REJECTION AND KEY-PHRASE SPOTTING TECHNIQUES USING A MUMBLE MODEL IN A CZECH TELEPHONE DIALOG SYSTEM

Luděk Müller, Filip Jurčiček, and Luboš Šmíd
University of West Bohemia in Pilsen, Department of Cybernetics, Univerzitní 22, 306 14 Pilsen, Czech Republic, muller@kky.zcu.cz, smidl5@kky.zcu.cz, filip@kky.zcu.cz

ABSTRACT
This paper describes an implementation of a mumble model in a real–time Czech telephone dialog system, and its contribution to speech recognition. A short overview of the Czech telephone dialog system with a special focus on its speech recognition module is given. Furthermore, the structure of a mumble model is described. A new rejection technique based on a local time comparison between a mumble’s score and a word’s score, evaluated by a Viterbi search is explained. A key–phrase spotting technique using a mumble model is shown. The both techniques have been evaluated with a Czech telephone database. The results show the 25.1% equal error rate of the techniques have been evaluated with a Czech telephone spotting technique using a mumble model is shown. The both evaluated by a Viterbi search is explained. A key–phrase rejection, key–phrase spotting technique using a mumble model is shown. The both techniques have been evaluated with a Czech telephone database. The results show the 25.1% equal error rate of the technique and 15.5% equal error rate of the key–phrase spotting technique.

Keyword: Czech telephone dialog system, mumble model, rejection, key–phrase spotting.

1. INTRODUCTION
Nowadays speech recognition techniques are not still perfect. Thus for almost each practical speech dialog system, it is desirable to have a technique enabling us to detect a recognition error that has occurred, so that subsequent action for the doubtful phrase can be taken. We call this technique a rejection technique, because the subsequent action usually causes the rejection of the uncertain phrase. There are several approaches to determine the probable recognition error.

We used a rejection method based on a mumble model. The main idea resides in a comparison between the word score and the mumble score within the word boundaries during recognition. The confidence measure of the recognized word is defined as the difference between these two scores. At the utterance level the average value of word confidence measures is usually compared to a predefined threshold. Recently, the recursive mumble model method was used and reported [3] for a stack decoder, where the word boundaries are readily available during recognition. To implement the mumble confidence measure in a recognition system equipped with a Viterbi time–synchronous search, it is necessary to keep the word and mumble scores for each word and for each time in which the outgoing transition from the word is chosen by the Viterbi algorithm. It generally escalates requirements on a computer memory and a processor time. In this article a new rejection method for a Viterbi decoder without substantial memory and computational demands is presented. The method is based on a time local distance between the mumble score and the word score. In the next part of the article our implementation of a key–phrase spotting technique using a mumble model and finite state grammar is described. The mumble model is here used to capture and absorb non–key–phrase parts of an acoustic waveform. Moreover, it will be elucidated how the backward loop transition of a mumble model can impact on the key–phrase spotting process and how an improvement can be reached through the redistribution of the mumble model backward loop transition probability to the rest of mumble model transitions.

2. DIALOG SYSTEM OVERVIEW
The Czech telephone dialog system [6] has been designed and developed at the Department of Cybernetics, University of West Bohemia in Pilsen. Even though the main purpose of this system is laboratory and experimental usage, it is fast enough to operate with several telephone lines on a PC in real time.

2.1. The dialog system architecture
The system comprises three main parts – a speech engine, a dialog manager, and a dialog application. The dialogue application is a task–oriented module keeping knowledge on a lexicon and a dialog structure. The dialog manager controls a communication between a user and the system.

The system uses a telephone interface board to capture or generate an analog telephone signal. Figure 1 illustrates the dialog system architecture.

2.2. Speech Recognition Engine
The speech recognition engine is based on a statistical approach. It comprises a front–end, an acoustic model, a language model, and a decoding block, that provides a search for the best word sequence matching the acoustic signal.

Acoustic Modeling: As a basic speech unit of our recognition system a triphone is used. Each triphone is represented by a 3 state left–to–right HMM with a continuous output probability
density function assigned to each state. Each density is expressed as a mixture of multivariate Gaussians with a diagonal covariance matrix. The Czech phonetic decision trees were used to tie states of Czech triphones.

**Front-end:** The speech signal is digitized through a telephone board at 8 kHz sample rate and converted to the mu-law 8-bit resolution format. The parameterization process used in our system is as follows: Firstly the pre-emphasized acoustic waveform is segmented into 25 millisecond frames every 10 ms. A Hamming window is applied to each frame and 13 MFCCs (including the energy coefficient \( C_0 \)) are computed. The RASTA filter is applied. The first-order and second-order derivatives of MFCCs are computed and appended to the static MFCCs of each speech frame.

