

# A Comparison of Lexicon-Building Methods for Subword-Based Speech Recognisers

Trym Holter and Torbjørn Svendsen

Department of Telecommunications  
Norwegian University of Science and Technology  
E-mail: holter@tele.unit.no

**ABSTRACT:** *A comparison of different algorithms for training of pronunciation dictionaries for use with subword-based speech recognisers is given. An extension to existing sub-optimal solutions is presented, and is shown to give results close to the maximum likelihood solution. The DARPA Resource Management (RM) database was used for evaluating the lexicon-building algorithms. When compared to the initial lexicon derived from the DARPA RM-distribution, improvements of recognition rates have been obtained for all lexicons trained with the different criteria. The maximum likelihood solution resulted in an 11.5% reduction in word error rate, compared to the 10.5% reduction offered by the proposed sub-optimal method.*

## 1. INTRODUCTION

Modern large vocabulary speech recognisers employ subwords as the basic modelling units. This implies that in order to recognise words (or sentences), a lexicon which defines the composition of the vocabulary words in terms of the basic units must be available to the recogniser. In most cases, linguistically defined subwords are chosen as the basic recognition units, typically phonemes or phone-like units.

In a standard HMM based speech recogniser, the subword models are optimised from the training data, utilising an objective criterion. This is in contrast to the baseform creation, which is commonly done by human experts or extracted from a standard pronunciation dictionary. Particular problems with these approaches arise, e.g., in cases where the pronunciation variations of many speakers with different dialects need to be represented by one or a small number of lexical entries. Also, the traditional approaches cannot readily be applied to systems employing non-linguistic recognition units.

Some efforts to automatically optimise the process of baseform selection have been reported over the past decade. These include attempts to find the phonetic transcription given the orthographic spelling [1, 2] or utilising the text-to-sound rules

from a speech synthesis system [3]. Recently, there have been proposed methods based on a maximum likelihood formulation without prior information. In [4, 5], the search space is constrained before the approximate maximum likelihood solution is found. This is in contrast to the algorithm presented in [6], where a method for finding the overall maximum likelihood solution is given. This algorithm is also used in [7], where a combined optimisation of baseforms and subword models is investigated. However, even if this algorithm is of reasonable computational efficiency, it is costly as it comes to memory requirements. In certain cases, the algorithm suffers from memory shortage. This problem is increasing as the number of subword units are increased. For these reasons a fall-back procedure is necessary, and sub-optimal solutions should be investigated.

In this paper we extend the methods given in [4, 5] utilising the N-best paradigm [8]. In Section 2 this extension is described after a review of the existing methods. In Section 3 results from experiments on the DARPA RM database [9] are reported. These results are discussed in Section 4 and finally conclusions are reported in Section 5.

## 2. LEXICON BUILDING METHODS

The maximum likelihood solution to the baseform selection problem can be formulated as follows: Given a set of sample utterances of a word,  $U = \{U_1, U_2, \dots, U_L\}$ , and a Hidden Markov Model,  $\lambda = \{\lambda_1, \lambda_2, \dots, \lambda_P\}$ , describing all the subword units  $p = \{p_1, p_2, \dots, p_P\}$ , find the most likely string of subword units,  $\hat{S}$ , over the set of all possible strings,  $S$ , subject to some given decoding grammar

$$\hat{S} = \operatorname{argmax}_S \{P(S|U, \lambda)\}. \quad (1)$$

By Bayes rule,

$$\begin{aligned} \hat{S} &= \operatorname{argmax}_S \left\{ \frac{P(U|S, \lambda)P(S|\lambda)}{P(U|\lambda)} \right\} \\ &= \operatorname{argmax}_S \{P(U|S, \lambda)P(S|\lambda)\}. \end{aligned} \quad (2)$$

Assuming that all strings are equally probable, this is equivalent to maximising the joint likelihood of the utterances, i.e.,

$$\begin{aligned} \hat{S} &= \operatorname{argmax}_S \{P(U|S, \lambda)\} \\ &= \operatorname{argmax}_S \prod_{i=1}^L P(U_i|S, \lambda). \end{aligned} \quad (3)$$

If (3) is to be optimised over only one utterance, the problem can be easily solved by Viterbi decoding. However, the number of sample utterances per word will need to be large enough to produce word models that are robust to both inter- and intra-speaker variations yielding a non-trivial optimisation problem. In the next two sections existing solutions to this problem are reviewed. The methods differ in how the search space of possible strings,  $S$ , is defined. In Section 2.3 we propose an extension to the method presented in Section 2.2.

