UNSUPERVISED LEXICAL CLUSTERING OF SPEECH SEGMENTS USING FIXED-DIMENSIONAL ACOUSTIC EMBEDDINGS

Herman Kamper[1,2]*, Aren Jansen*[3]*, Simon King*[1]*, 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, simon.king@ed.ac.uk, sgwater@inf.ed.ac.uk

ABSTRACT

Unsupervised speech processing methods are essential for applications ranging from zero-resource speech technology to modelling child language acquisition. One challenging problem is discovering the word inventory of the language: the lexicon. Lexical clustering is the task of grouping unlabelled acoustic word tokens according to type. We propose a novel lexical clustering model: variable-length word segments are embedded in a fixed-dimensional acoustic space in which clustering is then performed. We evaluate several clustering algorithms and find that the best methods produce clusters with wide variation in sizes, as observed in natural language. The best probabilistic approach is an infinite Gaussian mixture model (IGMM), which automatically chooses the number of clusters. Performance is comparable to that of non-probabilistic Chinese Whispers and average-linkage hierarchical clustering. We conclude that IGMM clustering of fixed-dimensional embeddings holds promise as the lexical clustering component in unsupervised speech processing systems.

Index Terms— Lexical clustering, unsupervised learning, fixed-dimensional embeddings, lexical discovery.

1. INTRODUCTION

In the last few decades, considerable advances have been made in supervised speech recognition for several languages. However, for most of the approximately 6 500 languages spoken in the world, speech applications are not being developed. Despite audio data collection efforts for under-resourced languages,[1] the transcription of audio remains a major obstacle in system development. *Unsupervised methods* that can learn linguistic structure directly from the speech signal would allow 'zero-resource' technology [1] to be developed without transcriptions or pronunciation dictionaries.

Work from two different communities is relevant to this problem. In speech technology, unsupervised techniques have been applied to tasks such as phonetic discovery [2, 3], lexical discovery [4, 5, 6], spoken document retrieval [7] and query-by-example search [8, 9]. In this community, lexical discovery involves finding repeated word-sized patterns while treating the rest of the data as background [4]. 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 [10]. 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 [11, 12] or phonetic [13, 14] symbol sequences rather than speech audio.

In addition to the desired full-coverage segmentation, these cognitive models can learn a language model during segmentation; this has been shown to greatly improve the accuracy of both segmentation and lexical discovery [11, 14]. Our ultimate goal is therefore to extend this approach to the continuous speech domain, developing what is essentially an unsupervised speech recognition system. That is, our envisioned architecture would jointly (i) hypothesize a complete segmentation of the input speech into word-like segments (tokens); (ii) cluster the tokens into lexical items (word types) and relate them to the underlying acoustics; and (iii) estimate a language model over the discovered types. Compared to current unsupervised speech technology (which mostly focus only on finding repeated snippets), the proposed system would enable tasks such as query-by-example search and unsupervised speech indexing (grouping together related utterances in a speech database) to be performed in a manner similar to their supervised counterparts. Furthermore, the system could be used as a cognitive model that learns from acoustic input rather than transcribed speech, and could thus be used to test hypotheses about the connections between phonetic and lexical learning in infants.

Some of the subtasks needed for this complete system have been investigated elsewhere. For example, [2] and [3] considered tasks (i) and (ii), but at the phone rather than word level; an additional layer would be required for the full model. [13], [15] and [16] considered (iii), but took phone lattices from supervised systems as input. The most complete work similar to what we envision is [17], but their approach was tested on small-vocabulary prompted speech and requires pre-specifying the number of lexical items; we are working towards a probabilistic model for conversational speech.

In this paper we consider the lexical clustering compo-

[1]See e.g. www.oralliterature.org and www.endangeredlanguages.com.

nent (ii) in isolation, assuming perfect segmentation (i). The aim of the clustering component is to cluster unidentified segmented word tokens (i.e. different realizations of words) according to the word types (i.e. the unique word-forms) to which they belong. Since the clustering component provides a mapping from the vectorized speech signal to a categorical structure, it could be described as the *acoustic model* of an unsupervised system (operating at the word level, in our case).

