Automatic segmentation and clustering of speech using sparse coding and metaheuristic search

Wiehan Agenbag and Thomas Niesler

Department of Electrical and Electronic Engineering, Stellenbosch University, South Africa

wagenbag@sun.ac.za, trn@sun.ac.za

10.21437/Interspeech.2015-641

Abstract

We propose a constrained shift and scale invariant sparse coding model for the purpose of unsupervised segmentation and clustering of speech into acoustically relevant sub-word units for automatic speech recognition. We introduce a novel local search algorithm that iteratively improves the acoustic relevance of the automatically-determined sub-word units from a random initialization by repeated alignment and subsequent re-estimation with the training material. We also contribute an associated population-based metaheuristic optimisation procedure related to genetic approaches to achieve a global search for the most acoustically relevant set of sub-word units. A first application of this metaheuristic search indicates that it yields an improvement over a corresponding local search. Using a subset of TIMIT for training, we also find that some of the automatically-determined sub-word units in our final dictionaries exhibit a strong correlation with the reference phonetic transcriptions. Furthermore, in some cases our sub-word transcriptions yield a compact set of often-used pronunciations. Informal listening tests indicate that the clustering is robust, and provides optimism that our approach will be suited to the task of generating pronunciation dictionaries that can be used for ASR.

Index Terms: segmentation, clustering, sparse coding, genetic algorithms, metaheuristic search, sub-word units

1. Introduction

We investigate the application of a sparse coding and dictionary learning framework to the task of unsupervised discovery of sub-word acoustic units in speech. This task is motivated by the need for building ASR's for under-resourced languages.

Previous work in the field of automatic segmentation by Torbati et al [1] demonstrates promising results using a Hierarchical Dirichlet Process HMM. Singh et al [2] attempt to obtain sub-word acoustic models and associated transcriptions using a maximum likelihood approach, conditioned on the orthographic transcriptions, acoustic models and resulting pronunciations, while Lee et al propose a hierarchical Bayesian model which jointly infers acoustic units and grapheme encoding [3]. Another approach to segment clustering using segment-level Gaussian Posteriorgrams is taken by Wang et al [4]. Sub-word units have also been derived by clustering HMM states associated with context-dependent graphemes [5, 6]. Sparse coding has previously been used in speech primarily for feature extraction [7–10].

In this study we present a novel implementation of sparse coding that is constrained to non-overlapping acoustic units to make it appropriate for segmenting speech. We also embed our sparse coding and dictionary learning algorithm inside a modified genetic search to obtain a hybrid metaheuristic algorithm that explores the solution space more extensively.

2. Background

Sparse coding attempts to reconstruct some input signal using a linear combination of the smallest possible number of basis functions taken from a finite set. This set is called the *dictionary* in the sparse coding literature and should not be confused with the term *pronunciation dictionary*. A sparse code $\boldsymbol{x}$, can therefore be seen as a solution to

$$\arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_0 \quad \text{such that} \quad \boldsymbol{y} = \boldsymbol{D}\boldsymbol{x} \tag{1}$$

where $\boldsymbol{y} \in \mathbb{R}^{N\times 1}$ is the signal we are trying to reconstruct, $\boldsymbol{D} \in \mathbb{R}^{N\times M}$ is the set of basis functions, packed column-wise, and $\boldsymbol{x} \in \mathbb{R}^{M\times 1}$ where $\|\boldsymbol{x}\|_0$ represents the number of nonzero values in the vector $\boldsymbol{x}$. In the context of speech, we may consider a typical utterance to be our input signal, which we wish to code using a highly sparse selection of sub-word phonemic units, which serve as basis functions.

