

AN EMBEDDED SEGMENTAL K-MEANS MODEL FOR UNSUPERVISED SEGMENTATION AND CLUSTERING OF SPEECH

Herman Kamper¹, Karen Livescu², Sharon Goldwater³

¹Electrical and Electronic Engineering, Stellenbosch University, South Africa

²Toyota Technological Institute at Chicago, United States

³ILCC, School of Informatics, University of Edinburgh, United Kingdom

kamperh@sun.ac.za, klivescu@ttic.edu, sgwater@inf.ed.ac.uk

ABSTRACT

Unsupervised segmentation and clustering of unlabelled speech are core problems in zero-resource speech processing. Most approaches lie at methodological extremes: some use probabilistic Bayesian models with convergence guarantees, while others opt for more efficient heuristic techniques. Despite competitive performance in previous work, the full Bayesian approach is difficult to scale to large speech corpora. We introduce an approximation to a recent Bayesian model that still has a clear objective function but improves efficiency by using hard clustering and segmentation rather than full Bayesian inference. Like its Bayesian counterpart, this embedded segmental K -means model (ES-KMeans) represents arbitrary-length word segments as fixed-dimensional acoustic word embeddings. We first compare ES-KMeans to previous approaches on common English and Xitsonga data sets (5 and 2.5 hours of speech): ES-KMeans outperforms a leading heuristic method in word segmentation, giving similar scores to the Bayesian model while being 5 times faster with fewer hyperparameters. However, its clusters are less pure than those of the other models. We then show that ES-KMeans scales to larger corpora by applying it to the 5 languages of the Zero Resource Speech Challenge 2017 (up to 45 hours), where it performs competitively compared to the challenge baseline.¹

Index Terms— Zero-resource speech processing, word segmentation, unsupervised learning, language acquisition

1. INTRODUCTION

The growing area of *zero-resource speech processing* aims to develop unsupervised methods that can learn directly from raw speech audio in settings where transcriptions, lexicons and language modelling texts are not available. Such methods are crucial for providing speech technology in languages where transcribed data are hard or impossible to collect, e.g., unwritten or endangered languages [1]. In addition, such methods may shed light on how human infants acquire language [2, 3].

Several zero-resource tasks have been studied, including acoustic unit discovery [4–6], unsupervised representation learning [7–9], query-by-example search [10, 11] and topic modelling [12, 13]. Early work mainly focused on unsupervised term discovery, where the aim is to automatically find repeated word- or phrase-like patterns in a collection of speech [14–16]. While useful, the discovered patterns are typically isolated segments spread out over the data, leaving much speech as background. This has prompted several studies on *full-coverage* approaches, where the entire speech input is segmented and clustered into word-like units [17–21].

Two such full-coverage systems have recently been applied to the data of the Zero Resource Speech Challenge 2015 (ZRSC’15), giving a useful basis for comparison [22]. The first is the Bayesian embedded segmental Gaussian mixture model (BES-GMM) [23]: a probabilistic model that represents potential word segments as fixed-dimensional acoustic word embeddings, and then builds a whole-word acoustic model in this embedding space while jointly doing segmentation. The second is the recurring syllable-unit segmenter (SylSeg) [24], a cognitively motivated, fast, heuristic method that applies unsupervised syllable segmentation and clustering and then predicts recurring syllable sequences as words. These two models are representative of two methodological extremes often seen in zero-resource systems: either probabilistic Bayesian models with convergence guarantees are used [6, 19, 23, 25–27], or heuristic techniques are used in pipeline approaches [18, 24].

Here we introduce an approximation to BES-GMM that falls in between these two extremes. The *embedded segmental K -means model* (ES-KMeans) uses hard clustering and segmentation, rather than full Bayesian inference. Nevertheless, it has a clear objective function, in contrast to heuristic methods such as SylSeg. Compared to BES-GMM, it has fewer hyperparameters and a simpler optimization algorithm since probabilistic sampling is not necessary; ES-KMeans is therefore more efficient, while still having a principled objective.

Hard approximations have been used since the start of probabilistic modelling in supervised speech recognition [28–30], and also in more recent work to improve the efficiency of an

¹https://github.com/kamperh/recipe_zs2017_track2

unsupervised Bayesian model [31]. We are therefore following in a long tradition of using hard approximation. However, all of these studies applied it in frame-by-frame modelling approaches, while our approach operates on embedded representations of whole speech segments. There is a growing focus on such *acoustic word embedding* methods [11, 32–38], since they make it possible to easily and efficiently compare variable-duration speech segments in a fixed-dimensional space.

We conduct two sets of experiments. First, we analyze how the hard approximations of ES-KMeans affect speed and accuracy relative to the original BES-GMM and the SylSeg method. On English and Xitsonga data (used in ZRSC’15), we show that ES-KMeans outperforms SylSeg in word segmentation and gives similar scores to BES-GMM, while being 5 times faster. However, the cluster purity of ES-KMeans falls behind that of the other two models. We show that the higher purity for BES-GMM results from a tendency towards smaller clusters which, unlike in ES-KMeans, can also be varied using hyperparameters. In the second set of experiments—conducted as part of the Zero Resource Speech Challenge 2017 (ZRSC’17)—we show that ES-KMeans can also scale to larger corpora by applying it to the data sets of up to 45 hours for 5 languages.

