Fully Unsupervised Small-Vocabulary Speech Recognition Using a Segmental Bayesian Model

Fully Unsupervised Small-Vocabulary Speech Recognition Using a Segmental Bayesian Model

Herman Kamper[1,2], *Aren Jansen*[3], *Sharon Goldwater*[2]

[1]CSTR and [2]ILCC, School of Informatics, University of Edinburgh, UK
[3] HLTCOE and CLSP, Johns Hopkins University, USA

h.kamper@sms.ed.ac.uk, aren@jhu.edu, sgwater@inf.ed.ac.uk

Abstract

Current supervised speech technology relies heavily on transcribed speech and pronunciation dictionaries. In settings where unlabelled speech data alone is available, unsupervised methods are required to discover categorical linguistic structure directly from the audio. We present a novel Bayesian model which segments unlabelled input speech into word-like units, resulting in a complete unsupervised transcription of the speech in terms of discovered word types. In our approach, a potential word segment (of arbitrary length) is embedded in a fixed-dimensional space; the model (implemented as a Gibbs sampler) then builds a whole-word acoustic model in this space while jointly doing segmentation. We report word error rates in a connected digit recognition task by mapping the unsupervised output to ground truth transcriptions. Our model outperforms a previously developed HMM-based system, even when the model is not constrained to discover only the 11 word types present in the data.

Index Terms: unsupervised speech processing, word discovery, speech segmentation, unsupervised learning, segmental models

1. Introduction

Large amounts of speech audio data are being made available online every day, even for severely under-resourced and endangered languages.[1] However, most of this speech data is unlabelled. To take advantage of these resources, we must develop unsupervised speech processing methods that can learn linguistic structure directly from raw speech audio without access to transcriptions or pronunciations. Such techniques are also essential in modelling how infants acquire language from speech input.

Recent studies in the speech processing community have applied unsupervised techniques to tasks such as phonetic discovery [1, 2, 3, 4], lexical discovery [5, 6, 7], spoken document retrieval [8] and query-by-example search [9, 10]. In this community, lexical discovery, also referred to as unsupervised term discovery (UTD), involves finding repeated word-sized patterns while treating the rest of the data as background [5]. Meanwhile in the scientific cognitive modelling community, unsupervised techniques are used to model how infants learn phonetic categories and a lexicon for their native language [11]. Here, models of lexical discovery perform full-coverage segmentation of data into a sequence of words (proposing word boundaries for the entire input), but take as input phonemic [12, 13] or phonetic [14, 15] symbol sequences rather than speech audio.

Our goal is a system which performs full-coverage segmentation of continuous speech into hypothesized word units. Compared to current unsupervised speech technology (which mostly aims to find repeated snippets), the proposed system would allow unsupervised tasks such as query-by-example search and speech indexing (grouping together related utterances in a database) to be solved in a manner similar to their supervised counterparts. Furthermore, such a system could be viewed as a cognitive model that learns from acoustic input rather than transcribed speech.

Three recent studies share this goal of full-coverage speech segmentation. Chung et al. [16] models discovered subword units as hidden Markov models (HMMs) and learns a lexicon by iterating between unsupervised decoding and parameter reestimation. In addition to phone- and word-layers, the model of Lee [17, Ch. 3] includes a syllable layer and a noisy channel model for capturing pronunciation variability. Unfortunately, evaluation was performed using spoken term detection, which does not evaluate the full-coverage output of the system.

Walter et al. [18] also followed a two-step iterative approach of subword and then word discovery. Every discovered word type is modelled as a whole-word discrete HMM with a multinomial emission distribution over subword units, accounting for variation. They evaluated their system in an 11-word connected digit recognition task, and constrained the system to only discover 11 types. Using a mapping of unsupervised output to ground truth reference transcriptions, they reported unsupervised word error rates: a random initialization resulted in an error rate of 32.1%; using UTD [5] to provide initial word identities and boundaries, 18.1% was achieved. This convincing study shows that careful heuristic design makes unsupervised speech recognition on a small vocabulary task possible. It also provides useful baselines on a small but standard dataset, and gives a reproducible evaluation method in terms of classic word error rate.

