A PHONE RECOGNIZER HELPS TO RECOGNIZE WORDS BETTER

Georg Stemmer, Viktor Zeissler, Christian Hacker, Elmar Nöth, Heinrich Niemann

Universität Erlangen–Nürnberg
Lehrstuhl für Mustererkennung (Informatik 5)
Martensstr. 3, 91058 Erlangen, Germany
stemmer@informatik.uni-erlangen.de

ABSTRACT

For most speech recognition systems dynamic features are the only way to incorporate temporal context into the output distributions of the HMMs. In this paper we propose an efficient method to utilize a large context in the recognition process. State scores of a phone recognizer which runs in parallel to the word recognizer are computed. Integrating these scores in the HMMs of the word recognizer makes their output densities context-dependent. The approach is evaluated on a set of spontaneous utterances which have been recorded with our spoken dialogue system. A significant reduction of the word error rate has been achieved.

1. INTRODUCTION

It is a common method in pattern recognition to improve the performance of a classifier by the incorporation of (more) context. A well-known weakness in HMMs is that the feature vectors are dependent only on the states which generated them, not on the neighboring feature vectors. Context is only represented by the dynamic features, e.g. delta coefficients of the Mel-frequency cepstral coefficients. However, most types of dynamic features are only limited to a few subsequent feature vectors and do not represent long-term variations. The main objective of the paper is to introduce a new way to utilize context in the output distribution of HMMs.

In the following we will describe how context can be incorporated into HMMs by simply taking into account a phone recognizer, which runs in parallel to the word recognizer. The state scores of the phone recognizer are computed with the beam search algorithm. They depend on all feature vectors that have been observed so far; the fact that the HMMs of the phone recognizer are based on the Markov assumption is not relevant. This makes the state scores of the phone recognizer a valuable additional information source for each state of the word recognizer. The state scores do not replace the Gaussian densities, but are used as an additional information source to increase the discrimination ability between sub-phonetic speech events.

The most successful ways to enhance the use of context in HMMs that can be found in literature are based on improvements of the extraction of temporal features [1], but this is beyond the scope of this paper. A number of studies to overcome the so-called conditional independence assumption of HMMs based on an improvement of the *model* are described in [2]. The concept of segment models is also related to this topic, please refer to [3] for an overview. Most of the approaches perform direct modeling of segments of speech frames, others assume that the output distribution of the HMM does not only depend on the current state but also on one or several previous frames [2]. A major disadvantage of most of these methods is that the parameter space increases dramatically, even if only one neighboring feature vector is considered.

The number of free parameters can be reduced by representing the context with a discrete random variable ([4], p. 409). This is similar to the approach described in this paper, as the context is also represented by a single discrete random variable. However, the context is not limited to a few feature vectors and the computation scheme for the output distribution has much less free parameters. Another major advantage of the approach introduced below is that the algorithms for training and decoding are not changed, so there is no increase in the complexity of the computation.

2. MATHEMATICAL FORMALISM

2.1. Output Density

In a standard (semi-)continuous HMM the density function $b_i(\mathbf{x}_t)$ for the output of a feature vector $\mathbf{x}_t$ by the state i at time t is computed by a sum over all codebook

A part of this work was funded by the German Federal Ministry of Education, Science, Research and Technology (*BMBF*) in the framework of the SmartKom project under Grant 01 IL 905 K7. The responsibility for the content lies with the authors.

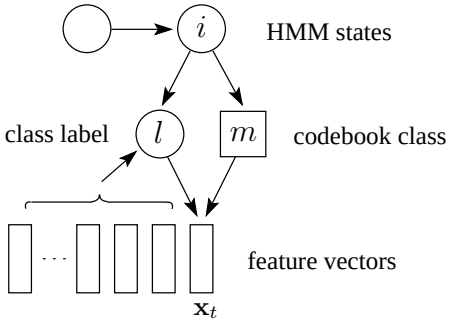

Fig. 1. Output of the feature vector $\mathbf{x}_t$ by the HMM state i. The arrows symbolize statistical dependencies between random variables, not state transitions.

classes $m \in M$:

$$b_i(\mathbf{x}_t) = \sum_m c_{i,m} \cdot p(\mathbf{x}_t|m,i) \approx \sum_m c_{i,m} \cdot p(\mathbf{x}_t|m) \quad (1)$$

The probability for a certain codebook class m, given a state i is represented by $c_{i,m}$. The second part in Eq. 1 corresponds to the transition from continuous to semi-continuous HMMs. A Gaussian pdf $\mathcal{N}(\mathbf{x}_t|\mu_m, \mathbf{\Sigma}_m)$ is typically used to represent $p(\mathbf{x}_t|m)$.

