A HIERARCHICAL SYSTEM FOR WORD DISCOVERY EXPLOITING DTW-BASED INITIALIZATION

*Oliver Walter, Timo Korthals, Reinhold Haeb-Umbach**

Department of Communications Engineering
University of Paderborn
Germany
{walter,haeb}@nt.uni-paderborn.de

Bhiksha Raj

Language Technologies Institute
Carnegie Mellon University
5000 Forbes Avenue Pittsburgh, PA 15213
United States
{bhiksha}@cs.cmu.edu

ABSTRACT

Discovering the linguistic structure of a language solely from spoken input asks for two steps: phonetic and lexical discovery. The first is concerned with identifying the categorical subword unit inventory and relating it to the underlying acoustics, while the second aims at discovering words as repeated patterns of subword units. The hierarchical approach presented here accounts for classification errors in the first stage by modelling the pronunciation of a word in terms of subword units probabilistically: a hidden Markov model with discrete emission probabilities, emitting the observed subword unit sequences. We describe how the system can be learned in a completely unsupervised fashion from spoken input. To improve the initialization of the training of the word pronunciations, the output of a dynamic time warping based acoustic pattern discovery system is used, as it is able to discover similar temporal sequences in the input data. This improved initialization, using only weak supervision, has led to a 40% reduction in word error rate on a digit recognition task.

Index Terms— Unsupervised, word discovery, acoustic units

1. INTRODUCTION

Unsupervised language acquisition is the task of acquiring the building blocks of a language without any supervision. Techniques to discover these components directly from audio recordings of continuous speech are known under the term zero resource speech technologies and are currently an area of active research [1]. The problem may be subdivided in two tasks: the phonetic and lexical discovery. The first aims at discovering the phonetic building blocks of speech and building an acoustic model for each of them. Since the acoustic front ends used in standard (supervised) acoustic model training were not particularly successful in the unsupervised setting, researchers have come up with alternative acoustic representations and models, such as a large Gaussian Mixture Model as universal background model, from which the subword unit models are derived by clustering [2], or hidden Markov Model (HMM) based self-organizing units [3]. However, the speaker-dependence of the units remains an important concern [1]. To overcome this issue it has been proposed to use weak top-down constraints as can be provided by spoken term discovery techniques [4], which capture the similar temporal alternation of utterances of the same word even if spoken by different speakers.

*The work was in part supported by Deutsche Forschungsgemeinschaft under contract no. Ha 3455/9-1 within the Priority Program SPP1527 "Autonomous Learning" and by the NSF grant 1017256.

Errors in this phonetic discovery stage may have detrimental effects on the second stage, the lexical discovery, whose goal it is to discover words or phrases. Typically, the output of the phonetic discovery unit is a sequence of tokens and is thus of categorial nature. This sequence, however, will be noisy, and two utterances of the same word may result in different token sequences. Techniques for unsupervised word segementation usually assume an error-free input character or phone sequence [5] and break down in the presence of errors.

One way to overcome this is to abandon the two-stage approach and discover word-sized patterns directly from the input speech [6, 7]. While these techniques have found much success in spoken term discovery tasks, they lack the insight gained from the two-stage approach that is closely linked to the notion, that language has a deep structure: while the observed acoustic signal can be represented as draws from a finite set of atomic sound units (the phonemes), the sound units do not directly map to semantics. Only the sequence of sound units, which make up words, may be given a semantic interpretation. Finally, dynamic time warping (DTW) techniques are a non-parametric, template-matching approach and may not exhibit the power and flexibility of parametric statistical models, such as HMMs.

In this contribution we therefore adhere to a hierarchical, two-stage approach to reveal the building blocks of a language from spoken input. The approach is rooted in the method proposed in [8] for audio classification, however extended to better capture the sequential nature of speech. In the first stage acoustic subword units are discovered by segmenting the input speech, clustering the segments and training HMMs on the clusters. These units are meant to capture acoustically consistent phenomena and are called *acoustic unit descriptors* (AUD) in the following, as proposed in [9]. The AUD label sequence is the input to the "word" discovery stage. Here, repeating sequences of AUDs are modeled as HMMs whose emission probabilites are multinomial distributions over the AUDs. Unlike [8], where all states of a HMM shared the same emission probability, we use state-specific probabilities to capture the temporal struture of a word. It turns out that this significantly improves recognition performance. The output of this second stage is thus a pronunciation dictionary, whose entries are discrete HMMs. This probabilistic lexicon is able to absorb recognition errors in the AUD discovery stage.