2.1. Shift and scale invariance

Sub-word acoustic units are generally much shorter than the speech signal under analysis. Furthermore, they may vary in length considerably. It is therefore necessary to insist on shift and scale invariant sparse coding. To obtain shift-invariance, the dictionary-code product $\boldsymbol{D}\boldsymbol{x}$ in Equation 1 is replaced by a dictionary-code convolution

$$\boldsymbol{\Phi} * \boldsymbol{S} = \sum_{j=1}^{M} \boldsymbol{\phi}^j * \boldsymbol{s}^j, \tag{2}$$

where $\boldsymbol{\Phi} \in \mathbb{R}^{N_\phi \times M}$ is the convolutional dictionary, and $\boldsymbol{S} \in \mathbb{R}^{M\times N}$ are the coefficient sequences [7, 11]. The quantities $\boldsymbol{\phi}^j$ and $\boldsymbol{s}^j$ refer to the j^{th} column and row of $\boldsymbol{\Phi}$ and $\boldsymbol{S}$ respectively. Each basis function $\boldsymbol{\phi}^j$ is now associated with a coefficient sequence $\boldsymbol{s}^j$ that indicates not just whether a basis function is being used, but also at which time offset in $\boldsymbol{y}$. To obtain scale-invariance, each basis function is represented across a range of time-scales in the dictionary that is presented to the sparse coding algorithm [12].

2.2. Sparse coding as an optimization problem

The exact sparse coding formulation as given in Equation (1) is intractable. Moreover, in our case the pursuit of an exact recovery of each input signal is certain to be futile, since we will restrict the coefficient sequences to the use of basis functions that do not overlap in time. Most authors choose to cast the problem into an optimization framework with the cost function given by

$$C(\boldsymbol{\Phi}, \{\boldsymbol{S}_k\}) = \sum_{k=1}^{K} \|\boldsymbol{y}_k - \boldsymbol{\Phi} * \boldsymbol{S}_k\|_2^2 + \beta\tau(\boldsymbol{S}_k), \tag{3}$$

where $\boldsymbol{S}_k$ refers to the coefficient sequences used to code the k^{th} input signal $\boldsymbol{y}_k$ [7, 12–15]. The cost function can be seen as a weighted sum of the reconstruction error and a code diversity measure $\tau(\boldsymbol{S})$. This latter term yields small values when the code is sparse, and large values when it is not. The l_0 pseudo-norm used in Equation (1) is one possible diversity measure, but others that are differentiable have been proposed [12, 16].

3. Implementation

In order to ensure that discovered basis functions can correspond to useful sub-word units, we introduce two new constraints, leading to what we believe to be a novel class of sparse codes. Firstly, we restrict the coefficient sequences to use basis functions that do not overlap in time. Secondly, we insist that all code coefficients are non-negative. These constraints enable the sparse codes to be unambiguously interpretable as a sequential alignment of the input signal with a set of basis functions and also lead to a new approach to the optimal solution to the constrained problems.

3.1. Finding the optimal alignment of the basis functions

We now find the best possible alignment $\boldsymbol{S}_k$ of the input utterance $\boldsymbol{y}_k$ with basis functions $\boldsymbol{\phi}_j$ from our dictionary $\boldsymbol{\Phi}$. In order to quantify how good an alignment is, we use the cost function in Equation (3), with the l_0 pseudo-norm as a sparsity constraint:

$$C(\boldsymbol{\Phi}, \boldsymbol{S}_k) = \|\boldsymbol{y}_k - \boldsymbol{\Phi} * \boldsymbol{S}_k\|_2^2 + \beta \|\boldsymbol{S}_k\|_0 . \tag{4}$$

Since we enforce the constraint that no basis function may overlap with another in the reconstruction, the optimal choice of coefficient given a time offset and basis function is unambiguous. This enables the alignment invariant calculation of a delta cost matrix $\Delta\boldsymbol{C}_k$, with $\Delta\boldsymbol{C}_k[j, n]$ being the reduction in the cost function when basis function $\boldsymbol{\phi}_j$ is activated at time offset n. The path through the cost matrix that yields the largest total reduction in cost can then be found by dynamic programming. Given the input signal $\boldsymbol{y}_k$ and the time-scale of the basis function L_j, the optimal coefficient for basis function $\boldsymbol{\phi}_j$ at the time-offset n can be calculated as

$$\boldsymbol{S}_{k,\text{opt}}[j, n] = \boldsymbol{y}_k[n : n + L_j] \cdot \boldsymbol{\phi}_j / \left\| \boldsymbol{\phi}_j \right\|_2^2 , \tag{5}$$

