is carried out by the models for syllabic consonants (SC) which are treated in the same way as models for ICC, RUD and SUF. Again these units can appear only in a distinct order which altogether is represented by the synatx graph depicted in Fig. 1 and which contains all paths for the Viterbi recognition. The main coarse of the syntax graph in Fig. 1 (bold arcs) represents the "normal" sequence in a loop from ICC to SUF and back again to ICC etc. Besides that, the path with SC indicates a syllabic consonant. In this case a reduced syllable nucleus has been detected; the Viterbi algorithm decides between both choices. In most cases a syllable consonant is followed only by a suffix, as e.g. in the word "zweifelt" (/tsv ai f l. t/). If also rudiments should be allowed (which are few cases), the syntax graph has to be extended accordingly. At the beginning or at the end of a sentence the graph gets rather simple; the graph has to to be entered with an ICC and has to be left at the syllable boundary which is located in front of an ICC.

As shown in Fig.1, there are copies of the same phonotactic classes (and therefore, of the items in that class) needed to treat the different predecessors. Of course, these copies somewhat increase the necessary amount of memory and computation.

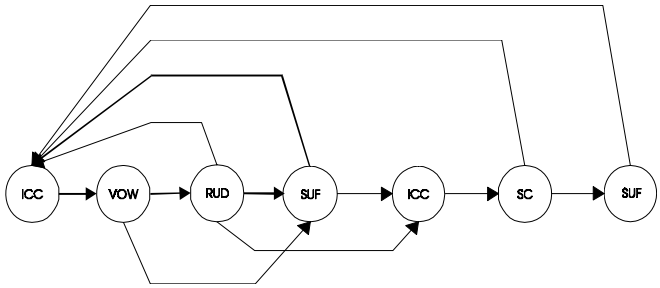

Fig. 1: micro syntax

MODELING OF SUBWORD UNITS

As mentioned above, the subword units are modeled by semicontinuous Hidden Markov Models, which have shown to be a robust modeling approach.

As is well known, the probability density of a given observation vector $\vec{x}_t$ in state i of the semicontinuous HMM is given by

$$p_i(\vec{x}_t) = \sum_{k=1}^{CBE} C_{ik} \cdot p(\vec{x}_t | \varphi_k)$$

where CBE is the number of codebook entries and φ_k denotes the codebook distribution number k.

The computation of initial model parameters is based on a modified version of our algorithm described in [4]: In that paper, we presented a dynamic segmentation procedure for giving a first estimate of the model parameters. The basic principle of that procedure is to measure the stationarity of the training pattern by computing the Euclidean distance between every two subsequent feature vectors: a small distance indicates stationary parts of the pattern, while a

large distance indicates instationarity. Using the accumulated distances in a given state, a simple threshold decision was made to proceed to the next state of the model. A drawback of this method is, that the feature vectors must be all available for the computation of the euclidean distance. As far as continuous HMMs are used, the feature vectors are always available during the training procedure. On semicontinuous HMMs however, one might want to perform the vector quantization first an then deal with the corresponding codeword sequence only. Therefore, a different distance measure has to be used: If the top-1-codeword number remains the same for a sequence of frames, a stationary segment can be assumed. Therefore, a simple measure of stationarity can be derived by observing the rank of the top-1-codeword between subsequent frames: Given the top-1-codeword at a given time frame t, its rank at the next time slot t+1 is used as a distance measure, e.g, if the codeword has fallen to the top-3 position, a distance of 3 between the two feature vectors is assumed. This distance measure allows directly the use of the state segmentation procedure as described in [4]. Given this initial state segmentation, the transition probabilities and mixture coefficients can be estimated using the reestimation formulae of the Viterbi training procedure.

For estimation of the mixture coefficients, the Viterbi training procedure with a-posteriori weighting of observations is used as shortly described below.

A product codebook of three diagonal-covariance codebooks is applied for the features (log-)spectra, delta-spectra and energy. The number of codebook-entries is 256, 128 and 64, respectively. The log-scores of these three codebooks are added after multiplication with an empirically determined weighting factor. In addition to these modeling parameters, the total duration of the subword unit is represented by a single gaussian distribution.

