

MAXIMUM LIKELIHOOD MODELLING OF PRONUNCIATION VARIATION

*Trym Holter*¹ and *Torbjørn Svendsen*²

¹Department of Radio and Signal Processing, SINTEF Telecom and Informatics

²Department of Telecommunications, Norwegian University of Science and Technology

E-mail: Trym.Holter@informatics.sintef.no or svendsen@tele.ntnu.no

ABSTRACT

This paper addresses the problem of generating lexical word representations that properly represent natural pronunciation variations for the purpose of improved speech recognition accuracy. The current work is based on a procedure for data-driven optimisation of the pronunciation dictionary which creates a single baseform per word in the vocabulary, subject to a maximum likelihood (ML) criterion [1]. In the current approach, we extend the ML formulation in order to achieve optimal modelling of pronunciation *variations*. Since different words will not in general exhibit the same amount of pronunciation variation, the procedure allows words to be represented by a different number of baseforms. The method improves the sub-word description of the vocabulary words, and has been shown to improve recognition performance on the DARPA Resource Management (RM) task.

1. INTRODUCTION

Most contemporary speech recognisers employ some kind of sub-word unit 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. Most current ASR systems utilise pronunciation lexica comprised of a single baseform for each word in the vocabulary. In this paper, we address the problem of generating lexica that properly represent natural pronunciation variations with the purpose of improved speech recognition accuracy. In order to achieve this with a deterministic lexicon, multiple baseforms should be allowed for each word.

Cohen [2] estimated that for carefully pronounced (e.g., read) speech, a lexicon containing *one* baseform per word will cover about 80% of all realisations. He also showed that it is crucial not to generate too many alternative baseforms for each word, as this will lead to larger confusability. In spontaneous speech, the pronunciation variability is much larger, and this is often

assumed to be one source of difficulty in recognition of unscripted speech.

Usually, the lexica used in ASR systems are manually generated, either by use of standard pronunciation dictionaries or by human experts. 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. We have in previous contributions proposed a data-driven algorithm which can automatically generate optimal baseforms [1, 3]. This procedure finds a single baseform per word in a manner which maximises 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. In principle, this algorithm is capable of finding the N most likely baseforms given a set of training utterances. However, these baseforms do not model pronunciation *variations*, as they are generally very similar and do not complement each other.

In the current work, we extend the ML formulation in order to allow more than one baseform per word in the vocabulary. The proposed algorithm allows words to be represented by a different number of baseforms, in such a manner that words which exhibit little inter- and intra-speaker variability are represented by one or a small number of baseforms. On the other hand, words that are exposed to large variability are modelled by a larger set of baseforms, with each baseform modelling one typical pronunciation of the word. The optimal number of baseforms per word is automatically determined by the algorithm.

This paper is organised as follows: In section 2, the optimisation criterion utilised for lexicon generation is presented. In section 3, the ML-based baseform optimisation procedure is briefly revisited, and in section 4, the ML clustering approach is described. In section 5, results from the experiments are presented and finally some concluding remarks are given in section 6.

2. THE OPTIMISATION CRITERION

Assume that the pronunciation lexicon L_V for a vocabulary of size V contains a set of baseforms

$$L_V = \left\{ B_v^{(i)} \right\}, \quad \begin{array}{l} i = 1, \dots, I_v, \\ v = 1, \dots, V, \end{array} \quad (1)$$

where $B_v^{(i)}$ is the i 'th baseform for word v , represented by a total of I_v baseforms. We define the likelihood of the training data as

$$p(\mathcal{T}|L_V) = \prod_{v=1}^V \prod_{n=1}^{N_v} \max_{i=1, \dots, I_v} p(\mathcal{X}_v^{(n)}|B_v^{(i)}), \quad (2)$$

where \mathcal{T} denotes the entire training set and $\mathcal{T}_v = \{\mathcal{X}_v^{(n)}\}$, $n = 1, \dots, N_v$, are the training samples of word v . The likelihood function $p(\mathcal{T}|L_V)$ is obviously a non-decreasing function of the total number of baseforms in the lexicon, and an unconstrained maximisation of equation 2 would result in a lexicon containing one baseform for each training token, i.e., $I_v = N_v$ (not deducting for the possibility of identical baseforms for different tokens), which is *not* the desired representation. Instead, we propose to perform the maximisation under the constraint that the total number of baseforms in the lexicon should not exceed a predetermined threshold I_{L_V} :

$$L_V = \operatorname{argmax}_{\{I_v, B_v^{(i)}\}} p(\mathcal{T}|L_V, B_v^{(i)}) \wedge \sum_{v=1}^V I_v \leq I_{L_V}. \quad (3)$$

This optimisation task can be recognised as a clustering problem, and in section 4 we will present a procedure based on the K -means algorithm, which is able to create a lexicon according to the criterion in equation 3.

