Unsupervised Mining of Acoustic Subword Units With Segment-level Gaussian Posteriorgrams

Haipeng Wang[1], *Tan Lee*[1], *Cheung-Chi Leung*[2], *Bin Ma*[2], *Haizhou Li*[2]

[1]Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong
[2]Institute for Infocomm Research, A*STAR, Singapore

{hpwang,tanlee}@ee.cuhk.edu.hk, {ccleung,mabin,hli}@i2r.a-star.edu.sg

Abstract

We consider the problem of unsupervised acoustic unit mining from unlabeled speech data. One typical method involves two steps: *unsupervised segmentation* and *segment clustering*. This paper proposes to improve *segment clustering* with segment-level Gaussian posteriorgram representation, which is generated by averaging the frame-level Gaussian posterior probabilities within each segment. Stacking together the segment-level Gaussian posteriorgrams of all the speech data, a Gaussian-by-segment data matrix is constructed. Given the Gaussian-by-segment matrix, we have the flexility to cluster either the Gaussian components or the segments into different acoustic unit categories. We have investigated both normalized cut and non-negative matrix factorization approaches on the data matrix for the clustering purpose. We carried out experiments to measure the quality of the clustering results with reference to manual phoneme labels. Experimental results show that the proposed methods consistently outperform a traditional vector quantization method and a Gaussian mixture model labeling method.

Index Terms: unsupervised acoustic unit mining, segment-level posteriorgrams, Gaussian-by-segment matrix, normalized cut, non-negative matrix factorization

1. Introduction

In recent years, the problem of unsupervised speech modeling has received increasing attention. It refers to the process of building a speech recognizer from unlabeled speech data. This is particularly important to low-resource languages, for which labeled data is very limited or even does not exist. Unsupervised speech modeling has been applied to, for examples, speech recognition [1], topic classification [2], spoken term detection [3], *etc*.

This study focuses on a fundamental issue in unsupervised speech modeling, namely automatic mining of acoustic units from unlabeled speech data. Specifically, we aim at discovering phoneme-like units in an unsupervised manner. Traditionally this was done in two steps [4, 5]. The first step is *unsupervised segmentation*, which exploits the temporal continuity of speech and determines the time boundaries that separate an utterance into variable-length segments. The second step is *segment clustering*, in which the acoustic similarity information is utilized to cluster the segments resulted from the first step. After segment clustering, each segment is labeled as a specific cluster, which corresponds to a discovered acoustic unit.

We propose a novel segment clustering approach based on segment-level Gaussian posteriorgram representation. Segment-level Gaussian posteriorgrams are generated by averaging frame-level Gaussian posterior probabilities within each segment. Pooling together all segment-level Gaussian posteriorgrams, we construct a Gaussian-by-segment data matrix. Based on the data matrix, we investigate two types of clustering algorithms: normalized cut [6] and non-negative matrix factorization [7]. Compared with spectral features, *e.g.*, MFCC, there are two advantages of using Gaussian posteriorgrams [8] for segment clustering. First, Gaussian posteriorgrams are more robust and more informative than spectral features. Second, using Gaussian posteriorgrams allows the flexibility to perform clustering on either speech segments or Gaussian components.

2. Related work

One commonly used approach to segment clustering is the conventional vector quantization [4, 9, 10]. Each segment is represented by a mean vector of the spectral features from its constituent frames. Another approach is the use of segmental Gaussian Mixture Model (SGMM) [2, 11], which represents each segment with a polynomial function of time. In [12], a GMM labeling approach was adopted. A GMM model is trained and used to label each segment with the index of the Gaussian component that scores the highest on the segment.

It is worth noting that the effectiveness of segment clustering can usually be improved with an iterative procedure [2, 10]. The iterative procedure refines the segment boundaries, segment labels, and the model parameters by alternately performing HMM training and decoding. The iterative refinement is guaranteed to increase the overall likelihood towards local optima [13], and its performance is sensitive to the initialization. Our study focuses on segment clustering, and can be viewed as a new initialization approach to the iterative procedure.

Apart from the segmentation-clustering scheme, other paradigms have also been proposed. In [14, 15], the successive state splitting algorithm was used to build the subword unit inventory. In [16], a non-parametric Bayesian approach was proposed to jointly learn the segment boundaries, segment labels, and HMM parameters. One highlight of this approach is the ability to automatic estimation of the unit inventory size. In [17, 18], the authors tackled this task by first discovering upper-level units (*e.g.*,words), and then conduct Gaussian component clustering with top-down constraints.