HMM TRAINING

For parameter estimation, we use the Viterbi training procedure as described in [4]. Let $p(\vec{X}|\lambda)$ be the joint probability density of the concatenated training set observations $\vec{X} = \left\{\vec{X}^{(1)}, \vec{X}^{(2)}, \ldots, \vec{X}^{(H)}\right\}$ of all HMMs and let $\lambda = \left\{\lambda^{(1)}, \lambda^{(2)}, \ldots, \lambda^{(H)}, \lambda^{(CB)}\right\}$ be the complete parameter set of all H HMMs and the codebook CB, then optimization of the joint probability yields the following reestimation formula for the mixture coefficients

$$C_{ik}{}^{*} = \frac{1}{M_i} \cdot \sum_{m=1}^{M_i} P_i\left(\varphi_k \middle| \vec{x}_m\right)$$

where M_i denotes the number of observation vectors associated with state i by the Viterbi algorithm. Note that every observation vector $\vec{x}_m$ is splitted across the mixture coefficients according to its a-posteriori probability $P_i\left(\varphi_k \middle| \vec{x}_m\right)$ instead of simply incrementing the counter for the mixture corresponding to the top-1 codeword.

Update of the codebook parameters can be performed by the following reestimation formula:

$$\vec{\mu}_k^{\,*} = \frac{\sum_{h,i,m} P_i^{(h)}\left(\varphi_k \middle| \vec{x}_m\right) \cdot \vec{x}_m}{\sum_{h,i,m} P_i^{(h)}\left(\varphi_k \middle| \vec{x}_m\right)}$$

$$R_k^{*} = \frac{\sum_{h,i,m} P_i^{(h)}\left(\varphi_k \middle| \vec{x}_m\right) \cdot \left(\left(\vec{x}_m - \vec{\mu}_k\right) \cdot \left(\vec{x}_m - \vec{\mu}_k\right)^T\right)}{\sum_{h,i,m} P_i^{(h)}\left(\varphi_k \middle| \vec{x}_m\right)}$$

where $\vec{\mu}_k$ and R_k denote the mean vector and the covariance matrix of codebook distribution number k, $\vec{x}_m$ the observation vector number m, and $P_i^{(h)}\left(\varphi_k \middle| \vec{x}_m\right)$ again represents the probability that observation vector $\vec{x}_m$ belongs to the codebook distribution φ_k when observed in state i of HMM number h.

The above formulae show that the a-priori probability of a given HMM h in the training data will have an important impact on its representation in the updated codebook. That is, a subword unit having a high occurrency in the training data will be well represented in the updated codebook parameters, while a subword unit which has a low occurence in the training data will have hardly any impact on the reestimation process. This effect may be of importance if the distribution of subword unit occurrencies in the training set significantly differs from the distribution of the test set.

IMPLEMENTATION

In our implementation, only the top-K codewords were used for training and testing to reduce the necessary amount of computation. The actual number of codewords used is determined every frame by a threshold decision: If a codeword score falls below a given threshold relatively to the score of the top-1 codeword, it is pruned. Thus, the actual number of codewords may vary from frame to frame and is not fixed as implied by the reestimation formulae.

For our tests, a full Viterbi search is performed without any pruning of hypotheses. This is possible since there are only a few node copies required according to the syntax graph given in Fig. 1.

In order to compute a lattice of subword units, we first implemented an exact N-best search algorithm to get the N-best transcription hypotheses, but as often observed with N-best search algorithms [5], our experiments showed that most of the hypotheses varied in the most ambiguous

vowels of the utterance and thus N had to be chosen very large to yield useful alternatives.
For this reason, we decided to use the top-1 segmentation of the utterance computed by the Viterbi algorithm and to compute the N-best labels within this given segmentation. All these labels were now chosen from the phonotactic class determined by the top-1 segmentation. However, the main drawback of using the top-1 segmentation is that the alternative subword units are constrained to the phonotactic class given by this segmentation. For example, if a rudiment /p/ was found by the top-1 decision, there can never be an alternative recognition of a /t/, since /t/ belongs to the suffixes.