An important issue is the initialization of the HMM training in the second stage. In the zero resource setting, both the pronunciation lexicon and the transcription are not available. As a substitute for the orthographic word transcription, we adopt the output of a DTW-based pattern discovery unit. This is in spirit of [4], who showed in

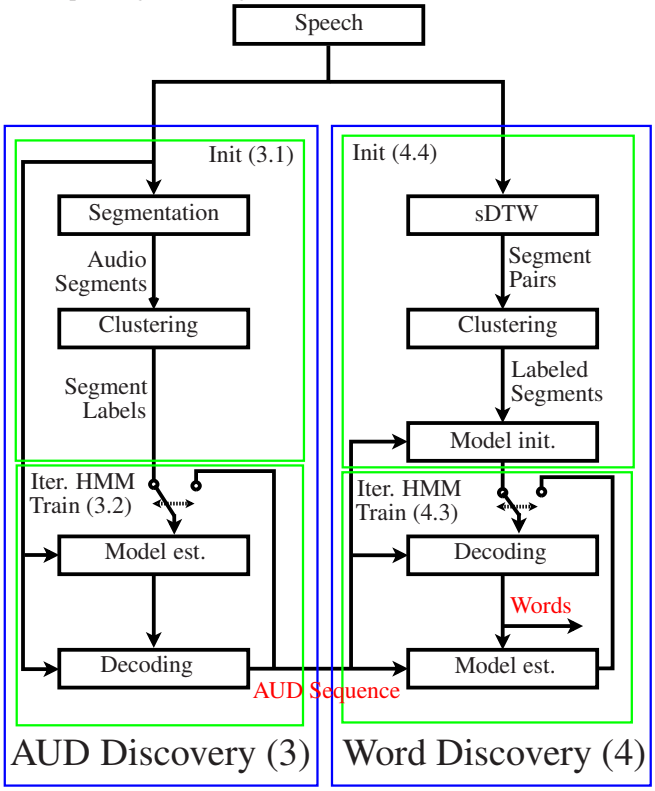
a proof-of-concept study that the weak top-down constraints gathered from a (perfect) DTW-output are helpful in finding subword unit models that are more speaker-independent. Here, we run DTW to find similar segment pairs in the audio data, cluster them and employ the cluster labels to obtain an initial partial transcription for the HMM training.

This paper is organized as follows: In the next section an overview of the hierarchical system for word discovery is presented, followed by a description of the training of the individual building blocks, the acoustic unit discovery in Section 3 and the word discovery in Section 4. Section 5 presents experimental results demonstrating the effectiveness of temporal constraints and the DTW initialization. The paper finishes with conclusions drawn in Section 6.

2. HIERARCHICAL SYSTEM FOR WORD DISCOVERY

Fig. 1 gives an overview of the hierarchical system for the task of word discovery. It is based on the audio semantic analysis system proposed in [8] and consists of two levels. On the first level (left box in Fig. 1), AUDs, i.e., the basic acoustic building blocks, are trained from the continuous audio recordings. The output of this stage is a transcription of the input speech in terms of AUD sequences. On the second level (right box) word pronunciations are discovered by finding consistent sequences of AUDs. To allow for variations in their realization, word pronunciations are modeled by discrete HMMs. The switch in each iterative HMM training block is initially in the left position for initialization, and is turned to the right position for the iterative training. In the following we describe the system in more detail.

Fig. 1. Block diagram of the hierarchical system for word discovery (corresponding sections given in braces)



3. ACOUSTIC UNIT DESCRIPTOR TRAINING

3.1. Segmentation and Custering

First, the speech input is segmented into chunks according to a local (cosine) distance measure between the mean representative of the current segment and the next feature vector using a constraint on the minimum length of a segment. As input MFCC feature vectors were used.

The goal of clustering is to group the obtained segments according to acoustic consistency. Then, to each utterance a sequence of cluster labels can be assigned which will serve as the initial label sequence for the iterative training of the AUD models.

