

# AUTOMATIC GENERATION OF PHONE SETS AND LEXICAL TRANSCRIPTIONS

R. Singh, B. Raj and R. M. Stern

Department of Electrical and Computer Engineering and School of Computer Science  
Carnegie Mellon University  
5000 Forbes Avenue, Pittsburgh, PA 15213

## ABSTRACT

Large vocabulary automatic speech recognition systems model words as sequences of a small set of basic sub-word units (the *phoneset*), which the systems are trained to classify. All words in the system's vocabulary are transcribed in terms of this set in a *dictionary*. The phoneset and dictionary are specific to a language and are typically designed manually. The system's performance is critically dependent on the quality of the phoneset and the accuracy of the dictionary.

In this paper we attempt to generate the phoneset and dictionary automatically, using only the training data and their transcriptions. We treat this as a joint optimization problem with a maximum *a posteriori* solution for the dictionary and a maximum likelihood solution for the phoneset and its acoustic models. Experiments with the DARPA Resource Management corpus show that the automatically generated phoneset and dictionary result in recognition accuracies close to those obtained using manually designed ones.

## 1. INTRODUCTION

Medium and large vocabulary speech recognition systems model a small number of sub-word units rather than entire words as the basic units of speech. These are usually phonetically motivated and represented as a set of symbols in the ASR system. Words are represented as sequences of these subword units in the dictionary used by the system. Traditionally, both the sub-word units and the dictionary are hand crafted and the same phoneset is used for all tasks within a given language with minor manually effected variations. While this is an adequate approach, a fixed phoneset may not be optimal under different acoustic conditions and for different task domains, even within the same language. It may be therefore be advantageous to derive the optimal sub-word units automatically from the training corpus.

Automatic derivation of pronunciations of words for a given set of predefined sub-word units has been attempted by several researchers in the past (*e.g.*[1], [3]). In this paper we address the problem of automatic derivation of the sub-word units themselves, as well as the dictionary. The phoneset and the dictionary are jointly optimized over the acoustic training data using likelihood of the training data as an optimization criterion. The following section outlines the problems involved. In Section 3 we present our solutions. We present our experimental results in Section 4.

## 2. PROBLEM DESCRIPTION

The simultaneous generation of a phoneset and dictionary is a highly unconstrained joint optimization problem. The optimal number of phones needed to represent the language, as represented by the training data is unknown. Independently of this, the number of

phones in any of the words is unknown. The word boundaries in the training data are also unknown.

For an ASR system, ideally the objective function to be optimized should be the recognition performance. However, the recognition performance is obtainable only at the end of a tedious training and testing process and it would be extremely time consuming to optimize over it. We therefore use the likelihood of the training data as an optimization criterion as follows: Let  $\Phi$  be phoneset of size  $n_\phi$ . Let the dictionary transcribing the words in terms of  $\Phi$  be denoted as  $\mathbf{D}_\phi$ . Let the parameters of the statistical models for  $\Phi$ , *i.e.* the *acoustic models*, be denoted as  $\lambda_\phi$ . Let the acoustic training data and their transcriptions be jointly denoted by  $\mathbf{T}$ . We note here that the knowledge of the acoustic models of the sub-word units,  $\lambda_\phi$  implies that the sub-word units,  $\Phi$ , are also known. If we have a statistical, or rule based model,  $\Gamma_P$ , that places constraints on how phones can follow each other, this can be used to constrain the problem. For non-ideographic languages, it may be possible to obtain a statistical or rule-based model,  $\Gamma_S$ , that relates the spellings of words to their pronunciations, this can also be used to constrain the problem. We incorporate these constraints and formulate our problem as :

$$\lambda_\phi, \mathbf{D}_\phi = \sup_{\Lambda, \{\phi\}} \{P(\mathbf{T}, \{\phi\} | \lambda, n_\phi, \Gamma_P, \Gamma_S)\} \quad (1)$$

The equation above results in a Maximum *a posteriori* (MAP) estimate of  $\mathbf{D}_\phi$  and a Maximum Likelihood (ML) estimate of  $\lambda_\phi$ .

