Published in the proceedings of Eurospeech'91, Genova, Italy, September 24-26, 1991, Vol. 2, pp. 793-796

Improving Short-Time Speech Frame Recognition Results by Using Context

Kari Torkkola, Mikko Kokkonen, Mikko Kurimo, and Pekka Utela
Helsinki University of Technology, Laboratory of Information and Computer Science
TKK-F, Rakentajanaukio 2 C, SF-02150 Espoo, Finland
email: kari.torkkola@hut.fi

ABSTRACT

This paper focuses on comparing three approaches to improve the accuracy of classifying short-time speech frames into phoneme classes by taking into account the classifications of nearby frames, also individually classified. We investigate whether this improvement has an effect to the accuracy of transcribing speech into phoneme sequences using two different decoding schemes, one based on simple durational rules, and the other on hidden Markov models (HMMs). The experiments indicate that recognition accuracies can indeed be improved significantly by taking the local context into account.

1 INTRODUCTION

"More is to be gained by discovering suboptimal ways of handling context than by discovering optimal ways of handling the local structure", Haralick stated in [2]. In this paper we explore some options to take advantage of local context in improving classification accuracy. Our framework is a phonemic speech recognizer based on classifying short-time feature vectors. The viewpoint taken is that of static pattern recognition. Individual or concatenated feature vectors are classified independently into phoneme classes. By utilizing context, the classifications are forced to be more consistent, or conforming to what is expected from speech.

Here, we present a comparison of three such methods, one that needs no training, namely relaxation labeling, and two others trained on the basis of examples, one based solely on symbolic processing, and the other on multilayer feedforward networks (MFNs).

The language used in this work is Finnish. Some important characteristics of this language are: 1) it contains 21 phonemes in its original form, 2) graphemes and phonemes correspond to each other almost identically, 3) the words are highly inflected, for example, a single verb can have over 1000 different inflected forms in which the baseform of the word can also change. Compared to the English language, the number of phonemes in Finnish is quite small. Furthermore, a large proportion of the phonemes are quasistationary (they have a stationary part), and thus are relatively easy to recognize. The choice of phonemes as the first recognition level is therefore natural. The approach used in this work is not limited to Finnish only. Its applicability to Japanese has been demonstrated [8]. Other phonemic-like languages, such as Italian or Spanish, might also be suitable for the suggested approach.

2 TRANSCRIBING SPEECH INTO PHONEMES BY MEANS OF STATIC PATTERN RECOGNITION

The starting point of this work is the phonemic speech recognition system developed at Helsinki University of Technology. This system has been presented in detail in references [8], [5], and [13]. A short description will be given below.

The system is based on adaptive vector quantization methods of Kohonen, especially on the *Learning Vector Quantization (LVQ)* [6]. The aim of vector quantization here is not to represent speech feature vectors with the smallest possible distortion, but with the best possible separation between phoneme classes. The LVQ accomplishes this by modifying the codebook vectors adaptively so that the borders between phoneme classes (midpoints between codebook vectors of different classes) approximate Bayes' decision surfaces. Good initial codebook vectors for the LVQ are obtained by using the Self-Organizing Maps (SOMs) [6] to cluster the initial training data. These maps have also the property of preserving the topological ordering of the input data, but this property is not taken advantage of, except for display purposes [7].

The feature vectors used in the system are 20-component mel-spaced cepstral coefficients liftered by raised sine [13]. They are computed every centisecond from a window of two centiseconds. Vector quantization by LVQ results in a string of phoneme labels at centisecond intervals, which is called the *quasiphoneme* string.

Context-independent duration-based rules are then used to compress the quasiphoneme sequence into a sequence of phonemes [7]. A decoding alternative giving higher accuracies is to use phoneme-based HMMs [13]. Coarticulation effects and other systematic errors are corrected at phoneme symbol level by context-dependent transformation rules created from examples by using a method called Dynamically Expanding Context (DEC) [4]. The error rate of the system as the sum of inserted, deleted, and misrecognized phonemes varies between 5% and 10% depending on the speaker and on the speech material.