3. AUTOMATIC BASEFORM GENERATION

The data-driven procedure for automatic baseform generation relies on an ML criterion [1]:

$$\hat{B} = \operatorname{argmax}_{B \in \mathcal{B}} \prod_{n=1}^N p(\mathcal{X}^{(n)}|B). \quad (4)$$

This optimisation problem can be interpreted as a search through a sub-word grammar \mathcal{B} . For maximum flexibility we employ a null grammar, i.e., a grammar that allows any number of sub-word units in any order to constitute a baseform. The single baseform that maximises the joint likelihood of all training utterances available for the present word can be found by a procedure known as the modified Tree-Trellis algorithm [1].

4. THE ML CLUSTERING APPROACH

For each word v , the relevant probability model is as follows: The baseform population consists of an unknown number I_v sub-populations. In order to fit the procedure to the criterion in section 2, a *classification likelihood* approach is utilised in the clustering. The cluster analysis is *model-based*, i.e., the properties of each cluster are described by its model; in this case the baseform which maximises the likelihood of the cluster members.

The scheme which is proposed here for optimising L_V proceeds in two stages. In the first stage, each word v is treated individually. As the optimal number of baseforms, I_v , is unknown, a number of candidate *baseform sets* are found through a divisive clustering procedure,

$$\left\{ \hat{B}_{v,j}^{(i)} \right\}, \quad \begin{array}{l} i = 1, \dots, j, \\ j = 1, \dots, I_{\max}, \end{array} \quad (5)$$

where $\hat{B}_{v,j}^{(i)}$ denotes baseform candidate i out of the j candidates in baseform set j for word v .

The number I_{\max} is a predefined limit which determines the maximum allowed number of baseforms for a single word. As such, I_{\max} is a heuristically defined limit, but if it is chosen large enough it will not restrict the optimisation process.

In the second stage of the optimisation algorithm, the lexicon is composed on basis of the candidate baseform sets found during stage 1. The number of baseforms for each word is chosen so that the likelihood of the entire training set is maximised, while the total number of baseforms in the lexicon is fixed to the predetermined limit, I_{L_V} . The details of the two stages are presented next.

Stage 1: For each word, $v = 1, \dots, V$:

1) *Initialisation:* Assign all N_v training tokens of word v to \mathcal{K}_1 (cluster 1). Find the ML estimate of the baseform describing the cluster, i.e.,

$$\hat{B}_{v,1}^{(1)} = \operatorname{argmax}_{B \in \mathcal{B}} \prod_{n=1}^{N_v} p(\mathcal{X}_v^{(n)}|B). \quad (6)$$

Calculate the likelihood of the training data given $\hat{B}_{v,1}^{(1)}$,

$$l_v(1) = \prod_{n=1}^{N_v} p(\mathcal{X}_v^{(n)}|\hat{B}_{v,1}^{(1)}). \quad (7)$$

Set $J = 2$.

2) *Splitting:* Find the cluster with the lowest total likelihood. Identify the two cluster members which are spaced furthest apart. The distance between two tokens is defined as the string distance between the best

baseform candidate for each of them. The baseform candidate for each token is found by Viterbi decoding, using the grammar \mathcal{B} . The string matching procedure is based on dynamic programming and utilises the Levenshtein distance [4]. Let the baseforms corresponding to the two tokens be the initial baseforms for two new clusters and remove the old cluster.