3. Data Representation

3.1. Segment-level Gaussian Posteriorgram Representation

Posterior features have been successfully applied to the problem of speech template matching [19, 20]. Given the spectral feature vector of a speech frame, its posterior feature vector consists of the posterior probabilities with respect to a set of predefined classes. Stacking together all posterior feature vectors of an utterance, we obtain the so-called *posteriorgram*. Gaussian

posteriorgrams are derived in the same manner by setting the predefined classes to be Gaussian components. Gaussian posteriorgram captures the temporal variation of frame-level posterior probabilities with respect to a set of Gaussian components.

Formally, we represent a speech utterance by $\boldsymbol{O} = [\boldsymbol{o}_1, \boldsymbol{o}_2, ..., \boldsymbol{o}_T]$, where $\boldsymbol{o}_t$ is the spectral feature vector of the t_{th} frame. We use $\{C_1, C_2, ..., C_M\}$ to denote the M Gaussian components. The corresponding Gaussian posteriorgram $\boldsymbol{GP}$ is,

$$\boldsymbol{GP} = [\boldsymbol{q}_1, \boldsymbol{q}_2, ..., \boldsymbol{q}_T], \tag{1}$$

where $\boldsymbol{q}_t$ is the posterior probability vector of the t_{th} frame:

$$\boldsymbol{q}_t = [p(C_1|\boldsymbol{o}_t), p(C_2|\boldsymbol{o}_t), ..., p(C_M|\boldsymbol{o}_t)]^T. \tag{2}$$

After segmentation, $\boldsymbol{O}$ is divided into K segments: $\boldsymbol{O} = [\boldsymbol{S}_1, \boldsymbol{S}_2, ..., \boldsymbol{S}_K]$, where $\boldsymbol{S}_k$ is the k_{th} segment. Let b_k and e_k denote the beginning and ending time of the k_{th} segment. Then,

$$\boldsymbol{S}_k = [\boldsymbol{o}_{b_k}, \boldsymbol{o}_{b_k+1}, ..., \boldsymbol{o}_{e_k}]. \tag{3}$$

With the segmentation boundaries and the Gaussian posteriorgram, the segment-level Gaussian posteriorgram $\boldsymbol{SGP} = [\tilde{\boldsymbol{q}}_1, \tilde{\boldsymbol{q}}_2, ..., \tilde{\boldsymbol{q}}_K]$ can be derived by:

$$\tilde{\boldsymbol{q}}_k = \frac{1}{e_k - b_k + 1} \sum_{t=b_k}^{e_k} \boldsymbol{q}_t. \tag{4}$$

Stacking the segment-level Gaussian posteriorgrams of all utterance forms a Gaussian-by-segment matrix $\boldsymbol{X}$ ($\boldsymbol{X} \in \mathbb{R}^{M \times N}$):

$$\boldsymbol{X} = [\underbrace{\tilde{\boldsymbol{q}}_1, \tilde{\boldsymbol{q}}_2, ..., \tilde{\boldsymbol{q}}_{K_1}}_{\text{utterance 1}}, \underbrace{\tilde{\boldsymbol{q}}_{K_1+1}, \tilde{\boldsymbol{q}}_{K_1+2}, ..., \tilde{\boldsymbol{q}}_{K_1+K_2}}_{\text{utterance 2}}, ..., \tilde{\boldsymbol{q}}_N], \tag{5}$$

where N is the total number of segments. One may consider the Gaussian-by-segment matrix $\boldsymbol{X}$ to be similar to the term-by-document matrix that is used in document clustering [21].

3.2. Duality of Gaussian Component Clustering and Segment Clustering

The matrix $\boldsymbol{X}$ has the dimension of $M \times N$, where M and N are both larger than the number of acoustic units to be discovered. With $\boldsymbol{X}$, we can perform clustering either on the row vectors or on the column vectors, which leads to Gaussian component clustering (GCC) or segment clustering (SC), respectively. GCC separates the M Gaussian components into different clusters which can establish a set of GMMs as acoustic models, while SC groups the N speech segments into different clusters which can induce corresponding segment transcriptions.