As a similarity measure between segments the (length normalized) dynamic time warping distance $d_{a,b}$ between two segments S_a and S_b is employed, using the cosine distance. To avoid the huge computational effort from computing all pairwise distances the clustering is carried out on a representative subset of the segments. To find such a subset we adopt the approach from [10], which applies the k-means++ algorithm [11] to determine K elements of the set of segments $\mathcal{S}$ such that the diversity in the data is well represented:

1. Set $k = 1$. Choose the first segment $S^{(k)}$ uniformly at random from the set $\mathcal{S}$.
2. Compute the DTW distances $d(S^{(k)}, S_i)$ between the chosen segment $S^{(k)}$ and all other $N-1$ segments in $\mathcal{S}$, and store the distances in the vector $\mathbf{d}_{\text{min}}$.
3. Increment k and choose the next seed value $S^{(k)} \in \mathcal{S}$ with probability proportional to its distance in $\mathbf{d}_{\text{min}}$.
4. Compute the DTW distances between $S^{(k)}$ and all other segments and replace an entry in the minimum distance vector $\mathbf{d}_{\text{min}}$ if the computed distance is smaller than the stored value.
5. Go to 3. until K representatives are found.

The idea behind this kind of seed selection is to prevent elements of $\mathcal{S}$ from being drawn which are very close to the set of already drawn segments. On the other hand, although insignificant outliers in $\mathcal{S}$ may have a great distance to the set of previously drawn elements, the probability to draw one of them is small, since the overall number of outliers is by definition small.

Clustering is now carried out on a sparse distance matrix containing only the distances between the chosen K segments and all other segments. As a clustering algorithm the graph clustering algorithm by Newman is used, which iteratively maximizes the modularity of the clustered graph, i.e. the ratio of the edges connecting vertices within a cluster to the edges connecting vertices of different clusters [12]. The adjacency matrix is computed from the distance matrix using $e^{-d_{a,b}}$ with diagonal elements and elements with no distance assigned set to zero. Finally a set of $\hat{K}$, where $\hat{K} \ll K$, clusters is obtained at the maximum modularity.

Thus, through the choice of K representatives, the number of distance computations is reduced from the order of $\mathcal{O}(N^2)$ to $\mathcal{O}(K \cdot N)$. Also the graph clustering algorithm runs significantly faster on the resulting sparse distance matrix.

3.2. Iterative AUD HMM Training

The cluster labels are now interpreted to be AUD labels, and the cluster labels obtained in the last step are used as a initial label se-

quence, which we denote as transcription $T_d^{(0)}$, $d = 1, \ldots, D$. Here, d is the index of the training utterance and D is the total number of training utterances. For each AUD $A \in \mathcal{A}$ we define a HMM λ_A and refer to the set of all AUD models as $\Lambda_{\mathcal{A}}$. Every model is a 3-state left-to-right HMM with Gaussian mixture output densities.

Let $T_d^{(i)}$ denote the transcription of the d-th training utterance in the i-th iteration, and let $\mathbf{X}_d = (\mathbf{x}_{d,1}, \ldots \mathbf{x}_{d,\tau_d})$ denote the MFCC feature vector sequence of the d-th utterance. The maximum likelihood estimation of the HMM parameters $\Lambda_{\mathcal{A}}$ leads to the following iterative EM algorithm, which alternates between reestimation of the AUD parameters, eq. (1), and decoding, i.e., reestimation of the label sequence, eq. (2) [8, 3]:

$$\Lambda_{\mathcal{A}}^{(i+1)} = \underset{\Lambda_{\mathcal{A}}}{\operatorname{argmax}} \prod_{d=1}^{D} p(\mathbf{X}_d | T_d^{(i)}; \Lambda_{\mathcal{A}}) \tag{1}$$

$$T_d^{(i+1)} = \underset{T}{\operatorname{argmax}}\, P(T | \mathbf{X}_d; \Lambda_{\mathcal{A}}^{(i+1)}). \tag{2}$$

Here we made use of the Viterbi approximation instead of employing the Forward-Backward algorithm.