We present a novel segmental Bayesian model for solving the full-coverage speech segmentation task. Our approach differs from previous studies in several respects. Instead of operating directly on acoustic frames, our model uses a fixed-dimensional representation of whole segments: any potential word segment of variable length is mapped to a fixed-length vector. This approach has found success in other unsupervised studies [19, 20, 21] since search and clustering tasks are more efficient in a fixed-dimensional space. Our model is also developed in a Bayesian framework (based on Goldwater et al. [12]), which is in contrast to the maximum likelhood approaches of [16, 18]. Finally, in contrast to all the above-mentioned studies, our model has no explicit subword modelling layer. We show that our model substantially outperforms the digit recognition baseline of Walter et al. [18], and does so within a single computational framework that does not rely on a UTD system for initialization.

2. The segmental Bayesian model

We first provide an intuitive overview of our complete model, and then describe the different components in more detail.

[1] E.g. www.oralliterature.org and www.endangeredlanguages.com.

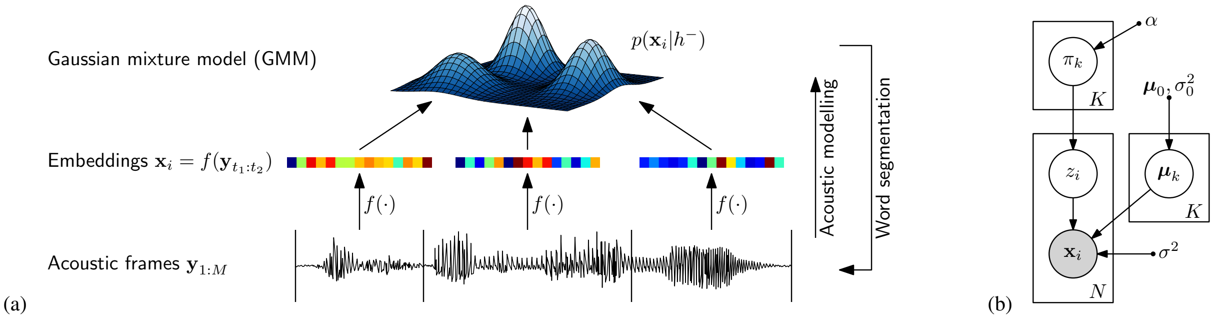

Figure 1: (a) *Overview of our unsupervised word segmentation approach.* (b) *The Bayesian GMM with fixed spherical covariance.*

2.1. Model overview

In our approach, a potential word segment (of arbitrary length) is mapped to a point in a fixed-dimensional space $\mathbb{R}^d$. The idea is that word instances of the same type[2] should lie close together in this space. We model different hypothesized word types in this d-dimensional space using a Gaussian mixture model (GMM) with Bayesian priors. Every mixture component corresponds to a word type; the component mean can be seen as an average embedding for that word. However, we do not know the identities of the word types to which the components correspond.

Assume such an ideal GMM exists. Given a new unsegmented unlabelled utterance of acoustic feature frames $\mathbf{y}_{1:M} = \mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_M$, the aim is to hypothesize where words start and end, and to which word type (GMM mixture component) every word segment belongs. Given a proposed segmentation hypothesis, we can embed every word segment, calculate a likelihood score for each embedding under the GMM, and obtain an overall score for the hypothesis. This is illustrated in Figure 1(a). The aim then is to find the optimal segmentation under the current GMM, which can be done using dynamic programming.

In our actual model, the Bayesian GMM is built up jointly while performing segmentation: the GMM provides the likelihood terms required for segmentation, while the segmentation hypothesizes the boundaries for the word segments which are then clustered using the GMM. The GMM (details in Section 2.3) can thus be seen as an acoustic model which discovers the underlying word types of a language, while the segmentation component (Section 2.4) discovers where words start and end. We use a Gibbs sampler for inference, and provide the details below.