2. EMBEDDED SEGMENTAL K-MEANS

Starting from standard K -means, we describe the embedded segmental K -means (ES-KMeans) objective and algorithm.

2.1. From K -means to ES-KMeans objective function

Given a speech utterance consisting of acoustic frames $\mathbf{y}_{1:M} = \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M$ (e.g., MFCCs), our aim is to break the sequence up into word-like segments, and to cluster these into hypothesized word types.

If we knew the segmentation (i.e., where word boundaries occur), the data would consist of several segments of different durations, as shown at the bottom of Fig. 1. To cluster these, we need a method to compare variable-length vector sequences. One option would be to use an alignment-based distance measure, such as dynamic time warping. Here we instead follow an *acoustic word embedding* approach [11, 32, 33]: an embedding function f_e is used to map a variable length speech segment to a single embedding vector $\mathbf{x} \in \mathbb{R}^D$ in a fixed-dimensional space, i.e., segment $\mathbf{y}_{t_1:t_2}$ is mapped to a vector $\mathbf{x}_i = f_e(\mathbf{y}_{t_1:t_2})$, illustrated as coloured horizontal vectors. The idea is that speech segments that are acoustically similar should lie close together in \mathbb{R}^D , allowing segments to be efficiently compared directly in the embedding space without alignment. Various such embedding methods have been proposed, ranging from graph-based [32] to unsupervised recurrent neural approaches [34]. Here we use a very simple method: any segment is uniformly downsampled so that it is represented by the same fixed number of vectors, which are then flattened to obtain the embedding [32]. But note that

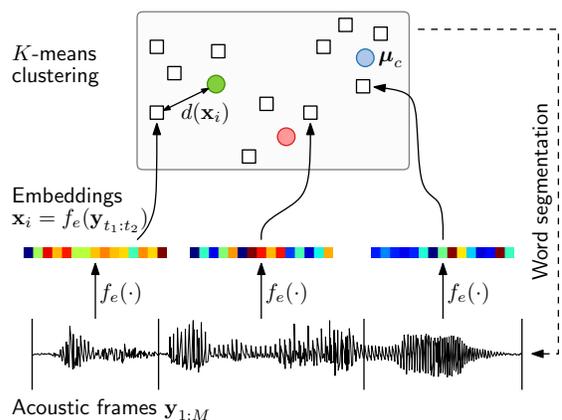


Fig. 1. The embedded segmental K -means model for unsupervised segmentation and clustering of speech.

ES-KMeans is agnostic to the embedding method, so future improvements in embeddings can be incorporated directly.

Embedding all segments in the data set would give a set of vectors $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$, which could be clustered into K hypothesized word classes using K -means, as shown at the top of Fig. 1. Standard K -means aims to minimize the sum of squared Euclidean distances to each cluster mean: $\min_{\mathbf{z}} \sum_{c=1}^K \sum_{\mathbf{x} \in \mathcal{X}_c} \|\mathbf{x} - \boldsymbol{\mu}_c\|^2$, where $\{\boldsymbol{\mu}_c\}_{c=1}^K$ are the cluster means, \mathcal{X}_c are all vectors assigned to cluster c , and element z_i in \mathbf{z} indicates which cluster \mathbf{x}_i belongs to. The standard algorithm alternates between reassigning vectors to the closest cluster means, and then updating the means.

Standard K -means would be appropriate if the segmentation was known, but this is not the case in this zero-resource setting. Rather, the embeddings \mathcal{X} can change depending on the current segmentation. For a data set of S utterances, we denote the segmentations as $\mathcal{Q} = \{\mathbf{q}_i\}_{i=1}^S$, where \mathbf{q}_i indicates the boundaries for utterance i . $\mathcal{X}(\mathcal{Q})$ is used to denote the embeddings under the current segmentation. Our aim now is to jointly optimize the cluster assignments \mathbf{z} and the segmentation \mathcal{Q} . Under what objective should these be optimized?

One option would be to extend the standard K -means objective and optimize $\min_{(\mathcal{Q}, \mathbf{z})} \sum_{c=1}^K \sum_{\mathbf{x} \in \mathcal{X}_c \cap \mathcal{X}(\mathcal{Q})} \|\mathbf{x} - \boldsymbol{\mu}_c\|^2$, where $\mathcal{X}_c \cap \mathcal{X}(\mathcal{Q})$ are embeddings assigned to cluster c under segmentation \mathcal{Q} . But this is problematic: imagine inserting no boundaries over an utterance, resulting in a single embedding and a single term in the summation; any other segmentation would result in more terms in the summation, likely giving an overall worse score—even if all embeddings are close to cluster means. Therefore, instead of assigning a score *per segment*, we assign a score *per frame*. This score is given by the score achieved by the segment to which that frame belongs, implying that segment scores are weighed by duration:

$$\min_{\mathcal{Q}, \mathbf{z}} \sum_{c=1}^K \sum_{\mathbf{x} \in \mathcal{X}_c \cap \mathcal{X}(\mathcal{Q})} \text{len}(\mathbf{x}) \|\mathbf{x} - \boldsymbol{\mu}_c\|^2 \quad (1)$$

where $\text{len}(\mathbf{x})$ is the number of frames in the sequence on which embedding \mathbf{x} is calculated.