3 IMPROVING CLASSIFICATION ACCURACY BY USING CONTEXT

In this paper we examine and compare three different methods to take context into account already at short-time feature vector classification before phonemic decoding. The methods fall into two distinct categories. *Relaxation labeling*, the only method in the first category, is a method widely used in image analysis to improve the consistency of contradictory or inconsistent classification [3], [2]. To apply it to quasiphoneme sequences, probabilistic classification information must first be derived from the LVQ. The probabilities are updated iteratively in order to increase the consistency of the labeling. The labelings of neighboring units are supposed to become compatible with each other.

The methods in the second category differ from relaxation in that they are trained by examples to map the quasiphoneme sequence closer to the ideal one. By the ideal quasiphoneme sequence we denote here a sequence, that has only labels of the correct phoneme inside the phoneme borders. We assume that the deviations from this kind of perfect sequences are systematic, mostly caused by coarticulation, and can be learned from examples.

The two mapping methods are quite different in nature. *Dynamically Focusing Context (DFC)* operates solely on symbols, in this case, on the quasiphoneme sequence. It generates automatically a large set of context-dependent transformation rules to do the mapping. On the other hand, the *multi-layer feed-forward network (MFN)* based mapping scheme uses the same kind of probabilistic labeling information that is fed to relaxation, too.

In the rest of this section we concentrate on describing relaxation labeling as we have applied it to quasiphoneme sequences. The two mapping methods have been described in detail in [12] and [14]; they are described here only superficially.

3.1 Relaxation labeling

In this section, speech feature vectors are denoted as consecutive units s_i. Each s_i has m labels with corresponding initial labeling probabilities $p_i(\lambda), \lambda = 1, \cdots, m$ given by a pattern recognizer. We have derived these probabilities from the distances between the feature vector s and the closest codebook vectors of LVQ for each class. The response w_i of a codebook vector f_i $(i = 1, \cdots, M)$ to s is derived from Euclidean distance. x_i is the ratio of that response to the best response.

$$w_i = \frac{1}{1 + \mathrm{distance}(s, f_i)} \tag{1}$$

$$x_i = \frac{w_i}{\max_j(w_j)} \tag{2}$$

The best x_i from each group of codebook vectors representing the same pattern class is chosen to be the likelihood of that label x_λ. The approximate labeling probability is x_λ divided by the sum of the likelihoods of all labels.

$$p(\lambda) = \frac{x_\lambda}{\sum_{\lambda'=1}^{m} x_{\lambda'}}, \ \lambda = 1, \cdots, m \tag{3}$$

Another alternative is to have "soft" labels attached to codebook vectors. Each codebook vector has then already a label probability vector. The final probability vector is computed as the weighted average of the probability vectors of N closest codebook vectors [9].

In order to eliminate unlikely combinations the relaxation algorithm attempts to use compatibility relationships $r_{ij}(\lambda, \lambda')$ predefined over each different pair of labels attached to neighboring units i and j. $r_{ij}(\lambda, \lambda')$ is a kind of heuristic correlation coefficient having values in the range $[-1, 1]$. The compatibility between the labelings of two units i and j can be defined by combining the transition probability matrix of the phonemes, the lengths of the phonemes and the time difference of the units. The latter was defined as a monotonically decreasing function of the difference $|i - j|$. In other words, the closer two units are, the larger the compatibility, and the more often two labels occur close together, the larger the compatibility.

A credibility coefficient for the labeling of unit i is a sum of compatibilities of different labelings λ' of other units j weighted by their initial probabilities.

$$q_i(\lambda) = \sum_{j=1}^{n} \sum_{\lambda'=1}^{m} r_{ij}(\lambda, \lambda') p_j(\lambda') \tag{4}$$

The credibility can be intuitively understood as the opinion of the context on unit i having the label λ. These credibilities for different labelings can be used to iteratively update the labeling probabilities [11]:

$$p_i^{new}(\lambda) = \frac{p_i(\lambda)[1 + q_i(\lambda)]}{\sum_{\lambda'=1}^{m} p_i(\lambda')[1 + q_i(\lambda')]} \tag{5}$$

Other formulations also exist to iterate the recognition probabilities [3], [10].