Dynamic time warping (DTW) is a standard way of measuring the similarity between speech segments of differing duration. However, Levin et al. [18] recently showed that embedding variable-length speech segments in a fixed-dimensional space can yield more efficient similarity comparisons that are at least as accurate as DTW. A major advantage of embedding in a fixed-dimensional space is that a wide variety of standard clustering algorithms are then applicable. Two scenarios were considered in [18]: in one a set of unidentified word exemplars are available, in the other the exemplar labels are known.

We evaluate several clustering algorithms on embeddings of segmented word tokens in these two settings. We focus on probabilistic approaches since they can eventually be integrated into existing word segmentation models [11, 12] in the probabilistic framework. For completeness and to better understand the embeddings, we compare these to standard non-probabilistic clustering algorithms. The aim of this paper is to show that clustering fixed-dimensional acoustic embeddings of word segments is viable as a lexical clustering model.

2. EMBEDDING SPEECH SEGMENTS IN A FIXED-DIMENSIONAL SPACE

The goal in [18] was to find a function that maps variable-length acoustic speech segments to a high-dimensional continuous space in which smaller distances correspond to greater similarity of linguistic content. We use the notation $Y = \mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_T$ to denote a vector time series of arbitrary length, where each $\mathbf{y}_t \in \mathbb{R}^K$ is the frame-level acoustic feature representation of the signal (e.g. MFCCs or PLPs). Given two series Y_1 and Y_2, the goal is to find a mapping f into $\mathbb{R}^D$ such that the resulting vectors $f(Y_1)$ and $f(Y_2)$ are near each other if and only if series Y_1 and Y_2 are linguistically similar. As in [18], we restrict ourselves to word segments in this study. Two relevant embedding scenarios were considered:

1. *UnsupTrain*: A set of unidentified exemplars are available $\mathcal{Y}_{\text{train}} = \{Y_i\}_{i=1}^{N_{\text{train}}}$.
2. *SupTrain*: In addition to $\mathcal{Y}_{\text{train}}$, the true word type of every exemplar is known. An unsupervised word discovery system [4, 6] could be used to construct such a labelled set.

In both cases a similar strategy is followed. For each vector time series Y in $\mathcal{Y}_{\text{test}}$, a reference vector is constructed by calculating the DTW alignment cost to every exemplar in a reference set $\mathcal{Y}_{\text{ref}} \subseteq \mathcal{Y}_{\text{train}}$. Applying dimensionality reduction to this reference vector then gives the desired embedding in $\mathbb{R}^D$. The associated projection maps are estimated only on $\mathcal{Y}_{\text{train}}$, and are then applied to the target out-of-sample set $\mathcal{Y}_{\text{test}}$. Although this still requires calculating a potentially large number of DTW alignment costs, the number of calculations is linear in the size of the test set if the reference set size is fixed.

Several dimensionality reduction techniques were compared in [18] using the *same-different task* [19]. This task quantifies the ability of a speech representation to associate words of the same type and to discriminate between words of different types. The same-different score of standard DTW on the original speech feature vectors were used as a baseline for comparing the different fixed-dimensional representations resulting from different dimensionality reduction operations. For *UnsupTrain*, Laplacian eigenmaps with a kernel-based out-of-sample extension (a non-linear graph embedding technique) performed similarly to DTW with 15-dimensional frequency-domain linear prediction features as input. For *SupTrain*, where the true word labels were used, linear discriminant analysis outperformed DTW with 39-dimensional perceptual linear prediction (PLP) features as input. In both cases cosine distance outperformed Euclidean distance as similarity measure. In this study we use these best approaches from [18]. We normalize all embeddings to the unit sphere so that Euclidean is equivalent to cosine distance up to a static non-linearity.