The overall ES-KMeans algorithm initializes word boundaries randomly, and then optimizes (1) by alternating between optimizing segmentation \mathcal{Q} while keeping cluster assignments \mathbf{z} and means $\{\boldsymbol{\mu}_c\}_{c=1}^K$ fixed (top to bottom in Fig. 1), and then optimizing the cluster assignments and means while keeping the segmentation fixed (bottom to top in the figure).

2.2. Segmentation

Under a fixed clustering \mathbf{z} , the objective (1) becomes

$$\min_{\mathcal{Q}} \sum_{\mathbf{x} \in \mathcal{X}(\mathcal{Q})} \text{len}(\mathbf{x}) \|\mathbf{x} - \boldsymbol{\mu}_{\mathbf{x}}^*\|^2 = \min_{\mathcal{Q}} \sum_{\mathbf{x} \in \mathcal{X}(\mathcal{Q})} d(\mathbf{x}) \quad (2)$$

where $\boldsymbol{\mu}_{\mathbf{x}}^*$ is the mean of the cluster to which \mathbf{x} is currently assigned (according to \mathbf{z}), and $d(\mathbf{x}) \triangleq \text{len}(\mathbf{x}) \|\mathbf{x} - \boldsymbol{\mu}_{\mathbf{x}}^*\|^2$ is the ‘‘score’’ of embedding \mathbf{x} (lower d is better).

Eq. (2) can be optimized separately for each utterance: we want to find the segmentation \mathbf{q} for each utterance that gives the minimum of the sum of the scores of the embeddings under that segmentation. This is exactly the problem addressed by the shortest-path algorithm (Viterbi) which uses dynamic programming to solve this problem efficiently [39, §21.7].

Let q_t be the number of frames in the hypothesized segment (word) that ends at frame t : if $q_t = j$, then $\mathbf{y}_{t-j+1:t}$ is a word.² We define forward variables $\gamma[t]$ as the optimal score up to boundary position t : $\gamma[t] \triangleq \min_{q:t} \sum_{\mathbf{x} \in \mathcal{X}(q:t)} d(\mathbf{x})$, with $q:t$ the sequence of segmentation decisions (the q ’s) up to t . These can be recursively calculated [39, §21.7]:

$$\gamma[t] = \min_{j=1}^t \{d(f_e(\mathbf{y}_{t-j+1:t})) + \gamma[t-j]\} \quad (3)$$

Starting with $\gamma[0] = 0$, we calculate (3) for $1 \leq t \leq M - 1$. We keep track of the optimal choice (arg min) for each $\gamma[t]$, and the overall optimal segmentation is then given by starting from the final position $t = M$ and moving backwards, repeatedly choosing the optimal boundary.

2.3. Cluster assignments and mean updates

For a fixed segmentation \mathcal{Q} , the objective (1) becomes

$$\min_{\mathbf{z}} \sum_{c=1}^K \sum_{\mathbf{x} \in \mathcal{X}_c \cap \mathcal{X}(\mathcal{Q})} \text{len}(\mathbf{x}) \|\mathbf{x} - \boldsymbol{\mu}_c\|^2 \quad (4)$$

When the means $\{\boldsymbol{\mu}_c\}_{c=1}^K$ are fixed, the optimal reassignments (5) follow standard K -means, and are guaranteed to improve (1) since the distance between the embedding and its assigned cluster mean never increases:

$$z_i = \arg \min_c \{\text{len}(\mathbf{x}_i) \|\mathbf{x}_i - \boldsymbol{\mu}_c\|^2\} = \arg \min_c \|\mathbf{x}_i - \boldsymbol{\mu}_c\|^2 \quad (5)$$

²For an utterance with frames $\mathbf{y}_{1:M}$, a sequence of q ’s ending with q_M specifies a unique segmentation; boldface \mathbf{q} is this sequence of q ’s.

Finally, we fix the assignments \mathbf{z} and update the means:

$$\boldsymbol{\mu}_c = \frac{1}{\sum_{\mathbf{x} \in \mathcal{X}_c} \text{len}(\mathbf{x})} \sum_{\mathbf{x} \in \mathcal{X}_c} \text{len}(\mathbf{x}) \mathbf{x} \approx \frac{1}{N_c} \sum_{\mathbf{x} \in \mathcal{X}_c} \mathbf{x} \quad (6)$$

The exact equation is the mean of the vectors assigned to cluster c weighed by duration, and is guaranteed to improve (1). We use the approximation, which is exact if all segments have the same duration, to again match standard K -means, with N_c the number of embeddings currently assigned to cluster c .

The complete ES-KMeans algorithm is given below. Since the segmentation, clustering and mean updates each improve (1), the algorithm will converge to a local optimum.

Algorithm 1 The embedded segmental K -means algorithm.

