

Experimental comparison of HMM and DP-matching methods for phoneme-based Finnish word recognition

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

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

fad@ee.oulu.fi, tapio.seppanen@ee.oulu.fi, pav@ee.oulu.fi

[†]VTT Electronics, Oulu Finland

Johannes.Peltola@vtt.fi

Abstract

In this paper we present and compare two isolated-word recognition systems for Finnish words. Both approaches are phoneme-based. Since there is almost a one-to-one match between phonemes and letters in the official written Finnish language, the words in the dictionary are represented as a sequence of phonemes. In the first method, a DP-matching is used to generate an N-best list and then a phoneme n-gram matching is designed to extract the correct solution. In the second method, the words are first detected from the audio stream, then the word models are tested against the detected words by the Viterbi algorithm and the best matching model is selected as the recognized word. A primary advantage of the two systems is their ability to easily modify the dictionary without retraining the ASR models. Our results suggest that, for small amount of training in the phoneme recognition component the second method is far better to use. On the other hand, when enough training material is available, then it may be better to use the first method, especially for large dictionaries where it performs much faster.

1. Introduction

In order to overcome the problem of large and changing dictionaries, several authors have developed systems where the word recognition component does not need to be retrained (see e.g. [9]). In particular, we have developed two methods [5,6] where the word recognition is phoneme-based and does not require any retraining of the language models.

The purpose of the present paper is to extend those works to different sets of material and provide a comparison.

For both methods, since the official written form of Finnish language is accurate enough to be used as the phonetic reference transcription, we take the outputs of the phoneme recognizer as input and the phoneme sequences in the dictionary as reference. The first method (method A) operates in two stages. First a dynamic programming (DP-matching) is used to find the list of the N most likely (N-best) matches from the utterances in the given dictionary. Then a phoneme n-gram method is used to select one of the N-best list as the recognized utterance.

In the second approach (Method B), the word models are assembled from the phoneme models by concatenating corresponding models into one word model. Then, the words are first detected from the audio stream, and the word models are tested against the detected words by the Viterbi algorithm

and the best matching model is selected as the recognized word.

From the above (simplified) presentations of the methods it is obvious that method A (that enjoys a dynamic programming algorithm of comparing two phoneme sequences) is much faster, especially for large dictionaries, than method B (where state probabilities have to be computed for every model).

In this paper we first present the two methods in more details. Then, we give results of experiments for different training data of the phoneme recognition as well as different sizes of the dictionary.

2. The phoneme recognition

The phoneme recognition is performed for discrete dictated words. The recognizer uses conventional approach to speech recognition by using MEL frequency cepstrum coefficient (MFCC) [1, 2] based speech features and discrete Hidden Markov Models (HMM) [3, 4] to model phonemes.

Sixteen Finnish phonemes were chosen as phoneme models. In addition there are also models for airflow at the end of a word and one model for the silence and all the stop consonants of the Finnish language. The recognizer is not capable to recognize plosives and it also can't detect them if they are located in the beginning or end of the word. All models were five-state left-right HMM models. The models are using three codebooks. One codebook contains the power and the differenced power, the other two are used for the MFCC coefficients and for the differenced MFCC values. The models were trained separately with hand labeled training data. Samples of Finnish words were broken down into labeled phonetic occurrences and the Baum-Welch algorithm was used to train phoneme models with corresponding occurrences.

In the recognition stage, spoken word utterances are automatically segmented and classified to symbolic phoneme sequences by the Viterbi algorithm. The Viterbi-trellis is constructed by representing each 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 'silence' have been the best state for 700 ms, which marks a silence between words. After that the best path through the trellis is determined by back tracking the best transitions. The phoneme sequence is obtained by identifying the corresponding models that lie in the best path. Results on the phoneme recognition will be discussed in Section 5.

3. Method A: The DP and n-gram method

In this method phonemes are recognized first using the phoneme recognition of section 2. Then, the words are recognized based on the recognized phoneme sequences. As stated in the introduction, we can exploit the fact that there is almost a one-to-one relationship between the letters and the phonemes in Finnish to represent the words in the dictionary as phoneme sequences. If w were the word utterance that was used in the phoneme recognizer to obtain the phoneme sequence p , we would like to recover the best match to w for the given p by first computing a Levenshtein type minimum edit distance between p and each utterance in a given dictionary (DP-matching, see [7] for details). If there is one utterance w_l in the dictionary whose phoneme sequence has the least minimum edit distance value it will be chosen as the recognized word utterance. If, on the other hand, there is more than one utterance with the least minimum edit distance value then we have a list of N-best solutions that will be processed by phoneme n-gram matching.

