

Combined Optimisation of Baseforms and Model Parameters in Speech Recognition Based on Acoustic Subword Units

Trym Holter and Torbjørn Svendsen

Department of Telecommunications
Norwegian University of Science and Technology

Abstract - A major challenge in speech recognition is creating a lexicon which is robust to inter- and intra-speaker variations. This is even more so in speech recognisers based on non-linguistic units, e.g., acoustic subword units (ASWUs), since no standard pronunciation dictionaries are available. Thus the baseforms describing the vocabulary words in terms of the recognition units need to be generated from training data. In this paper we propose an algorithm for ASWU-based speech recognition which performs a combined optimisation of the baseforms and the subword models. The resulting system has been tested on the DARPA Resource Management task, and is shown to perform comparable to a baseline phoneme based system.

1 Introduction

Most contemporary speech recognisers employ some kind of phonemic units as the basic modelling entity. Such recognisers need a lexicon of baseforms describing the composition of the vocabulary words in terms of the recognition units, e.g., as a concatenation of phonemic units. In most cases, these lexica are manually generated, either by use of pronunciation dictionaries or by human experts, based on the speech corpus for which the recogniser is to be trained. This method of lexicon generation is in contrast to the way in which other parameters of the recogniser are obtained, namely by careful optimisation based on objective criteria. Thus, the lexica typically employed will not necessarily be optimal, nor robust to inter- and intra-speaker variations.

We have in previous contributions proposed a data-driven method which can automatically generate optimal baseforms [1, 2]. This procedure finds the baseforms which maximise the likelihood of the training data, and we have demonstrated that by employing the automatically generated baseforms in a speech recognition system based on phonemic units, we can obtain higher recognition rates than by using a standard lexicon.

Most automatic speech recognisers employ phonemic subword units which are based upon an abstract linguistic description of the language. However,

the analysis of the actual speech signal is acoustically based. The resulting system is thus neither phonetically nor acoustically consistent, but is instead a hybrid of two methodologies. In order to create a consistent acoustic framework, there have been several attempts to utilise acoustically based subword units (ASWUs) over the last ten years (see e.g., [3, 4, 5, 6, 7]).

Of course, since there is no known mapping between the acoustic subword units and linguistic subword units, a major challenge has been to design a method for lexicon generation for speech recognisers employing ASWUs. In the present paper we propose such a method, based on the methods described in [1, 2, 8], and demonstrate its use on the DARPA Resource Management task.

2 Basic training scheme

For design of our ASWU based speech recogniser, the training involves the following main steps:

1. Data-driven definition of the recognition units including time aligned labelling of the training corpus.
2. Estimation of initial acoustic models.
3. Computation of initial baseforms.
4. Re-optimisation of model parameters and baseforms.

The final step of the procedure will typically be performed iteratively until some convergence criterion is met. The first two steps of the basic training scheme is similar to that proposed in [4, 6]. The major differences lie in how the baseforms are generated. In this work, we argue that step 2–4 should be viewed as stages in a scheme for combined optimisation of baseforms and sub-word model parameters.

In the proposed system, the first step of the procedure is detailed as follows:

i) Initial segmentation of speech utterances into acoustically stationary segments is performed by Constrained Clustering Segmentation [9]. This problem can be formulated as finding the set of segment boundaries $\{b_0, b_1, \dots, b_J\}$ that minimises the total distortion

$$D_{tot} = \sum_{j=0}^{J-1} \sum_{n=b_j}^{b_{j+1}-1} d(\mathbf{x}_n, \bar{\mathbf{x}}_j), \quad (1)$$

where $\bar{\mathbf{x}}_j$ is the centroid of the j 'th segment consisting of the feature vectors $\{\mathbf{x}_{b_j}, \dots, \mathbf{x}_{b_{j+1}-1}\}$, and $d(\mathbf{x}, \mathbf{y})$ is the distance between vectors \mathbf{x} and \mathbf{y} .

ii) Representing each of the acoustic segments by its centroid, the LBG-algorithm [10] is employed to cluster the segments into S clusters, and to create a corresponding codebook. The resulting codebook C contains S codewords, c_i , $i = 0, 1, \dots, S-1$, where S is the predefined number of subword units used in our system.