- 1: Initialize segmentation \mathcal{Q} randomly.
 - 2: Initialize cluster assignments \mathbf{z} randomly.
 - 3: **repeat** ▷ Optimization iterations
 - 4: **for** $i = \text{randperm}(1 \text{ to } S)$ **do** ▷ Select utterance i
 - 5: Calculate γ ’s using (3). ▷ Segmentation variables
 - 6: $t \leftarrow M_i$
 - 7: **while** $t \geq 1$ **do** ▷ Perform segmentation
 - 8: $q_t \leftarrow \arg \min_{j=1}^t \{d(f_e(\mathbf{y}_{t-j+1:t})) + \gamma[t-j]\}$
 - 9: $t \leftarrow t - q_t$
 - 10: **end while**
 - 11: Assign new embeddings $\mathcal{X}(\mathbf{q}_i)$ to clusters using (5).
 - 12: Update means using (6).
 - 13: **end for**
 - 14: **until** convergence
-

2.4. The Bayesian embedded segmental GMM

In previous work [20, 23], we proposed a very similar model, but instead of K -means, we used a Bayesian GMM as whole-word clustering component (top of Fig. 1). This Bayesian embedded segmental GMM (BES-GMM) served as inspiration for ES-KMeans; we briefly discuss their relationship here.

A Bayesian GMM treats its mixture weights $\boldsymbol{\pi}$ and component means $\{\boldsymbol{\mu}_c\}_{c=1}^K$ as random variables rather than point estimates, as is done in a regular GMM. We use conjugate priors: a Dirichlet prior over $\boldsymbol{\pi}$ and a spherical-covariance Gaussian prior over $\boldsymbol{\mu}_c$. All components share the same fixed covariance matrix $\sigma^2 \mathbf{I}$. The model is then formally defined as:

$$\boldsymbol{\pi} \sim \text{Dir}(a/K \mathbf{1}) \quad (7) \quad \boldsymbol{\mu}_c \sim \mathcal{N}(\boldsymbol{\mu}_0, \sigma_0^2 \mathbf{I}) \quad (9)$$

$$z_i \sim \boldsymbol{\pi} \quad (8) \quad \mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}_{z_i}, \sigma^2 \mathbf{I}) \quad (10)$$

Under this model, component assignments and a segmentation can be inferred jointly using a collapsed Gibbs sampler [40]. Full details are given in [20], but the Gibbs sampler looks very similar to Algorithm 1: the Bayesian GMM gives likelihood terms (‘‘scores’’) in order to find an optimal segmentation, while the segmentation hypothesizes boundaries for the word segments which are then clustered using the GMM. However, for BES-GMM, component assignments and segmentation are sampled probabilistically, instead of making hard decisions.

The link between the two models emerges asymptotically. It can be shown that standard K -means results from a GMM as the variances approach zero [41, §20.3.5], [42]. In a similar way it can be shown that the Gibbs sampling equations for segmentation and component assignments for BES-GMM (as given in [20]) approach (3) and (5), respectively, in the limit $\sigma^2 \rightarrow 0$, when all other hyperparameters are fixed.

Without giving a full complexity analysis, we note that because ES-KMeans only considers the closest cluster, it is more efficient than BES-GMM, where all components are considered when assigning embeddings to clusters and during segmentation (since embedding “scores” are obtained by marginalizing over all components). ES-KMeans can also be trivially parallelized, since both segmentation and cluster assignment can be performed in parallel for each utterance. This parallelized algorithm is still guaranteed to converge, though possibly to a different local optimum than Algorithm 1 since updates are in a different order. Parallelizing BES-GMM is also possible, but the guarantee of converging to the true posterior distribution is lost [31].

2.5. Heuristic recurring syllable-unit word segmentation

We also compare to the ZRSC’15 submission of Räsänen et al. [24]. Their system, which we refer to as SylSeg, relies on a novel cognitively motivated unsupervised method that predicts boundaries for syllable-like units, and then clusters these units on a per-speaker basis. Using a bottom-up greedy mapping, recurring syllable cluster sequences are then predicted as words.

SylSeg is much simpler in terms of computational complexity and implementation than ES-KMeans or BES-GMM. But, in contrast to the heuristic methodology followed in SylSeg, both ES-KMeans and BES-GMM have clear overall objective functions that they optimize, the one using hard clustering, the other using Bayesian inference.

3. EXPERIMENTS

We perform two sets of experiments. First, we compare ES-KMeans to SylSeg and BES-GMM on the ZRSC’15 data. SylSeg and BES-GMM have both previously been applied to the relatively small ZRSC’15 corpora, making it ideal for a comparative analysis. Second, we apply ES-KMeans to the ZRSC’17 data. Although there are no previous results to compare to, the ZRSC’17 data sets are considerably larger than those of ZRSC’15, providing a useful test of scalability.

3.1. Experimental setup and evaluation

As in [20, 23, 43], we use several metrics to evaluate against ground truth forced alignments. By mapping every discovered word token to the ground truth token with which it overlaps most and then mapping every cluster to its most common word, average cluster purity and unsupervised word error rate (WER)

can be calculated.³ By instead mapping every token to the true phoneme sequence with which it overlaps most, the normalized edit distance (NED) between all segments in the same cluster can be calculated; lower NED is better, with scores from 0 to 1. Word boundary precision, recall and F -score evaluates segmentation performance by comparing proposed and true word boundaries; similarly, word token precision, recall and F -score measures the accuracy of proposed word token intervals. Word type precision, recall and F -score compares the set of unique phoneme mappings (obtained as for NED) to the set in the true lexicon. See [43] for full details. Cluster purity and WER are not considered in the ZRSC, so for some systems these results are not reported below.

