An Evaluation of Unsupervised Acoustic Model Training for a Dysarthric Speech Interface

Oliver Walter, Vladimir Despotovic,
Reinhold Haeb-Umbach

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

Jort F. Gemmeke, Bart Ons,
Hugo Van hamme

ESAT-PSI
KU Leuven, Belgium
jort.gemmeke@esat.kuleuven.be

Abstract

In this paper, we investigate unsupervised acoustic model training approaches for dysarthric-speech recognition. These models are first, frame-based Gaussian posteriorgrams, obtained from Vector Quantization (VQ), second, so-called Acoustic Unit Descriptors (AUDs), which are hidden Markov models of phone-like units, that are trained in an unsupervised fashion, and, third, posteriorgrams computed on the AUDs. Experiments were carried out on a database collected from a home automation task and containing nine speakers, of which seven are considered to utter dysarthric speech. All unsupervised modeling approaches delivered significantly better recognition rates than a speaker-independent phoneme recognition baseline, showing the suitability of unsupervised acoustic model training for dysarthric speech. While the AUD models led to the most compact representation of an utterance for the subsequent semantic inference stage, posteriorgram-based representations resulted in higher recognition rates, with the Gaussian posteriorgram achieving the highest slot filling F-score of 97.02%.

Index Terms: unsupervised learning, acoustic unit descriptors, dysarthric speech, non-negative matrix factorization

1. Introduction

It is often said that a speech interface, e.g., to control household devices, is particularly helpful for physically challenged people [1, 2]. Unfortunately, a significant fraction of this group of users also suffers from speech impairments, such as dysarthria, a motor speech disorder, which makes their speech sound quite differently compared to speech uttered by people without speaking impairments. As a consequence, off-the-shelf automatic speech recognition (ASR) systems exhibit unacceptably high error rates for dysarthric speech [3]. The deviations from normal speech utterances are usually quite severe and conventional speaker adaptation approaches, such as Maximum-a-posteriori (MAP) or Maximum Likelihood Linear Regression (MLLR) adaption are able to compensate for these deviations to adapt the acoustic models to some extent to reduce the error rates for impaired speech. A significant amount of research has therefore been devoted to the characterization and recognition of dysarthric speech [4, 5, 6, 7, 8, 9, 10].

An alternative to the adaptation of a speaker-independent system is the training of a speaker-dependent recognizer. This asks for the availability of labeled training data, i.e., recordings of the user's spoken utterances and the corresponding text files, together with a pronunciation lexicon . In particular in the case of dysarthric speech, pronunciations can be quite different from the standard [11, 12, 13, 14], such that the appropriateness of canonical transcriptions is questionable.

In order to avoid the effort for providing an appropriate pronunciation lexicon and transcribing the training data, the ALADIN project follows a different route [15, 16]. It is concerned with the development of a self-learning vocal interface for a home automation system, where the learning of the acoustic models is done in a "zero-resource" scenario, requiring neither the transcription of the training data nor a pronunciation lexicon. The user still has to follow a training session, but this is only to learn the mapping of the user's commands, which he can choose freely, to the action to be carried out in the home automation system. To this end, only weak supervision is required – an action label assigned to an utterance – while no literal transcription of the user's utterance is needed. Thus the system is maximally adapted to the particular artifacts of the user's (dysarthric) speech and to the preferred wording of the user.

This approach, while attractive to the user, poses several challenges, such as unsupervised acoustic model training and the learning of the mapping between the user's utterance and the actions to be performed, using only weak supervision. While the latter has been discussed in [17], where a Non-negative Matrix Factorization (NMF) based approach was employed to effectively solve the semantic inference problem, this paper is concerned with the first issue. In the past years several unsupervised acoustic model training methods have been developed, including Gaussian posteriorgrams [18], hidden Markov model-based self-organising units [19], and non-parametric Bayesian estimation of HMMs [20]. In [21] we have adopted a hierarchical approach, which has originally been developed for the semantic analysis of the audio track of multimedia data [22], to the unsupervised learning of speech representations. On the first hierarchic layer, we learned Acoustic Unit Descriptors (AUDs), phone-like units, which are similar to the HMM-based self-organising units in [19]. The second layer is concerned with the discovery of word-like units, which manifest themselves as recurring sequences of AUDs. This approach showed very good performance on the TiDigits corpus with recognition rates coming close to a supervised training [21].