Typically $\boldsymbol{X}$ is very sparse, and the results of GCC and the results of SC are closely related. Segments are grouped into the same cluster because they have high posterior probabilities on several common Gaussian components. On the other hand, Gaussian components are grouped together because they appear with high posterior probabilities in many common segments. In the ideal case, we would expect that GCC and SC can directly lead to each other. In other words, the GMMs established by GCC can decode the speech data into acoustic unit sequences, that are similar to the SC results. And the segment transcriptions obtained by SC can be used to train acoustic models that represent similar distributions to the GMMs obtained by GCC.

For the purpose of segment clustering, besides direct column clustering on the data matrix $\boldsymbol{X}$, we can perform row clustering to partition Gaussian components, and then use the groups of Gaussian components to label the segment. This saves a lot of computation when the number of segments is much larger than that of the Gaussian components.

4. Clustering Algorithms

4.1. Normalized Cut approaches

We first introduce the GCC-based approach as shown below in Algorithm 1. Given the data matrix $\boldsymbol{X}$, we construct the similarity matrix by inner product, and apply the technique of normalized cut (NC) [22] for clustering. The clustering result forms a set of GMMs. After obtaining the R GMMs, each segment is labeled with the index of the GMM that scores the highest on it.

Algorithm 1: NC-GCC

Input: Data matrix $\boldsymbol{X}$, cluster number R.
Output: R GMMs and cluster membership of each segment.

1. Form inner-product similarity matrix $\boldsymbol{W}$:

$$\boldsymbol{W} = \frac{1}{N}\boldsymbol{X}\boldsymbol{X}^T. \tag{6}$$

2. Compute the normalized matrix $\boldsymbol{L}$:

$$\boldsymbol{L} = \boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}. \tag{7}$$

where $\boldsymbol{D}$ is the diagonal matrix with $\boldsymbol{D}_{ii} = \sum_j \boldsymbol{W}_{ij}$.
3. Derive matrix $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_R]$ that contains the eigenvectors corresponding to the R largest eigenvalues of $\boldsymbol{L}$.
4. Normalize every row of $\boldsymbol{U}$ to have unit L2-norm.
5. Perform k-means on the M row vectors of $\boldsymbol{U}$ to get the cluster membership.
6. Each cluster forms a GMM by assigning equal weights to the corresponding Gaussian components.
7. Label each segment with the index of the GMM that scores the highest on it.

Segment clustering (SC) can be done in the similar way as GCC using NC and the inner-product similarity matrix. However, in practice the number of segments N is often much larger than the number of Gaussian components M. In this case, directly computing the similarity matrix of segments becomes difficult because of the high memory cost. Thus we try to avoid the computation of the similarity matrix, and derive the eigenvector matrix $\boldsymbol{U}$ in a more efficient way. The implementation of segment clustering is formulated as in Algorithm 2. It is easy to show that the matrix $\boldsymbol{U}$ computed in step 4 of Algorithm 2 consists of the R eigenvectors of $\boldsymbol{D}^{-1/2}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{D}^{-1/2}$ with the largest eigenvalues.

4.2. Non-negative Matrix Factorization approaches

Non-negative Matrix Factorization (NMF) [7] has been successfully applied to clustering tasks [23]. Given the data matrix $\boldsymbol{X} \in \mathbb{R}^{M \times N}$, whose elements are all non-negative, NMF seeks to factorize $\boldsymbol{X}$ into non-negative matrix $\boldsymbol{F} \in \mathbb{R}^{M \times R}$ and non-negative matrix $\boldsymbol{G} \in \mathbb{R}^{N \times R}$. There are several possible objective functions that can be used for the factorization. In this paper, we use the squared Frobenius norm, with which the factorization problem is formulated as,

$$\min_{\boldsymbol{F},\boldsymbol{G}} \| \boldsymbol{X} - \boldsymbol{F}\boldsymbol{G}^T \|_F^2, \quad s.t.\ F \geq 0, G \geq 0. \tag{10}$$

To ensure the uniqueness and rigorous clustering interpretation of the factorization [24], the orthogonality constraint is added to the above formulation, *i.e.*,

$$\min_{\boldsymbol{F},\boldsymbol{G}} \| \boldsymbol{X} - \boldsymbol{F}\boldsymbol{G}^T \|_F^2, \quad s.t.\ F \geq 0, G \geq 0, G^TG = I. \tag{11}$$