4. WORD PRONUNICIATION DISCOVERY

The Bayesian decision rule applied to ASR asks for finding that word sequence $\hat{W}$ that maximizes the posterior probability given the acoustic evidence $\mathbf{X}$ or, equivalently,

$$\hat{W} = \underset{W}{\operatorname{argmax}}\ \{P(W)p(\mathbf{X}|W)\} \tag{3}$$

with the acoustic model $p(\mathbf{X}|W)$ and the language model $P(W)$. Usually, the acoustic model of a word is obtained by concatenating the HMMs of the subword units that make up the word according to a pronunciation dictionary. With T denoting the subword unit sequence we have

$$\hat{W} = \underset{W}{\operatorname{argmax}} \left\{ P(W) \sum_{T} p(\mathbf{X}|T, W) P(T|W) \right\} \tag{4}$$

With a deterministic pronunciation dictionary each word is assigned a unique subword unit sequence, such that $P(T|W)$ reduces to a dirac delta function. Here, however, we allow for multiple transcriptions and train a HMM $P(T|W)$ for each word W. Note that the term "word" refers to a consistent sequence of AUDs which need not be identical with the linguistic notion of a word.

In principle, the same iterative algorithm for training the HMM parameters is applied as in the previous section. However, the input data are now of categorial nature: the AUD token sequences T_d, $d = 1, \ldots, D$, delivered by the first stage. In the following we first describe the model and how it is trained, before we turn to the important issue of initialization in section 4.4.

4.1. Language Model

Since in our setting the words are not known in advance one might wonder if one has any a priori knowledge about them at all. We assume that the unigram word probabilities adhere to a power law distribution, the Zipf law, which holds ubiquitously across many languages [13].

Let $w^{(k)}$, $k = 1, \ldots, N$ denote the words, where k denotes the rank of the word in a probability table ordered according to descending frequency of occurrence, and where N is the lexicon size. According to Zipf's law the unigram probabilities are given by

$$P(w^{(k)}; s) = \frac{1/k^s}{\sum_{i=1}^{N} 1/i^s}. \tag{5}$$

The parameter s will be estimated on the data. Here we assume that either the lexicon size N is known or the number N of different words to be discovered is fixed in advance.

4.2. HMM Topology

The experiments reported later revealed that the Bakis left-to-right HMM topology, that is commonly used in (supervised) ASR was too unconstrained and there was a need to limit the flexibility and guide the training process more. This led us to adopt a strong length constraint.

The length n of a realization of the word $w^{(k)}$ (in number of AUDs) is modelled to be a draw from the Negative Binomial (NB) distribution with word-specific parameters (r_k, p_k), where the Negative Binomial distribution is a generalization of the Poisson distribution, which is often used to model lengths:

$$n \sim P_{\mathrm{NB}}(n; r_k, p_k), \quad \text{where} \tag{6}$$

$$P_{\mathrm{NB}}(n; r_k, p_k) = \binom{n+r-1}{n} p^n (1-p)^r. \tag{7}$$

Fig. 2 depicts the HMM topology. The transition probabilities can be readily computed from the NB distribution, e.g.: $a_{1,\text{out}} = \mathrm{NB}(1; r, p)$, $a_{1,2} = 1 - \mathrm{NB}(1; r, p)$, etc..

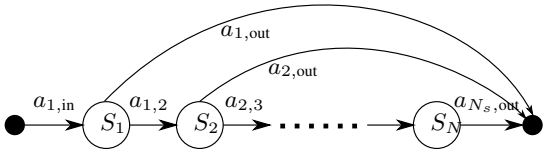


Fig. 2. HMM topology of word model.

The emission probabilities are modeled as multinomial probabilities: $\Phi_{k,l,m}$, where $k = 1, \ldots, N$ and $l = 1, \ldots, L_k$, $m = 1, \ldots, M$, is the probability that AUD m is emitted by state l of word w_k.

4.3. Iterative Training

Let $T_d = (T_{d,1}, \ldots, T_{d,t}, \ldots, T_{d,N_d})$ be the observed AUD sequence of the d-th utterance of length N_d AUDs. Here, $T_{d,t} \in \mathcal{A} = \{A^{(1)}, \ldots, A^{(M)}\}$ will be called the t-th observation.

In the E-step, the posterior probability $\gamma_t(q_{k,l})$ of being in the l-th HMM state of word w_k for the t-th observation, given the whole observed sequence T_d, is computed by the Forward-Backward algorithm:

$$\gamma_t(q_{k,l}) = \frac{\alpha_t(q_{k,l}) \cdot \beta_t(q_{k,l})}{\sum_{m,n} \alpha_t(q_{m,n}) \cdot \beta_t(q_{m,n})}, \tag{8}$$

where $\alpha_t(q_{k,l})$ is the forward probability, the probability of being in state $q_{k,l}$ at time t and having observed $(T_{d,1}, \ldots, T_{d,t})$, and $\beta_t(q_{k,l})$ denotes the backward probability of being in state $q_{k,l}$ at time t given the future observations $(T_{d,t+1}, \ldots, T_{d,N_d})$.