iii) The labelling of acoustic segments into subword classes is performed on basis of the codebook C . The optimal label i' is found according to:

$$i' = \underset{i \in \{0, 1, \dots, S-1\}}{\operatorname{argmin}} \sum_{n=b_j}^{b_{j+1}-1} d(\mathbf{x}_n, \mathbf{c}_i), \quad \mathbf{c}_i \in C. \quad (2)$$

Each partition of the feature space resulting from the above clustering represents an ASWU. Step 2 of the basic training scheme implies that, for each subword, initial Hidden Markov Models (HMMs) should be trained from the acoustic segments contained in the corresponding cluster. These are used for creating initial baseforms. Finally, combined optimisation of the baseforms and the subword HMMs [8] is performed by the algorithm described in section 3. The baseform optimisation procedure is based on a Maximum Likelihood (ML) formulation [2] and relies on the modified Tree-Trellis algorithm [1].

3 Combined optimisation of baseforms and subword models

In subword based automatic speech recognition (ASR), the word model is usually constructed as a concatenation of subword models defined by the HMM parameter set θ . For each word, the baseform is defined in terms of the basic modelling units, $B = (u_1 \oplus u_2 \oplus \dots \oplus u_w)$. The model ϕ_v for word v is therefore defined by $\phi_v = \{\theta, B_v\}$. For a vocabulary of size V , the model parameters describing the entire lexicon is defined by $\lambda = \{\phi_1, \dots, \phi_V\} = \{\theta, L_V\}$, where L_V is the lexicon containing the V baseforms, $L_V = \{B_1, \dots, B_V\}$. To obtain optimal word model parameters, the HMMs and baseforms that jointly maximises the likelihood must thus be found:

$$\lambda' = \underset{\lambda \in \Lambda}{\operatorname{argmax}} \{p(\lambda|\mathcal{T})\} = \underset{\lambda \in \Lambda}{\operatorname{argmax}} \{p(\mathcal{T}|\lambda)p(\lambda)\}, \quad (3)$$

where \mathcal{T} denotes the entire training set and Λ is the total parameter space. No known procedure exists which is capable to perform the joint optimisation described by equation 3. However, assuming that the prior, $p(\lambda)$, is uniform, the criterion of equation 3 can be approximated by an iterative procedure consisting of two main steps. Assume that the previous estimate of λ' exists, $\hat{\lambda}^{[i-1]} = \{\hat{\theta}^{[i-1]}, \hat{B}_1^{[i-1]}, \dots, \hat{B}_V^{[i-1]}\}$. The two steps then consist of HMM and lexicon re-estimation according to equations 4 and 6:

- HMM parameter-estimate i is found according to

$$\hat{\theta}^{[i]} = \underset{\theta \in \Theta}{\operatorname{argmax}} \left\{ p \left(\mathcal{T} | \theta, \hat{B}_1^{[i-1]}, \dots, \hat{B}_V^{[i-1]} \right) \right\}, \quad (4)$$

where Θ is the total HMM parameter space.

- The i 'th estimate of the lexicon is found according to

$$\hat{L}_v^{[i]} = \operatorname{argmax}_{L_v \in \mathcal{L}_v} \left\{ p \left(\mathcal{T} | L_v, \hat{\theta}^{[i]} \right) \right\} = \operatorname{argmax}_{L_v \in \mathcal{L}_v} \left\{ \prod_{v=1}^V p \left(\mathcal{T}_v | L_v, \hat{\theta}^{[i]} \right) \right\}, \quad (5)$$

where \mathcal{L}_v is the parameter space describing all valid lexica and \mathcal{T}_v are the available training utterances for word v . Given the HMM parameters, the baseforms describing different words can be optimised independently:

$$\hat{B}_v^{[i]} = \operatorname{argmax}_{B \in \mathcal{B}} \left\{ p \left(\mathcal{T}_v | B, \hat{\theta}^{[i]} \right) \right\}, \quad v = 1, \dots, V, \quad (6)$$

where \mathcal{B} is the set of all valid baseforms.

