A Czech Large Vocabulary Recognition System for Real-Time Applications

Jan Nouza

Technical University of Liberec, Department of Electronics and Signal Processing,
Halkova 5, 461 17 Liberec, Czechia
jan.nouza@vslib.cz

Abstract. In the paper we propose two methods for speeding up discrete-utterance recognition in vocabularies with hundreds to several thousands of words. We show that acceptable results as well as short response time can be achieved if the words are represented by concatenated monophone models (multi-mixture HMMs). In such case, the computation load of the classic Viterbi procedure can be reduced significantly if a proper caching scheme is used. In several experiments done with test vocabularies containing hundreds and thousands of Czech words we demonstrate that the recognition procedures can be speeded up by a factor of 50 to 100 without a loss of accuracy. The method is well suited for voice controlled systems with a large branching factor and low syntax, i.e. in voice portals, telephone directory assistance, etc.

1 Introduction

Automated voice-operated services, mainly those allowing access over telephone, become more and more popular [1]. As an example, we can mention system Infocity developed at our university that has been in public use since March 1999 and recently it registers about 50 to 100 calls per day. The system offers information about culture, sport, transport and commerce in Liberec in form of a simple dialogue between a caller and the computer [2]. Up to now the speech input to the system was limited to a small vocabulary with tens to hundreds of words. For many commercially interesting applications, like a telephone directory assistance, automatic call routing, time-table inquiries, etc, larger vocabularies with several thousands entries are necessary.

These services to be practically applicable need a speech recognition module that proves a high accuracy and fast response. On the other side they need not to be designed necessarily for continuous speech processing. They often take input in form of short answers, either isolated words or phrases. A typical example is an automated telephone directory inquiry service: A caller is asked to specify the surname of the wanted person, his/her first name, the city name and the street name. At each step the recognition system must search within a large vocabulary, which may or may not be partly constrained by context following from the previous step.

In this paper we propose methods for speeding up classification from large lists of candidates in an isolated-word recognition task and we test them for Czech language.

2 Speech Databases for Training and Testing

A speech recognition system that is to be flexible must operate with an inventory of sub-word units. In our case these units are phonemes. To train their statistic models we had to design, record and annotate a large corpus of spoken Czech. For testing purposes we prepared several databases that try to address practical tasks.

2.1 A database for training Czech sub-word units

There are several strategies to make a good training database. One, described in [3], takes an available corpus of spoken utterances and attempts to select those fulfilling some statistical criteria. We used another approach proposed in [4]. A special set of 80 sentences was constructed in the way that each Czech phoneme was represented by a sufficient number of samples. The goal was to cover all biphones occurring in Czech and thus create a chance for training triphones by a merging or cloning method [5, 6].

The 80-sentence set covers all 41 single phonemes defined in [7]. The less frequent phonemes have at least 16 occurrences while the vowels, for example, occur in several hundred variants. The set covers almost 99 % Czech biphones and about 3500 different triphones. Up to now the sentences have been recorded by some 35 people, some provided multiple recordings. The database contains almost 10 hours of speech signal that has been segmented and labeled at the phoneme level [8].

2.2 Testing databases

For evaluating new methods and testing potentially interesting applications we have prepared and collected several databases. Their features are summarized in Table 1.

Table 1. Databases of Czech words used for testing

Database name	Vocabulary size	∅ phonemes per word	Confusability
Names	360	6.3	high
Surnames	800	7.5	middle
City names	5 346	8.7	middle
Voc10K	10 000	6.9	high

The first set includes all Christian names in the Czech calendar, the second is a selection of surnames from Liberec telephone directory, the third contains all local (city

and village) names in Czechia and the last consists of the most frequent 10 thousand words in a 80-M-word newspaper corpus. Each database has its specific features. For example, the Names set includes quite a lot of minimal pairs or groups, like *Jan-Jana*, *Daniel-Daniela*, *Alena-Ilona-Elena-Helena, Miroslav-Miloslav-Jaroslav*. In the City name set there are both very short and very long items, like *Aš* vs. *Blatnice_pod_Svatým_Antonínkem*. The Voc10K set often contains several morphological forms of the same word, from which some cannot be distinguished on acoustic level, e.g. *volal-volala-volali-volaly-volalo*.

Each database consists of 1000 test words recorded by 10 speakers through a standard front-end employed in our recognition engine. Hence, each recorded item passed through the same endpoint and parameterization procedure like in a live test.