## 3. SOLVING THE PROBLEM

The solution for the optimal lexical representation as given by Equation (1) requires the joint estimation of  $\mathbf{D}_\phi$ ,  $\lambda_\phi$  and  $n_\phi$ . We attempt to solve the problem by decomposing it into two parts: estimating the size of the optimal phone set  $n_\phi$ , and *jointly* estimating  $\mathbf{D}_\phi$  and  $\lambda_\phi$ .

In the following paragraphs, for notational simplicity, we omit the subscript in  $\mathbf{D}_\phi$  and  $\lambda_\phi$  and write these as  $\mathbf{D}$  and  $\lambda$  instead.

### 3.1. Joint estimation of $\mathbf{D}$ and $\lambda$

We reduce the joint estimation in Equation (1) to an iterative solution:

$$\lambda_i = \sup_{\Lambda} P(\mathbf{T} | \mathbf{D}_i, n_\phi, \Lambda, \Gamma_P, \Gamma_S) \quad (2)$$

$$\mathbf{D}_{i+1} = \sup_{\{\phi\}} P(\{\phi\} | \mathbf{T}, n_\phi, \lambda_i, \Gamma_P, \Gamma_S) \quad (3)$$

where the subscript  $i$  represents the iteration number. It can be easily shown that each step of the iterations described above results in an increase in the likelihood of the data as given by Equation (1).

The size of the phone set  $n_\phi$  is implicit in the dictionary  $\mathbf{D}_\phi$ . Similarly, the likelihood of the data  $\mathbf{T}$ , given the dictionary  $\mathbf{D}_\phi$  is

independent of any constraints on the dictionary. As a result, the above equations can be modified to

$$\lambda_i = \sup_{\Lambda} P(\mathbf{T}|\mathbf{D}_i, \Lambda) \quad (4)$$

$$\mathbf{D}_{i+1} = \sup_{\{\phi\}} P(\{\phi\}|\mathbf{T}, \lambda_i, \Gamma_P, \Gamma_S) \quad (5)$$

We refer to Equations (4) and (5) as the *model update* step and the *dictionary update* step respectively. The model update step is clearly the maximum likelihood solution for the statistical models for the phones and can be obtained by the Baum-Welch algorithm when these models are HMMs.

The dictionary update step is more complicated since the boundaries of the individual words in the training corpus are not known. There are several ways of segmenting each utterance into as many segments as there are words in the utterance. We refer to each of these segmentations as a *word segmentation*  $w_s$  and to the set of word segmentations for all utterances in the training corpus  $\mathbf{T}$  as the set  $\{w_s\}$ . Of the possible word segmentations for an utterance, only one corresponds to the correct word boundaries. Ideally, Equation (5) would have to be optimized over all possible word segmentations. However, we simplify this process as

$$\mathbf{D}_{i+1}, \{w_s\}' = \sup_{\{\phi\}, \{w_s\}} P(\{\phi\}, \{w_s\}|\mathbf{T}, \Lambda, \lambda, \Gamma_P, \Gamma_S) \quad (6)$$

where  $\{w_s\}'$  represents the jointly optimal word segmentation. The dictionary update step can now, once again, be obtained as an iterative solution of the form:

$$\{w_s\}_j = \sup_{\{w_s\}} P(\{w_s\}|\mathbf{D}_{i+1,j}, \mathbf{T}, \lambda_i, \Gamma_P, \Gamma_S) \quad (7)$$

$$\mathbf{D}_{i+1,j+1} = \sup_{\{\phi\}} P(\{\phi\}|\{w_s\}_j, \mathbf{T}, \lambda_i, \Gamma_P, \Gamma_S) \quad (8)$$

Each step of this iteration can be shown to result in an increase in  $P(\{\phi\}, \{w_s\}|\mathbf{T}, \lambda_i, \Gamma_P, \Gamma_S)$ .