Algorithm 2: NC-SC

Input: Data matrix $\boldsymbol{X}$, cluster number R.
Output: Cluster membership of each segment.
1. Compute $\boldsymbol{d} = \boldsymbol{X}^T(\boldsymbol{X1})$, where $\boldsymbol{1}$ is the all-ones vector. Let $\boldsymbol{D}$ be the diagonal matrix with $\boldsymbol{d}$ on its diagonal.
2. Transform $\boldsymbol{X}$ to $\tilde{\boldsymbol{X}}$:

$$\tilde{\boldsymbol{X}} = \boldsymbol{X}\boldsymbol{D}^{-1/2}. \quad (8)$$

3. Compute the similarity matrix of row vectors:

$$\tilde{\boldsymbol{W}} = \tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}^T. \quad (9)$$

3. Derive matrix $\tilde{\boldsymbol{U}} = [\tilde{\boldsymbol{u}}_1, \tilde{\boldsymbol{u}}_2, ..., \tilde{\boldsymbol{u}}_R]$ that contains the eigenvectors corresponding to the R largest eigenvalues of $\tilde{\boldsymbol{W}}$.
4. Compute $\boldsymbol{U} = \tilde{\boldsymbol{X}}^T\tilde{\boldsymbol{U}}$.
5. Normalize every row of $\boldsymbol{U}$ to have unit L2-norm.
6. Perform k-means on the N row vectors of $\boldsymbol{U}$ to get the cluster memberships of the segments.

This problem can be solved using a multiplicative method as derived in [24]. To initialize $\boldsymbol{G}$, we first run k-means clustering on the mean vectors of the segments. Then $\boldsymbol{F}$ is initialized as the centroids of the k-means results, and $\boldsymbol{G}$ is constructed from the assignment of the k-means results. To smooth $\boldsymbol{G}$, we simply add a small constant to $\boldsymbol{G}$. This initialization is similar to the one suggested in [25]. After factorization, the matrix $\boldsymbol{G}$ is used to determine the membership of each segment. We summarize the implementation as in Algorithm 3.

Algorithm 3: NMF-SC

Input: Data matrix $\boldsymbol{X}$, cluster number R
Output: Cluster membership of each segment
1. Initialize $\boldsymbol{F}$ and $\boldsymbol{G}$.
2. Solve Equation 11 by alternately using the following updating rules:

$$\boldsymbol{F}_{mr} \leftarrow \boldsymbol{F}_{mr}\frac{(\boldsymbol{X}\boldsymbol{G})_{mr}}{(\boldsymbol{F}\boldsymbol{G}^T\boldsymbol{G})_{mr}} \quad (12)$$

$$\boldsymbol{G}_{rn} \leftarrow \boldsymbol{G}_{rn}\sqrt{\frac{(\boldsymbol{X}^T\boldsymbol{F})_{rn}}{(\boldsymbol{G}\boldsymbol{G}^T\boldsymbol{X}^T\boldsymbol{F})_{rn}}} \quad (13)$$

3. Determine the membership. Specifically, the n_{th} segment is assigned to cluster c if $c = \arg\max_r \boldsymbol{G}_{rn}$.

5. Experimental Setup

5.1. Data processing

Experiments were carried out on the $stories$ part of the OGI Multi-language Telephone Speech Corpus [26]. The data involves six languages, namely English (EN), German (GE), Hindi (HI), Japanese (JA), Mandarin (MA) and Spanish (SP). Experiments were conducted on the data of each language independently. Manual phoneme transcriptions were provided for the $stories$ part of this corpus. Since silence segments are relatively easier to be detected, the clustering results of silence segments are much higher than others. So the quite frequent occurrences of silence segments would significantly bias the results. Thus we removed the silence frames according to the manual transcriptions for all subsequent experiments.

The spectral feature vector is composed of 39-dimensional MFCC. The MFCC features were post-processed by utterance-level mean and variance normalization (MVN) and vocal tract length normalization (VTLN). For unsupervised segmentation, we used the approach as described in our previous work [12].

5.2. Evaluation metric

The clustering results were measured with reference to frame-level manual phoneme labels. Details of the reference phoneme sets can be found in [27]. For each language, the number of clusters R was made equal to the number of phonemes in the corresponding manual transcriptions.