In the M-Step the multinomial emission probabilities are reestimated as follows

$$\Phi_{k,l,m} = \frac{\sum_{d=1}^{D} \sum_{t=1}^{N_d} \gamma_t(q_{k,l}) \delta(T_{d,t} = A^{(m)})}{\sum_{m'=1}^{M} \sum_{d=1}^{D} \sum_{t=1}^{N_d} \gamma_t(q_{k,l}) \delta(T_{d,t} = A^{(m')})}, \tag{9}$$

for all $k = 1, \ldots, N$, $l = 1, \ldots, L_k$, $m = 1, \ldots, M$. Here $\delta(\cdot)$ denotes the Kroneker delta symbol which takes the value one if the argument is true and zero else.

The NB parameters (r_k, p_k) for each word w_k are estimated by maximizing the word dependent likelihood

$$L_k = \prod_{i=1}^{m_k} P_{\text{NB}}(n_i; r_k, p_k); \quad k = 1, \ldots, N. \tag{10}$$

Here, m_k denotes the number of occurrences of word w_k in the training corpus, and n_i the length of the i-th occurrence.

Finally, the parameter s of the Zipf distribution is estimated as the slope of the best fit line between the log-expected frequencies $\mathbf{y} = (\ln E_{w_1}, \ldots, \ln E_{w_N})^T$, where $E_{w_k} = \frac{m_k}{\sum_j m_j}$, and its log-rank $\mathbf{x}$ using linear regression:

$$s = -\mathbf{x}^+\mathbf{y}. \tag{11}$$

In eq. (9) word and state dependent multinomial distributions are estimated. As an alternative one could use tied states, where all HMM states of a word model share the same emission probability. This results in the bag-of-AUD model that was proposed in [8]. To achieve this, eq. (8) is replaced by

$$\gamma_t(k) = \frac{\sum_l \alpha_t(q_{k,l}) \cdot \beta_t(q_{k,l})}{\sum_{m,n} \alpha_t(q_{m,n}) \cdot \beta_t(q_{m,n})} \tag{12}$$

where the additional summation is over all HMM states of word w_k. The reestimation formula for the multinomial emission distribution then becomes

$$\Phi_{k,m} = \frac{\sum_{d=1}^{D} \sum_{t=1}^{N_d} \gamma_t(k)\delta(T_{d,t} = A^{(m)})}{\sum_{m'=1}^{M} \sum_{d=1}^{D} \sum_{t=1}^{N_d} \gamma_t(k)\delta(T_{d,t} = A^{(m')})}, \tag{13}$$

where $\Phi_{k,m}$ is the probability of observing AUD $A^{(m)}$ in word w_k.

4.4. Initialization by Pattern Discovery

In the unsupervised setting neither the pronunciation dictionary nor the transcription of an utterance in terms of a word sequence is given. To obtain an appropriate initialization we employ a dynamic time warping (DTW) approach. In [4] it has been argued that word-level patterns are fairly stable across speakers. Thus acoustic pattern discovery techniques, such as DTW, are suitable to discover word or phrase sized patterns even if uttered by different speakers.

Here, the segmental DTW algorithm of [6] was employed. It delivered a collection of segement pairs and a similarity measure for each pair. Then, the found segment pairs were clustered using the same graph clustering algorithm as in Section 3.1 and a predefined number of the N biggest clusters was returned.

DTW, however, is usually unable to find all occurrences of a word in a corpus. Thus there will be parts of the acoustic signal for which no similar pattern has been found and which will therefore not be assigned a cluster label. Also, the speech may contain more words/phrases than one is able to discover by DTW.

Thus the output of the clustering is unable to deliver a complete transcription of the audio data. Instead of initializing by a label sequence, the HMM training is initialized with an initial model: For each segment found by DTW, the AUD sequence within this segment was extracted, and a multinomial emission probability was estimated from the AUD sequences of all segments belonging to the same cluster. By this, an initial model is estimated and the iterative training can start.