Using Bayes' theorem and assuming that all possible word segmentations are equally likely *a priori*, Equation (7) can be modified to:

$$\{w_s\}_j = \sup_{\{w_s\}} P(\mathbf{T}|\{w_s\}, \mathbf{D}_{i+1,j}, \lambda_i, \Gamma_P, \Gamma_S) \quad (9)$$

This equation can be maximized for  $\{w_s\}$  very simply, using the Viterbi algorithm. Note here that if the *correct* word segmentation  $\{w_s\}'$  were given, the above estimation becomes unnecessary. In this case it is sufficient to solve for Equation (8) and the iterations over  $\{w_s\}$  can be avoided altogether.

Once a word segmentation is given, the boundaries of the various words in the training data are also given. Hence the dictionary need not be jointly optimized for Equation (8) - it is sufficient to optimize the pronunciation of each word in the lexicon. Thus, Equation (8) reduces to

$$\phi^{max} = \sup_{\phi} P(\phi|W_{data}, \lambda_i, \Gamma_P, \Gamma_S) \quad (10)$$

$$\mathbf{D}_{i+1,j+1} = \{\phi^{max}\} \quad (11)$$

where  $\phi$  refers to the pronunciation of the word  $W$  in the lexicon.  $W_{data}$  refers to the set of segmented acoustic realizations for the word  $W$ .

Equation (10) requires us to search over every possible pronunciation  $\phi$  to identify  $\phi^{max}$ , for each word in the lexicon. Since

there are an infinite number of possible pronunciations in the absence of any constraint, this is clearly infeasible. For any *single* instance  $W_k$  of a word  $W$ , however, it is straight forward to obtain

$$\phi_k^{max} = \sup_{\phi} P(W_k|\phi, \lambda_i) \quad (12)$$

using the Viterbi algorithm. We therefore obtain  $\phi_k^{max}$  for every instance of the word in the training data  $\mathbf{T}$ , resulting in a set of pronunciations  $\{\phi^{max}\}_W$  for the word  $W$ . This set of pronunciations can be collapsed into a graph [5] as shown in Figure 1. As

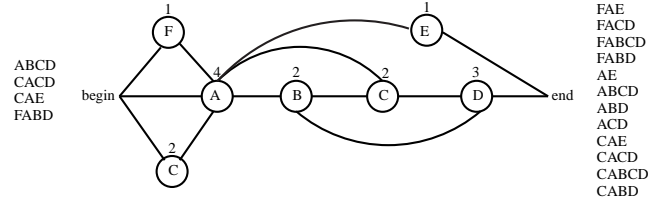


Figure 1: In this figure four hypothetical pronunciations for a word (left) have been collapsed into a single graph. The weight associated with any node (shown on top of the node) is proportional to the number of times the node has been visited in this set of four pronunciations. On the right of the graph are listed twelve pronunciations which have been generated from the graph.

we can see from this figure, the graph enables us to generate many more possible pronunciations for the word than the original set of pronunciations  $\{\phi^{max}\}_W$  than were used to create the graph. We generate a list of pronunciations  $\{\phi\}_{graph}$  from this graph, and restrict our search for the optimal pronunciation in Equation (10) to this set of pronunciations<sup>1</sup>.

$$\phi^{max} = \sup_{\phi \in \{\phi\}_{graph}} P(\phi|W, \lambda_i, \Gamma_P, \Gamma_S) \quad (13)$$

Using Bayes' theorem in conjunction with the fact that the spelling and phonemic constraints  $\Gamma_P$  and  $\Gamma_S$  only apply to the pronunciation  $\phi$  of the word, and not to  $W$  or  $\lambda_\phi$ , the above equation can be simplified to

$$\phi^{max} = \sup_{\phi \in \{\phi\}_{graph}} P(W|\phi, \lambda_i) P(\phi|\Gamma_P) P(\phi|\Gamma_S) \quad (14)$$

$P(W|\phi, \lambda_i)$  is the likelihood of the observed acoustic data for the word, for the phone sequence  $\phi$ . If the statistical models for the phones are HMMs,  $P(W|\phi, \lambda_i)$  can be easily obtained using the Baum-Welch algorithm.  $P(\phi|\Gamma_P)$  is the probability of the phone sequence  $\phi$  given the phonemic constraints  $\Gamma_P$ . We use a statistical N-gram model derived from a phonetic decoding of the training corpus  $\mathbf{T}$ .