2.2. Fixed-dimensional representation of speech segments

Our model requires that any acoustic speech segment in an utterance be represented as a fixed-dimensional vector. To obtain this mapping, we follow the embedding approach developed in [19].

The notation $Y = \mathbf{y}_{1:T}$ is used to denote a vector time series, where each $\mathbf{y}_t$ is frame-level acoustic features (e.g. MFCCs). We need a mapping function $f(Y)$ that maps time series Y into a space $\mathbb{R}^d$ in which the distance between mappings indicate similar linguistic content; in our case, we desire smaller distances between embeddings of word instances of the same type. In [19], mapping f is performed as follows. For a target speech segment, a reference vector is constructed by calculating the dynamic time warping (DTW) alignment cost to every exemplar in a reference set $\mathcal{Y}_{\text{ref}} = \{Y_i\}_{i=1}^{N_{\text{ref}}}$. Applying dimensionality reduction to the reference vector yields the embedding in $\mathbb{R}^d$. As in [19], we use Laplacian eigenmaps [22] for dimensionality reduction. Intuitively, this finds a mapping such that speech segments that are nearest neighbours in the reference set are projected to similar regions in $\mathbb{R}^d$. To embed a segment Y which is not an element of $\mathcal{Y}_{\text{ref}}$, a kernel-based out-of-sample extension is used [23].

2.3. Acoustic modelling: discovering word types

Given a segmentation hypothesis of a corpus (indicating where words start and end), the acoustic model needs to cluster the hypothesized word segments (represented as fixed-dimensional vectors) into groups of hypothesized word types. Formally, given the embedded word vectors $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$ from the current segmentation, the acoustic model needs to assign each vector $\mathbf{x}_i$ to one of K clusters, each corresponding to a hypothesized type.

As acoustic model we use a Bayesian GMM with fixed spherical covariance. This model treats its mixture weights $\boldsymbol{\pi}$ and component means $\{\boldsymbol{\mu}_k\}_{k=1}^K$ as random variables rather than point estimates, as is done in a regular GMM. In [20] we showed that the Bayesian GMM performs significantly better in clustering word embeddings than a GMM trained with expectation maximization. The former also fits naturally within the sampling framework of our complete model. We use conjugate priors: a Dirichlet prior over $\boldsymbol{\pi}$ and a diagonal-covariance Gaussian prior over $\boldsymbol{\mu}_k$. The model, illustrated in Figure 1(b), is formally defined as:

$$
\begin{aligned}
\boldsymbol{\pi} &\sim \text{Dir}\left(\alpha/K\mathbf{1}\right) & \boldsymbol{\mu}_k &\sim \mathcal{N}(\boldsymbol{\mu}_0, \sigma_0^2\mathbf{I}) \\
z_i &\sim \boldsymbol{\pi} & \mathbf{x}_i &\sim \mathcal{N}(\boldsymbol{\mu}_{z_i}, \sigma^2\mathbf{I})
\end{aligned}
$$

with z_i indicating which of the K components $\mathbf{x}_i$ belongs to. We use $\boldsymbol{\beta} = (\boldsymbol{\mu}_0, \sigma_0^2, \sigma^2)$ to denote the hyperparameters.

Given $\mathcal{X}$, we infer the component assignments $\mathbf{z} = (z_1, z_2, \ldots, z_N)$ using a Gibbs sampler [24]. Since we chose conjugate priors, we can marginalize over $\boldsymbol{\pi}$ and $\{\boldsymbol{\mu}_k\}_{k=1}^K$ and only need to sample $\mathbf{z}$. This is done in turn for each z_i conditioned on all the other current component assignments:

$$
\begin{aligned}
&P(z_i = k|\mathbf{z}_{\backslash i}, \mathcal{X}; \alpha, \boldsymbol{\beta}) \\
&\qquad \propto P(z_i = k|\mathbf{z}_{\backslash i}; \alpha) p(\mathbf{x}_i|\mathcal{X}_{k\backslash i}; \boldsymbol{\beta}) \qquad (1)
\end{aligned}
$$

where $\mathbf{z}_{\backslash i}$ is all latent component assignments excluding z_i and $\mathcal{X}_{k\backslash i}$ is all the embedding vectors assigned to component k without taking $\mathbf{x}_i$ into account. Both terms in (1) can be analytically calculated: the first can be seen as a discounted unigram language modelling probability [25, p. 843], while the second is the posterior predictive under the k^{th} Gaussian distribution [26].

2.4. Word segmentation of speech

Here we describe how acoustic modelling is performed jointly with word segmentation, using a blocked Gibbs sampler with dynamic programming [13] for inference. Our model can be seen as an extension of the Bayesian model of Goldwater et

[2] 'Word type' refers to distinct words, e.g. the entries in a lexicon.

al. [12]; their model took transcribed phoneme sequences as input, while our model operates on continuous speech audio.

Given acoustic data $\{\mathbf{s}_i\}_{i=1}^{D}$, where every utterance $\mathbf{s}_i$ consists of acoustic frames $\mathbf{y}_{1:M_i}$, we need to hypothesize word boundary locations and a word type (mixture component) for each hypothesized segment. The blocked Gibbs sampler, which samples a segmentation utterance-wide, is given in Algorithm 1: an utterance $\mathbf{s}_i$ is selected; the embeddings from the current segmentation $\mathcal{X}(\mathbf{s}_i)$ are removed from the acoustic model; a new segmentation is sampled; and finally the embeddings from this new segmentation are added back into the Bayesian GMM.

To sample a new segmentation for $\mathbf{s}_i$ (line 6), the forward filtering backward sampling algorithm is used [27]. Forward variable $\alpha[t]$ is defined as the density of the frame sequence $\mathbf{y}_{1:t}$, with the last frame the end of a word: $\alpha[t] \triangleq p(\mathbf{y}_{1:t})$. The α's can be calculated recursively [13], as:

$$\alpha[t] = \sum_{j=1}^{t} p(\mathbf{y}_{t-j+1:t}|h^{-})\alpha[t-j] \tag{2}$$

The embeddings and component assignments for all words not in $\mathbf{s}_i$, and the hyperparameters of the GMM, are denoted as $h^{-} = (\mathcal{X}_{\backslash s}, \mathbf{z}_{\backslash s}; \alpha, \boldsymbol{\beta})$. We calculate (2) for $1 \leq t \leq M-1$.

In a frame-based supervised setting, the $p(\mathbf{y}_{t-j+1:t}|h^{-})$ term in (2) would be calculated as the product of the density values of a GMM for the frames involved. However, we work at a whole-word level, and our acoustic model is defined over a whole segment. Let $\mathbf{x}_h = f(\mathbf{y}_{t-j+1:t})$ be the word embedding calculated on acoustic frames $\mathbf{y}_{t-j+1:t}$. We then define $p(\mathbf{y}_{t-j+1:t}|h^{-}) \triangleq \left[p\left(\mathbf{x}_h|h^{-}\right)\right]^{j}$, where $p\left(\mathbf{x}_h|h^{-}\right)$ is the marginal of $\mathbf{x}_h$ under the current GMM, calculated by marginalizing over the right-hand side of (1). The marginal is thus repeated for each of the j frames in the segment. This per-frame scaling is used to include a density term for every acoustic frame of which a segment is composed, as is done in frame-based supervised systems; here the same density term is just repeated.

Once all α's have been calculated, a segmentation can be recursively sampled backwards from position t [13], using:

$$P(q_t = j|\mathbf{y}_{1:t}, h^{-}) \propto p(\mathbf{y}_{t-j+1:t}|h^{-})\alpha[t-j] \tag{3}$$

We calculate (3) for $1 \leq j \leq t$ and sample while $t - j \geq 1$.