5. EXPERIMENTS

We performed our experiments on the TIDIGITs Database, downsampled to 16 kHz, both in a speaker-dependent (SD) and speaker-independent (SI) setting. The training set comprises 112 speakers and 77 digit sequences per speaker. The dataset contains 11 distinct words, the numbers 'oh' and 'zero' to 'nine'. We used the ETSI standard front-end to extract 13 Mel-frequency cepstral coefficients (MFCC) from the audio data and additionally the first and second order derivatives, resulting in a 39 dimensional feature vector per 10 ms frame. Finally cepstral mean and variance normalization (CMVN) was done on the full feature vector.

5.1. AUD Discovery Performance

First we evaluated the performance of the AUD extraction algorithm by calculating the average purity (AP) and precision-recall break-even (PRB) as proposed by [14] on the AUDs and, for comparison, on the MFCCs extracted from the TIDIGITS database. The average purity is a measure of how well a representation is able to capture the (desired) variability between different subword units while being insensitive to variations due to different realisations of the same subword unit. It has been shown that AP has a high correlation with the phoneme recognition rate [14].

Table 1 shows the results for the AP and PRB for the speaker dependent and speaker independent setups. We extracted $\hat{K} = 128$ AUDs and used $K = 1024$ seed values in the clustering step. The discovered AUD sequences were compared with the ground truth (GT) word transcriptions that were obtained from a forced alignment on the MFCCs. As the distance measure between AUD sequences we used the length normalized edit distance.

Table 1. AP and PRB in SD and SI case for AUDs and MFCCs extracted form TIDIGITs database in %

	AP		PRB	
Setup	AUD	MFCC	AUD	MFCC
SD	94.7	92.6	83.3	85.9
SI	64.6	61.7	60.0	57.5

As a reference, Table 1 also shows the AP and PRB when using MFCCs and the length normalized cosine distance. It can be seen that the AP and PRB of the AUDs is comparable to the MFCCs. The results show that the AUD extraction algorithm in fact delivers meaningful features which allow to discriminate between words. It can also be seen that the AP and PRB in the SD case are higher than in the SI case because of the lower variability in words uttered by the same speaker.

Table 2 shows the average duration of the AUDs extracted in the SD and SI case. For comparison the average duration of a phoneme as given by the ground truth labels is shown as well. It can be seen, that the average length of the AUDs is about half as long as the average length of a phoneme. This indicates that AUDs may not be directly be equated with a phoneme.

Table 2. Average AUD length in SD and SI case compared to average phoneme length on GT labels extracted from TIDIGITs database

SD	SI	GT
74 ms	62 ms	126 ms

5.2. Word Discovery Performance

Next the performance of the word discovery method presented in Section 4 was evaluated. This was done by comparing the decoding result with the ground-truth word transcription, where the HMMs were given that word as label that led to the overall smallest word error rate. We set our algorithm to extract $N = 11$ words. The Zipf parameter s was estimated to be $s \approx 0.5$ by the algorithm. For the digit recognition task under investigation, with equal probability of all words, the true value is $s = 0$. Setting $s = 0$ did not change the discovery performance considerably in our case.

Table 3 lists the word discovery results for three different HMM setups: The first two columns show the word accuracy (ACC $=$ $1-$WER) for the HMM topology of Fig. 2 with $N_s = 7$ states in the SD case and $N_s = 11$ states in the SI case. The column heading 'tied' indicates that all states of a HMM share the same multinomial emission probability, while 'untied' corresponds to the training of different emission probabilities for each HMM state. As a further comparision, a Bakis left-to-right ('l-r') HMM topology with $N_s = 7$ states in the SD case and $N_S = 8$ states in the SI case and untied emission probabilities was also tested.

Table 3. ACC (in %) for SD and SI case and for different setups

Setup	tied	untied	l-r
SD	74.5	62.5	12.5
SI	57.3	67.9	15.4

We can draw several conclusions from these results. First we can observe that the use of state-specific emission probabilities increases the performance in the speaker independent case significantly from 57% to 68%, while the performance in the speaker-dependent case suffers compared to using tied emission probabilities. Not surprisingly, the temporal order of subword units is an important characteristic of a word, which is lost when using a bag-of-AUD model as is done in the 'tied' case. This information can only be taken advantage of if the models can be trained reliably, as the reduced accuracy in the SD case is most likely attributed to the lack of sufficient training data. This precludes the reliable esimation of state-specific probabilities in the SD case.