In this paper we discuss the suitability of unsupervised

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

Vladimir Despotovic is supported by an Erasmus Mundus Action 2 scholarship within the EUROWEB scholarship programme.

The research of Jort F. Gemmeke and Bart Ons was funded by the IWT-SBO project ALADIN (contract 100049)

acoustic learning approaches for dysarthric speech. We concentrate on two representative approaches, Gaussian posteriorgrams computed from MFCC features, and the suprasegmental AUDs. Furthermore, we will also evaluate posteriorgrams of AUDs.

The paper is organized as follows: In the next section we give a brief overview of the vocal user interface developed in the ALADIN project. In Section 3 we introduce the different feature representations under investigation. The GMM based posteriorgram is described in subsection 3.1 while the AUD based representation is given in subsection 3.2. The database is presented in Section 4, followed by the section on experimental results. We finish with a discussion and conclusion in section 6.

2. Vocal User Interface

This section gives a brief overview of the architecture of the Vocal User Interface (VUI) that has been developed in the framework of the ALADIN project, e.g. for the purpose of controlling a home automation system. The main target group are people who suffer from speech impairment, hence the system should be able to adapt to voice pathologies [23]. The system is also designed to learn and adapt to unconstrained spoken commands, where the user can formulate a command in the words of his choice.

The mapping between the voice command and an action on the device's user interface is learned during the training phase. An action is represented by a semantic frame, a data structure that is composed of slots, which in turn contain slots or values. For example, a semantic frame can contain the slots <device> and <action>, with the corresponding values <television, radio> and <on, off>, respectively.

During the training phase, recurrent acoustic patterns are determined from the spoken commands using NMF [15]. NMF decomposes a non-negative matrix that represents training data into two lower rank matrices, i.e., a dictionary matrix containing recurrent acoustic patterns and a matrix of activations of these patterns. This process is weakly supervised by augmenting the representation of the user's utterance with labels indicating the slot values the utterance is referring to.

During the decoding process the vector describing the user's utterance is again decomposed via NMF, and the decomposition is compared with the trained dictionary, which included the grounding information. By finding the closest match, an estimate of the slot values is obtained. This recognized command is represented by the semantic frame, and finally sent to the target device [24].

As the mapping of the input utterance to a user command is carried out using NMF, each utterance of the user has to be represented by a vector of fixed size. The compilation and size of these vectors depend on the acoustic representation of the utterance, which will be described in the following section.

3. Acoustic Representations

The first step in the audio processing chain is the extraction of Mel Frequency Cepstral Coefficient (MFCC) feature vectors, which are augmented with the log energy and first and second-order temporal difference features to arrive at a 39-dimensional feature vector. Note that cepstral mean and variance normalization is carried out per utterance.

The following different representations of the input dysarthric speech were learned.

3.1. Gaussian Posteriorgrams

Here, the MFCC feature vector is transformed into a vector of posterior probabilities of Gaussians forming a codebook, using soft vector quantization. To this end, a 100 component full-covariance Gaussian Mixture Model is trained on MFCC vectors. The code book training starts off from a single cluster describing all training data. It is then split along the dominant eigenvector of its covariance matrix, followed by iterations of the Expectation Maximization algorithm. This process is repeated until a desired number of mixture components is obtained.

The posterior probability of each Gaussian mixture component is then computed for each MFCC vector. From the posteriors so-called histograms of acoustic co-occurrences (HACs) are constructed. The HAC is an estimate of the joint posterior probability of two acoustic events happening at a predefined time lag [25, 26].

