

Phoneme-based Recognition of Finnish Words with Dynamic Dictionaries

F. Seydou^{*}, T. Seppänen^{*}, J. Peltola[†], P. Väyrynen^{*}

^{*}University of Oulu, Information Processing Laboratory, Oulu, Finland

[†]VTTElectronics, Oulu Finland

Abstract: In this paper we present an isolated word recognition system using first HMM to recognize the underlying sequence of phonemes, then DP and phoneme-gram matching technique to determine the corresponding nearest idealized phoneme sequences in the dictionary. Our approach is based on the observation that there is almost a one-to-one match between phonemes and letters in the official written Finnish language and a representation of the words in the dictionary as phoneme sequences. A significant advantage of our system lies in its ability to easily modify the dictionary without retraining the ASR models. Speaker dependent word recognition experiments can show an achievement of 95% recognition rates.

1. Introduction

It seems to be a fact that in most ASR applications it is essentially impossible for the (predefined) dictionary to cover all the words. For this reason it is highly desirable to implement an ASR system with a dynamically changing dictionary where the recognition is made in several steps: a first step that is independent of a (given) dictionary and higher steps (post-processing) with a dictionary-dependent recognizer.

The post-processing technique differs from the existing approaches where multiple knowledge sources are combined into a single search (see, e.g. [4]). Several authors have studied this approach for continuous speech (see, e.g. [1, 10]). In those methods the second stage is processed using statistical methods. For isolated word recognition a number of recent studies were made (see [2] for speaker-dependent and [7] for speaker-independent recognition systems) using Neural Network in the first stage and dynamic programming (DP) as a post-processing algorithm.

In the present paper we divide the ASR system into three stages: (1) we use HMM to recognize the sequence of phonemes that make up the word to be recognized; (2) since the official written form of Finnish language is accurate enough to be used as the phonetic reference transcription, we take the output of the phoneme recognizer as input, the phoneme sequences in the dictionary as reference data and a Levenshtein type DP-matching (see [3], p. 154) is used for generating the N-best candidates; (3) finally a phoneme-based n-gram matching is used to select one of the N-best candidates of the second stage as the recognized phoneme sequence.

In the following, we first present the method of the phoneme recognition component. We then describe the word recognition algorithm and finally report on experimental results of the system.

2. The method for phoneme recognition

The phoneme recognition is performed for discrete dictated words. The recognizer utilizes discrete hidden Markov models (HMM) and each phoneme has a separate model [8,9]. The output of the HMM decoding is a sequence of phoneme symbols.

Sixteen Finnish phonemes were chosen as phoneme models. In addition separate models for the silence and the air flow at the end of a word were created. Plosive phonemes /k, g, p, b, t, d/ are not recognized at all, instead the silence model is trained to include all of them. All models are five-state left-right HMM models. The models are using a multiple codebook approach [6] to produce discrete output symbols. Three codebooks are used for every frame of the speech features. One codebook contains the power and the power difference, the other two are used for the MFCC coefficients and for the difference of the MFCC values. The length of the delay for calculating the difference values was 80ms.

The phonetic Hidden Markov Models were trained separately with hand-labeled training data by using the Baum-Welch algorithm. In the recognition stage, spoken word utterances are automatically segmented and classified to symbolic phoneme sequences. The trellis for the Viterbi algorithm is constructed by representing every phoneme in a single trellis. The path can move from the last state of any phoneme to the first state of every other phoneme. The trellis is updated until the model's silence have been the best state for 700ms, which marks a silence between words. After that the best path through the trellis is determined by backtracking the best transitions. The phoneme sequence is obtained by identifying the corresponding model states in the best path.