Two evaluation metrics were used for the evaluation: 1) F-measure; 2) normalized mutual information (NMI). Details of the evaluation metrics can be found in [28]. These two metrics are widely used for clustering tasks. For both metrics, the larger the value, the better the clustering performances.

6. Experimental Results and Analysis

6.1. Baseline approaches

As baseline approaches, the conventional vector quantization (VQ) approach [4, 10] and the GMM labeling approach [12] were implemented. The VQ approach was implemented with two steps: 1) representing each segment by the mean vector of its MFCC feature vectors; 2) running k-means on the segment mean vectors to obtain the clustering memberships. Note that in this paper, k-means was always implemented using the centroid splitting strategy. We found that this strategy provided more reliable performances than random initialization. The GMM labeling approach was implemented with three steps: 1) training a GMM with the component number to be the target cluster number; 2) using the GMM to score each segment; 3) labeling each segment with the index of the gaussian component which scores the highest on the segment.

The VQ approach serves as the basline approach to the NC-SC and NMF-SC approaches because they all perform segment clustering directly, while the GMM labeling approach is baseline to the NC-GCC approach because they both first build acoustic models and then assign labels to the segments by computing the acoustic likelihood. Table 1 shows the baseline performances. As can be seen, VQ generally performed better than GMM labeling approach. This may be explained by the difference between the objectives of these two approaches. While the objective of VQ is to find good partition of the segments, the GMM training aims at maximizing the likelihood.

Table 1: Performances (F-measure/NMI) of VQ and GMM labeling approaches for the six evaluation languages. The last row shows the averaged (Avg.) performances.

	VQ	GMM Labeling
EN	0.264 / 0.284	0.244 / 0.266
GE	0.239 / 0.269	0.213 / 0.256
HI	0.261 / 0.296	0.243 / 0.287
JA	0.327 / 0.283	0.273 / 0.270
MA	0.233 / 0.274	0.209 / 0.253
SP	0.308 / 0.305	0.261 / 0.300
Avg.	0.270 / 0.285	0.241 / 0.272

Table 2: Performances (F-measure/NMI) of NC-GCC approach. M is the number of the Gaussian components. The last row shows the averaged (Avg.) performances.

M	128	256	512	768	1024	2048	3072	4096
EN	0.214 / 0.268	0.233 / 0.272	0.236 / 0.286	0.237 / 0.293	0.241 / 0.301	0.248 / 0.303	0.252 / **0.306**	**0.258** / 0.304
GE	0.200 / 0.253	0.215 / 0.270	0.210 / 0.270	0.225 / 0.278	0.216 / 0.275	**0.234** / 0.283	0.225 / **0.287**	0.229 / **0.287**
HI	0.199 / 0.262	0.226 / 0.295	0.248 / 0.302	0.249 / 0.306	0.254 / 0.310	0.257 / **0.319**	**0.262** / 0.318	0.257 / 0.317
JA	0.295 / 0.288	0.309 / 0.298	0.314 / 0.305	**0.329 / 0.313**	0.307 / 0.308	0.310 / 0.303	0.324 / 0.310	0.311 / 0.310
MA	0.200 / 0.253	0.219 / 0.269	0.223 / 0.267	0.225 / 0.271	0.219 / 0.271	0.222 / 0.276	**0.230** / 0.277	0.228 / **0.279**
SP	0.260 / 0.294	0.267 / 0.297	0.287 / 0.317	0.285 / 0.325	0.279 / 0.325	0.294 / 0.333	0.295 / 0.333	**0.339 / 0.340**
Avg.	0.228 / 0.270	0.245 / 0.284	0.253 / 0.291	0.258 / 0.298	0.253 / 0.298	0.261 / 0.303	0.265 / 0.305	**0.270 / 0.306**

6.2. The proposed approaches

In this section, we first examined the NC-GCC approach (Algorithm 1). Its performance was evaluated as a function of the number of Gaussian components which is denoted by M in the Gaussian-by-segment representation as in Section 3. Table 2 shows the results. Bold values indicate the best performances achieved by NC-GCC for each language. When M goes from 128 to 768, a larger set of Gaussian components would lead to better performances for all the six languages. Different languages might have different optimal choices of M to get the best performances. On average, a larger M tends to provide more reliable performances. Compared with the averaged performances of GMM labeling, NC-GCC provides relative improvements of 12.0% on F-measure and 12.5% on NMI.