### 2.1. Maximum Likelihood Method

To achieve the overall maximum likelihood solution, the search space  $S$  must be unconstrained. A natural choice is the simple free subword decoding (e.g., free-phoneme decoding) grammar shown in figure 1, where any number of subword units in any order can constitute a string. In figure 1, “ $\emptyset$ ” denotes a null transition, i.e., a transition that is not associated with an observable event.

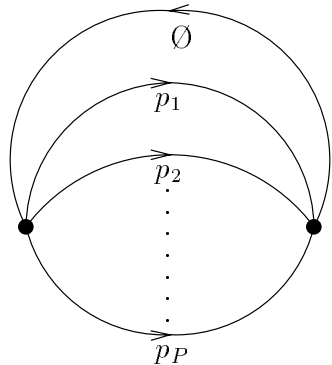


Figure 1: Free subword decoding grammar.

In [6] a modification of the tree-trellis based fast search [10] is proposed. This modified tree-trellis algorithm in principle guarantees finding the optimal baseforms.

The tree-trellis algorithm is based on the A\*-algorithm for finding the optimal path(s) through a tree [11]. In order to provide the A\*-search with a cost estimate, a time synchronous Viterbi search is performed in the (time-)forward direction, storing accumulated likelihood scores for every time

instance for all word terminal states. The A\*-search is then performed time asynchronously in the backward direction to find the  $N$  best strings. In order to maximise (3), the tree-trellis algorithm need some modifications.

The modified tree-trellis algorithm commences by computing a likelihood map of each of the  $L$  observations. Then, a frame synchronous Viterbi trellis search is performed in the forward direction according to the given decoding grammar.

The Viterbi search is executed using the token passing paradigm to accommodate length variations in the string hypotheses. For every observation, the Viterbi search also generates a partial path map which for every frame and grammar node contains the accumulated likelihood scores of all partial paths leading to the grammar node,

$$\max_{q_{1 \rightarrow t-1}} P(O_{1 \rightarrow t}^{(l)}, q_{1 \rightarrow t-1}, q_t = i(\lambda_{k_n}) | \Lambda_{l,n}), \quad (4)$$

where  $i(\lambda_{k_n})$  is the final state of the  $k$ th arc leading to the  $n$ th grammar node and the model  $\Lambda_{l,n}$  is a string model constructed by concatenating  $n$  subword models.  $q_{1 \rightarrow t-1}$  denotes the state sequence from time 1 to  $t-1$  and  $O_{1 \rightarrow t}^{(l)}$  denotes the  $t$  first observation vectors of sample utterance  $l$ . In addition, an ordered list of the rank of each partial path is produced.

After completing the forward Viterbi search, a backward, frame asynchronous tree search is performed. The search finds the phoneme string that maximises the combined likelihood scores for the  $L$  tokens of a word. If so desired multiple lexical entries can be found using this algorithm, producing a rank ordered list of transcriptions based on the corresponding likelihood scores. For some words with large variations in acoustic realisation (intra- or inter-speaker variations) the modified tree-trellis algorithm may not be able to produce a single representative baseform. These words can give rise to fan-out problems in the backward tree-search, resulting in computer memory shortage.