Second we can observe that a simple left-right HMM model does not give useful results. This indicates that the modelling strength of the left-right model is weak with respect to pattern discovery. The proposed models do incorporate more stringent word length constraints which helps in guiding the training process and discriminating between different words.

5.3. DTW Initialization

The initialization of the parameters in an EM learning algorithm is well known to have significant impact on the quality of the learnt models. The simplest option is to initialize the parameters uniformly at ramdom, as it was done for the previous experiments reported in Table 3.

For the results of Table 4 the word pronunciation training was initialized with the help of a DTW based pattern discovery algorithm as described in Section 4.4. We present the results for the speaker independent case only, using state-specific emission probabilities, as this is the most relevant setup.

For DTW, only a subset of the whole database was employed, which contained all 112 speakers but only 7 randomly selected utterances per speaker. On this dataset DTW was run to find similar segment pairs. A high acceptance threshold was chosen to make sure that the segments found indeed showed high similarity. Doing this the selected segments made up only 3.5% of the whole dataset.

For each segment in a cluster, its start and end points were used to extract an AUD label sequence that was used for the initial estimation of the emission probabilities of the word model corresponding to the cluster.

The assignment of a word label to a DTW cluster is necessary to carry out an evaluation w.r.t. word error rate. A cluster was given the word label of that word that was most often represented in the segments of the cluster. This resulted in a high cluster purity of 98%. Note that this resembles a weakly supervised setup where the segments and clusters are discovered in an unsupervised manner and only the class labels have to be assigned using some other knowledge source.

There is, however, an issue with this assignment. The number of clusters obtained from DTW has been set to equal the number of words N for which models are to be developed. However, it turned out that no clusters corresponding to the digits 'two', 'eight' and 'oh' had been found. Because of the three missing classes we only initialized $N - 3 = 8$ clusters according to the extracted AUD sequences while the remaining three clusters were still initialized using random values.

Table. 4 shows that the DTW-based initialization improved the word accuracy from 67.9% to 81.9% compared to random initialization, a reduction in error rate by more than 40%. As a comparision, the ACC is given for perfect initialization, i.e. given the correct transcription in terms of word labels and the correct word boundaries for the whole database (denoted 'ground truth' (GT)). It can be seen that the DTW initialization comes close to the ground truth even though it uses much less segments for initialization and even though only 8 of 11 clusters were initialized using discovered segments. The algorithm was also able to successfully recover the 3 remaining randomly initialized words. The Zipf parameter s was estimated to be $s \approx 0.3$ when using DTW initialization and $s \approx 0.1$ when using the GT initialization which is closer to the true value $s = 0$ compared to the case without initialization.

Table 4. ACC (in %) for different initialization strategies

random	DTW	GT
67.9	81.9	88.1

5.4. Automatic Speech Recognizer Training

In a last step we used the discovered transcriptions of the word discovery algorithm as initialization for a automatic speech recognizer training. We trained whole-word HMMs with Gaussian Mixture emission probabilities on MFCC input features for each discovered cluster. We used the discovered label sequence of the word discovery algorithm as the transcription for the acoustic model training and again alternated between decoding and model estimation. In each iteration we used the decoding result of the previous iteration as the new transcription.

Table 5 shows the results of the iterative training in terms of word accuracy for the different iterations using random and DTW initialization. Iteration 0 is the result delivered by the word discovery algorithm. The further iterations are the results of the iterative training.

The results show that the discovered transcriptions can in fact be used for an automatic speech recognizer training. The iterative

Table 5. ACC (in %) for iterative speech recognizer training over iterations and for different initialization strategies on training set.

Iter.	0	1	3	5	7
random	67.9	80.8	82.9	84.4	84.7
DTW	81.9	96.6	98.4	98.5	98.5

training improves the accuracy from iteration to iteration. The improvement from the 0th to the 1st iteration is a result of cleaning the transcriptions and especially removing insertions in the recognition result. For the completely unsupervised transcriptions we achieve an accuracy of 84.7% after 7 iterations. For the transcriptions discovered using the DTW based initialization, which uses weak supervision as was explained in Section 5.3, we achieve an accuracy of 98.5% which comes close to the accuracy of 99.4% using a completely supervised training.

To test if the trained acoustic models generalize on unseen test data, we used the acoustic models to decode the up to now unused test set of the TIDIGITS database. Table 6 shows the results.