In the rest of this paper we will consider different types of probability density functions which make it possible to integrate a large context $\mathbf{x}_1^{t-1}$ into the HMM output density. $\mathbf{x}_1^{t-1}$ stands for the context $\mathbf{x}_1, .., \mathbf{x}_{t-1}$ of feature vectors which have been observed so far. If we try to integrate the context $\mathbf{x}_1^{t-1}$ directly into b_i this results in a large amount of additional computational effort.

Therefore we introduce a new hidden random variable l, which we call the class label. Each of the class labels $l \in L$ may correspond to a phone symbol, for instance. From now on each state i does not only choose between the codebook classes $m \in M$, but at the same time also takes an independent decision for the class label l. The class label l itself is a discrete representation of the complete history of feature vectors $\mathbf{x}_1^{t-1}$. The integration of l into the output density makes b_i dependent on the history $\mathbf{x}_1^{t-1}$. Unlike the approaches which have been mentioned in the literature review, we do not entirely abandon the conditional independence assumption of HMMs: the new model still assumes that $\mathbf{x}_t$ is independent from the history $\mathbf{x}_1^{t-1}$ when l and m are known. The process of feature vector generation according to the new model is illustrated in Fig. 1. The probability term $b_i(\mathbf{x}_t|\mathbf{x}_1^{t-1})$ has to be expanded as follows:

$$b_i(\mathbf{x}_t|\mathbf{x}_1^{t-1}) = \sum_{l,m} p(\mathbf{x}_t|l,m,i) \cdot P(l,m|i,\mathbf{x}_1^{t-1}) \quad (2)$$

As $\mathbf{x}_1^{t-1}$ is the same for all states i at time t, there is no increase in the computational complexity of the algorithms for training and decoding.

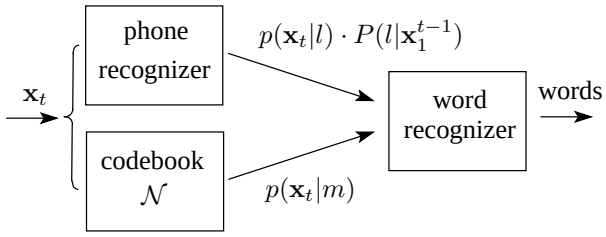

Fig. 2. A phone recognizer as an information source for the word recognizer.

2.2. Simplifying Assumptions

The representation of $b_i(\mathbf{x}_t|\mathbf{x}_1^{t-1})$ requires the estimation of too many parameters if we do not make additional simplifications. Firstly we can use the following approximation since the decisions for m and l are independent and m does not depend on $\mathbf{x}_1^{t-1}$:

$$P(l,m|i,\mathbf{x}_1^{t-1}) = c_{i,m} \cdot P(l|i,\mathbf{x}_1^{t-1}) \quad (3)$$

Secondly we can split $P(l|i,\mathbf{x}_1^{t-1})$ into two parts under the assumption that i is independent from $\mathbf{x}_1^{t-1}$:

$$P(l|i,\mathbf{x}_1^{t-1}) \propto P(l|i) \cdot P(l|\mathbf{x}_1^{t-1}) \quad (4)$$

$P(l|i)$ is estimated during the Baum-Welch training, while the computation of $P(l|\mathbf{x}_1^{t-1})$ is different for each type of class labels that are employed. Finally, we can compute the output density values of the models separately as m does not depend on l:

$$p(\mathbf{x}_t|l,m,i) \propto p(\mathbf{x}_t|m) \cdot p(\mathbf{x}_t|l) \quad (5)$$

To summarize, $b_i(\mathbf{x}_t|\mathbf{x}_1^{t-1})$ is computed by

$$b_i(\mathbf{x}_t|\mathbf{x}_1^{t-1}) \approx \left[\sum_l P(l|i) \cdot P(l|\mathbf{x}_1^{t-1}) \cdot p(\mathbf{x}_t|l)\right]^w \cdot \left[\sum_m c_{i,m} \cdot p(\mathbf{x}_t|m)\right]^{1-w} \quad (6)$$

The weighting factor w is introduced to control the influence of the different knowledge sources on $b_i(\mathbf{x}_t|\mathbf{x}_1^{t-1})$. We will optimize w on the development test set.