The phoneme n-grams of a sequence are its sub-sequences of n adjacent phonemes. For example, the sequence 'juoda' has as 2-grams (bi-grams) *ju*, *uo*, *od*, *da*. The n-gram matching between two sequences is to find the number of n-grams common to both. Here 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. Method B: HMM based word recognition

The phoneme recognizer presented in Section 2 can also be operated in a word recognition mode. The word models are assembled from the phoneme models by concatenating corresponding models into one word model. This way the simplicity of a word model based speech recognizer is achieved, but the dictionary is not confined to the training words [6].

Word recognition mode operates in two phases. First it detects the words from the audio stream, which is done by measuring the power of the incoming audio signal. After that all the word models are tested against the detected words and the best matching model is selected as the recognized word. Like in Section 3, word models are constructed from the written form of the word. When the phonetic description has been formed, concatenating corresponding phoneme models into one left-right model creates a word model. Figure 1 shows an example of the Finnish word model 'auto' that has been build by arranging corresponding phonemes /a/, /u/, /t/ and /o/ consecutively. The only valid initial state for the word model is the first state of the first phoneme. The successive phoneme models are tied together by allowing a state transition to occur only between the last state of the model to the first state of the next model. HMM models are tested against the array of the acquired speech features by using the Viterbi algorithm. The probability value of the last state of the word model declares

how well the audio signal represents the model in the dictionary when compared to other models. The best matching model is chosen as the recognition result by comparing the values in the last state and selecting the highest one.

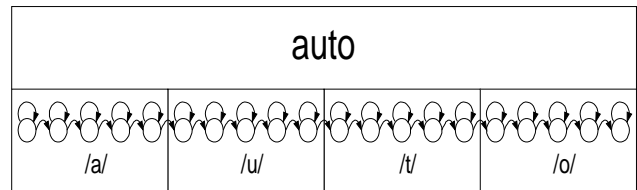


Figure 1: Word model 'auto' that has been build from four Finnish phoneme models

5. Results and discussion

For testing the two methods (A and B) we have used three different materials of the phoneme recognition. The full training material contained 1400 words (set 1), which consists of a Finnish text of 300 words read aloud three times and another text of 500 words read aloud once. All instances were over 50 ms long and the amount of instances per phoneme was between 299 and 1100. Then we used two other materials of 700 (set 2) and 350 (set 3) words taken from set 1.

The same 300 words were read aloud once more for generating a test material. The percentage of correctly recognized phonemes and recognition errors (distributed in deletion, insertion and replacement rates) are summarized in Table 1 below..

For set 3, the error rate is too high to expect good results in the word recognition. This will be seen below when we discuss the word recognition.

Looking at set 1 and set 2, we see that, while the error rates are close (15.6% and 18.8% respectively), there is at least two times more deletions in set 2. From this remark it is obvious- and our results will confirm this-that Method A will not give good results.

Table 1. The rate (in percentage) of the correctly recognized phonemes (CRP) and the recognition errors (the total (T) which is distributed in deletion (D), insertions (I) and replacement (R)), for different training materials to recognize 300 dictated words.

Number of words in the phoneme training.	CRP	Recognition error			
		T	D	I	R
Set 3 = 350	74.1	34.6	33.6	25	41.4
Set 2 = 700	85.5	18.8	41.7	22.8	35.6
Set 1 = 1400	91.9	15.6	18.1	41.5	40.3

For recognizing the word utterances, our test data consisted of 157 words randomly selected from the 300-word test material discussed above. Given the significance of compound words in Finnish language (65%, according to [8]), we decided to further divide the test set into short (length less or equal to 7) and long (length greater than 7) words. We implemented experiments for each of the training sets described above using different sizes of the dictionary. The full size of the dictionary

is about 1400 different words taken from a text corpus. It was further decreased to 700 and 200 words, with the test words being selected from the active dictionary. The results of the word recognition rates are summarized in Table 2 below.

Before discussing the recognition results some remarks are in order. First, we recall from the introduction that method A is a much faster algorithm for large dictionaries than method B. When the dictionary size is very large method B can be very time consuming. Next, as we mentioned above, it must be noted that when most of the phonemes within a word utterance are deleted method A can not give any reasonable result since the DP matching will generate many utterances in the N-best list that has nothing to do with the words that we are trying to recover. It is also obvious that, for both methods, the word recognition rates will increase when either the dictionary size decreases or the amount of training material for the phoneme recognition increases. This is clearly seen in Table 2. Now, let us look at the results for each training material more closely.