Our implementation of ES-KMeans follows as closely as possible that of BES-GMM in [23]. Both use uniform downsampling as embedding function f_e : a segment is represented by flattening 10 equally spaced MFCCs, with suitable interpolation. Both models use unsupervised syllable pre-segmentation [24] to limit allowed word boundaries. For BES-GMM we use simulated annealing, an all-zero vector for μ_0 , $\sigma_0^2 = \sigma^2/\kappa_0$, $\kappa_0 = 0.05$, $a = 1$ and $\sigma^2 = 0.001$.

3.2. Results: Comparison to other systems and analysis

In the first set of experiments, we use the two ZRSC’15 data sets: an English corpus of around 5 hours of speech from 12 speakers, and a Xitsonga corpus of 2.5 hours from 24 speakers [44]. We also use a separate English set of 6 hours for development. In order to compare to previously published results [23, 24], all systems here are applied in a speaker-dependent setting, and results are averaged over speakers. As in [23], for both ES-KMeans and BES-GMM, K is set to 20% of the number of first-pass segmented syllables. Word candidates are limited to span at most 6 syllables, and must be at least 200 ms in duration. We do not parallelize ES-KMeans here, in order to keep it as close as possible to BES-GMM.

Table 1(a) shows the performance of the three models on the English and Xitsonga corpora. Some of the SylSeg scores are unknown since these were not part of the ZRSC’15 evaluation [24]. Compared to BES-GMM, ES-KMeans achieves worse purity, WER and NED, but similar boundary, token and type F -scores. This comes with a 5-time improvement in runtime. ES-KMeans achieves worse NED than SylSeg, but much better word boundary, token and type F -scores. SylSeg, however, is twice as fast.

ES-KMeans is therefore competitive in terms of word segmentation scores (boundary, token F -scores) and lexicon quality (type F -score), but falls behind in the purity-based metrics (purity, WER, NED). The difference with BES-GMM is particularly interesting since σ^2 is set to be quite small, and ES-KMeans results from BES-GMM in the limit $\sigma^2 \rightarrow 0$. To understand the discrepancy in purity, we analyzed ES-KMeans and BES-GMM on a single English development speaker.

³We allow more than one cluster to be mapped to the same word.

Table 1. (a) Performance of models on the two test corpora. Lower NED is better. Runtimes for SylSeg* are rough estimates, obtained from personal communication with the authors [24]. (b) English development set performance (%) of BES-GMM as the variance is varied.

	Cluster purity	WER	NED	Boundary F	Token F	Type F	Runtime (s)		σ^2	Cluster purity	WER
<i>English (%)</i>								(b)	0.00001	56.1	68.9
SylSeg	-	-	71.1	55.2	12.4	12.2	100*		0.0001	55.7	69.0
ES-KMeans	42.8	73.2	71.6	62.2	18.1	18.9	193		0.001	56.9	67.7
BES-GMM	56.1	68.3	55.5	62.2	17.9	18.6	1052		0.0015	51.9	69.8
<i>Xitsonga (%)</i>									0.00175	41.1	75.8
SylSeg	-	-	62.8	33.4	2.7	3.3	20*		0.002	35.1	86.8
ES-KMeans	40.5	80.3	70.4	42.1	3.7	4.9	44				
BES-GMM	49.8	71.6	58.4	43.1	4.0	5.2	196				

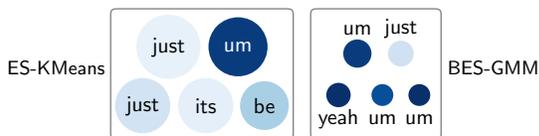


Fig. 2. The 5 biggest clusters for ES-KMeans and BES-GMM. Circle radii are according to cluster size; shading indicates purity. The cluster-to-true-word mapping is also shown.

For a qualitative view, Fig. 2 shows the 5 biggest clusters for the two models. BES-GMM outputs more smaller clusters with a higher purity (often separating the same word over different clusters) compared to ES-KMeans. By listening to tokens assigned to the same cluster by ES-KMeans, we found that, although tokens overlap with different ground truth labels, cluster assignments are qualitatively sensible, capturing similarity in acoustics or prosody. E.g., Fig. 3 shows spectrograms for tokens assigned to the “be” cluster in Fig. 2. The ground truth word labels with maximal overlap are also shown. For the “seventy” and “already” tokens, the segments only cover part of the true words (bold), and the “that you” token is actually pronounced in context as [dh uh]. So despite mapping to different true labels, these segments form a reasonable acoustic group. Nevertheless, they are penalized under purity and WER.