3. INTEGRATION OF THE PHONE RECOGNIZER

3.1. Combination of Information Sources

The improvements which may be achieved with our approach depend to a large part on the specific choice of the class labels l and the corresponding density functions $p(\mathbf{x}_t|l) \cdot P(l|\mathbf{x}_1^{t-1})$. As we decided to use a phone recognizer as information source, the labels l represent states of

phone models. The density value can be computed from the probability that the current state s_t of a phone HMM is equal to l:

$$p(\mathbf{x}_t|l) \cdot P(l|\mathbf{x}_1^{t-1}) := p(\mathbf{x}_t|l) \cdot P(s_t = l|\mathbf{x}_1^{t-1}) \quad (7)$$

where $P(s_t = l|\mathbf{x}_1^{t-1})$ is calculated from the forward score:

$$P(s_t = l|\mathbf{x}_1^{t-1}) = \frac{P(s_t = l, \mathbf{x}_1^{t-1})}{\sum_j P(s_{t-1} = j, \mathbf{x}_1^{t-1})} \quad (8)$$

The latter is approximated by the viterbi score of the state which is computed during beam search decoding in the phone recognizer. $p(\mathbf{x}_t|l)$ is the output density value of state l and is modeled as a mixture of Gaussian pdfs. The system architecture is illustrated in Fig. 2.

3.2. State Clustering

As the phone recognizer has about 300 different states (including the models for pauses, filled pauses and nonverbal sounds), we reduce the number of parameters by clustering similar states into groups. The use of state clusters for the class labels in contrast to individual states also increases robustness w.r.t. phone recognition errors.

A symmetric distance $D'(i,j)$ between two states i, j for semi-continuous HMMs can be computed from the Kullback-Leibler distance $D(i|j)$ of their output densities:

$$D(i|j) = \sum_m c_{i,m} \cdot \ln \frac{c_{i,m}}{c_{j,m}} \quad (9)$$

$$D'(i,j) = \frac{1}{2}D(i|j) + \frac{1}{2}D(j|i) \quad (10)$$

We apply the clustering algorithm from [5], p. 143: The size of a cluster C is defined as the maximum distance between any two states in C:

$$size(C) = \max_{i,j \in C} D'(i,j) \quad (11)$$

Initially each cluster contains exactly one state. The pair of clusters which when combined would form the smallest resultant cluster are merged. We repeat this step until the desired total number of clusters is reached.

In all experiments which are described below, the class label l stands not for a single HMM state but for a state cluster C_l. The probability of a specific label l is computed by averaging the scores of all states s_t which are in the same cluster C_l.

3.3. Reversed Phone Recognizer

We extend the context of the output density $b_i(\mathbf{x}_t|\mathbf{x}_1^{t-1})$ to $b_i(\mathbf{x}_t|\mathbf{x}_1^{t-1}, \mathbf{x}_{t+1}^T)$ with an additional discrete random variable r which takes the 'future' feature vectors $\mathbf{x}_{t+1}^T = \mathbf{x}_{t+1}, \mathbf{x}_{t+2}, .., \mathbf{x}_T$ into account. r is equivalent to l, but corresponds to the state clusters of a phone recognizer which runs from right to left on the time axis:

$$p(\mathbf{x}_t|r) \cdot P(r|\mathbf{x}_{t+1}^T) := p(\mathbf{x}_t|r) \cdot P(s_t = r|\mathbf{x}_{t+1}^T) \quad (12)$$

A second weighting factor v is introduced in order to integrate r into the computation of the output density $b_i(\mathbf{x}_t|\mathbf{x}_1^{t-1}, \mathbf{x}_{t+1}^T)$, which is also optimized on the development test set. The weights sum up to 1.

4. DATA

Acoustic models are trained on a part of the EVAR data set. It consists of 7438 utterances, which have been recorded by phone with our conversational train timetable information system. A detailed description of this system can be found in [6]. Nearly all utterances are in German language. The total amount of data is $\approx$ 8 hours. 4999 utterances have randomly been selected for training, the development test set contains 441 utterances. The rest of 1998 utterances is available for testing. The speakers of the training and the test sets are disjunct.

5. BASELINE SYSTEM

The system which has been used for the experiments is a speaker independent continuous speech recognizer. It is based on semi-continuous HMMs, the output densities of the HMMs are full-covariance Gaussians. Please refer to [7] for a detailed description of the speech recognizer. If the baseline system is only trained on the training data set described in the next section and no other data is used for training or initialization of the acoustic models, it achieves a word error rate of 26.0% on the test data.

6. EXPERIMENTAL RESULTS

6.1. Training and Weighting Factor Optimization

The training of the whole system is done in two steps: Firstly the phone recognizers which generate the class labels l, r are trained. As we do not have a phone-level annotation of the training data, we simply replace each word in the transcription by its canonic phone representation. The phone recognizer achieves a phone error rate of 43.9% on the test data. The reversed phone recognizer has a phone error rate of 44.5%.