3.2. AUD based representation

Acoustic subword units are meant to capture acoustically consistent phenomena and will be referred to as *acoustic unit descriptors* (AUDs) [27]. They represent similar recurring sequences of feature vectors.

The discovery of AUDs is done in two steps. In the initialization step input speech is segmented and the segments are clustered to generate an initial transcription of the input speech in terms of sequences of segment labels. The second step is the iterative training of hidden Markov models (HMMs) for the discovered clusters. The block diagrams of these steps are depicted in figure 1 and will be briefly described in the following. For a more detailed description the reader is referred to [21].

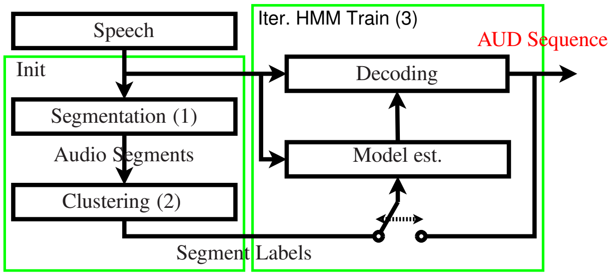

Figure 1: AUD discovery algorithm

3.2.1. Segmentation

In the segmentation step the speech input is segmented into consistent speech segments according to a local distance measure between the mean representative of the current segment and the next feature vector. If the value of the local distance measure is greater than a threshold, a new segment is created. A constraint on the minimum segment length is used to prevent the generation of short segments. These parameters are chosen such that the average segment length corresponds to the expected length of a phoneme. As the local distance measure the cosine distance is employed.

3.2.2. Clustering

In the clustering step the segments from the segmentation step are grouped to obtained clusters according to acoustic consistency. Clustering is carried out on a (sparse) adjacency matrix derived from the distances between representative segments and all other segments. The (length normalized) dynamic time warping distance between two segments is employed, using the cosine distance as the local distance measure. The representative segments are chosen according to the kmeans++ initialization [28, 29]. Finally the unsupervised graph clustering algorithm by Newman [30] is used for clustering. As output for each utterance, a sequence of cluster labels is assigned which will serve as an initial label sequence for the iterative training of the AUD models.

3.2.3. Iterative AUD HMM Training

In the iterative HMM training step, the cluster labels are interpreted to be AUD labels and used as an initial transcription $T_d^{(0)}$ for the d-th input speech utterance. For each AUD we define a 3-state left-to-right HMM with Gaussian mixture output densities and refer to the set of all AUD models as $\Lambda_{\mathcal{A}}$. We use a zerogram language model to connect the AUDs. The HMM parameters $\Lambda_{\mathcal{A}}$ and the transcriptions T_d are updated by alternating between re-estimation of the AUD parameters, eq. (1), and decoding of the input speech, eq. (2) [19, 22]:

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

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

Here, i is the iteration index and $\mathbf{X}_d$ denotes the MFCC feature vector sequence of the d-th utterance. D is the total number of utterances.

3.2.4. Mapping of utterance to fixed-length vector

The AUD sequences, which describe the utterance, are not directly amenable to NMF. They need to be mapped to a representation of fixed dimension, in which linearity holds, i.e., that the utterance-level speech representation is approximately equal to the sum of the speech representations of the acoustic patterns it contains [25]. This vector is created by replacing each AUD in the recognized sequence of AUDs of an utterance by an indicator vector, where the element of the vector representing the AUD is set to one and all other elements to zero. Using all vectors of an utterance a histogram of occurrences and co-occurrences is built and used as input to the NMF.

4. DOMOTICA-3 Database

In this work, we employ the DOMOTICA-3 database that has been collected in the framework of the ALADIN project [16]. The DOMOTICA-3 database is a collection of recordings of Flemish dysarthric speakers controlling a home automation system. Recordings were collected in two phases. During the first phase users were asked to command 26 distinct actions in a simulated 3D computer animation of a home environment, in order to ensure an unbiased choice of words and grammar by the user. In the second phase speakers were recorded reading these commands to obtain enough repetitions of each spoken command.

