Design of Speech Recognition Engine*

Luděk Műller, Josef Psutka, and Luboš Šmídl

University of West Bohemia,
Department of Cybernetics,
Univerzitní 22, 306 14 Plzeň, Czech Republic
muller@kky.zcu.cz, psutka@kky.zcu.cz, smidl5@kky.zcu.cz

Abstract. This paper concerns a speaker independent recognition engine of Czech continuous speech designed for Czech telephone applications and describes the recognition module as an important component of a telephone dialog system being designed and constructed at the department of Cybernetics, at University of West Bohemia (UWB). The recognition is based on a statistical approach. The left-to-right three-state HMMs with an output probability density function expressed as multivariate Gaussian mixture are used to model triphones as basic units in acoustic modeling and stochastic regular grammars are implemented to reduce a task perplexity. A real time recognition process is supported by a very computation cost reduction approach estimating log-likelihood scores of Gaussian mixtures and also by a beam pruning used during Viterbi decoding. Presented paper concerns the main part of the engine - a speaker independent recognition engine for continuous Czech speech.

1 Introduction

This paper concerns the speaker independent recognition of Czech continuous speech for the telephone application and describes a recognition module designed for this purpose at the University of West Bohemia. The task of this module is to recognize and transcribe continuous Czech telephone speech to a word sequence restricted by a stochastic regular grammar. The paper is structured as follows. The short review of the dialog system is given in section 2. The section 3 describes the speech recognition engine, which is based on a statistical approach. The statistical approach means that the recognition module incorporates front-end, acoustic model, language model and decoding block providing the search for the best word sequence matching the incoming acoustic signal with respect to the grammar. The acoustic model is depicted in section 3.1. The remaining parts of the article deal in detail with the individual subsystems of the speech recognition engine. How the telephone signal is captured and preprocessed in front-end is briefly given in section 3.2. Section 3.3 deals with the block that evaluates HMM's output probability density functions. The grammar implementation is presented in section 3.4. Finally, the results and conclusions are given in section 4.

* The work was partially supported by the Ministry of Education of the Czech Republic, project no. MSM235200004, and by the Grant Agency of the Czech Republic, project no. 102/96/K087.

2 Dialog System

At the department of Cybernetic UWB an endeavor has recently been exerted to develop a laboratory applicable telephone dialog system. The dialog system consists of three main parts - a speech engine, dialog manager, and dialog application. The dialog application is a task-oriented module keeping knowledge on a lexicon, dialog structure etc., and the dialog manager controls a communications between users and the system. The speech engine contains only a speech recognition module today; a speech synthesis module will be added to the engine in the close future. The digitization of an input analog telephone signal and/or a generation of a synthesized speech signal are provided by a telephone interface board. Figure 1 illustrates the used system architecture.

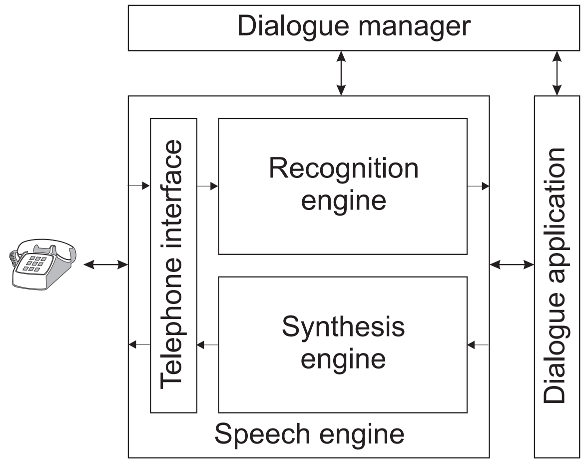

Fig. 1. Telephone dialog system

3 Speech Recognition Engine

The core of the speech recognition engine is implemented in C++ and designed as platform independent. The platform specific implementation layer has been built for an implementation on MS Windows NT/95/98/2000. Our goal was to design a very fast recognition module without decreasing recognition accuracy suitable for telephone dialog applications and after a memory optimization even for embedded systems. Several instances of the designed speech recognition engine can operate on one PC in a real time. Furthermore each engine module can be implemented as a set of several tasks (see Figure 2) each of them running as an individual process on a computer.