In this paper we use the same data as in [18]. Two sets $\mathcal{Y}_{\text{train}}$ and $\mathcal{Y}_{\text{test}}$ were extracted from the Switchboard English corpus, including only words of at least 6 characters and 0.5 s in duration. $\mathcal{Y}_{\text{train}}$ consists of 10 383 tokens with 5539 types. This set was constructed to cover several word types; it was extracted from 360 conversation sides with 156 unique speakers, with at most 5 tokens from any word type and each token for a given type from a different speaker. $\mathcal{Y}_{\text{test}}$ consists of 11 024 tokens with 3392 types. This set was constructed to reflect a content word distribution encountered in conversational speech; it was extracted from 360 conversation sides with 236 unique speakers, none in the training set, by taking all words with the minimum length requirement. The standard deviation of the number of tokens per type is 7.06 for this set. We use the full $\mathcal{Y}_{\text{train}}$ as reference set and embed the tokens in $\mathcal{Y}_{\text{test}}$ to obtain the set of embedding vectors $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{N}$. In all cases we use 50-dimensional embeddings.

3. CLUSTERING ALGORITHMS

Since our ultimate aim is a probabilistic unsupervised model of segmentation, lexical category learning and language modelling, we are primarily interested in probabilistic clustering approaches. However, for comparison and to investigate the properties of the embedding representation, we also consider state-of-the-art non-probabilistic clustering approaches.

3.1. Probabilistic approaches

We explore three types of Gaussian mixture models (GMMs). Each determines the posterior probability that an embedding

vector $\mathbf{x}_i$ belongs to the k^{th} (of K) mixture component by multiplying the mixture weight $\pi_k = p(z_i = k)$, where z_i indicates the latent cluster to which $\mathbf{x}_i$ is assigned, by the likelihood $p(\mathbf{x}_i|z_i = k)$. The three GMMs differ in their training algorithms and prior distributions over π_k.

3.1.1. EM Gaussian mixture model (EMGMM)

The finite GMM has the form:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{1}$$

where $\boldsymbol{\mu}_k$ is the mean vector and $\boldsymbol{\Sigma}_k$ the covariance matrix of the k^{th} Gaussian component. Treating $\boldsymbol{\pi} = \{\pi_k\}_{k=1}^K$, $\{\boldsymbol{\mu}_k\}_{k=1}^K$ and $\{\boldsymbol{\Sigma}_k\}_{k=1}^K$ as *parameters*, the expectation maximization (EM) algorithm can be used to fit the model under the maximum likelihood criterion on training data $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$. We refer to this model as the EMGMM (Figure 1, left).

3.1.2. Finite Bayesian Gaussian mixture model (FBGMM)

When dealing with high-dimensional data, fitting the parameters of the EMGMM can be difficult because of the many free parameters. A Bayesian approach can alleviate this problem by treating $\Theta = (\boldsymbol{\pi}, \{\boldsymbol{\mu}_k\}, \{\boldsymbol{\Sigma}_k\})$ as *random variables* and working with distributions over Θ rather than point estimates. Doing so requires specifying prior distributions on Θ. To allow us to marginalize over the parameters (see below), we use conjugate priors: a symmetric Dirichlet prior for the mixture weights $\boldsymbol{\pi}$ and a Normal-inverse-Wishart (NIW) prior for the component parameters $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$. The finite Bayesian Gaussian mixture model (FBGMM) is defined as:

$$\begin{aligned}
\boldsymbol{\pi} &\sim \text{Dir}\,(\alpha/K\mathbf{1}) && (2)\\
z_i &\sim \boldsymbol{\pi} && (3)\\
\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k &\sim \text{NIW}(\mathbf{m}_0, \kappa_0, \nu_0, \mathbf{S}_0) && (4)\\
\mathbf{x}_i &\sim \mathcal{N}(\boldsymbol{\mu}_{z_i}, \boldsymbol{\Sigma}_{z_i}) && (5)
\end{aligned}$$