In this study only speakers that have uttered at least five repetitions of each command were included. They will be referred to by unique id's 17, 28, 29, 30, 31, 34, 35, 41 and 44. The total number of utterances per speaker was in the range of 151 to 350, with an average of 238. The total size of the database is about 4 hours of speech. Speech intelligibility scores were obtained for all speakers by analysing their recorded speech using an automated tool [31], which led to the conclusion that all except two speakers (id's 17 and 44) were considered to utter dysarthric speech.

5. Experiments

We performed our experiments on the DOMOTICA-3 database using the NMF based command recognition framework. We used the following setups:

1. Gaussian posteriorgram based representation
2. AUD sequence based representation
3. AUD posteriorgram based representation
4. Phoneme sequences derived using a speaker independent general acoustic phoneme model

For setup 1 the results were produced by using 100 full-covariance Gaussians trained on all speech material available for that speaker as described in subsection 3.1. From these the histogram of occurrences and the HACs at four different lags, 2, 5, 9 and 20 frames, were computed. The resulting vector to be forwarded to the NMF-based semantic inference stage was of size $4 \times 100^2 + 100 = 40100$.

For setup 2 we learned speaker dependent acoustic models of the AUDs with and additional silence HMM on the speech material available for that speaker. Each state has one 39-dimensional Gaussian emission density with a diagonal covariance matrix. A zerogram language model was used. The number of AUDs per speaker varied between 22 and 98 AUDs, depending on the outcome of the unsupervised clustering algorithm described in 3.2.2. We then produced lattices over AUDs for each audio recording and used the algorithm described in [32] to learn a 4-gram language model in an unsupervised way over the sequence of AUDs and output a refined sequence of AUDs. We then used the discovered sequence of AUDs, with silence HMMs removed, as input to the command recognition algorithm by computing the histogram of occurrences and a HAC with lag 1. The resulting vector was on average of size $50^2 + 50 = 2550$.

For setup 3 two different posteriorgram representations were derived using the acoustic models of the discovered AUDs. The first representation (AUD/GMM) was derived by concatenating the Gaussians learned for each state of the HMM to one GMM to again calculate a posteriorgram similar to setup 1. Each mixture component was assigned the same weight. For the second representation (AUD/HMM) we used the posterior probabilities of being in a certain state of the HMM calculated with the Forward-Backward algorithm. In both cases we did sum the probabilities of all the states belonging to one HMM to generate an AUD based posteriorgram, similar to a phoneme posteriorgram. Vectors in which silence had the highest probability were removed. From the resulting AUD based posteriorgrams a histogram of occurrences and HACs at four different lags, 2, 5, 9 and 20 frames, were computed. The resulting vector, that was input to the command recognition framework, was on average of size $4 \times 50^2 + 50 = 10050$.

For setup 4, which served as a baseline, a pre-trained speaker independent general acoustic model and a zerogram language model were used to decode the audio recordings and produce lattices for each recording. The speaker independent general acoustic model was trained on Dutch speech. A sequence of phonemes was then generated using again the algorithm of [32] and learning a 4-gram language model in an unsupervised way. The sequence of phonemes was used to compute

Table 1: F-scores of the different setups; for explanation see text.

Speaker	44	17	34	31	29	28	35	30	41	Average
# Utterances	166	350	335	235	181	214	284	223	151	238
# AUDs	98	56	59	38	58	30	53	22	32	50
Setup 1: Gaussian posteriorgrams	99.35	99.74	98.76	92.09	99.39	93.99	97.53	93.26	97.95	97.02
Setup 2: AUD sequences	95.49	96.92	90.38	79.88	92.74	76.18	94.31	85.31	90.78	89.49
Setup 3: AUD/HMM posteriorgrams	93.03	96.06	91.30	86.48	95.00	79.99	91.38	88.66	93.48	90.75
Setup 3: AUD/GMM posteriorgrams	96.29	99.24	97.67	90.50	98.12	89.51	95.65	93.22	94.58	95.30
Setup 4: Phoneme recognizer	90.75	87.17	78.69	66.32	84.84	54.23	80.99	56.16	64.81	74.70

a HAC-based representation in the same way as was done with the AUDs above and then forwarded to the command recognition framework.