3. The word recognition algorithm

As shown in the previous section the phoneme recognition system (which can be described as an operator \mathcal{A} that consists of the phoneme recognizer as well as the dialect of the speaker, the speaker condition, the acoustic environment, the microphone, etc.) is applied to a word utterance w and produces a phoneme sequence p . The phoneme sequence p is expected to describe fairly well the utterance w . In Finnish language p and w are expected to be almost identical. That, however, will not be the case because of the components that the operator \mathcal{A} contains. Our goal is now "how to build a model so that we can recover w from p ?"

To this end, since we can exploit the fact that there is a one-to-one relationship between the letters and the phonemes in Finnish, we shall first represent the words in the dictionary as phoneme sequences. Then, we present a straightforward technique (post-processing) for detecting and correcting recognition errors in two stages: (1) we implement an optimization method based on DP-matching to generate a list of N -best phoneme sequences in the dictionary that match the phoneme sequence p ; (2) we use a phoneme-based n -gram to select one of the N -best lists as the recognized phoneme sequence. The whole architecture is shown in Figure 1.

DP-matching: First, for each phoneme sequence w in the dictionary and a given phoneme sequence p from the phoneme recognizer, we compute a distance matrix $ed[i,j]$ by the following Dynamic programming (DP) algorithm (see [11] for details):

```

Start
 $ed[0,0] := 0;$ 
foreach column  $i$  from 1 to  $length(w)$   $do$   $ed[i,0] := ed[i-1,0] + IC(w[i]);$ 
foreach row  $j$  from 1 to  $length(p)$   $do$   $ed[0,j] := ed[0,j-1] + DC(p[j]);$ 
foreach column  $i$  from 1 to  $length(w)$   $do$ 
  foreach row  $j$  from 1 to  $length(p)$   $do$  start
     $m1 := ed[i-1,j] + IC(w[i]);$ 
     $m2 := ed[i,j-1] + DC(p[j]);$ 
     $m3 := ed[i-1,j-1] + SC(w[i],p[j]);$ 
     $ed[i,j] = \min(m1, m2, m3);$ 
  end
end

```

where, in the algorithm, $w[i]$ and $p[j]$ are the i th and j th symbols for these sequences w and p , respectively; $IC(s)$ and $DC(s)$ represent the costs for insertion and deletion of the symbol 's', respectively, and $SC(s_1, s_2)$ is the cost for the substitution of the symbol 's₁' by 's₂'. For handling plosive phonemes, the silence model trained to include all of them (see Section 2) is replaced by the symbol '?'. In this paper we adapt a Levenshtein type costs. In particular, $IC = DC = 1$ and $SC(s_1, s_2) = 0$ if either s_1 and s_2 are the same or s_1 is a plosive phoneme and s_2 is the symbol '?', otherwise $SC(s_1, s_2) = 1$. After computing the distance matrix we obtain the edit distance between each utterance w in the dictionary and the sequence p , $ED(w, p) := ed[length(w), length(p)]$. If there is one utterance w in the dictionary whose phoneme sequence has the least $ED(w, p)$ value it will be chosen as the recognized utterance. If, on the other hand, there is more than one utterance with the least $ED(w, p)$ value then we have a list of N -best utterances that will be processed by phoneme-gram matching.

Phoneme-gram matching. The phoneme-grams of a sequence are its sub-sequences of adjacent phonemes. For example, the sequence 'kopioida' has 2-grams (bi-grams) $ko, op, pi, io, oi, id, da$. Then -gram matching between two sequences is to find the number of n -grams common to both. Here we first replace the plosive consonants in the dictionary utterances by the symbol '?'. Then we first use bi-gram matching. The utterances in the N -best list from the DP matching that has the highest match with the phoneme sequence p generate another M -best list. If $M = 1$ the utterance will be chosen as the recognized word. Otherwise, we repeat the same procedure using 1-gram (uni-gram) matching. If, in the latter case, we do not get one single utterance in the M -best list, then the sequence p will be declared as unrecognized.