The speech recognition engine is based on a statistical approach. The task of this module is to recognize and transcribe continuous Czech telephone speech

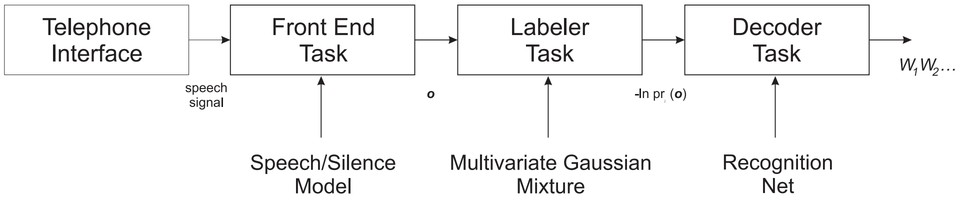

Fig. 2. Speech recognition engine

to a word sequence restricted by a stochastic regular grammar. The recognition module incorporates front-end, acoustic model, language model (represented by a stochastic regular grammar) and decoding block providing searching for the best word sequence matching the incoming acoustic signal with respect to the grammar.

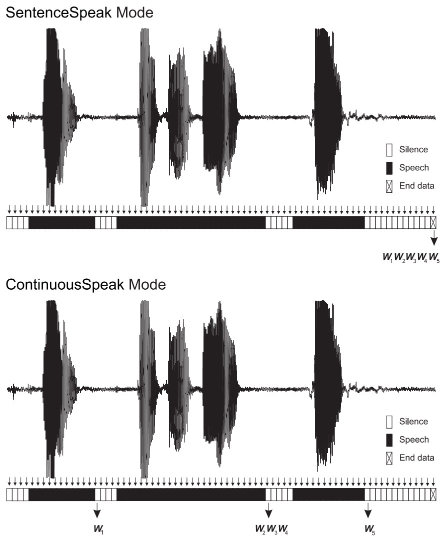

Fig. 3. SentenceSpeak vs. ContinuousSpeak mode

The speech recognition engine can run in one of two different modes - in an ContinuousSpeak mode or in a SentenceSpeak mode. If the engine is running in the SentenceSpeak mode, all tasks are working continuously until a signal of the end of a utterance is generated by the application and sent to the engine. Then the decoder gives a result. If the engine is running in the ContinuousSpeak mode, the decoder gives a result immediately after a silence of predefined length is detected by the silence detector. The both speak modes are illustrated in Figure 3.

3.1 Acoustic Modeling

The basic speech unit of our system is a triphone. A triphone is a phone situated between two specific phones. Each individual triphone is represented by a HMM with a continuous output probability density function assigned to each state. Each density of our model is given by mixture of multivariate Gaussians where each Gaussian has a diagonal covariance matrix. The number of mixtures for each model was obtained experimentally. We use 8 mixtures for each state of each model in our system now.

Because the number of Czech triphones is too large the phonetic decision trees were used to tie states of Czech triphones what gives us more robust estimate of HMM parameters. That is because all the data associated with each individual state can be pooled and after tying several states can share the same output probability distribution. For each state of each phone an individual phonetic tree was constructed and the states were clustered. Initially all states of a specific phone state position were associated with the root node of the tree. Depending on each answer attached to each tree node, the pool of states was split into two parts and this process of splitting continued until the states had dropped down to leaf nodes. All states in the same leaf node were then tied.

3.2 Front-end Task

The digitization of an input analog telephone signal and/or a generation of a synthesized speech signal are provided by a telephone interface board. In the current implementation the DIALOGIC D/21D board is used. Via this board the telephone system is able to operate up to 2 telephone lines. This is reached through the capability of the system to run several speech recognition engines simultaneously.

Since the DIALOGIC D/21D is only half-duplex board (at one moment we can either captur or play the signal) we had to implement a function allowing a fast start of recording after a playing has been performed.

The aim of the front-end processor is to convert a continuous speech signal into a sequence of feature vectors. The parameterization process used in our system is based on the Mel-Frequency Cepstral Coefficients (MFCCs). The pre-emphasised acoustic waveform (by applying the first-order high frequency amplification and the pre-emphasis coefficient of 0.97) is segmented into 25 millisecond frames every 10 ms. Hamming window is applied to each frame and MFCCs are computed to generate final acoustic vectors. We use a filterbank with 26 triangular filters, 13 MFCCs including the 0th coefficient. The 13 delta and 13 delta delta MFCCs are computed and appended to the static MFCCs of each speech frame.

The front-end includes a silence detector technique based on an energy of an acoustic signal. If the feature vector has been labeled as a silence then the labeler task does not compute the likelihood score and the decoder task is waiting for speech data.