The spelling constraints  $\Gamma_S$  are also statistical, and are obtained as the probability of phone sequences given the spelling of the word [4].

Using Equation (11) and Equation (14),  $\mathbf{D}_{i+1,j+1}$  can now be obtained as

$$\mathbf{D}_{i+1,j+1} = \left\{ \sup_{\phi \in \{\phi\}_{graph}} P(W|\phi, \lambda_i) P(\phi|\Gamma_P) P(\phi|\Gamma_S) \right\} \quad (15)$$

<sup>1</sup>If we include the corresponding pronunciation from  $\mathbf{D}_{i+1,j}$  in  $\{\phi\}_{graph}$ , the most likely pronunciation in  $\{\phi\}_{graph}$  is guaranteed to be at least as likely as the pronunciation in  $\mathbf{D}_{i+1,j}$ , thereby guaranteeing a non-decreasing likelihood for every iteration.

For the complete solution for the optimal  $\mathbf{D}_\phi$  for any  $n_\phi$ , Equations (2) and (3) are iterated until Equation (1) converges. Within each of these iterations themselves, the solution for Equation (3) is obtained by iterating Equations (7) and (8) until Equation (6) converges. In practice we iterate these steps until the recognition accuracy on a heldout data set converges. For a phoneset of any given size the Equations (7) and (8) typically converge in 3 iterations. Equations (2) and (3) converge in just 2 iterations.

### 3.2. Estimating $n_\phi$

Increasing the number of phones  $n_\phi$  results in an increase in the number of parameters representing the training data, and therefore an increase in the likelihood of the training data. The likelihood of the training data is, therefore, not a good metric to base the choice of the optimal  $n_\phi$  on. We therefore use the recognition accuracy of the optimal dictionary and phoneset for any  $n_\phi$  on a set of heldout data,  $\mathbf{T}_H$ , which is not part of  $\mathbf{T}$ , to estimate  $n_\phi$ . In other words, we attempt to obtain

$$n_\phi = \sup_n R(\mathbf{T}_H|n) \quad (16)$$

where  $R(\mathbf{T}_H|n)$  is the recognition accuracy of the heldout set on the acoustic models for the optimal phoneset of size  $n$ . Note that the optimal dictionary  $\mathbf{D}_{T,n}$  and the optimal statistical parameters  $\lambda_{T,n}$  have been obtained from  $\mathbf{T}$  and not  $\mathbf{T}_H$ .

We begin with a small value for  $n_\phi$ , and split the most frequently occurring phones in the dictionary. We do this by clustering the data segments corresponding to the phones into two clusters, while ensuring that all data segments belonging to a particular word stay together, and replacing the phone labels with the cluster labels in the dictionary. We increase the phone set in a phased manner until any increase in the number of phones does not result in increase in  $R(\mathbf{T}_H|n)$ .

### 3.3. Implementation of the algorithm

For any non-ideographic script, we initialize the dictionary in a rule-based manner using the corresponding alphabet. For example, the word CAT could be transcribed phonetically as “C A T”. Another initialization for the same word that is less dependent on the consistency of the script of the language could be “Y Y Y”. This initialization is non-committal in assuming only a relation relation only between the number of characters in the spelling and the length of the pronunciation. The complete algorithm is shown in the flowchart in Figure 2.

## 4. EXPERIMENTAL RESULTS

The phone definition and lexical generation algorithm presented in this paper was tested on the DARPA Resource Management (RM) database. A phoneset and dictionary were automatically generated using 2.7 hours of RM training data, and their corresponding transcriptions. Recognition performance with semi-continuous HMMs using these components was tested on a held out RM test set consisting of 1.58 hours of speech. The CMU SPHINX-III speech recognition system was used for all experiments. The training set covered a vocabulary of 987 words. The vocabulary of the held out set was 991 words, four of which were not covered by the training set.