Algorithm 1 gives the complete sampler. The algorithm's inside loop is also illustrated in Figure 1(a): lines 4 to 6 perform word segmentation, proceeding top-to-bottom in the figure; line 7 performs acoustic modelling, proceeding bottom-to-top.

Algorithm 1 Gibbs sampler for word segmentation of speech.

```
1: Choose an initial segmentation (e.g. random).
2: for $j = 1$ to $J$ do                            ▷ Gibbs sampling iterations
3:     for $i$ = randperm(1 to $D$) do               ▷ Select utterance $\mathbf{s}_i$
4:         Remove embeddings $\mathcal{X}(\mathbf{s}_i)$ from acoustic model.
5:         Calculate $\alpha$'s using (2).            ▷ Forward filtering
6:         Sample word boundaries for $\mathcal{X}(\mathbf{s}_i)$ using (3).
7:         Add new $\mathcal{X}(\mathbf{s}_i)$ into acoustic model using (1).
8:     end for
9: end for
```

2.5. Iterating the model

As explained in Section 2.2, the fixed-dimensional embedding extraction relies on a reference set $\mathcal{Y}_{\text{ref}}$. Intuitively, if we had true words in the exemplar set, unseen words would be consistently mapped to similar locations in the embedding space. In this unsupervised setting, however, we do not have such a set. We therefore start with exemplars extracted randomly from the data. We extract embeddings using this set, and then run our sampler in an unconstrained setup where it can discover many more clusters than the true number of word types. From the 15 biggest clusters discovered in this first iteration (chosen since these cover more than 90% of the data), we extract a new embedding set, which can be used to recalculate embeddings. We repeat this procedure for a number of iterations, resulting in a refined exemplar set $\mathcal{Y}_{\text{ref}}$.

3. Experiments

3.1. Experimental setup and evaluation

We perform evaluation on the TIDigits [28] connected digit corpus which has an official training set with 112 speakers (male and female) and 77 digit sequences per speaker, and a comparable test set. There is no speaker overlap between the two sets. We report results on both sets separately; in each case, unsupervised modelling and evaluation is performed on the same set. Since we do not train on one set and test on the other, we refer to the official training set as a *development set*. The data contains 11 distinct word types: 'oh' and 'zero' through 'nine'.

For evaluation, the unsupervised decoding output of a system is compared to the ground truth transcriptions; discovered word types are greedily mapped to ground truth types to obtain the smallest word error rate (WER). Since at most one cluster can be assigned to each true type, unassigned clusters are counted as errors (when there are more than 11 clusters). By comparing the word boundary positions proposed by our system to those from forced alignments of the data (falling within 40 ms), we also calculate boundary precision and recall, and report the F-scores.

We use $d = 15$ dimensional embeddings throughout, extracted using 30 nearest neighbours, a kernel width $\sigma_K = 0.04$, regularizer $\xi = 2.0$, and a reference set of size $N_{\text{ref}} = 5000$ (full details in [19]). 15-dimensional frequency-domain linear prediction features [29] are used as input to DTW calculations. As in [20], embeddings are normalized to the unit sphere. We found that some embeddings were close to zero, causing issues in the sampler; we therefore add low-variance zero-mean Gaussian noise before normalizing. Based on [20, 26, 30], we use the following hyperparameters for the acoustic model: all-zero vector for $\boldsymbol{\mu}_0$, $\alpha = 1$, $\sigma^2 = 0.005$, $\sigma_0^2 = \sigma^2/\kappa_0$ and $\kappa_0 = 0.05$.

To make the search problem in Algorithm 1 tractable, we impose several constraints. We impose 200 ms minimum and 1 s maximum duration on potential words; use simulated annealing [12]; and run 5 sampling chains in parallel [24] and report results for the chain with the highest marginal $p(\mathcal{X}, \mathbf{z}; \alpha, \boldsymbol{\beta})$ (taking per-frame scaling into account as explained in Section 2.4). As mentioned, [18] constrained their system to only discover 11 clusters (the true number). We do this as well (the *constrained* setting), but also run a model that is allowed to discover up to 100 clusters (the *unconstrained* setting).