By spreading out its discovered tokens more evenly over clusters (Fig. 2), BES-GMM produces a clustering that is better-matched to the evaluation metrics, although the ES-KMeans clustering might be subjectively reasonable. This spreading (or sparsity) of BES-GMM can be controlled through the fixed spherical covariance parameter σ^2 , which impacts both the soft assignments of an embedding to a cluster and the segmentation (§2.4). Table 1(b) shows performance on the development set as σ^2 is varied. There is a sweet spot: when σ^2 is too big, most tokens are sucked up by a number of large garbage clusters; when σ^2 is smaller, more tokens are assigned to separate clusters. In contrast, ES-KMeans has no σ^2 parameter and considers *only* the single closest cluster.

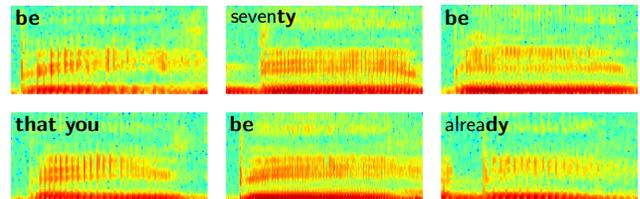


Fig. 3. Spectrograms for random tokens from the cluster of ES-KMeans mapped to “be” in Fig. 2. The portion of each true word which is covered by the segment is shown in bold.

When reading the results in Table 1(b) from bottom to top, note that the results of BES-GMM do not converge to ES-KMeans, even though σ^2 is tending towards 0. This is because, based on [45], we set the variance of the prior on the component means of BES-GMM as $\sigma_0^2 = \sigma^2 / \kappa_0$ (§3.1), so the prior variance on the component means is tied to the fixed data variance. Under these conditions, the asymptotic equivalence of BES-GMM and ES-KMeans no longer holds. Murphy [45] explains that this coupling is a sensible way to incorporate prior knowledge of the typical spread of data, and here we indeed show how this helps our Bayesian model; this principled way of including priors is not possible in ES-KMeans or SylSeg. When setting $\sigma_0^2 = 1$ (rather than tying it), the results of BES-GMM match ES-KMeans when $\sigma^2 = 0.00001$.

3.3. Results: Zero Resource Speech Challenge 2017

Next, we use ES-KMeans as our entry to ZRSC’17. For the challenge, data for three languages were released in advance with evaluation code, allowing participants to develop their systems. After registering their optimal hyperparameters, participants could apply their systems to two surprise languages and submit their system output to the organizers for evaluation. Thus, the challenge measures how well systems and hyperparameters generalize to completely unseen languages. The data consists of an English corpus of about 45 hours of speech from 69 speakers, a French corpus of 24 hours from 28 speakers, a

Table 2. Performance of the baseline system and ES-KMeans on the five languages of Zero Resource Speech Challenge 2017.

	Coverage	NED	Boundary			Token			Type			Runtime (min)
			Prec.	Rec.	<i>F</i>	Prec.	Rec.	<i>F</i>	Prec.	Rec.	<i>F</i>	
<i>English (%)</i>												
JHU-PLP	7.9	33.9	33.9	3.1	5.7	3.9	0.3	0.5	5.0	0.7	1.2	-
ES-KMeans	100	72.6	51.0	54.4	52.7	13.0	14.1	13.5	8.3	16.7	11.1	788
<i>French (%)</i>												
JHU-PLP	1.6	25.4	30.9	0.6	1.1	5.2	0.1	0.1	6.9	0.2	0.3	-
ES-KMeans	97.2	67.3	37.8	41.6	39.6	3.5	3.9	3.7	3.1	6.3	4.2	615
<i>Mandarin (%)</i>												
JHU-PLP	2.9	30.7	37.5	0.9	1.8	4.0	0.1	0.1	4.5	0.1	0.2	-
ES-KMeans	100	88.1	36.5	47.1	41.1	2.5	3.4	2.9	2.5	4.1	3.1	4
<i>Surprise Lang1 (%)</i>												
JHU-PLP	3.0	30.5	28.2	1.2	2.3	4.0	0.1	0.2	5.5	0.3	0.6	-
ES-KMeans	100	66.4	42.6	56.5	48.6	10.3	14.3	12.0	5.7	11.2	7.5	473
<i>Surprise Lang2 (%)</i>												
JHU-PLP	5.9	30.8	25.3	1.0	2.0	1.6	0.0	0.1	2.3	0.1	0.2	-
ES-KMeans	100	72.2	42.4	44.3	43.3	4.9	5.2	5.0	4.6	10.0	6.3	5

Mandarin corpus of 2.5 hours from 12 speakers, the surprise Lang1 corpus of 25 hours from 30 speakers, and the surprise Lang2 corpus of about 10 hours of speech from 24 speakers.⁴

To evaluate whether ES-KMeans scales, and in the spirit of the challenge which explicitly want to measure robustness to multiple speakers, we use a speaker-independent setup here.⁵ To deal with the increased computational load, we use more constrained hyperparameters than in the previous section: *K* is set to 10% of the number of first-pass segmented syllables, and word candidates are limited to at most 4 syllables. We also parallelize ES-KMeans over 15 CPUs. The only hyperparameter we explicitly tried to optimize was the probability with which word boundaries are initialized; the optimal values varied across languages and metrics, so we just randomly initialize boundaries with a 0.5 probability.