RECOGNITION EXPERIMENTS

This section presents some of the recognition experiments made with our subword unit models.
Direct evaluation on the phonetic string is somewhat difficult since there are cases in which more than one correct segmentation into subword units is possible.

Therefore, to evaluate the model performance, we decided to use automatically segmented and labeled data for testing. Thus, the HMMs were run within a given segmentation to avoid the problems stated above.
The training database consisted of a total of 503 sentences spoken by 5 male speakers, while the test database contained 250 sentences spoken by 5 different male speakers which are part of the German PHONDAT train time table information database. In this experiment, the standard Viterbi training procedure without codebook update was used for parameter estimation.
In Tab. 1, all experiments are given for top-1 decision and the top-N decision with N = 4. Note that recognition results for the top-4 decision are significantly higher. This implies that in most cases there is only a small difference in score between the correct subword unit and the wrong classified top-1 subword unit.
In all recognition experiments, a "hard decision" was made for error counting, that is, a subword unit is counted as an error if it does not match the given labeling even if some of its phonemes were correctly classified. It should be mentioned that the test sentences were spoken at a significantly higher speed (more than 10 % faster) than the sentences of the training set and therefore the test was carried out under relatively hard conditions.

	ICC	VOW	RUD	SUF
top-1	53 %	49 %	69 %	86 %
top-4	78 %	78 %	88 %	94 %

Tab. 1.: correctly recognized subword units

DISCUSSION

The syntax graph given in Fig. 1 shows that the resulting transcription can never contain such effects as multiple insertions of identical subword units. For example, the VOW class can never be followed by itself directly. However, this implies that every possible vowel clusters of the application under concern has to be present in the inventory of vowels and therefore the number of subword units needed can be relatively high which will require a large amount of training data. In addition, as mentioned above, the subword units are divided into 5 phonotactic classes. This implies that subword units that consist of identical phonemes or phoneme clusters will be treated as different subword units depending on their phonotactic position. For example, the phomeme sequence /ʃt/ belongs to the ICC class as well as to the SUF class. Therefore, the training data available for each model is further reduced. As a consequence, some of the larger subword units cannot be estimated due to the limited training data, even if our "smooth" training technique with a-posteriori probability weighting is used. In these cases, the subword unit will have to be represented by a concatenation of smaller subword units.

CONCLUSION

We presented a simple method of utilizing phonotactic constraints for improvement of automatic recognition / transcription of continuous speech. These constraints can easily be implemented with only a modest increase of computational complexity or memory requirements, but a large amount of training data is required or otherwise the subword units have to be concatenated by smaller units.

This work has been partly carried out within the **ASL/VERBMOBIL** project which is sponsored by the German **BMFT**.

REFERENCES

[1] G. Ruske, B. Plannerer and T. Schultz, Stochastic modeling of syllable-based units for continuous speech recognition, Proc. of ICSLP-92, Oct. 1992, Vol. 2, 1503-1506.
[2] F. Schiel and F. Wolfertstetter, Regelbasierte Erzeugung von robusten Aussprachemodellen und deren Darstellung im Silbenraster, 2. Konferenz "Elektronische Sprachverarbeitung" der TU Dresden, 28.-30.10.1991, Studientexte zur Sprachkommunikation ISSN 0940-6832 Heft 8, 173-182.
[3] R. Seck and G. Ruske, Structure of German syllable initial and final consonant clusters based on articulatory features, Speech Communication 5, 1986, 347-354.
[4] B. Plannerer and G. Ruske, Recognition of demisyllable based units using semicontinuous hidden Markov models, Proc. of ICASSP-92, March 1992, I-581 - I-584.
[5] V. Steinbiss, A search organization for large vocabulary Recognition based on N-best decoding, Proc. of EUROSPEECH-91, September 1991, Vol. 3, 1217 - 1220.