with negative coefficients set to zero. Since both the squared vector norm of the reconstruction residual and the l_0 pseudonorm of the coefficient sequences are element-wise summations, the change in the cost function is confined to the change in the norm of the reconstruction residual within the time interval that the basis function $\boldsymbol{\phi}_j$ is active. Thus, we can calculate the reduction in cost function as

$$\begin{aligned} \Delta\boldsymbol{C}_k[j, n] = &\left\| \boldsymbol{y}_k[n : n + L_j] \right\|_2^2 - \beta \\ &- \left\| \boldsymbol{y}_k[n : n + L_j] - \boldsymbol{S}_{k,\text{opt}}[j, n]\boldsymbol{\phi}_j \right\|_2^2 . \end{aligned} \tag{6}$$

The combination of the steps detailed in this section comprises a new algorithm for the optimal solution to the constrained sparse coding model we consider in this paper.

3.2. Determining the dictionary of basis functions

We now consider the task of obtaining the optimal set of basis functions given an alignment. If we define the set $\{\boldsymbol{y}_n\}$ to contain all segments of the input signals where the basis function $\boldsymbol{\phi}_j$ is used to reconstruct those segments, and $\{s_n\}$ the corresponding coefficients, then it can be shown that

$$\boldsymbol{\phi}_{j,\text{opt}} = \sum_n s_n \boldsymbol{y}_n / \sum_n s_n^2 . \tag{7}$$

Equation (7) does not take into account that we have made time-scaled versions of each basis function available for coding. The final step in updating the basis functions is therefore to reinforce this relationship. Suppose the set $\{\boldsymbol{\phi}_n^l\}$ contains the updated basis functions derived from the same prototype $\boldsymbol{\phi}^l$, resampled to a common time-scale and normalised to unit norm. Also let σ_n^l denote the number of times the scaled basis function $\boldsymbol{\phi}_n^l$ is aligned with the input utterances, and L_n^l the length of that scaled basis function. A good approximation to the optimal prototype basis function is then

$$\boldsymbol{\phi}^l = \frac{1}{Z_l} \sum_n L_n \sigma_n \boldsymbol{\phi}_n^l , \tag{8}$$

where Z_l is a normalising factor. The factor $L_n\sigma_n$ represents a very good approximation of the relative importance of $\boldsymbol{\phi}_n^l$ in reducing the cost function, since each instance in which a basis function is used approximately decreases the cost function by a constant value proportional to that basis function's length.

3.3. Improving dictionaries with search

Having determined procedures for finding an optimal alignment given a dictionary, and a close to optimal set of basis functions given an alignment, we can develop a local search procedure to improve from an initial dictionary through repeated alignment of the dictionary with the input data, and subsequent use of that alignment to update the dictionary. Although this procedure is capable of reliably improving initial dictionaries, the point at which it converges is likely to be at a local optimum.

We turn to the metaheuristic strategies detailed below to improve our chances of finding good solutions. As an ensemble, these strategies can be seen as a modified genetic algorithm, retaining the notion of maintaining a population of solutions, and the genetic operators of selection and mutation. The *chromosome* of each individual in the population is a dictionary of prototypical (i.e. scale invariant) basis functions, with the individual basis functions seen as the *genes*. The fitness of a particular chromosome is simply the cost (Equation (3)), of its optimal alignment with the training data.

3.3.1. Selection

When preparing a new generation of solutions, the first step is to select the individuals from the previous generation which are to be used as the basis of the new generation. Since we want to exploit those solutions that show promise, and abandon those that do not, more offspring should be derived from fitter solutions. However, applying this bias too aggressively results in a loss of genetic diversity and hence memory of areas in the fitness landscape that show promise.

We therefore use a scheme known as rank selection, where the fitnesses of the individuals in the previous generation are ranked with the best individual achieving rank N and the worst rank 1 [17]. The expected number of offspring of individual i is tied to this rank $R[i]$ such that

$$E[i] = m_i + (m_a - m_i)\frac{R[i] - 1}{N - 1}, \tag{9}$$

where m_a is the variable selection pressure, and is constrained such that $1 \leq m_a \leq 2$ and $m_i = 2 - m_a$. We use stochastic universal sampling (SUS) to perform the selection itself, which guarantees that the actual number of offspring of each individual $O[i]$ is constrained to $\lfloor E[i] \rfloor \leq O[i] \leq \lceil E[i] \rceil$ [18].