Table 2 compares the performance of ES-KMeans to the challenge baseline—an unsupervised term discovery system (JHU-PLP)—on the 5 languages. Term discovery systems like JHU-PLP [15] aim to find high-precision clusters of isolated segments, but do not cover all the data. Compared to ES-KMeans, it therefore performs much better on NED, which only evaluates the discovered patterns. In contrast to JHU-PLP, ES-KMeans performs full-coverage segmentation. It therefore achieves better recall, *F*-score and coverage performance on all 5 languages. For word boundary, token and type precision, JHU-PLP performs better on some of the Mandarin and French

metrics, but ES-KMeans scores better on the other languages.

Apart from word boundary detection, the scores of all metrics in Table 2 are very low for both systems, showing that zero-resource speech processing is still far from mature. Nevertheless, hyperparameters seem to generalize across languages, with ES-KMeans performing better on Lang1 and Lang2 than on either French or Mandarin. Moreover, ES-KMeans can be trained in a reasonable time on realistically sized corpora, taking 13 hours to train on the 45-hour English corpus.

4. CONCLUSION

We introduced the embedded segmental *K*-means model (ES-KMeans), a method that falls in between the fully Bayesian embedded segmental GMM (BES-GMM) and the cognitively motivated heuristic SylSeg method. Its word segmentation performance is on par with BES-GMM and superior to SylSeg, but cluster purity is worse than both other methods. In terms of efficiency, it is 5 times faster than BES-GMM, but half as fast as SylSeg. Despite using hard clustering and segmentation, ES-KMeans still has a clear objective function and it is guaranteed to converge (to a local optimum), in contrast to SylSeg. It also has far fewer hyperparameters than BES-GMM, although we show that this is what gives the latter the upper hand. Due to its improved efficiency, we are also able to apply ES-KMeans to larger corpora, on which it performs competitively. In future work we will focus on improving the acoustic word embedding function, which ES-KMeans could directly incorporate since it is agnostic to the embedding method.

Acknowledgements: We would like to thank Okko Räsänen and Shreyas Seshadri for helpful feedback on the SylSeg model.

⁴<http://sapience.dec.ens.fr/bootphon/2017/>

⁵BES-GMM can also be applied to multiple speakers, as in [23]. However, on the 5-hour English data set of §3.2, the speaker-independent BES-GMM takes around 40 hours to train. Since we wanted to compare different models and hyperparameters in a reasonable time, we used speaker-dependent modelling in §3.2. Furthermore, SylSeg is also speaker-dependent [24].

5. REFERENCES

- [1] L. Besacier, E. Barnard, A. Karpov, and T. Schultz, “Automatic speech recognition for under-resourced languages: A survey,” *Speech Commun.*, vol. 56, pp. 85–100, 2014.
- [2] 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.
- [3] E. Dupoux, “Cognitive science in the era of artificial intelligence: A roadmap for reverse-engineering the infant language-learner,” *arXiv preprint arXiv:1607.08723*, 2016.
- [4] B. Varadarajan, S. Khudanpur, and E. Dupoux, “Unsupervised learning of acoustic sub-word units,” in *Proc. ACL*, 2008.
- [5] C.-y. Lee and J. R. Glass, “A nonparametric Bayesian approach to acoustic model discovery,” in *Proc. ACL*, 2012.
- [6] L. Ondel, L. Burget, J. Cernocky, and S. Kesiraju, “Bayesian phonotactic language model for acoustic unit discovery,” in *Proc. ICASSP*, 2017.
- [7] L. Badino, C. Canevari, L. Fadiga, and G. Metta, “An auto-encoder based approach to unsupervised learning of subword units,” in *Proc. ICASSP*, 2014.
- [8] D. Renshaw, H. Kamper, A. Jansen, and S. J. Goldwater, “A comparison of neural network methods for unsupervised representation learning on the Zero Resource Speech Challenge,” in *Proc. Interspeech*, 2015.
- [9] N. Zeghidour, G. Synnaeve, N. Usunier, and E. Dupoux, “Joint learning of speaker and phonetic similarities with Siamese networks,” in *Proc. Interspeech*, 2016.
- [10] Y. Zhang, R. Salakhutdinov, H.-A. Chang, and J. R. Glass, “Resource configurable spoken query detection using deep Boltzmann machines,” in *Proc. ICASSP*, 2012.
- [11] K. Levin, A. Jansen, and B. Van Durme, “Segmental acoustic indexing for zero resource keyword search,” in *Proc. ICASSP*, 2015.
- [12] 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.
- [13] S. Kesiraju, R. Pappagari, L. Ondel, and L. Burget, “Topic identification of spoken documents using unsupervised acoustic unit discovery,” in *Proc. ICASSP*, 2017.
- [14] 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.
- [15] A. Jansen and B. Van Durme, “Efficient spoken term discovery using randomized algorithms,” in *Proc. ASRU*, 2011.
- [16] B. Oosterveld, R. Veale, and M. Scheutz, “A parallelized dynamic programming approach to zero resource spoken term discovery,” in *Proc. ICASSP*, 2017.
- [17] M. Sun and H. Van hamme, “Joint training of non-negative Tucker decomposition and discrete density hidden Markov models,” *Comput. Speech Lang.*, vol. 27, no. 4, pp. 969–988, 2013.
- [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] C.-y. Lee, T. O’Donnell, and J. R. Glass, “Unsupervised lexicon discovery from acoustic input,” *Trans. ACL*, vol. 3, pp. 389–403, 2015.
- [20] H. Kamper, A. Jansen, and S. J. Goldwater, “Unsupervised word segmentation and lexicon discovery using acoustic word embeddings,” *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 24, no. 4, pp. 669–679, 2016.
- [21] M. Elsner and C. Shain, “Speech segmentation with a neural encoder model of working memory,” in *Proc. EMNLP*, 2017.
- [22] M. Versteegh, X. Anguera, A. Jansen, and E. Dupoux, “The Zero Resource Speech Challenge 2015: Proposed approaches and results,” in *Proc. SLTU*, 2016.
- [23] H. Kamper, A. Jansen, and S. J. Goldwater, “A segmental framework for fully-unsupervised large-vocabulary speech recognition,” *Comput. Speech Lang.*, vol. 46, pp. 154–174, 2017.
- [24] O. J. Räsänen, G. Doyle, and M. C. Frank, “Unsupervised word discovery from speech using automatic segmentation into syllable-like units,” in *Proc. Interspeech*, 2015.
- [25] N. Vanhainen and G. Salvi, “Word discovery with beta process factor analysis,” in *Proc. Interspeech*, 2012.
- [26] N. Vanhainen and G. Salvi, “Pattern discovery in continuous speech using block diagonal infinite HMM,” in *Proc. ICASSP*, 2014.