We consider two system initialization strategies, which were also used in [18]: (i) random initialization; and (ii) initialization from a UTD system. A UTD system (we use [7]) typically employs a DTW-method to find re-occurring word-sized patterns in a corpus, thus providing boundary positions and cluster assignments for the word snippets discovered. Walter et al. [18] used both the boundaries and word identities for initialization in the the UTD case, while we use only the word boundaries for initialization, and leave it up to the model to discover the clusters.

3.2. Results

As explained in Section 2.5, we use our model to iteratively rediscover a reference set $\mathcal{Y}_{\text{ref}}$ used for embedding extraction.

Table 1: *Development WERs for the unconstrained model. Each iteration provides an exemplar set for the next.*

Iteration	1	2	3	4	5
WER (%)	49.8	24.6	20.6	25.2	24.7

Performance of our unconstrained model, starting from a random initialization in each iteration, is shown in Table 1; this represents the most realistic setting where lexicon size is not known upfront. Despite being allowed to discover many more clusters (up to 100) than the true number of word types (11), the model achieves a WER of around 25% in the second iteration, with lowest overall WER in the third iteration. Although all 100 components of this third-iteration model are occupied, only the largest 16 components contain more than 100 word instances (out of a total of 28 329). The 5% higher WER in iterations 4 and 5 is due to a single 'error': the digit 'six' is split into two clusters, only one of which is then mapped to the ground truth label. Listening to these clusters (all more than 99.7% pure) revealed that they separate out faster- and slower-spoken instances of 'six'.

To compare with the HMM-based system of [18], we use the exemplar set discovered in iteration 3 and constrain our model to 11 components. Table 2 shows the unsupervised WERs on the training portion of TIDigits. Under random and UTD initialization, our constrained model performs 20% and 6% better absolute than the discrete HMM, respectively. Note again that the Bayesian models only use boundary information in the UTD condition, while the HMM additionally uses word identities for cluster initialization. The WER of the third-iteration unconstrained model (Table 1) is repeated in the last row of Table 2. Despite only mapping 11 out of 100 clusters to true labels, this unconstrained model still yields 11.5% absolute lower WER than the discrete HMM with the correct number of clusters. A completely supervised system achieves 0.6% WER [18].

3.3. Further analysis and discussion

Table 3 shows performance of randomly-initialized systems on both development and test data. Exemplar extraction and segmentation was performed separately on the two sets. As described in Section 3.1, we used 5 sampling chains for each of the Bayesian models; the 'avg.' rows in the table show average performance across these 5 runs, with performance of the model yielding highest $p(\mathcal{X}, \mathbf{z}; \alpha, \boldsymbol{\beta})$ given in the 'highest prob.' rows. For the 11-component Bayesian models (rows 2 and 3), there is a high variance in WERs from multiple runs, as seen by the big difference in average and optimal model performance. Despite this, Figure 2 illustrates that there is a strong correlation between performance and model score. In contrast to the 11-component models, the unconstrained Bayesian models (rows 4 and 5), yield consistent results over different chains, and also give better boundary segmentation F-scores than the 11-component models. It seems, then, that the 11-component sampler has a more difficult search task, possibly caused by the stricter constraints.