3.3.2. Mutation

After selecting parents, new offspring can be produced by mutation, whereby each gene is given a small probability μ of being

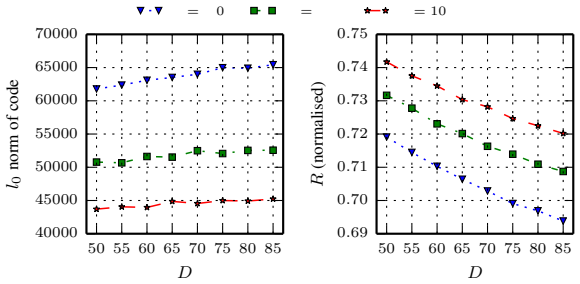

Figure 1: *Number of coefficients used and normalised reconstruction error for the elite individual for various values of β and D. The reference transcription contains 56377 phonemes.*

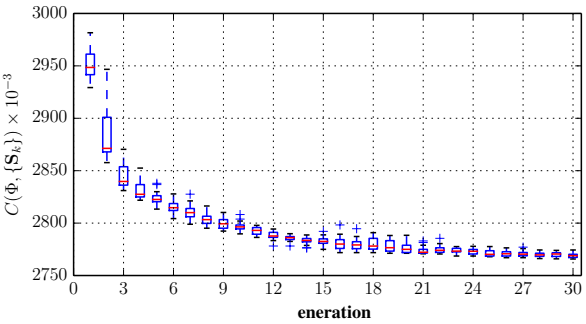

Figure 2: *Generational development of the population fitness distribution for an experiment with $\beta = 8.3$ and $D = 55$.*

modified. In order for the modification to represent a sensible new search direction, we replace the affected gene with a segment randomly drawn from the pool of blind segmentations created during initialization. In order to ensure that the fitness of each generation is at least as good as the previous one, we protect one instance of each of the ϵ best individuals from the previous generation from mutation.

3.3.3. Iterated search

The dictionaries developed through a random initialization, as well as those disrupted through mutation, are generally quite far from the minima of their basins of attraction. Even more problematically, they would not be at comparable distances from their eventual convergence point. When we compare the fitnesses of dictionaries, we would actually like to compare the minima of their respective basins of attraction. Failing that, we want all the dictionaries in the population to be roughly the same *distance* away from their minima. Anything else would lead to a scheme that cannot reliably distinguish between unfit individuals in deep basins and fit individuals in shallow basins.

Therefore, as a final step in producing a new individual, a local search is performed using the iterative realignment described at the start of Section 3.3. The search terminates after the absolute per-iteration improvement ΔC in the cost function falls below a certain threshold. In this study the threshold was initially infinite, and updated to the median of the population's terminal ΔC after each generation.

3.4. Initialising the dictionary

To initialise our dictionary discovery process we apply a blind segmentation algorithm to the feature vectors extracted from the speech signal. This creates a pool of candidate basis functions from which we can draw to construct plausible initial dictionaries. For this purpose, we employed the approach used by Ten Bosch, which inserts segment boundaries at locations where feature vectors change rapidly [19].

4. Experiments and results

4.1. Experimental setup and training overview

The 1386 SI training utterances of the TIMIT corpus were used for training. These are phonetically diverse sentences each spoken only once. This choice is motivated by the desire to avoid repetition which could bias the development of sub-word units that favour very specific contexts. The selected utterances are converted to 12-coefficient MFCC feature vectors using HTK [20]. The MFCCs are generated at a rate of one every 12 ms, with a window size of 19.2 ms.

A series of metaheuristic searches was applied to the training corpus in order to investigate the effect of tunable parameters. The parameters in question were the number of prototypical basis functions D and the diversity penalty β. The remaining parameters such as the mutation rate μ, the selection pressure m_a and the population size were fixed at values that appeared reasonable during initial informal testing.