and is shown in Figure 1 (middle), with $\boldsymbol{\beta} = (\mathbf{m}_0, \kappa_0, \nu_0, \mathbf{S}_0)$. Given $\mathcal{X}$, we infer the component assignments $\mathbf{z} = (z_1, z_2, \ldots, z_N)$ using a Gibbs sampler, which iteratively samples the value of each z_i in turn, conditioned on the current values of $(z_1, \ldots, z_{i-1}, z_{i+1} \ldots z_N)$ and marginalizing over Θ; see [20, §24.2.4], [21] for details. After convergence the sampler returns samples from the posterior $P(\mathbf{z}|\mathcal{X}, \alpha, \boldsymbol{\beta})$.

3.1.3. Infinite Gaussian mixture model (IGMM)

For both the EMGMM and FBGMM we need to specify the number of mixture components K. The infinite Gaussian mixture model (IGMM; Figure 1, right) is a non-parametric Bayesian model which extends the FBGMM so that the (potentially infinite) number of components is inferred automatically [22]. This is achieved by replacing the Dirichlet prior with a *Dirichlet process* prior. In the mixture model formulation, we define $\boldsymbol{\pi} \sim \text{GEM}(\alpha)$ to replace equation (2) (see [20, §25.2.3] for details); equations (3) to (5) are unchanged. GEM refers to the stick-breaking distribution, a particular construction of the Dirichlet process. Marginalizing over $\mathbf{z}$ reveals the infinite mixture model, with the form of (1) but with $K = \infty$.

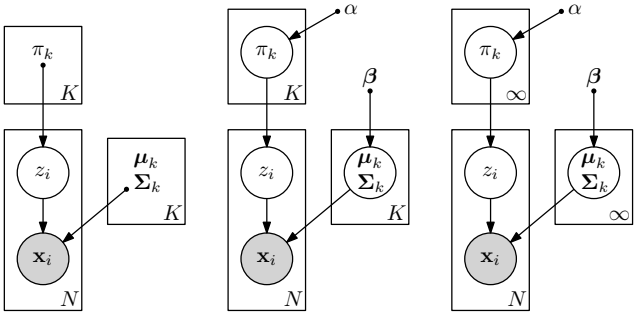

Fig. 1. From left to right: the standard EMGMM, the finite Bayesian GMM (FBGMM), and the infinite GMM (IGMM).

We again use collapsed Gibbs sampling for inference [20, §25.2], [21]. We found that both the FBGMM and IGMM converge after about 15 iterations, but we used 25 iterations for all experiments. For both models we set $\alpha = 1$, as in [21]. The interpretation of the parameters of the NIW prior in (4) is as follows [20, p. 133]: $\mathbf{m}_0$ is our prior mean for $\boldsymbol{\mu}_k$; κ_0 is how strongly we believe this prior; $\mathbf{S}_0$ is proportional to our prior mean for $\boldsymbol{\Sigma}_k$; and ν_0 is how strongly we believe this prior for $\boldsymbol{\Sigma}_k$. As in [21], we set $\mathbf{m}_0$ to the global sample mean and set $\kappa_0 = 0.05$. We set $\nu_0 = 1000$ after initial experiments showed stable performance for values in the range $\nu_0 \in [850, 3000]$. These values are large compared to those reported elsewhere [20, 21], which indicates that our data requires a strict prior on the cluster covariances. Performance was found to be most sensitive to the $\mathbf{S}_0$ parameter. Selection of this parameter is discussed in Section 4.3. The same hyperparameters were used for the FBGMM and IGMM.