Relaxation effectively reduces the number of labels attached to each unit until only locally consistent label set remains. However, it is not necessary to iterate the probabilities until one label achieves 100% probability. The iteration can be stopped when a label set dominates clearly the observation sequence. In the experiments, after 5 - 10 iterations the improvment was only marginal.

In the example of Figure 1 the initial distribution of the labeling probabilities is shown around the plosive in a word */jonka/*. Each column is a distribution of one quasiphoneme in per cents. As can be seen from the second distribution the profile of the distribution becomes sharper after relaxation. The probabilities below 1% are marked as dots.

```
INITIAL LABEL DISTRIBUTION
i:  4  5  5  3  4  2  2  2  2  2  2  2  2  2  2  2
e:  3  4  3  2  4  3  3  3  3  2  2  3  4  3  2  3
h:  5  6  6  5  7  9 12  6  7  4  2  2  3  3  3  6
k:  3  3  3  5 14 28 22 22 19 10 10  6  6  6  8  9
s:  1  2  2  2  5  5  4  3  3  1  1  1  1  1  1  1
r:  5  4  4  4  7  7  9  8  9 13 11  8  9  8  7 10
{:  2  3  2  2  4  4  4  4  4  3  2  3  4  4  3  7
y:  2  4  3  3  3  2  2  2  2  2  1  2  2  1  1  1
l:  6  5  7  5  5  3  3  4  5  8  5  6  6  5  7  4
|:  2  3  2  2  3  2  3  3  3  3  3  3  4  3  2  3
j:  5  5  5  8  6  4  3  2  2  2  1  2  2  1  2  2
n: 16 14 11  7  5  3  3  4  3  4  4  4  4  4  4  4
v: 10  7 11 26  9  5  5  5  5 12  7  6  7  8  9  6
o:  3  4  4  2  3  3  5  6  9 11 22 20 15 17 12  9
a:  3  3  3  2  5  6  9  9 11  9 14 16 15 19 18 19
m: 17 13 17 11  6  3  3  4  2  3  3  4  3  3  4  3
u:  5  7  6  3  3  2  2  3  3  4  3  4  4  3  5  3

MPL:m  n  m  v  k  k  k  k  k  r  o  o  o  a  a  a
(MPL = Most Probable Labeling)

LABEL DISTRIBUTION AFTER 3 ITERATIONS
i:  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
e:  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
h:  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
k:  5 15 24 36 54 73 74 71 62 49 36 18 11  5  3  1
s:  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
r:  .  .  .  .  1  1  2  2  2  3  2  2  2  1  1  .
{:  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
y:  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
l:  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
|:  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
j:  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
n: 57 47 32 19 13  6  2  .  .  .  .  .  .  .  .  .
v:  8  8 13 21 13  8  6  5  3  2  .  .  .  .  .  .
o:  .  .  .  .  .  .  3  6  9 15 19 20 17 17 15 11
a:  .  .  .  .  1  1  6 11  2 20 40 57 68 75 79 86
m: 27 25 27 20 13  6  2  1  .  .  .  .  .  .  .  .
u:  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .

MPL:n  n  n  k  k  k  k  k  k  k  a  a  a  a  a  a
```

Figure 1. The effect of three iterations of relaxation labeling to phonemic labeling probabilities of consecutive feature vectors. Labels { and | denote vowels /æ/ and /œ/, respectively.

Relaxation labeling can be considered quite speaker independent, because the best parameters were almost the same, and the improvement in recognition accuracy was similar for each speaker in our experiments.