- For training set 3 we see that both methods yield to poor recognition rates for short, long and mixed (short and long combined) words for large dictionaries. The results, as we mentioned earlier, get better as the size of the dictionary decreases. Method B starts giving fairly good results for mixed and especially long words with smaller size of the dictionaries. Method A gives much poorer results in all figures, which is due to the poor recognition rate in the phoneme recognition. Method B shows its high performance for small sizes of the dictionary and Method A exhibits its high sensitivity to the amount of training material.
- The word recognition, when using training set 2, still gives poor results with method A with any size of the dictionary and length of the words. This is, as we pointed out above, due to the amount of phoneme deletions in the phoneme recognizer (see Table 1). Method B, on the other hand, gives reasonable results for large dictionary with mixed words, excellent rates for mixed words in a small dictionary, and for long words regardless of the size of the dictionary. For short words it gives poor results only for a large dictionary.
- Finally, for training set 1, method B gives slightly better result than for set 2. This is due to the low difference in the phoneme recognition rates between the two sets. Method A, on the other hand, had a very good improvement for all figures because of the lower rates for deletions in the phoneme recognition. For small sizes of the dictionary, the recognition rates are very close to the rates of Method B. For large sizes of the dictionary the results are quite satisfactory. This shows that, with a good amount of training material for the phoneme recognition, method A performs very well because of the higher phoneme recognition rates in that case. Since it is a much faster algorithm, we can state that, for well-trained phonemes, it is preferred to Method B, especially for large dictionaries.

Table 2. The results of word recognition rates (in percentage) for methods A and B, for different number of words in the phoneme training sets (see Table 1) and different sized of the Dictionary **D**. The results are for short, long and mixed (short and long) words.

Sets	D	Word Recognition rates					
		Mixed		Short		long	
		(A)	(B)	(A)	(B)	(A)	(B)
Set 3	200	61	81	57	67	63	87
	700	43	72	35	53	48	81
	1400	41	68	32	49	46	78
Set 2	200	69	95	72	86	68	100
	700	54	93	55	84	53	97
	1400	48	88	52	74	46	95
Set 1	200	95	97	92	92	97	100
	700	87	93	78	82	92	98
	1400	83	90	70	74	92	97

6. Conclusion

In this paper we have presented two different ASR systems for recognizing Finnish words. Both methods are phoneme-based. In the first system, the phonemes are recognized first. Then, based on the recognized phoneme sequences, words are recognized using a DP and phoneme n-gram matching. For the second method, the word models are assembled from the phoneme models by concatenating corresponding models into one word model. Then, the words are first detected from the audio stream, and the word models are tested against the detected words by the Viterbi algorithm. An important property of the two methods is that the changes in the size and contents of the dictionary do not require any changes to the language model. When comparing the two methods we found that the second method is much slower to run for large sizes of a dictionary. However, it does, in general, perform better than the first one. It is also much less sensitive to the training material. For smaller training material of the phoneme recognizer it can still give satisfactory results even for large dictionaries. The first method, on the other hand, performs very poorly for smaller training material. When the material for training the phonemes get larger, it gives satisfactory results for large dictionaries and very good rates for smaller sizes of the dictionary.

We have also shown that Method B can be successfully extended to large dictionaries, with a computation labor. When a small material for training the phoneme recognition is used, there is no need to use method A. If, however, the training material is large enough it may be better to use method A since, in that case, it gives satisfactory results and is a much faster algorithm than Method B.

References

- [1] Tohkura Y. (1987) A weighted cepstral distance measure for speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 35, pp. 1414-1422.
- [2] Davis S. B., Mermelstein P. (1980) Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28, pp. 357-366.
- [3] Lee K-F. (1989) Automatic speech recognition, The development of the SPHINX system. Kluwer Academic Publishers.
- [4] Rabiner L. R., Levinson, & Sondhi M. (1983) On the application of vector quantization and hidden Markov models to speaker-independent isolated word recognition. *Bell System Technical Journal* 62, pp. 1075-1105.
- [5] Seydou, F., Seppänen, T., Peltola, J., Väyrynen P. Phoneme-based Recognition of Finnish Words with Dynamic Dictionaries. Submitted.
- [6] Peltola J., Plomp J., Seppänen T. (1999) A dictionary-adaptive speech driven user interface for a distributed multimedia platform. *Proceedings of the 25th EUROMICRO Conference. Vol. II* pp 326 – 332.
- [7] D. Jurafsky & J.H. Martin. *Speech and Language Processing*. Prince Hall. 2000.
- [8] Vesikansa, J., Miljoona sanaa,, Söderström, Porvoo (1978).
- [9] 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