3.2. Non-probabilistic approaches

We experimented with three non-probabilistic approaches:

K-means clustering: This method iteratively reassigns each vector to the cluster with the closest mean. The number of clusters must be specified beforehand. The algorithm results as a special case of the EMGMM (Section 3.1.1) as $\sigma^2 \to 0$ if all covariance matrices are set to $\sigma^2\mathbf{I}$ [23, p. 446]. We use the implementation in [24] which uses kmeans++ initialization.

Hierarchical clustering (HC): We use greedy agglomerative clustering on the normalized embeddings, merging until the desired number of clusters is reached The linkage function we found to work best is average linkage, defined between clusters A and B as $\frac{1}{|A|\,|B|}\sum_{\mathbf{a}\in A}\sum_{\mathbf{b}\in B} D(\mathbf{a}, \mathbf{b})$, where $D(\cdot,\cdot)$ is the (Euclidean) distance metric. Using cosine distance on

the unnormalized embeddings gave identical results.

Chinese whispers (CW) algorithm: The CW algorithm is a simple randomized graph clustering algorithm which has become popular in the NLP community [25]. The algorithm treats each word token as a node in a weighted undirected graph, beginning with each node in a cluster of its own. Nodes are then reassigned to the highest ranked cluster to which any of its neighbours belongs. The rank of a cluster with respect to a node is the sum of the edge weights from the cluster to that node. This process is repeated for a few iterations or until cluster assignments do not change. To construct a graph representation of the embedding vectors we follow [26]: an edge with weight $1/D_{\cos}(\mathbf{a}, \mathbf{b})$ is added between two word embedding vectors $\mathbf{a}$ and $\mathbf{b}$ if the weight exceeds some threshold. This threshold is the only parameter in the algorithm; as in the IGMM, the number of clusters is automatically inferred.

4. EXPERIMENTS

We apply all the clustering methods to the fixed-dimensional speech embeddings, focusing evaluation on the probabilistic methods but using the non-probabilistic methods to gain deeper insight into the former and the structure of the embedding space. For k-means, HC, the EMGMM and the FBGMM the number of clusters was set to the correct number of 3392 types. For CW and the IGMM, hyper-parameters were chosen to yield approximately the correct number of clusters (see Section 4.3).

4.1. Quantitative evaluation measures

Quantitative evaluation of a particular clustering is a non-trivial problem and several measures have been proposed [27, 28]. To gain a full picture of results, we use several measures to compare the induced lexical clusters to the ground truth word types (labelled clusters). Apart from the standard deviation of the cluster sizes, all values range between 0 and 1.

Cluster purity: Every cluster is mapped to the most common type in that cluster; the proportion of correctly mapped tokens is computed [29]. More than one cluster may be mapped to a type, resulting in higher scores when there are more clusters and a perfect score if every token is in its own cluster.

One-to-one mapping (1-to-1): Mapping clusters to true types is done greedily and constrained so that at most one cluster is mapped to any type (some clusters might be unassigned). The proportion of correctly mapped tokens is calculated.

Adjusted rand index (ARI): For each pair of tokens, a clustering predicts whether the pair belongs to the same or different word types. The proportion of correct predictions (rand index) is adjusted for chance to obtain the ARI [30]. This measure is especially useful when the number of clusters and the number of tokens are similar [28], as is the case here.

Standard deviation of cluster size (σ_{sizes}): The type frequency distribution of our test set is broad as is inherently the case in natural language. Thus we quantify nonuniformity of cluster size using standard deviation as an additional measure.

We also considered V-measure (VM), a widely-used information theoretic metric [29]. However, it assigns high scores to random cluster assignments when the average cluster size is small [28], and was found to be uninformative for our task.

We evaluated the metrics for trivial assignments: randomly assigning 11 024 tokens to 3392 word types gave purity, 1-to-1 and ARI scores of 0.30, 0.23 and 0.00, respectively; assigning each token to its own cluster gave scores of 1, 0.31, 0.00; assigning all tokens to one cluster gave low scores for all metrics.