Table 2: *Comparison of the unsupervised discrete HMM system of [18] to our segmental Bayesian model, in terms of development WER (%). The constrained column indicates whether the model was set to discover the true number of word types.*

Model	Constrained	Random	UTD
Discrete HMM [18]	yes	32.1	18.1
Segmental Bayesian	yes	**11.2**	**12.1**
Segmental Bayesian	no	20.6	-

Table 3: *Development and test set WERs for average and highest probability constrained and unconstrained systems over 5 sampling chains. Random initialization is used in all cases.*

Model	Develop (%)		Test (%)	
	WER	Seg. F	WER	Seg. F
Discrete HMM [18]	32.1	-	-	-
Avg. constrained	21.1	61.9	27.2	58.2
Highest prob. const.	11.2	69.6	20.8	66.7
Avg. unconstrained	20.7	75.7	32.3	69.7
Highest prob. unconst.	20.6	75.8	32.3	69.6

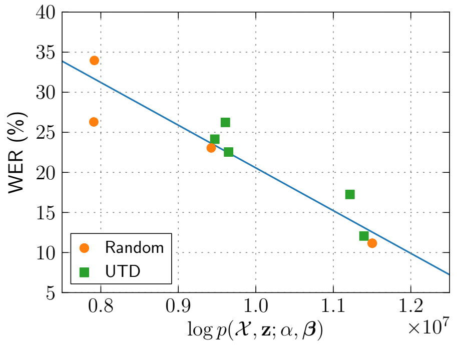

Figure 2: *WER against model probability for different sampling chains of the constrained segmental Bayesian models in Table 2.*

Table 3 also shows a discrepancy between development and test set performance using the Bayesian models, with test set WERs 10% absolute higher in most cases. To investigate this, we listened to the clusters of the constrained test system with 20.8% WER. Almost all of the errors were caused by two fail modes. First, the biggest cluster in this model is bimodal, containing the digits 'one' and 'nine'. Second, the digit 'five' is split across two clusters: one consisting mostly of instances of [f ay], the other of [ay v]. This is not an unreasonable 'error' to make. These trends seem to be a consequence of the 11-component constraint; in the unconstrained test model (32.3% WER), the digits 'one' and 'nine' are found in different clusters, while 'five' is split between two clusters. In the unconstrained development model (20.6%), 'five' is found in a single cluster, resulting in the better WER and F-score compared to the unconstrained test case.

Despite higher WER, the results in Table 3 show that the performance of the Bayesian model on the test data is still better than the HMM on the development data (test scores were not given for the discrete HMM in [18]). Compared to the latter, the unconstrained test model also achieves comparable WER (32.1% vs. 32.3%). When using UTD initialization on the test set, an 11-component Bayesian model achieves 14.2% WER, which is still better than the 18.1% of the development HMM (Table 2).

4. Conclusion

We introduced a novel Bayesian model, operating on fixed-dimensional embeddings of speech, which segments and clusters unlabelled continuous speech into hypothesized word-sized units. We applied our model to a small-vocabulary recognition task and compared performance to a more traditional HMM-based approach of a previous study. In most cases the segmental Bayesian model achieves improvements over the baselines of more than 10% absolute in error rate, and achieves improvements even when it is not constrained to discover the correct number of word types (as the HMM was).

5. References

[1] C.-y. Lee and J. R. Glass, “A nonparametric Bayesian approach to acoustic model discovery,” in *Proc. ACL*, 2012.

[2] L. Badino, C. Canevari, L. Fadiga, and G. Metta, “An auto-encoder based approach to unsupervised learning of subword units,” in *Proc. ICASSP*, 2014.

[3] G. Synnaeve1, T. Schatz, and E. Dupoux, “Phonetics embedding learning with side information,” in *Proc. SLT*, 2014.

[4] H. Kamper, M. Elsner, A. Jansen, and S. J. Goldwater, “Unsupervised neural network based feature extraction using weak top-down constraints,” in *Proc. ICASSP*, 2015.

[5] A. S. Park and J. R. Glass, “Unsupervised pattern discovery in speech,” *IEEE Trans. Audio, Speech, Language Process.*, vol. 16, no. 1, pp. 186–197, 2008.

[6] A. Jansen, K. Church, and H. Hermansky, “Towards spoken term discovery at scale with zero resources,” in *Proc. Interspeech*, 2010.

[7] A. Jansen and B. Van Durme, “Efficient spoken term discovery using randomized algorithms,” in *Proc. ASRU*, 2011.