The steps of the proposed optimisation scheme are described below:

1. Calculate initial estimates of the sub-word HMMs, $\hat{\theta}^{[0]}$.
2. Find the initial baseforms for each word in the vocabulary, $\hat{B}_v^{[0]}$, $v = 1, \dots, V$, according to equation 6. Set $i = 1$.
3. Re-estimate the HMM parameters according to equation 4.
4. Re-estimate the baseforms for every word in the vocabulary according to equation 6. Set $i = i + 1$.
5. Repeat from step 3 until some convergence criterion is met.

The criterion for HMM re-estimation described by equation 4 can generally not be fulfilled by any known algorithm. However, the Baum-Welch re-estimation procedure is known to converge towards a *local* maximum, and is extensively used in the ASR community. In practise, we perform a small number of iterations of the Baum-Welch procedure in each cycle of the combined optimisation scheme.

Equation 6 describes an ML based approach to baseform optimisation. This has been studied in [1, 2, 8]. The optimisation problem can be interpreted as a search through a sub-word grammar \mathcal{B} . For maximum flexibility we employ a null grammar \mathcal{B}_0 , i.e., a grammar that allows any number of sub-word units in any order to constitute a baseform. With this grammar, the solution to equation 6 fulfils the *exact* ML criterion. The modified Tree-Trellis algorithm [1] offers a solution for this search problem. Similar to the Baum-Welch procedure, this approach does guarantee a non-decreasing likelihood of the training data. Thus, the likelihood will increase (or stay constant) at each step of the training scheme, leading to iteratively refined word models. The proposed procedure will eventually converge.

With an increasing number of subword units, the search complexity increases, and for large subword inventories, the modified Tree-Trellis algorithm may require excessive amounts of memory. In the present work we

have utilised a procedure which approximates the exact ML solution, namely the Extended Multiple Candidate Method (EMCM) [2] in order to reduce the memory requirements. The idea of the EMCM is to constrain the grammar \mathcal{B} in order to reduce the search space. The first step of this algorithm consists in creating a set of baseform candidates. The N most likely baseforms for each of the K training tokens constitute this set. These candidates are then evaluated in a final search for the best baseform. The EMCM can be summarised as follows:

1. Perform an N -best search for each of the K tokens, $\mathcal{X}^{(k)}, k = 1, \dots, K$:

$$B'_k(i) = \underset{B \in \mathcal{B}_0, B \notin \mathcal{B}_{k,i-1}}{\operatorname{argmax}} \left\{ p(\mathcal{X}^{(k)}|B) \right\}, \quad \begin{array}{l} i=1,2,\dots,N, \\ k=1,2,\dots,K, \end{array} \quad (7)$$

where $\mathcal{B}_{k,j} = \{B'_k(i)\}, i = 1, \dots, j$, and $\mathcal{B}_{k,0}$ is the empty set.

2. Collect all candidate baseforms in $\mathcal{B}_{\text{emcm}} = \{\mathcal{B}_{k,N}\}, k = 1, \dots, K$.
3. Remove redundant baseform candidates from $\mathcal{B}_{\text{emcm}}$.
4. Utilise the modified Tree-Trellis algorithm to find the optimal baseform according to the given grammar,

$$B' = \underset{B \in \mathcal{B}_{\text{emcm}}}{\operatorname{argmax}} \left\{ \prod_{k=1}^K p(\mathcal{X}^{(k)}|B) \right\}. \quad (8)$$

Alternatively, the likelihood of each token can be calculated by the Viterbi algorithm for each of the candidate baseforms. The single baseform that maximises the joint likelihood of all tokens can then easily be found.

The size of N in the N -best search does affect the quality of the resulting lexicon. In [2], experiments on a phonemic subword based speech recogniser showed a significant gain by increasing N from 1 to 5. No additional gain was achieved by a further increase of N from 5 to 10. However, if the size of the subword inventory is increased, a larger variability is expected in $\mathcal{B}_{\text{emcm}}$ for a given number of candidates. It is therefore believed that with a larger subword inventory, $N = 5$ might not be sufficient to create baseforms that are close to the optimal. Thus, $N = 10$ was utilised in the experiments described in the next section.