As a performance measure the slot $F_{\beta=1}$ score of the action recognition was used, which is the harmonic mean of slot precision and slot recall. For its computation a five-fold cross validation procedure was used as described in [17], with four blocks for training and one for testing.

Figure 2 shows the recognized AUD sequences for two utterances of the sentence “ALADIN hoofdeinde op stand 1” by speaker 30. Note that the name assigned to an AUD is, of course, arbitrary, as no phonetic interpretation can be given to it in an unsupervised training. Same recognized AUD sequences are marked in color. The similarity between the two sequences is striking. The AUDs can be interpreted as phone-like units and sequences of it as word-like entities. Differences between the two recognized sequences can be viewed as recognition errors or pronunciation variations.

Example 1:
AJ AE AA AC B AF F BJ C H H AH AB AF AC AD BJ C AC F F AD E I AC H AH AB AF F
Example 2:
AJ AE AA AC B AF F BJ C H AH AB AF AC AD E C H BB F AD E I AC H AH AB AF F

Figure 2: Recognized AUD sequences of two utterances of the same sentence spoken by speaker 30

Figure 3 shows the posteriorgrams of the example utterances obtained by the Forward-Backward algorithm on the AUD/HMMs. A certain similarity can again be observed between the two posteriorgrams, indicating that posteriograms on AUDs are also a consistent representation of an utterance.

Table 1 shows the slot F-scores of the individual speakers and the average over all speakers (weighted by the relative number of utterances per speaker) for the different setups. Additionally the number of utterances per speaker and the number of discovered acoustic units is shown.

The table is ordered so that the left most speaker has the highest intelligibility score while the score decreases when going to the right. The Speakers 44 and 17 are considered as normal speakers.

6. Discussion and Conclusion

First of all, the results clearly show that all unsupervised modeling approaches deliver significantly better slot F-scores than the speaker-independent phoneme recognition baseline, showing the suitability of unsupervised acoustic model training for dysarthric speech. Of the unsupervised techniques, the Gaussian posteriorgrams come out first, followed by the posteriorgrams computed from AUDs, while the AUDs themselves performed clearly worse. It seems that the posteriorgrams are able to capture more information relevant for semantic inference than is available in the mere presence or absence of an AUD.

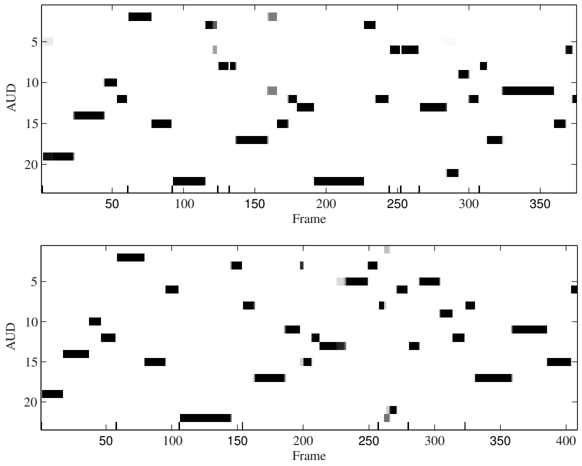

Figure 3: AUD/HMM: Example 1 (top), Example 2 (bottom)

Note, however, that the Gaussian posteriorgrams are the most expensive description of the utterance in terms of the vector length forwarded to NMF, which was 40100. The AUD-based posteriorgrams are coded in a vector of only one quarter of the size, and the AUDs in a vector of approximately one fifteenth of the size.

Another interesting observation is that the slot F-score does not monotonously decrease with the intelligibility of the speech. While the speakers are ordered in the table according to decreasing measured speech intelligibility score, the slot F-scores obtained from the various ASR variants are not ordered in the same way. Especially the speakers 29, 35 and 41 achieve higher results than one would expect from their rank according to speech intelligibility. One reason for this might be, that consistency in utterances is more important for the recognition task than intelligibility and that it is not measured in the intelligibility score.