3) Re-assign each token n to the cluster i' which maximises the likelihood,

$$i' = \operatorname{argmax}_{i=1, \dots, J} p(\mathcal{X}_v^{(n)} | \hat{B}_{v,i}^{(i)}). \quad (8)$$

4) Update the ML baseform estimates for each cluster,

$$\hat{B}_{v,i}^{(i)} = \operatorname{argmax}_{B \in \mathcal{B}} \prod_{\{n: \mathcal{X}_v^{(n)} \in \mathcal{K}_i\}} p(\mathcal{X}_v^{(n)} | B), \quad i = 1, \dots, J. \quad (9)$$

5) Calculate the likelihood for each cluster,

$$l_v^{(i)}(J) = \prod_{\{n: \mathcal{X}_v^{(n)} \in \mathcal{K}_i\}} p(\mathcal{X}_v^{(n)} | \hat{B}_{v,i}^{(i)}), \quad i = 1, \dots, J, \quad (10)$$

and the total likelihood of the training data,

$$l_v(J) = \prod_{i=1}^J l_v^{(i)}(J). \quad (11)$$

6) Loop to step 3 if the token assignment has changed and the number of performed iterations is below a pre-determined threshold.

7) If $J < I_{\max}$, increment J and loop to step 2.

Stage 2 is also an iterative procedure. The initial lexicon consists of a single baseform per word. The total number of baseforms is then increased by one in each cycle, each time in a manner that guarantees a maximum increase of the total likelihood.

Stage 2:

1) Set $I_v = 1$ and $l_v = l_v(1)$ for $v = 1, \dots, V$, where I_v is the number of baseforms for word v and l_v is the current joint likelihood for all utterances of word v . Set $I = V$, where I is the current number of baseforms in the lexicon.

2) Identify the word for which increasing the number of baseforms by one leads to the largest increase in log-likelihood, i.e.,

$$v' = \operatorname{argmax}_{v=1, \dots, V} \{\log[l_v(I_v + 1)] - \log[l_v]\}. \quad (12)$$

3) Increment $I_{v'}$ and I , and set $l_{v'} = l_{v'}(I_{v'})$.

4) If $I < I_{L_V}$, loop to step 2.

5) Compose the lexicon on basis of the candidate base-

form sets found during stage 1,

$$\hat{L}_V = \left\{ \hat{B}_{v,I_v}^{(i)} \right\}, \quad i = 1, \dots, I_v, \quad v = 1, \dots, V. \quad (13)$$

In the proposed scheme, total likelihood is used as a criterion function. Alternatively, the average likelihood per frame for all tokens could be used, in order to make the resulting number of baseforms insensitive to the number of training samples. However, using total likelihood and uniform thresholds for all words favour those words that occur frequently in the training data, thereby making it likely that the words modelled by a large number of baseforms have sufficiently many training tokens for reliable estimates. Also, training a large number of baseforms from a small set of training samples would increase the possibility of incorporating outlier baseforms in the lexicon, which is not the intention of the pronunciation variation modelling.

5. EXPERIMENTS

The proposed procedure was tested on the RM corpus. Simple context independent models (47 phones and two silence models) with a single Gaussian were generated from the speaker independent training set utilising the initial lexicon. This lexicon is manually generated and is distributed with the RM corpus. We used a standard front-end which parameterises the speech data into feature vectors of dimension 39 (12 liftered MFCCs and normalised log-energy in addition to first and second order delta parameters)[1, 3].

In order to ensure that the data-driven pronunciation modelling has a reasonable amount of training samples available, we chose to consider only those words that occurred at least ten times in the training data. This vocabulary subset accounts for 605 of the 991 words in the RM vocabulary. For the remaining 386 vocabulary words, the single-baseform lexical entries from the initial lexicon were maintained.

These experiments aim at investigating the relationship between the lexicon size and the corresponding recognition performance achieved for the RM task. In the first experiment, we examined the criterion function itself as the lexicon size was increased. In figure 1, the average log-likelihood per frame for both the training and test data is shown for 11 different lexicon sizes, described by the average number of baseforms per word, \bar{I}_{L_V} . As expected, the likelihood of the training data increases monotonically with the size of the lexicon. The same applies to the likelihood of the test data, even if this is not guaranteed by the optimisation criterion. Figure 1 shows that by allowing just a few words to be represented by more than one baseform,

a relatively large likelihood gain is achieved. The gain achieved by increasing \bar{I}_{L_V} from 1.0 to 1.1 (i.e., increasing the lexicon size by 99 baseforms) is larger than the additional gain achieved by increasing \bar{I}_{L_V} to 2.0. This indicates that for this corpus, a small set of words are insufficiently modelled by a single baseform.