[8] H. Gish, M.-H. Siu, A. Chan, and B. Belfield, “Unsupervised training of an HMM-based speech recognizer for topic classification,” in *Proc. Interspeech*, 2009.

[9] Y. Zhang and J. R. Glass, “Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams,” in *Proc. ASRU*, 2009.

[10] F. Metze, X. Anguera, E. Barnard, M. Davel, and G. Gravier, “The spoken web search task at MediaEval 2012,” in *Proc. ICASSP*, 2013.

[11] O. J. Räsänen, “Computational modeling of phonetic and lexical learning in early language acquisition: Existing models and future directions,” *Speech Commun.*, vol. 54, pp. 975–997, 2012.

[12] S. J. Goldwater, T. L. Griffiths, and M. Johnson, “A Bayesian framework for word segmentation: Exploring the effects of context,” *Cognition*, vol. 112, no. 1, pp. 21–54, 2009.

[13] D. Mochihashi, T. Yamada, and N. Ueda, “Bayesian unsupervised word segmentation with nested Pitman-Yor language modeling,” in *Proc. ACL*, 2009.

[14] G. Neubig, M. Mimura, S. Mori, and T. Kawahara, “Learning a language model from continuous speech,” in *Proc. Interspeech*, 2010.

[15] M. Elsner, S. J. Goldwater, N. Feldman, and F. Wood, “A joint learning model of word segmentation, lexical acquisition and phonetic variability,” in *Proc. EMNLP*, 2013.

[16] C.-T. Chung, C.-a. Chan, and L.-s. Lee, “Unsupervised discovery of linguistic structure including two-level acoustic patterns using three cascaded stages of iterative optimization,” in *Proc. ICASSP*, 2013.

[17] C.-y. Lee, “Discovering linguistic structures in speech: Models and applications,” Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, 2014.

[18] O. Walter, T. Korthals, R. Haeb-Umbach, and B. Raj, “A hierarchical system for word discovery exploiting DTW-based initialization,” in *Proc. ASRU*, 2013.

[19] K. Levin, K. Henry, A. Jansen, and K. Livescu, “Fixed-dimensional acoustic embeddings of variable-length segments in low-resource settings,” in *Proc. ASRU*, 2013.

[20] H. Kamper, A. Jansen, S. King, and S. J. Goldwater, “Unsupervised lexical clustering of speech segments using fixed-dimensional acoustic embeddings,” in *Proc. SLT*, 2014.

[21] K. Levin, A. Jansen, and B. Van Durme, “Segmental acoustic indexing for zero resource keyword search,” in *Proc. ICASSP*, 2015.

[22] M. Belkin and P. Niyogi, “Laplacian eigenmaps for dimensionality reduction and data representation,” *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, 2003.

[23] M. Belkin, P. Niyogi, and V. Sindhwani, “Manifold regularization: A geometric framework for learning from labeled and unlabeled examples,” *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, 2006.

[24] P. Resnik and E. Hardisty, “Gibbs sampling for the uninitiated,” University of Maryland, College Park, MD, Tech. Rep., 2010.

[25] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA: MIT Press, 2012.

[26] ——, “Conjugate Bayesian analysis of the Gaussian distribution,” 2007. [Online]. Available: http://www.cs.ubc.ca/~murphyk/mypapers.html

[27] S. L. Scott, “Bayesian methods for hidden Markov models,” *J. Am. Stat. Assoc.*, vol. 97, no. 457, pp. 337–351, 2002.

[28] R. G. Leonard, “A database for speaker-independent digit recognition,” in *Proc. ICASSP*, 1984.

[29] M. Athineos and D. P. W. Ellis, “Frequency-domain linear prediction for temporal features,” in *Proc. ASRU*, 2003.

[30] F. Wood and M. J. Black, “A nonparametric Bayesian alternative to spike sorting,” *J. Neurosci. Methods*, vol. 173, no. 1, pp. 1–12, 2012.