We then examined the performances of NC-SC and NMF-SC approaches by setting M to 4096. Table 3 shows the results. Bold values indicate the relatively better performances. Roughly speaking, NC-SC tended to give better NMI values, while NMF-SC tended to give better F-measure values. Compared with the performances of VQ shown in Table 1, these two approaches performed consistently better on both metrics. Compared with the performances of NC-GCC shown in Table 2, NC-SC and NMF-SC performed better except for the NMI values on Japanese and Spanish data. This might imply that even using a large set of GMM, its Gaussian components were still not well separable in terms of phonemes. This is in accordance with the observation that different phoneme acoustic models may share some common Gaussian components.

Table 3: Performances (F-measure/NMI) of NC-SC and NMF-SC approaches. M is set to 4096.

	NC-SC	NMF-SC
EN	**0.288 / 0.314**	0.276 / 0.305
GE	0.256 / **0.293**	**0.261** / 0.283
HI	**0.309 / 0.335**	0.265 / 0.300
JA	0.341 / 0.303	**0.374 / 0.311**
MA	0.252 / **0.290**	**0.264** / 0.279
SP	0.365 / **0.337**	**0.391** / 0.331
Avg.	0.302 / **0.312**	**0.305** / 0.302

To gain a clear impression of the clustering results, we analyze the confusion matrix between the phonemes and the cluster indexes. Fig. 1 shows the confusion matrix derived by the NC-SC approach on English data. Phonemes are divided into broad phonetic classes, namely vowel (vwl), diphthong (dip), semi-vowel (smv), stop (stp), fricative (frc) and nasal (nas) [29]. The silence class has been removed according to manual labels as introduced in Sec. 5.1. The first three classes are often merged into one big class as *vowel*. As can be seen, promising correspondences were observed between the clustering results and the phoneme units. However within each class, the phonemes are quite confusable. Significant inter-class confusion appears between vowel and dipthong, and between stop and fricative. Another observation is that compared with other phonetic classes, the phonemes in the class *stop* (*b, d, g, k, p, t*) are more difficult to be found out by the current approaches.

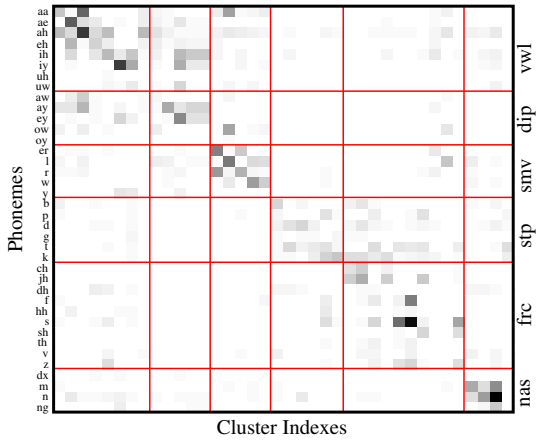

Figure 1: Confusion Matrix between phonemes and cluster indexes on English data. NC-SC approach is used to derive the confusion matrix. The darkness is scaled to the value of the corresponding element in the confusion matrix.

7. Conclusion and Future Work

In this paper, we have proposed to discover the phoneme-like acoustic units using the segment-level Gaussian posteriorgram representation. By representing each segment with the averaged Gaussian posterior probability vector, a Gaussian-by-segment data matrix is formed. Based on the Gaussian-by-segment data matrix, we have shown how to conduct clustering to discover the phoneme-like units. Three clustering algorithms, namely NC-GCC, NC-SC and NMF-SC have been investigated. Experiments were carried out on the OGI Multi-language Corpus. Experimental results show that our approaches outperform the baseline VQ appraoch and GMM labeling approach. For future work, we may consider more sophisticated clustering algorithms and automatic unit number estimation.

8. Acknowledgement

The authors would like to thank Dau-Cheng Lyu for providing the phoneme transcriptions [27] of the OGI Multi-language Telephone Speech Corpus. This research is partially supported by the CUHK-PKU Joint Centre for Intelligence Engineering, and the General Research Funds (Ref: 414010 and 413811) from the Hong Kong Research Grants Council.

9. References

[1] R. Singh, B. Raj, and R. Stern, "Automatic generation of subword units for speech recognition systems," *IEEE Trans. SAP*, vol. 10, no. 2, pp. 89–99, 2002.