3 Methods for efficient word recognition

The phonemes in our system are represented by three-state hidden Markov models (HMMs) with left-to-right structure and no skips, as shown in Fig.1. Output pdfs are multi-mixture gaussians defined in the 18-dimensional space of parameters derived from LP cepstrum. A word (or phrase) model is created by concatenating appropriate phoneme models. Monophones or triphones can be employed alternatively, as it is depicted in Fig.1

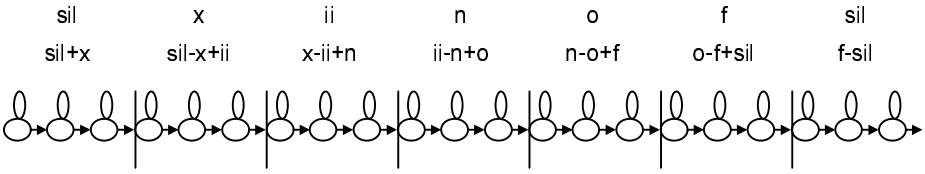

Fig. 1. A model of Czech word Chýnov created by concatenating monophone or triphone three-state HMMs.

After training models of different type and size and testing them on the Name set we got results that are summarized in Table 2. We can see that the score obtained with 16-mixture monophones is almost comparable with that of the triphones of the same size. However, the recognition time (measured on a PC 600 MHz) for the best models is too long (about 8 s for a word). A considerable reduction of the computation load is necessary if we want to employ the models in practical applications.

Table 2. Recognition results obtained for the Name set and different models

Model type	**Recognition score [%]**	**Average recognition time per word [ms]**
Monophones - 4 mixtures	87.1	2570

Monophones - 8 mixtures	88.2	4960
Monophones - 16 mixtures	91.5	8350
Triphones - 16 mixtures	91.9	8400

3.1 Reducing state likelihood computations

Anybody who implemented an HMM recognizer knows that most computation time is spent by calculating state output probabilities. If a signal represented by *F* frames is to be matched to *M* word models, each having (in average) *N* states, the total number of evaluations of multi-dimensional multi-mixture gaussian fuctions is equal to the product *F* x *M* x *N*. For typical values it may lead to 100 x 1000 x 20 = 2 M gaussian evaluations needed for the classification of a word. This large number can be reduced by adopting various schemes based on the beam search strategy. However, these schemes carry a potential danger of loss of accuracy, which may happen, particularly, in case when word endpoints are not determined correctly.

Our approach is based on reducing the number of likelihood calculations by caching the already computed values and reusing them if the frame is matched to the same state. The frequency of repeated calculations for the same frame-state pair is very high for monophones, because there are only 41 x 3 different states in a Czech system. Moreover, the cache hit rate increases with increasing the vocabulary. For illustration, if the caching scheme is used for the recognition within the Voc10k vocabulary, 99.94 % likelihood values can be fetched from the cache. Obviously, this rate is much lower for triphones, where there are several thousands of different states. The impact of the caching scheme on the computation time is demonstrated in Table 3.

3.2 Eliminating computations for same word sub-strings

Another savings in the matching (Viterbi) algorithm can be achieved if we realize that some words in the vocabulary share same sub-strings of phonemes, especially at initial parts. If the models of these words enter the matching procedure each after other, some of the accumulated likelihood values computed for the previous model can be reused for the next model. This is shown in fig.2

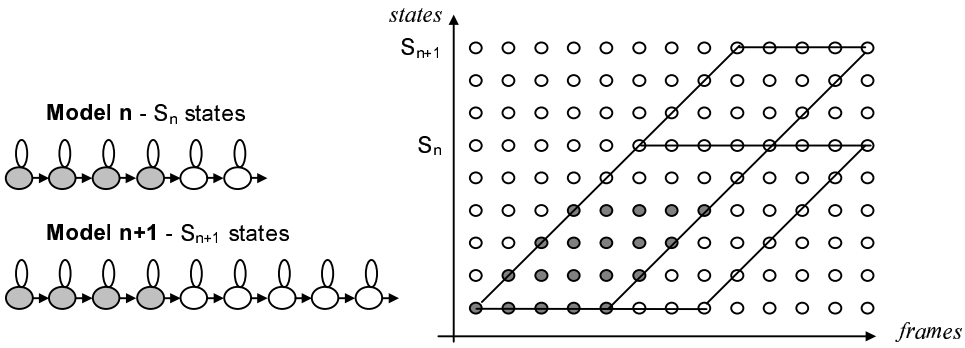

Fig. 2. An example of two models sharing the first 4 states. In the frame-state space there are points (filled by gray color) whose accumulated likelihood values computed for model *n* can be reused for model *n+1*. The demarcation lines delimit the Viterbi search space for each model.