Table 1. Clustering evaluation on the *SupTrain* embeddings.

Algorithm	Purity	ARI	1-to-1	σ_{sizes}	K
DTW HC	0.66	0.36	0.48	4.61	3392
EMGMM	0.67	0.17	0.42	2.43	3392
FBGMM	0.67	0.34	0.47	3.89	3199
IGMM	**0.67**	**0.40**	**0.49**	**5.63**	3411
K-means	0.66	0.17	0.41	2.49	3392
HC	0.69	**0.48**	**0.54**	5.39	3392
CW	**0.70**	0.44	0.53	**8.73**	3756

Table 2. Clustering evaluation on the *UnsupTrain* embeddings.

Algorithm	Purity	ARI	1-to-1	σ_{sizes}	K
EMGMM	0.59	0.19	0.38	3.37	3392
FBGMM	0.59	0.23	0.40	4.09	3379
IGMM	**0.60**	**0.27**	**0.41**	**5.54**	3564
K-means	0.59	0.17	0.37	3.22	3392
HC	0.59	**0.32**	**0.44**	6.87	3392
CW	**0.61**	0.25	0.43	**11.26**	3941

4.2. Results

Tables 1 and 2 show results for the *SupTrain* and *UnsupTrain* embeddings. K is the inferred number of clusters. Purity scores are high on both sets (about 70% and 60%, respectively) and similar for all algorithms, so we focus on the other metrics.

As a baseline, Table 1 shows HC performed using DTW costs calculated directly on the PLP vector time series of the word segments, i.e. without embedding. Compared to the best embedding clustering results using *SupTrain*, this baseline performs worse; however, performance is better than the best *UnsupTrain* results. Nevertheless, the fixed-dimensional embeddings have many useful properties that can be exploited in downstream tasks, e.g. representation in a probabilistic model.

Of the probabilistic algorithms the EMGMM gives poorest ARI and 1-to-1 scores on both sets. By marginalizing over the means and covariances, the FBGMM outperforms the EMGMM on both sets. Although $K = 3392$ is specified for the FBGMM, some clusters are emptied out, resulting in fewer components. By extending the FBGMM to allow for

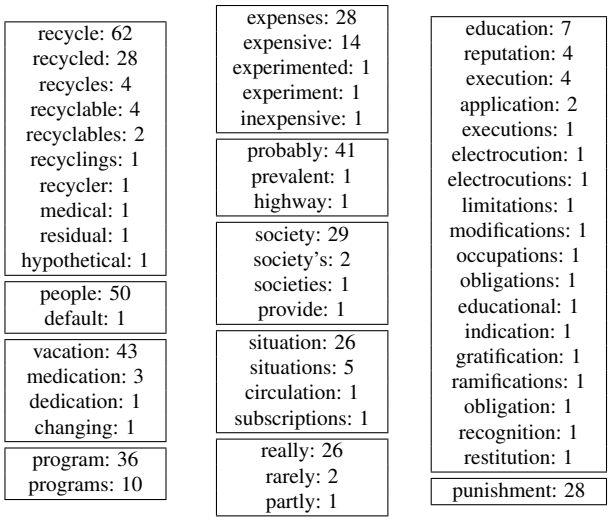

Fig. 2. The number of tokens for each type in the biggest clusters obtained using the IGMM on *SupTrain*.