- [27] T. Taniguchi, S. Nagasaka, and R. Nakashima, “Nonparametric Bayesian double articulation analyzer for direct language acquisition from continuous speech signals,” *IEEE Trans. Cogn. Developmental Syst.*, vol. 8, no. 3, pp. 171–185, 2016.
- [28] J. G. Wilpon and L. R. Rabiner, “A modified K-means clustering algorithm for use in isolated word recognition,” *IEEE Trans. Acoust. Speech Signal Process.*, vol. 33, no. 3, pp. 587–594, 1985.
- [29] L. R. Rabiner, J. G. Wilpon, and B.-H. Juang, “A segmental k-means training procedure for connected word recognition,” *AT&T Tech. J.*, vol. 65, no. 3, pp. 21–31, 1986.
- [30] B.-H. Juang and L. R. Rabiner, “The segmental K-means algorithm for estimating parameters of hidden Markov models,” *IEEE Trans. Acoust. Speech Signal Process.*, vol. 38, no. 9, pp. 1639–1641, 1990.
- [31] S. H. Shum, D. F. Harwath, N. Dehak, and J. R. Glass, “On the use of acoustic unit discovery for language recognition,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 24, no. 9, pp. 1665–1676, 2016.
- [32] 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.
- [33] H. Kamper, W. Wang, and K. Livescu, “Deep convolutional acoustic word embeddings using word-pair side information,” in *Proc. ICASSP*, 2016.
- [34] Y.-A. Chung, C.-C. Wu, C.-H. Shen, and H.-Y. Lee, “Unsupervised learning of audio segment representations using sequence-to-sequence recurrent neural networks,” *Proc. Interspeech*, 2016.
- [35] S. Settle and K. Livescu, “Discriminative acoustic word embeddings: Recurrent neural network-based approaches,” in *Proc. SLT*, 2016.
- [36] W. He, W. Wang, and K. Livescu, “Multi-view recurrent neural acoustic word embeddings,” in *Proc. ICLR*, 2017.
- [37] K. Audhkhasi, A. Rosenberg, A. Sethy, B. Ramabhadran, and B. Kingsbury, “End-to-end ASR-free keyword search from speech,” in *Proc. ICASSP*, 2017.
- [38] S. Settle, K. Levin, H. Kamper, and K. Livescu, “Query-by-example search with discriminative neural acoustic word embeddings,” in *Proc. Interspeech*, 2017.
- [39] J. Erickson, “Algorithms, etc.,” UIUC Lecture Notes, Jan. 2015.
- [40] P. Resnik and E. Hardisty, “Gibbs sampling for the uninitiated,” Tech. Rep., University of Maryland, College Park, MD, 2010.
- [41] D. Barber, *Bayesian Reasoning and Machine Learning*, Cambridge University Press, Cambridge, UK, 2013.
- [42] B. Kulis and M. I. Jordan, “Revisiting k-means: New algorithms via Bayesian nonparametrics,” in *Proc. ICML*, 2012.
- [43] B. Ludusan, M. Versteegh, A. Jansen, G. Gravier, X.-N. Cao, M. Johnson, and E. Dupoux, “Bridging the gap between speech technology and natural language processing: An evaluation toolbox for term discovery systems,” in *Proc. LREC*, 2014.
- [44] M. Versteegh, R. Thiollière, T. Schatz, X. N. Cao, X. Anguera, A. Jansen, and E. Dupoux, “The Zero Resource Speech Challenge 2015,” in *Proc. Interspeech*, 2015.
- [45] K. P. Murphy, “Conjugate Bayesian analysis of the Gaussian distribution,” 2007.