Secondly we run the phone recognizer on the training data in order to compute the labels and the corresponding density values which are then used for the training of the word recognizer with the Baum-Welch algorithm.

The weighting factor w is set to 0.5 for the Baum-Welch training of the word recognizer. The optimal choice for the

clusters	5	10	15	20	25	30
WER[%]	25.3	25.0	24.7	24.6	24.7	25.1

Table 1. Comparison of the error rates when the total number of clusters is varied. Only the forward phone recognizer is used. w is set to 0.5 during training; for decoding w has been optimized on the development test set.

phone recognizer	WER [%]	relative improvement [%]
none (baseline)	26.0	-
forward	24.4	6.2
reversed	25.6	1.5
both	24.0	7.7

Table 2. Comparison of the different information sources and relative improvements over the baseline. The number of clusters is set to 20, all recognizers are trained with weights which have been optimized on the development test set.

value of the weighting factor w during decoding is determined on the development test set; w is varied in the range of 0.05–0.45 in steps of 0.05. We got slight improvements, when we additionally used the optimized weighting factor to re-train the word recognizer from scratch.

6.2. Results and Discussion

In a first experiment the total number of clusters is varied. Two separate clusters are manually assigned to all states of HMMs for pauses and nonverbal sounds, the rest of the states is clustered with the algorithm described above. As shown in Tab. 1, the system performance is significantly better than the baseline for 15-25 state clusters. The optimal weight factors w are in the range between 0.1 and 0.2. We use 20 clusters (plus two for the pauses and nonverbal sounds) for all following experiments. We compare the word error rates (WER) for the forward and the reversed phone recognizer to the baseline in Tab. 2. The WER for the forward phone recognizer is slightly better than in Tab. 1, because we used the optimized weight ($w = 0.2$) for the training of the word recognizer. The reversed phone recognizer alone does only yield to a very small decrease in WER. However, we were pleased to find that the relative improvements gained by forward and the reversed phone recognizer sum up when both information sources are combined. The best result corresponds to an improvement of 7.7% relative (2 percent points) over the baseline.

The fact that the accuracy of a phone recognizer is typically quite low does not seem to hurt too much. It is possible that we have prevented this by using scores for state clusters and not the final output of the phone recognizer. The scores contain an interpolated representation of all possible hypotheses and not only the single best phone sequence.

7. CONCLUSION AND OUTLOOK

For most speech recognition systems, the only way to incorporate temporal context in the output distributions of the HMMs is to use dynamic features. We have introduced a new method to integrate context into the recognition process of a word recognizer: we simply use the state scores of a phone recognizer which is run in parallel to the word recognizer. Our experiments show that the proposed approach reduces the word error rate significantly.

Future work includes further investigations into the application of different types of sub-word units for the acoustic preprocessor. However, our approach is not limited to the use of states or state clusters for the class labels. Therefore we plan to evaluate if the performance of the word recognizer can be improved by using (context dependent) classifiers for other types of labels, e.g. for noisy or voiced/unvoiced speech.

8. REFERENCES

[1] N. Morgan, "Temporal Signal Processing for ASR," in *Proceedings of the Automatic Speech Recognition and Understanding Workshop (ASRU'99)*, 1999.

[2] J. Ming and F. J. Smith, "Modelling of the Interframe Dependence in an HMM Using Conditional Gaussian Mixtures," *Computer Speech and Language*, vol. 10, no. 4, pp. 229–247, 1996.

[3] M. Ostendorf, V. Digalakis, and O. Kimball, "From HMMs to Segment Models: A Unified View of Stochastic Modeling for Speech Recognition," *IEEE Trans. on Speech and Audio Processing*, vol. 4, no. 5, pp. 360–378, 1996.

[4] X. Huang, A. Acero, and H.-W. Hon, *Spoken Language Processing*, Prentice Hall, Upper Saddle River, New Jersey, 2001.

[5] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK Book (for HTK Version 3.0)*, Microsoft Corporation, 2000.

[6] F. Gallwitz, M. Aretoulaki, M. Boros, J. Haas, S. Harbeck, R. Huber, H. Niemann, and E. Nöth, "The Erlangen Spoken Dialogue System EVAR: A State–of–the–Art Information Retrieval System," in *Proceedings of 1998 International Symposium on Spoken Dialogue (ISSD 98)*, 1998, pp. 19–26.

[7] F. Gallwitz, *Integrated Stochastic Models for Spontaneous Speech Recognition*, Ph.D. thesis, University of Erlangen-Nuremberg, 2001.