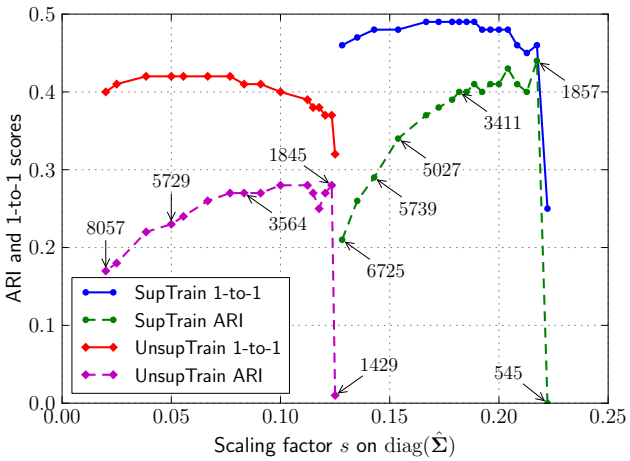

Fig. 3. ARI and 1-to-1 scores achieved by the IGMM as the scaling factor for setting the prior covariance $\mathbf{S}_0$ is varied. The annotations indicate the number of components of a model.

an arbitrary number of components, the IGMM performs best of the probabilistic approaches. K-means, like the EMGMM, performs poorly on both sets. HC outperforms all the other algorithms. The CW algorithm achieves the closest performance to HC on *SupTrain*, and scores similar to the IGMM on *UnsupTrain*. The three best algorithms (IGMM, CW, HC) also yield the largest variance in cluster size, with CW's variance even larger than the true value of 7.06.

The biggest clusters obtained on *SupTrain* using the 3411-component IGMM (Table 1) are shown in Figure 2. Although the righthand cluster overclusters several different types, substantial phonetic and morphological similarities are evident. Overall, despite some noise arising from large variations in surface forms in conversational speech, qualitatively the clusters are reasonable. HC yields similar biggest clusters. Despite the good scores of CW, its biggest clusters tend to contain more different types: for the 3756-component CW model in Table 1, the five biggest clusters have 12, 41, 57, 25, 53 different types.

4.3. Discussion

As noted in Section 3.1.3, we found that IGMM performance is sensitive to the hyper-parameter $\mathbf{S}_0$, which controls the prior on the cluster covariance matrices. Based on [20, p. 133] we set $\mathbf{S}_0$ such that the expected value of $\mathbf{\Sigma}_k$ is $s \cdot \text{diag}(\hat{\mathbf{\Sigma}})$, where $\hat{\mathbf{\Sigma}}$ is the global sample covariance and s is a scaling parameter. We varied s to find approximately the correct number of components, using values of $s = 0.18$ and $s = 0.08$ for the IGMMs in Tables 1 and 2, respectively. Figure 3 shows the sensitivity of ARI and 1-to-1 scores on *SupTrain* and *UnsupTrain* as s is varied. The annotations indicate the number of components of a model. We see that smaller covariance is required on *UnsupTrain* than on *SupTrain*. The drop-off in the scores for the 545- and 1429-component *SupTrain* and *UnsupTrain* models result from garbage clusters that capture the majority of the tokens. The annotations show that s strongly affects the number of components, yet performance remains relatively stable over a range of model sizes. In particular, on *SupTrain* the ARI and 1-to-1 scores of the IGMM are within 25% of the best performance for models ranging from 1857 to 5027 components; *UnsupTrain* models with 1845 to 5729 components give scores within 20% of the best performance.

By varying the graph construction threshold, the CW algorithm was also evaluated across different model sizes. As with the IGMM, adjusting the threshold results in very different cluster numbers, yet performance is again stable over a range of sizes: on *SupTrain*, CW models with 2755 to 7337 components give ARI and 1-to-1 scores within 25% of the best; on *UnsupTrain*, models with 3354 to 8536 components give scores within 30% of the best. HC also yields stable performance over a range of model sizes: on both *SupTrain* and *UnsupTrain* scores are within 20% of the best for models with 1000 to 5000 components. These results from the three best algorithms indicate that there are some easily distinguishable tokens which are consistently grouped correctly whether there are few or many components. To further verify this, we calculated the overlap of the tokens that are correctly one-to-one mapped for IGMM *SupTrain* models with 1857 to 5739 components: more than 74% of the tokens that are correct for the 3411-model are also mapped correctly by the other models.