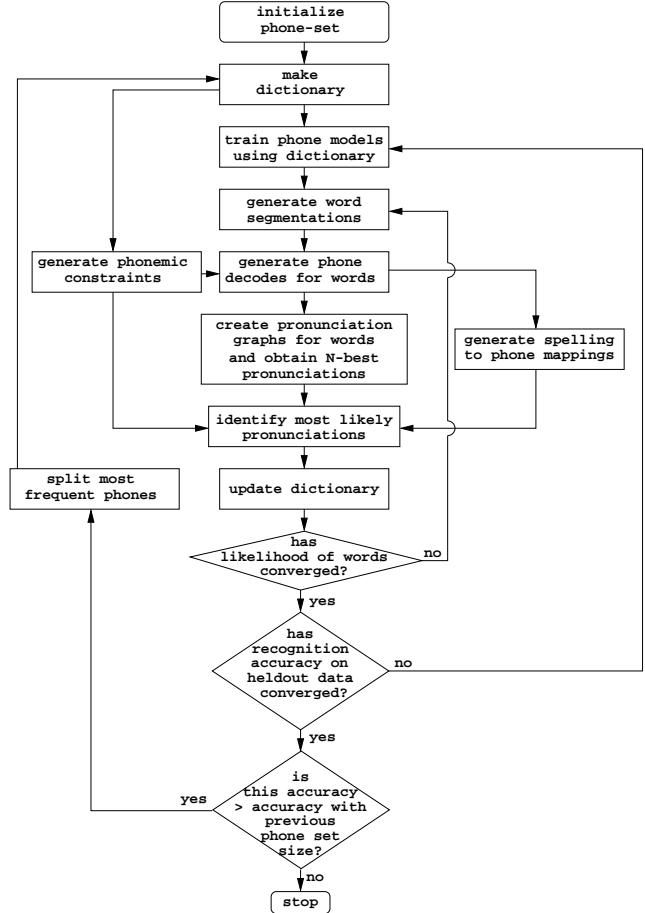


Figure 2: Flow chart for Phone definition and Lexical generation

A baseline was established using the CMUdict [6], which uses a set of 50 manually-designed phonetic units. Although the Resource Management task has a very constrained linguistic structure, the experiments took minimal advantage of it, by using a very low language weight for all experiments. The dictionary to be derived was initialized with the 26 symbol alphabet of the English language.

Figures 3 and 4 below show the results obtained during various stages of the experiment. In these figures the model update steps are indicated by Roman numerals (I,II,..), and the dictionary update steps are indicated by Arabic numerals (1,2,..). The phone set expansions (obtained by splitting phones) are indicated as  $a \rightarrow b$ , where  $a$  refers to the size of the phone set prior to splitting and  $b$  refers to the size of the phone set after splitting. Figure 3 shows that the likelihood of the training data increases monotonically with the model and dictionary updates and becomes equal to the baseline (with manually designed dictionary and phone set) with only 34 phones, increasing further over the baseline as the number of phones was increased to 42. From Figure 4 and Table 1 we see that the best word error rate obtained is for 34 phones. When the phone set size is increased to 42, the likelihood continues to increase but the word error rate observed for the held out set increases.

The resultant automatically-generated 34-symbol phoneset and the corresponding dictionary were evaluated by building context dependent semi-continuous HMMs with 2000 tied states. For comparison purposes, context-dependent models with 2000 tied states were also built for the baseline system. States were tied using decision trees. For both the baseline and the automatically derived phoneset the linguistic questions used in the decision trees were automatically generated [2]. Table 1 lists the word error rates obtained.