[2] M.-H. Siu, H. Gish, A. Chan, and W. Belfield, "Improved topic classification and keyword discovery using an HMM-based speech recognizer trained without supervision," in *Proc. INTERSPEECH*, 2010, pp. 2838–2841.

[3] H. Wang, T. Lee, C.-C. Leung, B. Ma, and H. Li, "Using parallel tokenizers with DTW matrix combination for low-resource spoken term detection," in *Proc. ICASSP*, 2013.

[4] C.-H. Lee, F. Soong, and B.-H. Juang, "A segment model based approach to speech recognition," in *Proc. ICASSP*, 1988, pp. 501–541.

[5] M. Bacchiani and M. Ostendorf, "Joint lexicon, acoustic unit inventory and model design," *Speech Communication*, vol. 29, no. 2, pp. 99–114, 1999.

[6] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. PAMI*, vol. 22, no. 8, pp. 888–905, 2000.

[7] D. Seung and L. Lee, "Algorithms for non-negative matrix factorization," *Advances in neural information processing systems*, vol. 13, pp. 556–562, 2001.

[8] Y. Zhang and J. Glass, "Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams," in *Proc. ASRU*, 2009, pp. 398–403.

[9] B. Ma, D. Zhu, and H. Li, "Acoustic segment modeling for speaker recognition," in *Proc. ICME*, 2009, pp. 1668–1671.

[10] J. Reed and C.-H. Lee, "A study on music genre classification based on universal acoustic models," in *Proc. ISMIR*, 2006, pp. 89–94.

[11] H. Gish and K. Ng, "A segmental speech model with applications to word spotting," in *Proc. ICASSP*, vol. 2, 1993, pp. 447–450.

[12] H. Wang, C.-C. Leung, T. Lee, B. Ma, and H. Li, "An acoustic segment modeling approach to query-by-example spoken term detection," in *Proc. ICASSP*, 2012, pp. 5157–5160.

[13] T. Hazen, M.-H. Siu, H. Gish, S. Lowe, and A. Chan, "Topic modeling for spoken documents using only phonetic information," in *Proc. ASRU*, 2011, pp. 395–400.

[14] B. Varadarajan and S. Khudanpur, "Automatically learning speaker-independent acoustic subword units," in *Proc. INTERSPEECH*, 2008, pp. 1333–1336.

[15] H. Singer and M. Ostendorf, "Maximum likelihood successive state splitting," in *Proc. ICASSP*, vol. 2, 1996, pp. 601–604.

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

[17] A. Jansen and K. Church, "Towards unsupervised training of speaker independent acoustic models," in *Proc. INTERSPEECH*, 2011, pp. 1693–1696.

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

[19] G. Aradilla, J. Vepa, and H. Bourlard, "Using posterior-based features in template matching for speech recognition," in *Proc. INTERSPEECH*, 2006, pp. 1186–1189.

[20] T. Hazen, W. Shen, and C. White, "Query-by-example spoken term detection using phonetic posteriorgram templates," in *Proc. ASRU*, 2009, pp. 421–426.

[21] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American society for information science*, vol. 41, no. 6, pp. 391–407, 1990.

[22] A. Ng, M. Jordan, Y. Weiss *et al.*, "On spectral clustering: Analysis and an algorithm," *Advances in neural information processing systems*, vol. 2, pp. 849–856, 2002.

[23] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," in *Proc. 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, 2003, pp. 267–273.

[24] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal non-negative matrix T-factorizations for clustering," in *Proc. 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006, pp. 126–135.

[25] C. Ding, T. Li, and W. Peng, "On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing," *Computational Statistics and Data Analysis*, vol. 52, no. 8, pp. 3913–3927, 2008.

[26] Y. Muthusamy, R. Cole, and B. Oshika, *The OGI multi-language telephone speech corpus*, 1994.

[27] S. Siniscalchi, D.-C. Lyu, T. Svendsen, and C.-H. Lee, "Experiments on cross-language attribute detection and phone recognition with minimal target-specific training data," *IEEE Trans. ASLP*, vol. 20, no. 3, pp. 875–887, 2012.

[28] C. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*. Cambridge University Press Cambridge, 2008.

[29] T. Sainath, D. Kanevsky, and B. Ramabhadran, "Broad phonetic class recognition in a hidden Markov model framework using extended Baum-Welch transformations," in *Proc. ASRU*, 2007, pp. 306–311.