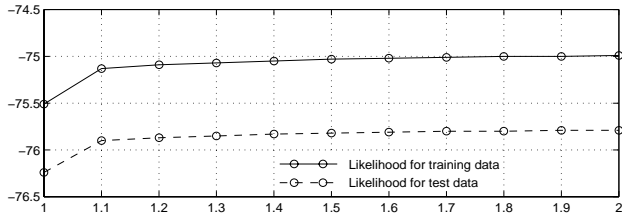


Figure 1: Average log-likelihood per frame versus average number of baseforms per word in the vocabulary.

Note that the average number of baseforms per word refers to the entire vocabulary, i.e., 991 words, while only the 605 words for which the optimisation procedure is applied can be represented by multiple baseforms. The maximum number of baseforms per word that has been allowed by the optimisation procedure is set to $I_{\max} = 4$. It is believed that this limit is chosen large enough to not restrict the optimisation process severely.

The recognition experiments were performed with a simple word-pair grammar. Four testdata subsets were used, “feb89”, “oct89”, “feb91”, and “sep92”, and the average word correct and word accuracy rates were calculated. In figure 2, the recognition rates are plotted versus \bar{I}_{L_V} . For comparison, we performed a baseline experiment utilising the same HMMs with the initial lexicon. This resulted in a word correct rate of 78.3% and a word accuracy rate of 75.5% [3].

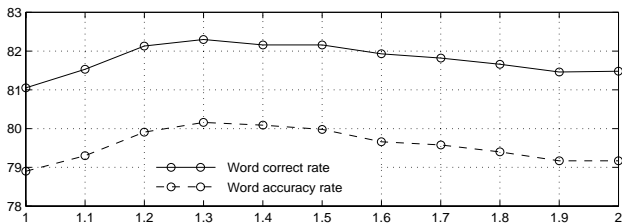


Figure 2: Word correct (solid) and accuracy rates (stapled) versus average number of baseforms per word in the vocabulary.

Figure 2 shows that the best results were achieved with an average of 1.3 baseforms per word in the vocabulary, i.e., a total of 1288 baseforms for the 991 words. In this lexicon, 13 words were represented by 4 baseforms, 45 words by 3 baseforms, 168 words by 2 baseforms, and the rest by a single baseform. When \bar{I}_{L_V}

was increased further, the performance started to drop off, which is in accordance with the previous statement that too many baseforms for each word in the vocabulary will lead to larger confusability and thus hurt the recognition rate. Comparing the results achieved with the proposed scheme at $\bar{I}_{L_V} = 1.3$ to the baseline system, an error rate reduction of 18.4% is found. In comparison to the system using the single-baseform ML-optimised lexicon ($\bar{I}_{L_V} = 1.0$), the resulting error rate reduction is 6.3%. Thus, significant performance gain is achieved at the cost of a moderate decoding complexity increase.

The RM corpus is a highly controlled task, i.e., it consists of read speech with a relatively small amount of dialectic variation. We believe that this kind of pronunciation variation modelling is even more important for less controlled tasks, and larger performance gain could be expected using the proposed scheme under other circumstances.

6. CONCLUSIONS

A novel approach to modelling of pronunciation variations in a deterministic lexicon is proposed. The procedure is based on an ML formulation and is completely data-driven. When the performance of the automatically generated, multiple-baseform pronunciation dictionary with an average of 1.3 baseforms per word is compared to the manually generated single-baseform lexicon, an error rate reduction of 18.4% is found. It is believed that this kind of pronunciation variation modelling is even more important for unscripted speech than for a controlled task like the RM corpus which was utilised in these experiments.

7. 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] M. Cohen, *Phonological Structures for Speech Recognition*. PhD thesis, University of California, Berkeley, 1989.
- [3] T. Holter, *Maximum Likelihood Modelling of Pronunciation in Automatic Speech Recognition*. PhD thesis, Norwegian University of Science and Technology, Dec. 1997.
- [4] S. Y. Lu and K. S. Fu, “A sentence-to-sentence clustering procedure for pattern analysis,” *IEEE Trans. Syst., Man, Cyb.*, vol. SMC-8, pp. 381–389, May 1978.