While a definite statement is certainly not possible from this limited dataset, these results nevertheless are encouraging, as they point to the potential of self-learning vocal user interfaces: not only are they superior to off-the-shelf speaker-independent ASR solutions, unsupervised learning approaches have the potential of performing on dysarthric speech as well as on normal speech.

7. References

[1] J. Noyes and C. Frankish, "Speech recognition technology for individuals with disabilities," *Augmentative and Alternative Communication*, vol. 8, no. 4, pp. 297–303, 1992.

[2] M. S. Hawley, S. P. Cunningham, P. D. Green, P. Enderby, R. Palmer, S. Sehgal, and P. O'Neill, "A voice-input voice-output communication aid for people with severe speech impairment," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 21, no. 1, pp. 23–31, 2013.

[3] M. S. Hawley, P. Enderby, P. Green, S. Cunningham, S. Brownsell, J. Carmichael, M. Parker, A. Hatzis, P. O'Neill, and R. Palmer, "A speech-controlled environmental control system for people with severe dysarthria," *Medical Engineering & Physics*, vol. 29, no. 5, pp. 586–593, 2007.

[4] K. Rosen and S. Yampolsky, "Automatic speech recognition and a review of its functioning with dysarthric speech," *Augmentative and Alternative Communication*, vol. 16, no. 1, pp. 48–60, 2000.

[5] H. Christensen, S. Cunningham, C. Fox, P. Green, and T. Hain, "A comparative study of adaptive, automatic recognition of disordered speech." in *INTERSPEECH*, 2012.

[6] K. T. Mengistu and F. Rudzicz, "Comparing humans and automatic speech recognition systems in recognizing dysarthric speech," in *Advances in Artificial Intelligence*. Springer, 2011, pp. 291–300.

[7] M. Hasegawa-Johnson, J. Gunderson, A. Penman, and T. Huang, "HMM-based and SVM-based recognition of the speech of talkers with spastic dysarthria," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 3. IEEE, 2006, pp. III–III.

[8] P. Green, J. Carmichael, A. Hatzis, P. Enderby, M. S. Hawley, and M. Parker, "Automatic speech recognition with sparse training data for dysarthric speakers." in *INTERSPEECH*, 2003.

[9] H. V. Sharma and M. Hasegawa-Johnson, "State-transition interpolation and MAP adaptation for HMM-based dysarthric speech recognition," in *Proceedings of the NAACL HLT 2010 Workshop on Speech and Language Processing for Assistive Technologies*. Association for Computational Linguistics, 2010, pp. 72–79.

[10] F. Rudzicz, "Acoustic transformations to improve the intelligibility of dysarthric speech," in *Proceedings of the Second Workshop on Speech and Language Processing for Assistive Technologies*. Association for Computational Linguistics, 2011, pp. 11–21.

[11] E. Sanders, M. B. Ruiter, L. Beijer, and H. Strik, "Automatic recognition of dutch dysarthric speech: a pilot study." in *INTERSPEECH*, 2002.

[12] W. K. Seong, J. H. Park, and H. K. Kim, "Multiple pronunciation lexical modeling based on phoneme confusion matrix for dysarthric speech recognition," *Advanced Science and Technology Letters*, vol. 14, pp. 57–60, 2012.

[13] K. T. Mengistu and F. Rudzicz, "Adapting acoustic and lexical models to dysarthric speech," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 4924–4927.

[14] S. O. Caballero-Morales and F. Trujillo-Romero, "Dynamic estimation of phoneme confusion patterns with a genetic algorithm to improve the performance of metamodels for recognition of disordered speech," in *Advances in Computational Intelligence*. Springer, 2013, pp. 175–187.

[15] J. F. Gemmeke, J. V. D. Loo, G. D. Pauw, J. Driesen, H. V. hamme, and W. Daelemans, "A self-learning assistive vocal interface based on vocabulary learning and grammar induction," in *Proc. INTERSPEECH*, 2012, pp. 1–4.