### 2.2. Multiple-Candidate Method

This method is in essence the approach taken in [4] for determining fenonic baseforms and in [5] for phonemic transcription. First, the search space is constrained by a maximum likelihood search for the best transcription for each utterance of the given word,

$$\hat{S}_i = \operatorname{argmax}_S \{P(U_i|S, \lambda)\}, \quad i = 1, 2, \dots, L. \quad (5)$$

The transcriptions  $\hat{S}_i$  can be found by a standard Viterbi decoding, utilising the free subword decoding grammar in figure 1. Now, the solution to the

baseform selection problem is the one transcription  $\hat{S}$  of the candidates  $S' = \{\hat{S}_i\}$  that has most likely produced all  $L$  utterances. In accordance with (3) this is given by

$$\hat{S} = \operatorname{argmax}_{S'} \prod_{i=1}^L P(U_i|S', \lambda). \quad (6)$$

The set of candidate transcriptions is now (in contrast to the overall maximum likelihood approach) finite. At most, the number of candidates equals  $L$ . If several sample utterances are decoded to the same string of subword units, the number of candidate transcriptions will be less than  $L$ . Most often,  $L$  is small and the baseform selection problem can be solved by an exhaustive search over the set  $S'$ .

### 2.3. Extended Multiple-Candidate Method

In this section we propose an extension to the approach in Section 2.2. The main difference between the two approaches already presented is the search space constraint that is imperative in the multiple-candidate method. To get a closer approximation to the overall maximum likelihood solution, we let  $S''$  be a larger subset of  $S$ . This is achieved by replacing the search for the best transcription for each utterance of the given word by an N-best search. This gives a set of candidate transcriptions

$$S'' = \left\{ \hat{S}_i^{(j)} \right\}, \quad \begin{array}{l} i = 1, \dots, L, \\ j = 1, \dots, N, \end{array} \quad (7)$$

where  $\hat{S}_i^{(j)}$  is the  $j$ 'th best transcription found for utterance number  $i$ . Now, the solution to the baseform selection problem is given by

$$\hat{S} = \operatorname{argmax}_{S''} \prod_{i=1}^L P(U_i|S'', \lambda). \quad (8)$$

This procedure increases the search space by approximately a factor  $N$ . At most the number of candidate transcriptions equals  $N \cdot L$ , but a number of those will in most cases be redundant. This results in a set that is still feasible for an exhaustive search.

## 3. EXPERIMENTS

The lexicon building methods in Section 2 all depend on reliable word-alignment of the training data, as each transcription in the lexicon should be trained from the given material. Since manual segmentation and labelling of speech data is expensive, hand-labelled training data is seldom

available. The training corpus can, however, be reasonably well segmented at the word-level utilising the HMM framework. Given an initial lexicon, a set of subword HMMs, and word-level transcriptions of the training corpus, the word-level segmentation can be obtained by Viterbi decoding. The automatic word-aligned data and the HMMs are then used as a basis for building a new lexicon by one of the methods in Section 2.

The training schemes have been tested on the DARPA RM speech corpus. Using the standard word transcriptions, simple context independent phoneme models with a single Gaussian have been generated from the speaker independent training set. 47 phone models and two silence models were used. The speech was first pre-emphasised with the filter  $H(z) = 1 - 0.97z^{-1}$ . Then feature vectors containing 39 parameters (12 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.

Experiments have shown that a minimum of 7-8 training tokens are necessary to obtain a representative transcription. In this work, the search for new transcriptions was performed only for those words that occurred ten times or more in the training set. Of the 991 words in the RM vocabulary, 605 words occur at least ten times in the training set. For the remaining 386 vocabulary words, the original phonemic transcriptions were used. In order to minimise the computational burden, a maximum of 100 (randomly selected) tokens were used in training. The same tokens were used in all experiments.

The modified tree-trellis algorithm may, as discussed in Section 2.1, not be able to produce a single representative baseform. For those words which suffered from fan-out problems the extended multiple-candidate approach was used with  $N = 10$ . Typically, this happened for long words, as these tended to increase the risk of memory shortage during the search for an optimal baseform. The fall-back procedure was necessary for a total of 34 words.