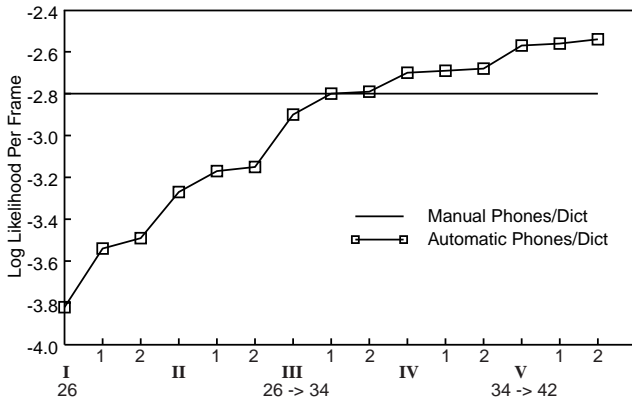


Figure 3: Likelihood as a function of iteration number

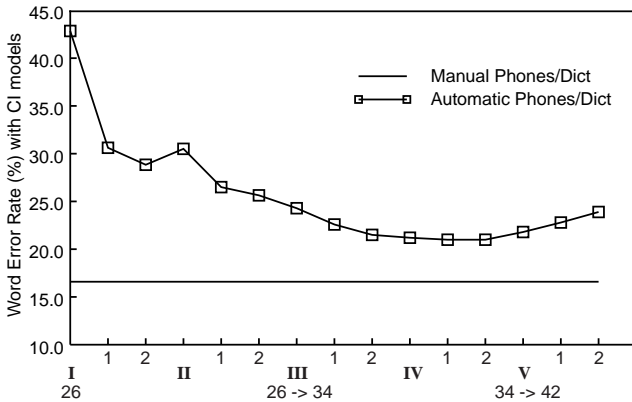


Figure 4: Word error rate as a function of iteration number

## 5. CONCLUSIONS

We demonstrated that it is possible to build up a complete phoneset, including the identities of the sub-word units as well as their dictionary pronunciations, in a completely automatic fashion given only a speech database and its correct orthographic transcription. The speech recognition accuracy obtained with the automatically-generated phoneset in English approached that obtained using a phoneset developed manually over many years by expert phoneticians. The performance of the automatically-generated phoneset was limited by the small size of the graphs used to search for optimal pronunciation. Also, the entire procedure was implemented using only context-independent phonetic models for economy of computation.

Type of phoneset	No. of phones	WER(%) with CI models	WER(%) with CD models
Automatic	26	26.1	
Automatic	34	21.2	12.6
Automatic	42	24.0	
Manual	50	17.2	9.2

Table 1: Word error rates with context-independent (CI) and context-dependent (CD) acoustic models as a function of the number of phones

Since the problem of generating a complete lexical representation is highly unconstrained in its current format, adequate training data are required to capture the optimal sub-word units. While the optimization problem has been formulated in a compact framework, the solution obtained is not optimal, since the objective function used – the likelihood of the training data – may not be suited to the problem. This is evidenced by the trend in Figure 3 where the likelihoods obtained become higher with the automatically generated phones than those with the hand-crafted ones in just a few iterations. Nevertheless, the recognition accuracy does not follow the same trend. We believe that these encouraging results can be improved on through the use of a more appropriate objective function as well as the use of context-dependent pronunciation modelling.

## 6. ACKNOWLEDGEMENT

This research was sponsored by the Department of the Navy, Naval Research Laboratory under Grant No. N00014-93-1-2005. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.

## 7. REFERENCES

- [1] Sloboda, T., and Waibel, A., *Dictionary learning for spontaneous speech recognition*, Proc. ICSLP 1996, Vol. 4, pp.2328-2331, 1996
- [2] Singh, R., Raj, B., and Stern, R. *Automatic clustering and generation of contextual questions for tied states in hidden markov models*, Proc. ICASSP'99, Phoenix, Arizona, March 1999.
- [3] Riley, M., Ljolje, A., Hindle, D., Pereira, F., *The AT&T 60,000 word speech-to-text system*, Proc. Eurospeech 95, Madrid, Spain, September 1995
- [4] Singh, R., Raj, B., Stern, R. M., *Probabilistic deduction of symbol mappings and automatic generation of phone sets and lexical transcriptions* To be submitted.
- [5] Nilsson, N. J., "Problem Solving Methods in Artificial Intelligence", Mc-Graw Hill, N.Y., 1971
- [6] <http://www.speech.cs.cmu.edu/cgi-bin/cmudict>