The better scores of the CW and HC algorithms over the IGMM demonstrate the limitations of the single Gaussian per type assumption made by the latter. Indeed, the centroid linkage function for HC, which makes a similar one centroid per type assumption, performs very poorly on our evaluation.

Finally, comparing the scores in Tables 1 and 2 for the three best algorithms, we see that ARI drops by 33–43% and 1-to-1 drops by about 20% when moving from *SupTrain* to

UnsupTrain. This suggests that weak supervision in the form of labelled exemplars could significantly improve the performance of unsupervised speech processing systems (this was also shown in [31]). However, this 'supervision' could actually be obtained in an unsupervised fashion from a high-precision spoken term discovery system, e.g. [4, 6].

5. CONCLUSIONS

We evaluated several methods for clustering fixed-dimensional embeddings of word segments. The best probabilistic model is an infinite Gaussian mixture model (IGMM), which automatically selects its number of components and would fit naturally into a complete probabilistic unsupervised model of segmentation, lexical and language learning. Slight improvements were obtained using the non-probabilistic Chinese Whispers and average-linkage hierarchical clustering algorithms. These approaches do not model every word type as a single Gaussian, as the IGMM does. When not including supervision in learning the embeddings, performance drops by around 35%. Such supervision could, however, be obtained using a high-precision word discovery system. In future work we aim to incorporate the IGMM in a joint model of word segmentation, lexical discovery and language modelling. Ultimately, this would allow zero-resource applications to be developed in languages for which collecting transcribed speech data is not possible.

6. REFERENCES

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

[2] C. Lee and J. R. Glass, "A nonparametric Bayesian approach to acoustic model discovery," in *Proc. ACL*, 2012.

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

[4] A. Park and J. R. Glass, "Unsupervised word acquisition from speech using pattern discovery," in *Proc. ICASSP*, 2006.

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

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

[7] 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.

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

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

[10] O. Räsänen, "Computational modeling of phonetic and lexical learning in early language acquisition: Existing models and future directions," *Speech Commun.*, vol. 54, 2012.

[11] 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.

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

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

[14] 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.

[15] J. Heymann, O. Walter, R. Haeb-Umbach, and B. Raj, "Unsupervised word segmentation from noisy input," in *Proc. ASRU*, 2013.

[16] J. Heymann, O. Walter, R. Haeb-Umbach, and B. Raj, "Iterative Bayesian word segmentation for unsupervised vocabulary discovery from phoneme lattices," in *Proc. ICASSP*, 2014.

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

[18] 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.

[19] M. A. Carlin, S. Thomas, A. Jansen, and H. Hermansky, "Rapid evaluation of speech representations for spoken term discovery," in *Proc. Interspeech*, 2011.

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

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

[22] C. E. Rasmussen, "The infinite Gaussian mixture model," in *Proc. NIPS*, 1999.

[23] D. Barber, *Bayesian Reasoning and Machine Learning*, Cambridge University Press, Cambridge, UK, 2012.

[24] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[25] C. Biemann, "Chinese whispers: An efficient graph clustering algorithm and its application to natural language processing problems," in *Proc. HLT-NAACL*, 2006.

[26] C. Biemann, "Unsupervised part-of-speech tagging employing efficient graph clustering," in *Proc. COLING*, 2006.

[27] C. Christodoulopoulos, S. J. Goldwater, and M. Steedman, "Two decades of unsupervised POS induction: How far have we come?," in *Proc. EMNLP*, 2010.

[28] "Scikit-learn documentation: Clustering," Online, Available: http://scikit-learn.org/stable/modules/clustering.html.

[29] A. Rosenberg and J. Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure," in *Proc. EMNLP-CoNLL*, 2007.

[30] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Is a correction for chance necessary?," in *Proc. ICML*, 2009.

[31] A. Jansen, S. Thomas, and H. Hermansky, "Weak top-down constraints for unsupervised acoustic model training," in *Proc. ICASSP*, 2013.