Table 1: *Improvement in cost function by metaheuristic search compared to pure local search as a multiple of the search termination threshold.*

	$D = 55$	$D = 65$	$D = 75$
$\beta = 6$	36.63	36.74	38.15
$\beta = 8.3$	46.36	20.39	21.62
$\beta = 10.6$	32.15	24.87	60.03

Figure 1 summarises the elite individual found for each pair (β, D) on the training grid in terms of the number of sub-word units used by that individual in its transcription of the training audio, as well as the resulting normalised mean reconstruction error. From the figure it is clear that the diversity penalty allows the intended trade off between reconstruction error and increased code sparsity. It is also observable that our search procedure is capable of using larger dictionaries to perform more accurate acoustic matching. Furthermore, as the dictionaries become larger, with β held fixed, the number of coefficients used increases. This implies that the learned basis functions start matching shorter acoustic events.

4.2. Performance of metaheuristic search

Figure 2 shows how the metaheuristic search influences the fitness distribution of a population over the course of 30 generations. It is apparent that the algorithm manages to produce populations that consistently achieve higher fitnesses than previous generations. The generally smooth $1/n$ progression may be attributed to the local search function combined with the adapting ΔC threshold. Sudden jumps in fitness (most clearly visible at generation 12 and 16) are the result of fortuitous mutation.

Having shown that the metaheuristic search is effective, we would also like to show that it represents an improvement upon randomly guessed initial dictionaries. In order to do this, we saved the initial population of dictionaries for a subset of the experiments described in Section 4.1. These initial dictionaries were individually optimised using an exclusively local search until the per-iteration reduction in the cost function fell below the terminal ΔC.

Table 1 shows the absolute improvement (as a multiple of the terminal ΔC) made in the cost function by comparing the elite individual from the metaheuristic search to the elite individual resulting from iteratively improving the initial population. It is clear that the metaheuristic search consistently finds

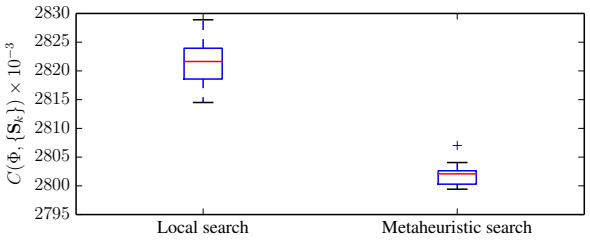

Figure 3: *Terminal fitness distributions for an experiment with* $\beta = 10.6$ *and* $D = 75$.

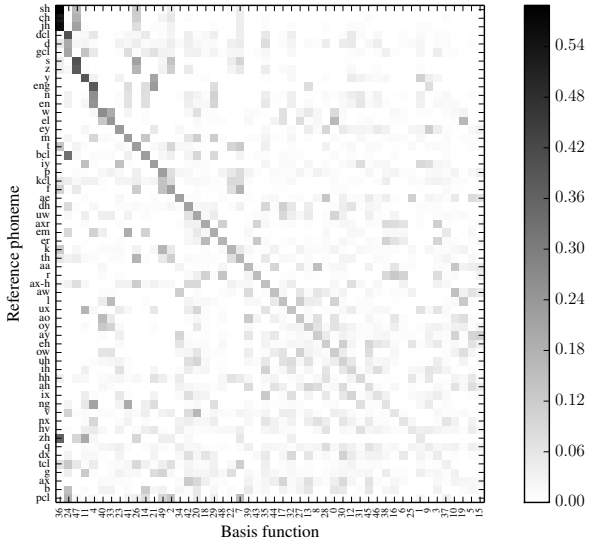

Figure 4: *Coincidence of learned basis functions with reference phonemes for* $\beta = 10.6$ *and* $D = 50$.

substantially better solutions. Figure 3 compares the terminal fitness distributions for the experiment that showed the greatest improvement when metaheuristic search was applied.

4.3. Evaluation of basis functions as sub-word units

In this section we evaluate whether the basis functions we learn in our experiments could be suited to the task of generating pronunciation dictionaries that can be used for ASR.