**Labeler:** The recognition algorithm uses 2510 different tied states, each of them represented by a mixture of 8 Gaussian distributions in the 39-dimensional space. Thus during a decoding it is necessary to compute a large number of log-likelihood scores (LLSs) every 10 ms. In order to perform the recognition in real time the number of calculations is reduced by applying a technique, which seeks to find and precisely determinate only the first 150 most probable LLSs. This technique efficiently uses relevant statistical properties of the Gaussian mixture densities combining them with "a priory hit" technique and the kNN method. This approach allows more than 90% reduction of a computation cost without substantial decrease of a recognition accuracy.

**Decoder:** The decoder uses a crossword context dependent HMM state network generated by a Net generator. The input of the Net generator is a text grammar format represented by an extended BNF with respect of the VoiceXML. The whole net consists of one or more in run-time connected regular grammars. A considerable part of the net is usually generated before the decoder starts but every part of the net can be generated on demand in run-time. The decoder utilizes a Viterbi search with a beam pruning.

### 3. THE MUMBLE MODEL TECHNIQUES

If a recognition error occurs, a method of obtaining a confidence measure of the recognition result is in demand. The confidence measure evaluates the belief that the recognition result is valid. Furthermore, the confidence measure can also be used to decide, if the utterance pronounced by the speaker is an admissible utterance, i.e. if the utterance belongs to a set of utterances defined by the vocabulary and the language model of a given recognition task. In our system, the admissible utterance is an utterance, which respects the finite-state-grammar rules. For example, if a speaker says an out-of-vocabulary word, the speech recognition engine must not select any sentence defined by the grammar, but it should inform the dialog manager that no admissible sentence matches the input. In order to solve this problem, many different techniques have been proposed [1,2,3].

#### 3.1. The Mumble Model Structure

The mumble model is constructed as a set of HMM models connected in a parallel fashion. Each HMM model is 3 state left-to-right and represents one context-independent phone. The structure of the mumble model is depicted in Figure 2. Actually the probabilities of emission of an observation vector in a given state are evaluated as the maximal emission probability of all corresponding states of context-dependent triphones. Thus neither additional HMM models nor additional training is required. The value of the backward loop probability \( BPr \) causes a various length of the phone sequence recognized by the network in Figure 2. While the higher value produces more insertions, the small value induces more deletions in the resulting phone sequence.

![Mumble model](image)

The basic idea of both the rejection and the key-phrase spotting techniques is that the score (i.e. log-likelihood) of the mumble is compared to the score of the network of HMM of triphones. Then, if the mumble score is higher then the score of the network of triphones the rejection is executed, respectively the phrase is not accepted as a key phrase.

Note, that if the backward loop probability \( BPr \) equals to one, then the mumble model score is always same or higher than the score of the network of triphones HMM. It means, that we should penalize the mumble model to avoid the situation, when all utterances would be rejected, respectively no key-phrase would be found.

Usually the penalization is performed just by decreasing the mumble model’s backward loop probability \( BPr \). But as it has been shown, a too small value of the backward loop probability can lead to a deletion errors in a phone sequence produced by the mumble model. It means, that the mumble model does not represent the acoustic signal accurately and the mumble model score can be negatively influenced. Therefore, the mumble model behavior can be improved by redistributing the probability \( BPr \) to the rest of mumble model’s transitions.

#### 3.2. The Rejection Technique

In [3] a rejection technique based on a confidence measure obtained by using a mumble model and a stack decoding technique is reported. Firstly, the confidence measure is evaluated for each recognized word. Then the confidence measure of the utterance is computed as the average value of word confidence scores augmented with the minimum word confidence score.
Our recognition engine uses a Viterbi algorithm and a finite-state-grammar as the language model. To keep the backtracking information a token passing algorithm with a word link record [4] is applied. To avoid a necessity of keeping the mumble and the word scores for each word that was passed by a Viterbi search and for each time in which a token has passed the word, we use a local confidence measure independent of word boundaries.

At each time frame, the log-likelihood score of the mumble model is evaluated as the maximum log-likelihood score from all mumble model states. Similarly, the log-likelihood score of the recognition network is taken as the maximum log-likelihood score from all network model states. Then the difference between these two maximal values is computed and saved into a buffer keeping score differences of the last N time frames. Let us denote that M is constant, (0 < M < N). In each time of recognition the difference between the last buffer element B[t] and the buffer element B[t-M] is evaluated and compared to some predefined threshold. Note that more values of M can be used, each of which compared to another threshold. If the thresholds are exceeded, then the recognition result is rejected.