4 Experiments

The proposed training scheme has been tested on the DARPA Resource Management speech corpus. For segmentation and labelling, feature vectors containing 14 LPC-cepstrum parameters were computed every 10 ms. The distortion (equations 1 and 2) was computed by Euclidean distance. The

initial segmentation was performed such that the average number of acoustic segments after the segmentation step was slightly less than twice the number of phonemes in the material, according to a standard pronunciation dictionary. For the segmentation, the duration of a subword was constrained to be at least 20 ms.

For modelling and recognition the speech was first pre-emphasised with the filter $H(z) = 1 - 0.97z^{-1}$. Feature vectors containing 39 parameters (12 lifted mel-frequency cepstral coefficients and normalised log energy, plus first and second order difference coefficients) were computed every 10 ms using a 25 ms Hamming window.

The size of the subword inventory was set to 128. Each of these units was modelled by a single-state HMM. Three sets of experiments were performed with one-, two-, and three-component Gaussian mixture pdfs for modelling of the state observation densities, respectively. For each model set we performed ten iterations of the combined optimisation scheme, including the initial parameter estimation.

The resulting lexicons and HMMs were tested on the four test sets “feb89”, “oct89”, “feb91”, and “sep92” with a total of 1200 sentences. A simple word-pair grammar was used for language modelling.

The word correct rates averaged over the four test sets are shown in figure 1. The integer labels in this figure correspond to the number of completed iterations of the training scheme. The intermediate points indicate the performance after only the HMM re-optimisation step of the iteration has been completed.

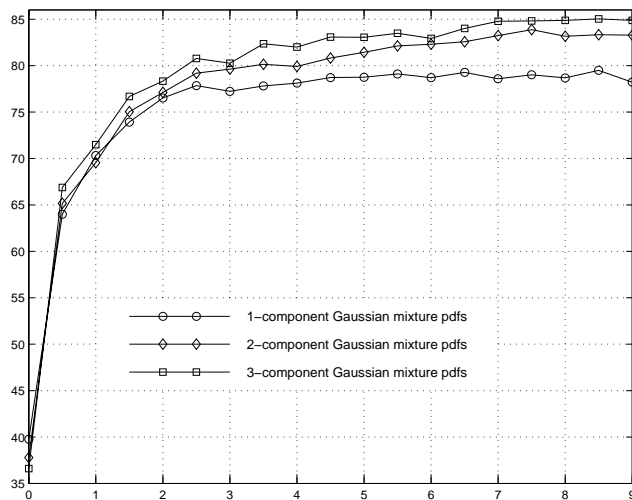


Figure 1: Average word correct rates versus iteration number.

For comparison, baseline HMM-systems using three-state context independent phoneme HMMs with one-, two-, and three-component Gaussian mixture pdfs were designed. Two sets of experiments were performed. In the first

experiment, the pronunciation dictionary was extracted from the DARPA Resource Management distribution. For the second experiment, the baseform optimisation procedure was applied, using the respective HMM-sets in order to improve the initial lexicon. As the subword inventory in this case is smaller than for the ASWUs, the modified Tree-Trellis algorithm was utilised for optimisation subject to the exact ML criterion. To ensure that the data-driven procedure found a sound foundation for the baseform estimates in the training data, the baseform optimisation scheme was only applied for words that occurred at least ten times in the database, i.e., approximately 60% of the vocabulary. The other lexical entries were left unchanged. This is in contrast to the experiments performed for the ASWUs, where baseforms necessarily must be created for each word in the vocabulary. The results from these baseline experiments are shown in table 1.

Number of mixtures	Initial lexicon	Optimised lexicon
1	78.3%	81.1%
2	84.3%	85.4%
3	87.2%	88.1%

Table 1: Recognition rates for baseline systems.