4.3.1. Coincidence with reference phonemes

Figure 4 shows how our learned basis functions coincide with the reference phonemes described by TIMIT. Since we did not attempt to infer an optimal alignment between our basis function transcriptions and TIMIT's phoneme transcriptions, we simply count the number of times where at least 50% of the span of one of our basis functions occurs within the time interval of a reference phoneme instance. Therefore we do not present a true confusion matrix, but rather a 2D coincidence histogram, where every row is normalised to show the fraction of each of our basis functions that coincide with a reference phoneme.

Figure 4, in conjunction with informal listening tests, indicates that the clustering and segmentation is reasonably robust and acoustically meaningful. Many reference phonemes are strongly represented using only one or two basis functions. In addition, in cases where a single basis function is used to represent multiple reference phonemes, it is often because those reference phonemes are acoustically similar.

4.3.2. Pronunciation consistency

For our learned sub-word units to be useful for the generation of pronunciation dictionaries, the transcription of spoken words

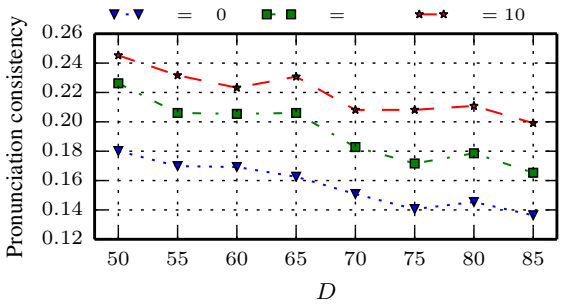

Figure 5: *Weighted average of the fraction of occurrences of the 20 most frequent words transcribed by one of their top 3 pronunciations. The reference transcription achieves 0.69.*

Table 2: *Pronunciation statistics for the most consistent set of sub-word acoustic units. Reference statistics are in parentheses.*

Word	# Occ.	# Pron.	Top pron.	Top 3 pron.	Top 5 pron.
the	508	55 (22)	14% (39%)	25% (81%)	34% (88%)
a	351	61 (20)	5% (36%)	15% (72%)	22% (81%)
to	269	86 (34)	7% (20%)	18% (43%)	26% (60%)
of	245	82 (25)	9% (35%)	20% (69%)	29% (80%)
and	226	102 (61)	8% (24%)	17% (36%)	24% (48%)
he	212	37 (10)	33% (63%)	51% (91%)	65% (96%)
in	184	75 (19)	8% (43%)	17% (71%)	25% (83%)
is	170	61 (17)	12% (46%)	29% (81%)	38% (88%)
are	92	51 (18)	8% (25%)	20% (57%)	28% (77%)

in terms of our units should be consistent. In order to evaluate how well our sub-word transcriptions perform in this respect, we calculate the cumulative fractions of the occurrences of each word that are transcribed using the top N pronunciations for that word. Figure 5 reports the weighted average fraction of occurrences for the top 3 pronunciations for a selection of the 20 most frequent words. In the most consistent case, an average of 25% of the occurrences of the selected words were represented using a set of just 3 pronunciations. This is quite poor compared to TIMIT's reference transcriptions, which achieve a fraction of 69%. However, it is high enough that we believe our approach shows promise, and work is ongoing to improve the achieved consistency.

Table 2 reports in-depth pronunciation statistics for the most consistent experiment in Figure 5. It is clear that even hand transcription by experts yields much phonetic variation. For example, the word 'and' occurs 226 times in the training corpus, and is pronounced in 61 different ways.

5. Summary and conclusions

In this paper, we have contributed a novel class of sparse codes for the specific task of unsupervised segmentation and clustering of speech. We also contributed the associated algorithms for the optimal calculation of the basis functions and the corresponding alignment with the audio features. Finally, we proposed a new optimisation strategy that embeds a local search within a metaheuristic search.

We found that the proposed metaheuristic search improved upon local search for the purpose of extracting acoustically relevant sub-word units from speech, on the basis of both an empirical cost function and informal listening tests. We are optimistic that, with further improvement, the automatically induced sub-word transcriptions will become useful for the development of ASR systems.

