Detection of Recognition Errors Based on Classifiers Trained on Artificially Created Data

Tomáš Bartoš and Luděk Müller

Department of Cybernetics
University of West Bohemia, Pilsen, Czech Republic
tbartos@kky.zcu.cz, muller@kky.zcu.cz

Abstract
This paper wishes to contribute to the solution of the problem occurring when an automatic speech recognition system does not recognize an input utterance correctly. The solution is usually based on a utilization of a confidence measure (CM) which is assigned to each recognized word and which informs a user or a higher level module on the belief that the recognized word has been really said. The task becomes more difficult if the vocabulary contains acoustically similar words which differ for example only in one phoneme. To cope with this problem, we introduced a new confidence measure based on the basic elementary unit for which the presented CM is investigated: a phone.

The first part of the article shortly describes the used speech recognition system and the previously used confidence measures. The main part of this article deals with description of creation of the new CM based on utilization of artificially created training data and also with description of the used classification features and the classifiers based on this CM.

A rejection technique based on the described CM was evaluated on the Czech yellow-pages database. Experimental results show that the proposed rejection technique achieves approximately 5% equal error rate (ERR) for phone rejection and about 6-16% EER for word rejection.

1. Introduction
Although a significant progress in speech recognition area has been done in last years the automatic speech recognition (ASR) systems are not still perfect. If the recognition block of an ASR system makes an error, the meaning of a whole sentence could be changed.

This problem mainly occurs in recognition of database entries, e.g. person names recognition in an automatic switchboard operator system. Some of the names are phonetically too close to each other, which makes the task of correct recognition complicated. The names can differ in only one phoneme, e.g. in Czech “Vrána” and “Brána” and the acoustic difference between them is consequently too small to precisely distinguish such words with current ASR systems, which results in recognition errors of such words.

Another problem is presence of out-of-vocabulary (OOV) word, which can be acoustically very close to a word from the system vocabulary or it can be a fragment of some vocabulary word. This makes the recognition task even more difficult. Therefore, for almost each practical speech recognition system, an efficient recognition error detection technique is in great demand. This technique is usually called a rejection technique because the words determined by this technique as low confident are subsequently rejected as incorrectly recognized.

For the sake to be able to distinguish between acoustically close words, the CM should be defined on the phone level. For that purpose the proposed CM utilizes a new classification feature set which is more adequate to the local acoustic dissimilarity of words. The CM is trained on results of ASR, i.e. on the recognizer output for correctly and wrongly recognized words. The most important issue is that the CM is trained by supervised learning provided with precise and detailed supervised information on which part of the recognizer output represents the wrongly or correctly recognized phone. To obtain such training data, we forced the recognizer to recognize out-of-vocabulary words and in this way we make it produce recognition errors. We call this training data artificial because the recognizer errors were made in an artificial manner. The CM for a word is calculated from CMs of the phones which the word consist of.

2. ASR system
All recognition experiments reported in Section 4 were performed with the recognition engine of the telephone dialog system [3] built at the Department of Cybernetics, University of West Bohemia. The recognition engine is a speaker independent CDHMMs based module. It incorporates front-end, acoustic model, language model and decoding block.

The front-end module provides acoustic signal preprocessing as follows: An analogue telephone signal is digitized at 8 kHz sample rate and converted to the mu-law 8-bit resolution format. The pre-emphasized signal is segmented into 25 millisecond frames and every 10 ms a feature vector consisting of 7 PLP cepstral static, 7 delta, and 7 accelerations coefficients is computed. The acoustic model is triphone based represented by three-states HMM with 8 Gaussian component mixture assigned to each state. The decoder uses a crossword context dependent HMM recognition network, which is generated by a net generator. The decoder utilizes the Viterbi search and beam pruning. In order to determine the recognition confidence the system is equipped with so-called Mumble Model (MM), fully described in [1], [2]. MM is constructed as a set of CDHMMs that are connected in a parallel fashion. Each CDHMM is 3 states left-to-right CDHMM and represents one context-independent phone.

The MM network (MN) and the HMMs recognition network (RN) are separated but work in parallel, and each of them has its own output; the output of each network is a matrix incorporating a cumulative output score of all HMM networks states in all decoding time frames. The matrix column indexes represent the HMM-states and matrix row indexes represent the time frames. The network score in time t is a minus log-likelihood of such surviving network state which likelihood is maximal in the time t. These scores are obtained through Viterbi search with beam pruning. Both
outputs are time synchronized; the output of one network in time $t$ corresponds to the output of the other network in the same time $t$. The entire recognizer output contains the MN matrix, the RN matrix, and the matrix containing for each time $t$ and each HMM state $s$ a non-cumulative score (i.e. minus logarithm of value of p.d.f. assigned to $s$ for the feature vector observed in time $t$) of each HMM state of each phone for each time frame, referred as the labeller ($L$) matrix.