Utilizing this fact and organizing the vocabulary in the way that the neighboring words enable the highest possible reduction factor we can save another significant portion of the computation load. When organizing the vocabulary we may use the alphabetic order but also we must take into account the length of the neighboring words (more precisely, the number of states of the corresponding models). The impact of this secondary cache scheme demonstrates Table 3.

Table 3. Recognition times achieved with the Names set for the original and the optimized computation schemes. The values represent average classification times per word. The C1 method is that described in section 3.1, the C2 is the combination of the two caching methods mentioned in this chapter.

Model type	Original time [ms]	C1 method time [ms]	C2 method time [ms]
Monophones - 4 mixtures	2570	160	90
Monophones - 8 mixtures	4960	190	110
Monophones - 16 mixtures	8350	250	160
Triphones - 16 mixtures	8400	2450	1780

4 Practical experiments

In the previous text we showed that it is possible to reduce the computation load of isolated-word recognition without a loss of accuracy. The reduction is considerable, in particular, for the system based on monophones. For triphones the caching scheme has only a small effect. Hence, a triphone-based system must rely on other paradigms

(e.g. the already mentioned beam search) that are sub-optimal and may introduce some additional errors.

We have implemented the methods into a word recognition system. Recently it operates with 16-mixture monophone models, that provide the same scores as the triphone ones, but in time 10 to 50 times shorter. The results from experiments with different test databases are summarized in Table 4. The recognition rate was significantly lower for the Voc10K database. This is because it contains many groups of words that can be distinguished only in a context.

Table 4. Recognition results from experiments with different test databases

Database type	Recognition score [%]	Recognition time per word [ms]
Names	91.5	160
Surnames	89.4	350
City names	87.1	1120
Voc10K	75.8	1260

5 Conclusions

Two methods are proposed for speeding up recognition of words or utterances spoken in isolated way. A considerable reduction of classification time is achieved, in particular, for a monophone-based system. In the recognition test with a 5-K word vocabulary (the City Names) the reduction factor was higher than 100 while, the score was still at an acceptable level for a practical task. It should be mentioned that the test data came from live recordings and therefore the reported results correspond with those obtained in on-line tests performed under computer room conditions.

Recently the recognition module based on the monophone models (adapted for telephone signals) have been built in automated information system InfoCity. In spite of the lower signal quality the system runs with average recognition rate about 90 %.

Acknowledgments. The research reported in this paper was supported by the Grant Agency of the Czech Republic (grant no.102/96/KO87), by the Czech Ministry of Education, through research goal project MSM 242200001 and by project Kontakt no. ME293.

References

1. Proc. of Workshop Voice Operated Telecom Services (Do they have a bright future?). Ghent, May 2000.

2. Nouza J., Holada M.: A Voice-Operated Multi-Domain Telephone Information System. Proc. of 25th Int. Conference on Acoustics, Speech and Signal Processing (ICASSP2000), Istanbul, June 2000 (to appear).
3. Radová V., Vopálka P.: Methods of Sentences Selection for Read-Speech Corpus Design. Proc. of the 1st Workshop on Text, Speech and Dialogue, Plze• 1999, pp.165-170.
4. Hájek D.: A Continuous Speech Recognition System. MSc thesis (in Czech). Technical University of Liberec, May 1998.
5. Hájek D., Nouza J.: A Quasi-Triphone Model Created by Merging Context-Specific Phone Models. In Studientexte zur Sprachkommunikation, Heft 14 (Elektronische Sprachsignal-verarbeitung), Cottbus 1997, pp.85-92
6. Ming J., O'Boyle O., Owens M., Smith F.J.: A Bayesian Approach for Building Triphone Models for Continuous Speech Recognition. IEEE Trans. Speech and Audio Processing, vol. 7, no. 6, 1999, pp. 678-684.
7. Nouza J., Psutka J., Uhlíø J.: Phonetic Alphabet for Speech Recognition of Czech. Radio-engineering, vol.6, no.4, Dec.1997, pp.16-20.
8. Nouza J., Myslivec M.: Creating and Annotating Speech Database for Continuous Speech Recognition. Proc. of. 4th ECMS Workshop, Liberec, May 1999, pp.147-151.