Table 6. ACC (in %) using trained acoustic models on test set.

random	DTW
84.3	98.3

It can be seen that the automatic speech recognizer delivers almost the same results on the test data set as during the discovery step using the training data.

6. CONCLUSIONS

We presented a hierarchical system for unsupervised word discovery consisting of acoustic unit and lexical unit discovery. We learned phone like acoustic unit descriptors (AUDs) in an unsupervised manner and evaluated the performance of the learnt AUDs in terms of AP and PRB showing that they deliver a comparable performance to MFCCs. The lexical unit discovery operated on the sequence of AUD labels and learnt HMMs with multinomial emission probabilities for each word. Doing this in a completely unsupervised fashion, a word accuracy of 68% was achieved, where the use of temporal information and a length constraint are important features of the presented algorithm. Further we employed DTW to initialize the word discovery algorithm using segments and clusters discovered in an unsupervised manner and using weak supervision to assign a word identity of each cluster. This led to an improved word accuracy of 82%. Finally, the transcriptions obtained from the word discovery algorithm were used for the training of an automatic speech recognizer delivering an accuracy of 85% in a completely unsupervised setup and close to ideal 99% accuracy in the weakly supervised setup.

For the future we are planning to use the proposed system on large vocabulary tasks. Then AUD based acoustic models instead of whole-word models will be trained for the speech recognizer. To this end, a canonical transcription of a word in terms of AUDs will be derived from the probabilistic lexicon or by clustering the AUD sequences found for a given word. Coupling the AUD discovery, the speech recognizer training and using a canonical transcription is expected to lead to more robust and speaker independent AUDs and therefore also better word discovery results.

7. REFERENCES

[1] A. Jansen et al., "A summary of the 2012 JHU CLSP workshop on zero resource speech rechnologies and models of early language acquisition," in *Proc. ICASSP*, Vancouver, Ca., May 2013.

[2] A. Jansen and K. Church, "Towards unsupervised training of speaker independent acoustic models," in *Proc. Interspeech*, Florence, Italy, 2011.

[3] M. Siu, H. Gish, A. Chan, W. Belfield, and S. Lowe, "Unsupervised training of an HMM-based self-organiing unit recognizer with applications to topic classification and keyword discovery," *Comput. Speech Lang.*, 2013.

[4] A. Jansen and H. Hermansky, "Weak top-down constraints for unsupervised acoustic model training," in *Proc. ICASSP*, Vancouver, Ca., May 2013.

[5] Daichi Mochihashi, Takeshi Yamada, and Naonori Ueda, "Bayesian unsupervised word segmentation with nested pitman-yor language modeling," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*. Association for Computational Linguistics, 2009, pp. 100–108.

[6] A. S. Park and James R. Glass, "Unsupervised pattern discovery in speech," *IEEE Transactions on Audio, Speech & Language Processing*, vol. 16, no. 1, pp. 186–197, 2008.

[7] Xavier Anguera, Robert Macrae, and Nuria Oliver, "Partial sequence matching using an unbounded dynamic time warping algorithm," in *Int. Conf. on Acoustics Speech and Signal Processing (ICASSP)*, 2010.

[8] Sourish Chaudhuri and Bhiksha Raj, "Unsupervised structure discovery for semantic analysis of audio," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1187–1195.

[9] Sourish Chaudhuri, Mark Harvilla, and Bhiksha Raj, "Unsupervised learning of acoustic unit descriptors for audio content representation and classification," in *INTERSPEECH*. 2011, pp. 2265–2268, ISCA.

[10] Joerg Schmalenstroeer, Markus Bartek, and Reinhold Haeb-Umbach, "Unsupervised learning of acoustic events using dynamic time warping and hierarchical k-means++ clustering," in *Interspeech 2011*, 2011.

[11] David Arthur and Sergei Vassilvitskii, "k-means++: the advantages of careful seeding," in *Proc. ACM-SIAM symposium on Discrete algorithms*, 2007, pp. 1027–1035.

[12] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, pp. 026113+, Feb. 2004.

[13] Christopher D Manning and Hinrich Schütze, *Foundations of statistical natural language processing*, MIT press, 1999.

[14] Michael A Carlin, Samuel Thomas, Aren Jansen, and Hynek Hermansky, "Rapid evaluation of speech representations for spoken term discovery.," in *INTERSPEECH*, 2011, pp. 821–824.