4. Results

4.1. Performance of the basic phoneme recognizer

The training material consisted of about 1400 words. A Finnish text of about 300 words was read aloud three times and another text of 500 words was read aloud

once. The total amount of training material was 12650 instances of phonemes for 18 models. All instances were over 50ms long and the amount of instances per phoneme was between 200 and 1100.

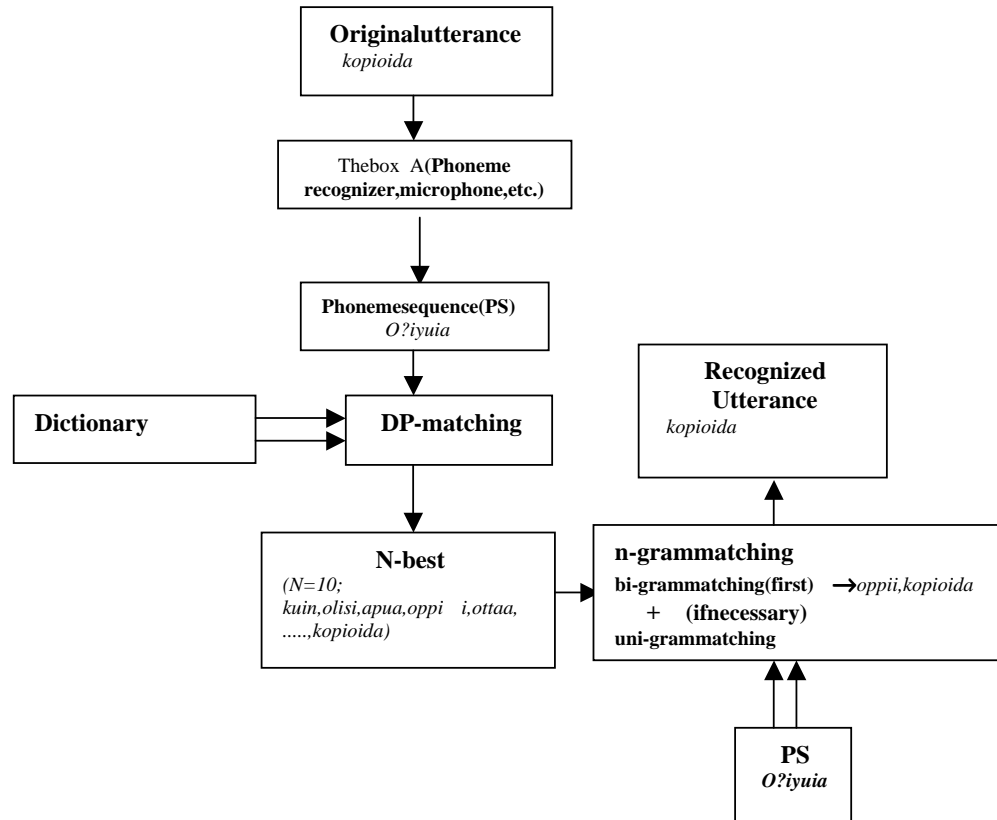


Figure 1. The system architecture

The same 300 -word text corpus was read aloud once more for generating test material for the phoneme recognizer. The test material contained a total of 2219 phonemes and the recognizer had technical prerequisites to recognize 1587 of them. The total number of correctly recognized phonemes is 1458, which leads to a percentage of correctly recognized phonemes to be 91.9%. The percentage of recognition errors is 15.6% and the distribution of the types of the recognition errors is 18.1% deletions, 41.5% additions and 40.3% replacements.

4.2. Performance of the word recognizer

For the recognition of the word utterances, our test data consists of 157 words randomly selected from the 3000-word test material discussed more closely in Section 4.1. The dictionary consists of 1200 different words from a text corpus.