3.3 Labeler Task

There are 1822 different tied states each of them represented in the space of dimension 39 (13 MFCCs + 13 delta + 13 delta delta) by the mixture of 8 Gaussian distributions in our telephone dialog system. During the decoding it is necessary to compute for an input vector o in each frame (every 10 ms) a large number of log-likelihood score (LLSs)

$$-\ln pr_j(o) = -\ln\left[\sum_{i=1}^{8} c_{ji} N(o, \mu_{ji}, \Sigma_{ji})\right] , \tag{1}$$

where c_{ji} is the weight of the i-th component in the j-th mixture, $N(., \mu, \Sigma)$ is a multivariate Gaussian distribution with mean vector μ and covariance matrix Σ, and $j = 1, ..., 1822$. To compute all 1822 LLSs entails nearly 1 million operations of multiplication, 0.5 million operations of addition, 1 thousand calculations of the exp function, and 1.5 thousand calculations of the log function.

Let us mention that in the case of simple recognition tasks with a small vocabulary it is necessary to compute in each frame period only several tens of LLSs. This means that the labeler can be run in the "on demand" mode - the decoder asks the labeler for the computation of only several particular LLSs. In the case of more complicated tasks with middle or large vocabulary it is necessary to compute during each frame period several hundreds (maybe all of 1822) LLSs. To implement a real time speech recognizer with a small computation cost, it is important to reduce or simplify the large number of calculations.

We solve this problem by applying a quite new technique, which makes possible to establish relatively exactly (in the original space of dimension 39) the first 50 or 100 best (most probable) LLSs. The proposed technique uses efficiently relevant statistical properties of the Gaussian mixture densities combining them with the so-called "a priory hit" technique and the kNN method. This approach allows more than 90% computation cost reduction without any substantial decrease of the recognition accuracy.

3.4 Decoder Task

The decoder uses a crossword context dependent HMM state network, which is generated by a Net generator. The input of the Net generator is a text grammar format represented by an extended BNF that respects the VoiceXML description. The whole net consists of one or more connected grammars. Grammars can be connected together in run-time. A considerable part of the net is generated before the decoder starts but every part of the net can be generated on demand in run-time. The decoder uses a Viterbi search technique with an efficient beam pruning.

The decoder contains a rejection technique to refuse incoming utterances that do not match any word sequence defined by the grammars. It is required that the speech recognition engine must not select any sentence from the set of sentences defined by the grammar and should inform the dialog manager that no sentence matches the input utterance.

Furthermore the decoder task can run in a key-phrase spotting mode too. The same Viterbi search technique is used. We must only enlarge the HMM decoding net. The recursive mumble models are added at the beginning and at the ending of the net or are put into the net in order to capture and absorb non-key-phrase parts inside the sentence.

4 Results and Conclusion

To evaluate the performance and reliability of our engine the following experiments were provided: The 100 utterances (sentences) were randomly chosen from the telephone speech corpus [3] from an economic area. The vocabulary size of the corpus was 475 words. The first test was performed with a zerogram model. The second one used a bigram language model [2]. The both models were implemented as a static regular grammar. For an evaluation of recognition results we used two standard measures: Correctness (Co) and Accuracy (Ac) defined as

$$Co = (N - D - S)/N \times 100\% = H/N \times 100\%, \quad (2)$$

$$Ac = (N - D - S - I)/N \times 100\% = (H - I)/N \times 100\%, \quad (3)$$

where N is the total number of evens (i.e. words) in the reference transcription, S is the number of substitution errors, D is the number of deletion errors, I is the number of insertion errors and $H = (N - D - S)$. The results of several recognition experiments are summarized below.

Model	Co	Ac	H	N	D	S	I
0-gram	82.13	76.87	1108	1349	79	162	71
2-gram	96.44	94.29	1301	1349	21	27	29

References

1. Műller, L., Psutka J.: The Speaker Independent Recognition Module for the Czech Telephony Dialogue System, ICSPAT99, Orlando, USA, (1999).
2. Jelinek, F.: Statistical Methods for Speech Recognition, MIT Press, Cambridge, (1997).
3. Radová, V., Psutka J., Šmídl L., Vopálka P., Jurčíček F.: "Czech Speech Corpus for Development of Speech Recognition Systems" In Proceedings of the Workshop on Developing language resources for minority languages, Athens 2000.