3.2 Dynamically Focusing Context

Dynamically Focusing Context (DFC) [12], [14] operates on symbolic quasiphoneme sequences to transform them closer to ideal sequences. Such quasiphoneme sequences facilitate easier and more accurate phonemic decoding. DFC is a derivative of the *Dynamically Expanding Context (DEC)* [4]. This method was developed to automatically derive context-dependent rules for the task of correcting phonemic transcriptions. The DFC is a modification of the DEC to better suit to fine-grained data such as quasiphoneme sequences.

The purpose of the DFC is to construct rules from examples to implement a mapping between symbolic data streams. In this application the source stream is the quasiphoneme sequence, and the target stream is the corresponding ideal sequence. The basic idea is to derive context-dependent rules that will be as general as possible. The generality here is defined as taking as low resolution view of the context as possible. At a low resolution level the condition parts of two mapping rules might look the same, but the production parts may be different. To solve this kind of conflicts between rules, the resolution by which the condition parts are determined is increased (focused) [12].

The algorithm may generate 10.000–20.000 mapping rules, when 900 words are used for training. The whole rule set is stored as a vast tree structure, whose depth is limited to, say, ten. The usage of the rule set for quasiphoneme mapping does not pose heavy computational loads due to this fast tree search.

3.3 Multi-Layer Feed-Forward Nets

The method based on MFNs is an enhanced version of the scheme presented in [14]. This scheme employs a feedforward network with one hidden layer of 200 units. The input to the network is composed of a labeling probability vectors. The network is trained to produce the ideal code for the center of the input window. There are 19 output units corresponding to different phonemic classes. The value of one of them is trained to be one, and the rest zero.

The network uses a multiple resolution input built from a 400 millisecond window sliding over the label probability vector sequence. The input is constructed by taking a series of gradually focusing windows of the sequence. These windows are presented to the network as parallel inputs. Each of the windows, independent of its width, feeds 19 input units dedicated to different phoneme classes that are distinguished by the LVQ. The input values for these units are constructed by averaging the 19-component label probability vectors within a window. The longest one of these windows comprise 160 msec, whereas the shortest ones include only one vector. The total number of these focusing windows is 16.

In the scheme presented in [14], instead of labeling probability vectors, we used only the best label derived from the LVQ. For each window, the 19 component input to the network was constructed as proportions of discrete labels within the window. The difference of these two schemes is significant, in favor of the probability vectors, as can be seen in Table 1.

4 EXPERIMENTS

Three male speakers were used to evaluate the presented schemes. Four independent repetitions of a set of 311 utterances were used. The amount of phonemes in one set was 1737. Three sets were used for training and the remaining one to evaluate the phonemic recognition accuracies by the leave-one-out principle. The recognition procedure after training was the same for all of the methods:

1. Classification of the feature vectors using LVQ. The result is a sequence of label probability vectors. The sequence of most probable labels can be written as the quasiphoneme sequence.
2. The sequence was transformed using one of the described methods.
3. The transformed quasiphoneme sequence was decoded into a phoneme sequence using identical duration-based rules for all of the methods, or by using phoneme based HMMs [13].

The original procedure against which the presented methods were compared consisted only of steps 1 and 3 (row *b-LVQ* in Table 1).

The unvoiced plosives were treated as a single class in these experiments. All other phonemes of the original Finnish language were recognized separately, giving 19 classes altogether. The results in table 1 exhibit the accuracy of combined detection and classification, that is, the accuracy of transcribing speech into phonemes. The beginnings and endpoints of utterances were assumed to be known, the figures thus do not contain errors due to their imperfect detection. All figures in the table are per cents of the correct number of phonemes in the test set. The column *errors* denotes the sum of insertion, deletion, and replacement errors, and the column *accuracy* denotes the amount of correctly recognized phonemes. All differences in the table are significant according to matched-pairs test [1].