In addition to the lexicons generated by the different training methods, the original lexicon was tested for reference purposes. The experiments were performed with the following lexicons:

- *Original*: The initial lexicon derived from the DARPA RM-distribution.
- $N_1$ : The lexicon generated by the multiple-candidate method.

- $N_2, \dots, N_{10}$ : The lexicons generated by the extended multiple-candidate method with  $N = 2, \dots, 10$ .
- $ML$ : The lexicon generated by the overall maximum likelihood method. The extended multiple-candidate method with  $N = 10$  was used as a fall-back procedure for 34 words.

The resulting lexicons were tested with the same HMMs on the four test sets “feb89”, “oct89”, “feb91”, and “sep92” with a simple word-pair grammar. Each of these test sets contains 300 sentences.

The results for the extended multiple-candidate method are plotted in figure 2 for the four test-sets to illustrate how the choice of  $N$  influences the recognition results (notice that the  $y$ -axis is not continuous). In figure 3 the average results are plotted. These results are also summarised in table 1.

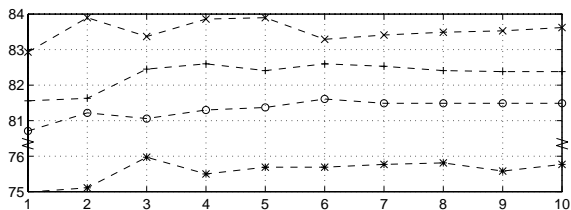


Figure 2: Word correct rates for lexicons trained with  $N = 1, \dots, 10$  on the four test-sets: feb89 (o), oct89 (+), feb91 (x), and sep92 (\*).

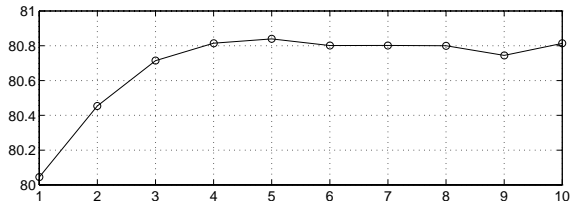


Figure 3: Average word correct rates for lexicons trained with  $N = 1, \dots, 10$ .

In addition to the recognition experiments, the transcriptions generated by the extended multiple-candidate method and the overall maximum likelihood method were compared directly. The overall maximum likelihood search succeeded for a total of 571 words. These transcriptions were compared to the corresponding transcriptions generated by the extended multiple-candidate method with  $N = 1, 2, \dots, 10$ . The percentage of identical transcriptions as a function of  $N$  is shown in figure 4.

Lexicon	Average recognition rates	
	Correct	Accuracy
<i>Original</i>	78.28%	75.46%
$N_1$	80.05%	77.79%
$N_2$	80.45%	78.38%
$N_3$	80.71%	78.64%
$N_4$	80.81%	78.81%
$N_5$	80.84%	78.94%
$N_6$	80.80%	78.91%
$N_7$	80.80%	78.92%
$N_8$	80.80%	78.90%
$N_9$	80.74%	78.81%
$N_{10}$	80.81%	78.88%
<i>ML</i>	81.05%	78.38%

Table 1: Average word recognition rates for different lexicons.

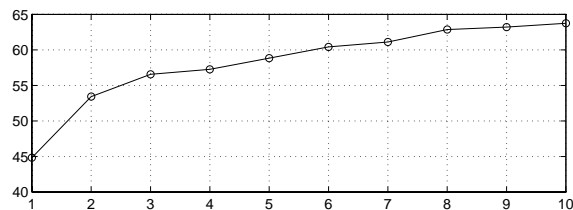


Figure 4: Percentage of transcriptions that are identical to the overall ML-solution for the extended multiple-candidate method with  $N = 1, \dots, 10$ .

#### 4. DISCUSSION

The results in Section 3 show that the performance increases when the lexicon is optimised from the training data, utilising an objective criterion. When compared to the initial lexicon derived from the DARPA RM-distribution, improvements of recognition rates have been obtained for all lexicons trained with the different schemes.