We used three values $M_1, M_2, M_3$ and tree thresholds $Th, k_1Th, k_2Th$, where $k_1$ and $k_2$ are constants obtained experimentally. If we denote the $B[t]$ as the buffer element of time frame t, then for an utterance of a length of T frames the rejection decision can be expressed as follows:

$$C = 0$$

FOR every $t_1, t_2, t_3$: $M_1 < t_1 \leq T, M_2 < t_2 \leq T, M_3 < t_3 \leq T$ DO:

IF ($B[t_1] - B[t_1-M_1] > Th$) THEN $C = C + 1$

IF ($B[t_2] - B[t_2-M_2] > k_1Th$) THEN $C = C + 1$

IF ($B[t_3] - B[t_3-M_3] > k_2Th$) THEN $C = C + 1$

END FOR

IF $C > CT$ THEN DO REJECTION,

where $CT$ is a predefined constant.

Note that the redistribution of the backward loop probability $BPr$ has no effect on the result of rejection (through the operation of subtraction two log-likelihood scores with the same offset), and it can be omitted.

### 3.3 The Key–Phrase Spotting Technique

The mumble model can also be used to absorb non-key–phrase parts of a spoken utterance during a key–phrase spotting. The recognition network contains a set of key–phrases in a parallel connection and a mumble model that is also connected to the key–phrase network in parallel. An example of a key–phrase recognition network with a mumble model is depicted in Figure 3.

During the recognition the Viterbi search finds out the best path through the recognition network. If some non–key–phrase part of an utterance has appeared then the mumble model will have better score than any key–phrase model, and the mumble word will be assigned to the non–key–phrase part of the utterance. In this way the mumble model catches the non key–phrase parts. Finally, the mumble words are omitted from the resulting recognized word sequence and only the key–phrases are in the output.

![Figure 3: Key–phrase recognition network](image)

The redistribution of the backward loop probability $BPr$ has effect on the result because no subtraction between log-likelihood scores of the mumble model is calculated.

### 4. EXPERIMENTAL RESULTS

The proposed methods have been tested using two Czech telephone databases [5]. A finite-state-grammar was used as the language model in all experiments.

#### 4.1 Experiment 1 and 2: Rejection

We use a telephone yellow–page database. The speech corpus comprises 357 speakers and 357 utterances (each utterance was spoken by a different speaker). Using the database we performed two experiments.

![Figure 4: Rejection result (716 sentences)](image)
corpus, and false acceptance and false rejection error rates were obtained for different values of the rejection thresholds. The results are shown in Figure 4. The word error rate is 5.4%. The intersection point between the false acceptance curve and the false rejection curve denotes the ERR (equal error rate).

In the second experiment the vocabulary contains 2719 words and the grammar accepts 2864 different two-word sentences. As well the test with a grammar which does not accept any utterance from the test speech corpus was provided. The results are shown in Figure 5. The word error rate is 12.4%. This rather high value is caused by a relatively small width of beam pruning during recognition.

![Figure 5: Rejection result (2864 sentences)](image)

The ERR is 19.4% for the experiment 1, and 25.1% for the experiment 2. The ERR increases with the number of sentences accepted by the recognizer and with the vocabulary size.

4.2 Experiment 3: Key–Phrase Spotting

The key–phrase spotting method has been tested using the Czech telephone database from economic area. 50 words were chosen as 50 key–phrases. 97 utterances from different speakers were chosen as a test speech corpus.

![Figure 6: No redistribution of BPr](image)

The average utterance length is 14 words and all utterances contain 450 different words. Results for the backward loop probability which is not redistributed are shown in Figure 6 (ERR = 16.7). Results for the redistributed backward loop probability are in Figure 7 (ERR = 15.5). The results show that the mumble model with the redistributed backward loop probability gives a better result of rejection.

![Figure 7: Redistributed BPr](image)

5. CONCLUSIONS

A rejection method based on a mumble model and a Viterbi search of the key–phrase spotting technique using a mumble model was performed on a telephone Czech database. The results show its 25.1% equal error rate. As well a test of the key–phrase spotting technique using a mumble model was performed on a telephone Czech database and 16.7% equal error rate was achieved. Moreover a refinement of mumble model parameters reduces this error rate to 15.5%. These results indicate that a usage of mumble model can improve the dialog system behavior.

6. ACKNOWLEDGMENTS

This work was supported by the Grant Agency of Czech Republic no. 10296/K087 and the Ministry of Education of Czech Republic, project no. VS97159.

7. REFERENCES