[16] J. F. Gemmeke, B. Ons, H. Van hamme, J. van de Loo, W. D. G. De Pauw, J. Huyghe, J. Derboven, L. Vugen, B. van Den Broeck, P. Karsmakers, and B. Vanrumste, "Self-taught assistive vocal interfaces : An overview of the ALADIN project," in *Proc. INTERSPEECH*, 2013, pp. 1–5.

[17] B. Ons, N. Tessema, J. van de Loo, J. Gemmeke, G. De Pauw, W. Daelemans, and H. Van hamme, "A Self Learning Vocal Interface for Speech-impaired Users," in *Proc. Workshop on Speech and Language Processing for Assistive Technologies (SLPAT)*, 2013.

[18] Y. Zhang and J. Glass, "Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams," in *IEEE Workshop on Automatic Speech Recognition Understanding (ASRU)*, Nov 2009, pp. 398–403.

[19] M. Siu, H. Gish, A. Chan, W. Belfield, and S. Lowe, "Unsupervised Training of an HMM-Based Self-Organising Unit Recognizer with Applications to Topic Classification and Keyword Discovery," *Comput. Speech Lang.*, vol. 28, no. 1, pp. 210–223, Jan. 2013.

[20] C.-Y. Lee and J. Glass, "A Nonparametric Bayesian Approach to Acoustic Model Discovery," in *Proc. of 50th Annual Meeting of the ACL*, Stroudsburg, PA, USA, 2012, pp. 40–49.

[21] O. Walter, T. Korthals, R. Haeb-Umbach, and B. Raj, "A hierarchical system for word discovery exploiting DTW-based initialization," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Olomouc, Czech Republic, Dec. 2013.

[22] S. Chaudhuri and B. Raj, "Unsupervised Structure Discovery for Semantic Analysis of Audio," in *Advances in Neural Information Processing Systems 25*, 2012, pp. 1187–1195.

[23] J. V. D. Loo, G. D. Pauw, J. F. Gemmeke, P. Karsmakers, B. Van, D. Broeck, W. Daelemans, and H. V. hamme, "Towards shallow grammar induction for an adaptive assistive vocal interface: a concept tagging approach," in *Proc. NLP4ITA*, 2012, pp. 27–34.

[24] J. Gemmeke and H. Van hamme, "NMF-Based Keyword Learning from Scarce Data," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Olomouc, Czech Republic, Dec. 2013.

[25] H. Van hamme, "HAC-models: a Novel Approach to Continuous Speech Recognition," in *Proc. INTERSPEECH*, 2008.

[26] M. Sun and H. V. HAMME, "Coding Methods for the NMF Approach to Speech Recognition and Vocabulary Acquisition." *Journal of Systemics, Cybernetics & Informatics*, vol. 10, no. 6, 2012.

[27] S. Chaudhuri, M. Harvilla, and B. Raj, "Unsupervised Learning of Acoustic Unit Descriptors for Audio Content Representation and Classification," in *Proc. INTERSPEECH*, 2011, pp. 2265–2268.

[28] D. Arthur and S. Vassilvitskii, "k-means++: the advantages of careful seeding," in *Proc. ACM-SIAM symposium on discrete algorithms*, 2007, pp. 1027–1035.

[29] J. Schmalenstroeer, M. Bartek, and R. Haeb-Umbach, "Unsupervised learning of acoustic events using dynamic time warping and hierarchical K-means++ clustering," in *Proc. INTERSPEECH*, 2011.

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

[31] C. Middag, "Automatic analysis of pathological speech," Ph.D. dissertation, Ghent University, Belgium, 2012.

[32] J. Heymann, O. Walter, R. Haeb-Umbach, and B. Raj, "Iterative Bayesian Word Segmentation for Unsupervised Vocabulary Discovery from Phoneme Lattices," in *Proc. ICASSP*, Florence, Italy, May 2014.