6. Acknowledgements

The presented study was sponsored in part by the National Research Foundation of the Republic of South Africa and by Telkom South Africa. Computations were performed using the University of Stellenbosch's Rhasatsha HPC.

7. References

[1] A. H. H. N. Torbati, J. Picone, and M. Sobel, "Speech acoustic unit segmentation using hierarchical Dirichlet processes." in *Proceedings of INTERSPEECH*, 2013, pp. 637–641.

[2] R. Singh, B. Raj, and R. Stern, "Automatic generation of subword units for speech recognition systems," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 2, pp. 89–99, Feb 2002.

[3] C.-y. Lee, Y. Zhang, and J. R. Glass, "Joint learning of phonetic units and word pronunciations for asr." in *Proceedings of EMNLP*, 2013, pp. 182–192.

[4] H. Wang, T. Lee, C.-C. Leung, B. Ma, and H. Li, "Unsupervised mining of acoustic subword units with segment-level gaussian posteriorgrams," in *Proceedings of INTERSPEECH*, 2013, pp. 2297–2301.

[5] W. Hartmann, A. Roy, L. Lamel, and J.-L. Gauvain, "Acoustic unit discovery and pronunciation generation from a grapheme-based lexicon," in *Proceedings of ASRU*, Dec 2013, pp. 380–385.

[6] M. Razavi *et al.*, "An HMM-Based Formalism for Automatic Subword Unit Derivation and Pronunciation Generation," in *International Conference on Acoustics, Speech and Signal Processing*, no. EPFL-CONF-206814, 2015.

[7] R. B. Grosse, R. Raina, H. Kwong, and A. Y. Ng, "Shift-invariance sparse coding for audio classification," *CoRR*, vol. abs/1206.5241, 2012.

[8] W. Smit and E. Barnard, "Continuous speech recognition with sparse coding," *Computer Speech & Language*, vol. 23, no. 2, pp. 200–219, 2009.

[9] G. S. V. S. Sivaram, S. Nemala, M. Elhilali, T. Tran, and H. Hermansky, "Sparse coding for speech recognition," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, March 2010, pp. 4346–4349.

[10] O. Vinyals and L. Deng, "Are sparse representations rich enough for acoustic modeling?" in *INTERSPEECH*, 2012.

[11] M. S. Lewicki and T. J. Sejnowski, "Coding time-varying signals using sparse, shift-invariant representations," 1998.

[12] W. Smit, "Sparse coding of single spoken digits," in *PRASA 2013*, 2013.

[13] M. Mørup, M. N. Schmidt, and L. K. Hansen, "Shift invariant sparse coding of image and music data," Tech. Rep., 2008. [Online]. Available: http://www2.imm.dtu.dk/pubdb/p.php?4659

[14] K. Kavukcuoglu, P. Sermanet, Y. lan Boureau, K. Gregor, M. Mathieu, and Y. Lecun, "Learning convolutional feature hierarchies for visual recognition."

[15] Y. Li and S. Osher, "Coordinate descent optimization for l_1 minimization with application to compressed sensing; a greedy algorithm," *Inverse Probl. Imaging*, vol. 3, no. 3, pp. 487–503, 2009.

[16] B. A. Olshausen and D. J. Fieldt, "Sparse coding with an overcomplete basis set: a strategy employed by v1," *Vision Research*, vol. 37, pp. 3311–3325, 1997.

[17] J. E. Baker, "Adaptive selection methods for genetic algorithms," in *Proceedings of an International Conference on Genetic Algorithms and their applications*. Hillsdale, New Jersey, 1985, pp. 101–111.

[18] ——, "Reducing bias and inefficiency in the selection algorithm," in *Proceedings of the second international conference on genetic algorithms*, 1987, pp. 14–21.

[19] L. ten Bosch and B. Cranen, "A computational model for unsupervised word discovery." in *Proceedings of INTERSPEECH*, 2007, pp. 1481–1484.

[20] S. Young, "The HTK Hidden Markov Model Toolkit: Design and Philosophy," 1994.