With the complete dictionary we found a recognition rate of 85% when DP and n-gram matching are used and 66% when only DP matching is used. Next, we decreased the size of the dictionary step by step, with the test words being selected from the active dictionary. We found that, as expected, the smaller the size the better performance we get in the recognition. The smallest dictionary size was 160 words, and the corresponding performance figures were 95% and 85% with our enhanced method and the basic DP method, respectively. For each size of the dictionary the use of DP and n-gram matching gives a significant increase of 10%–17% in the performance of the recognition compared to when we only use the basic DP matching. The recognition rates obtained for different sizes of the dictionary are summarized in Figure 2, where the rate is plotted against the size of the dictionary. The curve on top is when DP and n-gram matching are used and the one on the bottom is when only DP matching is used.

A further advantage of our recognizer is that when the dictionary is changed there is no need to perform many retrainings of the recognizer.

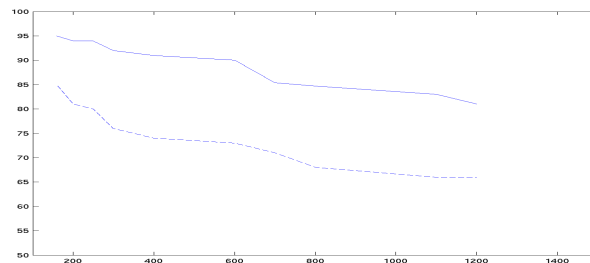


Figure 2. The word recognition rates (in percentage) against the dictionary size. On top when DP and n-gram matching are used. On bottom when n-gram is not used.

5. Conclusion

We have developed an ASR system for recognizing Finnish words. In the system phonemes are recognized first. Then, words are recognized by post-processing the output of the phoneme recognition. In the post-processing we first use DP matching to generate a list of N-best sequences and then a phoneme n-gram matching for selecting one of the N-best lists as the recognized word. An important property of our method is that the size and contents of a dictionary can be changed dynamically without changes required to the language models. We have found recognition rates between 85% and 95% depending on the size of the dictionary.

References

- [1] I. Bazzi & J. Glass. Heterogeneous Lexical Units for Automatic Speech Recognition: Preliminary Investigations. pp. 1257 - 1260. Proc. ICASSP 2000.
- [2] A. Hirai & A. Waibel. Phoneme-based Word Recognition by Neural Network - A Step Toward Large Vocabulary Recognition. Proc. IJCNN 1990. pp. 671 - 676.
- [3] D. Jurafsky & J. H. Martin. *Speech and Language Processing*. Prince Hall. 2000.
- [4] T. Kohonen. Expanding Context, with Application to the Correction of Symbol Strings in the Recognition of Continuous Speech. Proc. Of the 8th Int. Conf. On Pattern Recognition 1986. pp. 1148-1151
- [5] Kohonen T. Et Al. LVQ_PAK: The Learning Vector Quantization program package. Raportti A30, Otaniemi TKK, 1996.
- [6] Lee K -F. Automatic speech recognition, the development of the SPHINX system. Kluwer Academic Publishers, 1989.
- [7] Y. Matsuura, H. Miyazawa, T. Skinner. Word Recognition Using Neural Network and Phonetically Based DTW. *Neural Networks for Signal Processing [1994] IV. Proceedings of the 1994 IEEE Workshop.* Pp. 329 - 334
- [8] J. Peltola, Speech recognition with hidden Markov models. In Finnish. University of Oulu, Department of Electrical Engineering. MSEE thesis, 1998.
- [9] J. Peltola, Plomp J, Seppänen T. A dictionary-adaptive speech driven user interface for distributed multimedia platform. *Proceedings of the Euromicro workshop on multimedia and telecommunications 1999*, Volume II, Milan, Italy, September 8 - 10 1999, pp. 326 - 332.
- [10] E. K. Ringger & J. F. Allen. Error correction via a Post-processor for Continuous Speech Recognition. pp. 427 - 430. Proc. ICASSP 1996
- [11] R. A. Wagner & M. J. Fisher. The string-to-string correction problem. *Journal of the Association for Computing Machinery*, 1974, 21, 168 - 173.