The method to produce the quasiphonemes	rule-based decoding		HMM decoding	
	accuracy	errors	accuracy	errors
b-LVQ	90.6	14.1	94.5	7.7
p-LVQ + relaxation	91.1	12.1	93.2	9.0
b-LVQ + DFC-mapping	93.7	8.8	95.1	7.1
b-LVQ + MFN-mapping	94.6	7.0	95.7	5.8
p-LVQ + MFN-mapping	95.3	6.0	96.6	4.9

Table 1. The results of transcribing speech into phonemes. b-LVQ denotes that only the best label given by LVQ is utilized; p-LVQ denotes the probabilistic alternative, in which the probabilities of all labelings are used.

5 DISCUSSION

Trying first to recognize phonemes seems to be well suited to the speaker-dependent case as can be seen in the results of this work. In the experiments done with three male Finnish speakers it was found out that especially for the simpler of the decoding schemes, the durational rules, all of the approaches reduce significantly the error rate, at best 57 per cent. A 36 per cent reduction in the error rate was achieved even for the HMM-based decoding, which in itself already takes some context into account.

The performance of relaxation labeling fell short of that of the DFC and MFNs. Some improvement was achieved with the rule-based decoding, but the performance dropped with the HMM-based decoding. The reason for this might be that relaxation is basically a heuristic approach, and it does not seem to be straightforwardly applicable to data that has details of varying size, for example, short plosive bursts and long double vowels. Relaxation tends to favor long steady periods at the expense of short details. Under the circumstances, it seems better to use methods that can be trained by explicit examples to take these different kind of phenomena into account.

REFERENCES

[1] Gillick, L. and Cox, S. Some statistical issues in the comparison of speech recognition algorithms. In *Proceedings of the IEEE 1989 International Conference on Acoustics, Speech, and Signal Processing (ICASSP89)*, pages 532–535, Glasgow, Scotland, May 23-26 1989.

[2] Haralick, R. M. Decision making in context. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(4):417–428, July 1983.

[3] Hummel, R. A. and Zucker, S. W. On the foundations of relaxation labeling processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(3):267–287, May 1983.

[4] Kohonen, T. Dynamically Expanding Context, with application to the correction of symbol strings in the recognition of continuous speech. In *Proceedings 8th International Conference on Pattern Recognition (8th ICPR)*, pages 1148–1151, Paris, France, Oct. 27-31 1986.

[5] Kohonen, T. The 'Neural' Phonetic Typewriter. *IEEE Computer*, 21(3):11–22, March 1988.

[6] Kohonen, T. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

[7] Kohonen, T., Mäkisara, K., and Saramäki, T. Phonotopic maps – insightful representation of phonological features for speech recognition. In *Proceedings of the 7th International Conference on Pattern Recognition (7th ICPR)*, pages 182–185, Montreal, Canada, July 1984.

[8] Kohonen, T., Torkkola, K., Shozakai, M., Kangas, J., and Ventä, O. Phonetic Typewriter for Finnish and Japanese. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP88)*, pages 607–610, New York City, USA, April 1988.

[9] Laaksonen, J. A new reliability-based phoneme segmentation method for the neural phonetic typewriter. In *Proceedings of the 2nd European Conference on Speech Communication and Technology (Eurospeech91)*, Genova, Italy, September 23-26 1991.

[10] Peleg, S. A new probabilistic relaxation scheme. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4:362–369, 1980.

[11] Rosenfeld, A., Hummel, R. A., and Zucker, S. W. Scene labeling by relaxation operations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:420–433, 1976.

[12] Torkkola, K. A combination of neural network and low level AI-techniques to transcribe speech into phonemes. In *Proceedings of COGNITIVA-90*, pages 637–644, Madrid, Spain, November 20-23 1990.

[13] Torkkola, K., Kangas, J., Utela, P., Kaski, S., Kokkonen, M., Kurimo, M., and Kohonen, T. Status report of the Finnish phonetic typewriter project. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN-91)*, Espoo, Finland, June 24-28 1991.

[14] Torkkola, K. and Kokkonen, M. A comparison of two methods to transcribe speech into phonemes: A rule-based method vs. back-propagation. In *Proceedings of 1990 International Conference on Spoken Language Processing (ICSLP90)*, volume 1, pages 673–676, Kobe, Japan, November 18-22 1990.