As one would expect, the overall maximum likelihood method outperforms the sub-optimal procedures aiming at an approximate maximum likelihood solution. Even if the difference is consistent over all test-sets, the difference is not very large. This indicates that the sub-optimal procedures can be reasonable alternatives when the overall maximum likelihood solution for some reason can not be found.

It is also found that the extended multiple-candidate method ( $N > 1$ ) gives better results than what is achieved with the multiple-candidate method ( $N = 1$ ). This gives reason, as the extended multiple-candidate method will converge towards the maximum likelihood solution as  $N \rightarrow \infty$ . From figure 3 we conclude that choosing  $N = 5$  seems to be sufficient. In this case, an additional increase in  $N$  do not give any significant reduction in word error rate.

As  $N$  is increased, the percentage of transcriptions achieved from the extended multiple-candidate method identical to the maximum likelihood solutions increases steadily. This also applies when  $N$  increases from 5 to 10. In this area there is no corresponding effect on the word recognition rate. This indicates that the transcriptions that differ from the maximum likelihood solution, differ by only a few subword units, and also give a performance close to these.

An interesting question is what happens as the number of subword units increases. This will often be the case e.g. when acoustically defined subword units are utilised. Preliminary results indicate that the overall maximum likelihood solution often suffers from fan-out problems, which increases the need for good fall-back procedures. The choice of  $N$  in the extended multiple-candidate method under these circumstances is a subject for future research.

## 5. CONCLUSIONS

In this paper we have proposed an extension to an existing sub-optimal procedure for training of lexicons from the available speech data. This method has been compared to the maximum likelihood solution. When compared to the initial lexicon derived from the DARPA RM-distribution, the maximum likelihood solution gives a word-error-rate reduction of 11.5%, compared to the 10.5% reduction offered by the sub-optimal approach at best. The training schemes that are discussed are equally applicable to linguistically and non-linguistically defined subword units.

## 6. REFERENCES

- [1] J. Lucassen and R. Mercer, "An information theoretic approach to the automatic determination of phonemic baseforms," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Proc. (ICASSP)*, (San Diego, USA), pp. 42.5.1-42.5.4, IEEE, Mar. 1984.
- [2] L. R. Bahl *et al.*, "Automatic phonetic baseform determination," in *Proc. of the DARPA Workshop on Speech and Natural Language*, (San Mateo, USA), DARPA, June 1990.
- [3] A. Asadi, R. Schwartz, and J. Makhoul, "Automatic modeling for adding new words to a large vocabulary speech recognition system," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Proc. (ICASSP)*, (Toronto, Canada), pp. 305-308, IEEE, 1991.
- [4] L. R. Bahl, P. F. Brown, P. V. de Souza, 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.
- [5] R. Haeb-Umbach, P. Beyerlein, and E. Thelen, "Automatic transcription of unknown words in a speech recognition system," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Proc. (ICASSP)*, (Detroit, USA), pp. 840-843, IEEE, May 1995.
- [6] 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.
- [7] 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 Proc. and its Applications (ISSPA)*, (Gold Coast, Australia), Aug. 1996. (To be published).
- [8] R. Schwartz and Y.-L. Chow, "The N-best algorithm: An efficient and exact procedure for finding the N most likely sentence hypotheses," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Proc. (ICASSP)*, (Albuquerque, USA), pp. 81-84, IEEE, Apr. 1990.
- [9] P. Price, W. M. Fisher, J. Bernstein, and D. S. Pallet, "The DARPA 1000-word Resource Management database for continuous speech recognition," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Proc. (ICASSP)*, (New York, USA), pp. 651-654, IEEE, April 1988.
- [10] F. K. Soong and E.-F. Huang, "A trellis based fast search for finding the N best sentence hypotheses in continuous speech recognition," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Proc. (ICASSP)*, (Toronto, Canada), pp. 705-708, IEEE, May 1991.
- [11] N. Nilsson, *Problem-Solving Methods in Artificial Intelligence*. McGraw Hill, 1971.