In figure 2, the best results achieved with the ASWUs are compared to the performance of the two baseline systems. For a fair comparison, the complexity of the acoustic modelling in each system should be taken into account. Thus, the figure shows recognition rates versus the number of free parameters utilised by the sub-word HMMs.

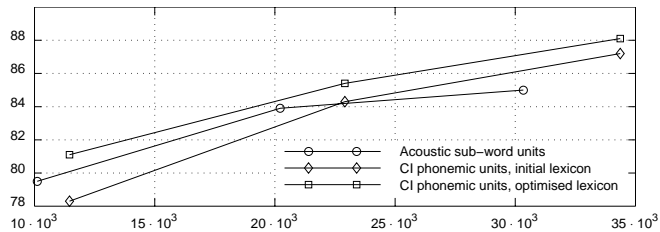


Figure 2: Word correct rates versus the total number of free parameters utilised for acoustic modelling.

The figure indicates that for the one- and two-mixture models, the ASWU-based system performs better than the phonemically based system with the initial lexicon and a comparable number of free parameters. However, increasing the number of mixtures from two to three for the proposed ASWU-based system does not give the same kind of performance gain as experienced for the phonemically based systems. We also note that the system based on phonemic units and an optimised lexicon performs best of all three systems in all experiments.

5 Conclusions

We have proposed an algorithm for designing a medium sized vocabulary speech recognition system based on acoustic subwords. Experiments on the Resource Management task have demonstrated that the proposed approach performs comparable to baseline HMM systems using phonemic units and a manually designed pronunciation dictionary. The performance of the ASWU system is in fact close to the performance obtained using optimised baseforms in the phonemic systems. To our knowledge, it is the first time acoustic subword units have been successfully applied to speaker independent recognition of vocabularies of this size.

References

- [1] T. Svendsen, F. K. Soong, and H. Purnhagen, "Optimizing baseforms for HMM-based speech recognition," in *Proc. European Conf. on Speech Commun. and Techn. (EUROSPEECH)*, (Madrid, Spain), pp. 783–786, Sept. 1995.
- [2] T. Holter and T. Svendsen, "A comparison of lexicon-building methods for subword-based speech recognisers," in *Proc. IEEE Region 10 Conf. on Digital Signal Proc. (TENCON)*, (Perth, Australia), pp. 102–106, IEEE, Nov. 1996.
- [3] J. G. Wilpon, B.-H. Juang, and L. R. Rabiner, "An investigation on the use of acoustic sub-word units for automatic speech recognition," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Proc. (ICASSP)*, (Dallas, USA), pp. 821–824, IEEE, Apr. 1987.
- [4] C.-H. Lee, F. K. Soong, and B.-H. Juang, "A segment model approach to speech recognition," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Proc. (ICASSP)*, (New York, USA), pp. 501–504, IEEE, Apr. 1988.
- [5] C.-H. Lee, B.-H. Juang, F. K. Soong, and L. R. Rabiner, "Word recognition using whole word and subword models," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Proc. (ICASSP)*, (Glasgow, Scotland), pp. 683–686, IEEE, May 1989.
- [6] T. Svendsen, K. K. Paliwal, E. Harborg, and P. O. Husøy, "An improved subword based speech recognizer," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Proc. (ICASSP)*, (Glasgow, Scotland), pp. 108–111, IEEE, May 1989.
- [7] L. R. Bahl, P. F. Brown, P. V. deSouza, R. L. Mercer, and M. A. Picheny, "A method for the construction of acoustic Markov models for words," *IEEE Trans. Speech and Audio Processing*, vol. 1, pp. 442–452, Oct. 1993.
- [8] T. Holter and T. Svendsen, "Combined optimisation of baseforms and subword models for an HMM based speech recogniser," in *Proc. The 4th Int. Symposium on Signal Processing and its Applications (ISSPA)*, (Gold Coast, Australia), pp. 321–324, Aug. 1996.
- [9] T. Svendsen and F. K. Soong, "On the automatic segmentation of speech signals," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Proc. (ICASSP)*, (Dallas, USA), pp. 77–80, IEEE, Apr. 1987.
- [10] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. 28, pp. 84–95, Jan. 1980.