3. Construction of confidence measure

This part briefly describes the rejection techniques we previously used, a new approach with a new confidence measure, training data creation, the types of used features, and finally the used classifiers.

3.1. Previous rejection techniques

In our previous work the rejection algorithm utilizes only scores of the RN and the MN. The algorithm worked as follows. In every time step the difference between the MN score and the RN score was computed. This difference was stored in a buffer $B$ containing the last $N$ computed score differences. We assumed that $M$ is such a constant that $0 < M < N$. In every time step $t > M$ the difference between the last buffer element $B[i]$ and the element $B[t-M]$ was computed and compared to a heuristically adjusted threshold. Note, that one can use more than one threshold and more time intervals $M$ (a use of 3 different $M$ and 3 thresholds is described in [1]). If the computed difference is higher than the heuristic threshold more than $n$-times, where $n$ is also a heuristic constant, then the recognized word is rejected.

Another approach led to the algorithm described in [2], where we used several so-called differential curves (each curve corresponds to $B[i] - B[t-M]$ for a some fixed value $M$) and their peaks were used as features for classification. The Karhunen-Loève Expansion (KLE) was employed to reduce the feature dimensionality.

The equal error rate (EER) of the method described in [1] was about 20%, and about 13% in case of the method reported in [2].

3.2. Improved approach

The origin of the new CM is motivated by the need to have some technique which can detect the recognition errors already on the phone-level. This is necessary in applications as person names recognition, where the names differ for each person. From that reason the features from our previous technique are not applicable on the phone level because they are computed from large time intervals (up to 400 ms), which usually contain more than one phone. The decision if the recognized phone was recognized correctly or not can be understood in terms of hypothesis testing. This approach manages two probabilistic models of generation of a feature vector

\[ p(\text{phone feature vector} | A_L) \text{ and } p(\text{phone feature vector} | A_R) \]

of generation of a feature vector by the acceptation and rejection model, respectively. The CM of the phone then can be expressed as the likelihood ratio:

\[ CM_{\text{phone}} = \frac{p(\text{phone feature vector} | A_L) \cdot P(A_L)}{p(\text{phone feature vector} | A_R) \cdot P(A_R)}, \quad (1) \]

where $P(A_L)$ and $P(A_R)$ are a priori probabilities of the acceptation and rejection model, respectively, and $c$ is a constant whose value determines the threshold of rejection. The models of $p(\text{phone feature vector} | A_L)$ and $p(\text{phone feature vector} | A_R)$ can be built from training data. Because the ratio of a priori probabilities $P(A_L) / P(A_R)$ is also a constant, its value multiplied by $c$ can be treated as a value which sets the rejection threshold, and therefore neither $P(A_L)$ nor $P(A_R)$ have to be known explicitly.

The acceptance and rejection models have to be trained from recognizer output data (i.e. MM, RN, and L matrices) for correctly and wrongly recognized phones, respectively.

The training data for the acceptation model are taken from such parts of the matrices which contain data corresponding to correctly recognized phones. Such places can be easily found by using only recognizer outputs for correctly recognized words (we made the assumption that all phones of correctly recognized words had to be also recognized correctly). To obtain the training data from the rejection model we have to find the RN matrix elements that correspond to wrongly recognized phones. The detection of such elements makes the training data acquisition task much more complicated. The training data used in our previous rejection technique did not contain the information on where (i.e. in which HMM states) recognition errors exactly occurred. They contained only information if the recognized word was recognized correctly or not. From that reason our previously used features were computed across the whole word and therefore they are not suitable for the phone-level rejection case. To acquire the training data for the rejection model, we used a new approach. It is based on artificially created training data by using ASR vocabularies specially designed for this task. We collected a set of recordings where each recording contained one word. For each recording we constructed a vocabulary which incorporated phonetic transcription of the recording and phonetic transcription of all words created from the recording by substituting, deleting or inserting one and only one of recording’s phone. Consequently, for each recording we obtained a vocabulary with hundreds words derived from the original recording. Afterwards, for each recording and each of its derived vocabulary word we let recognize the recording with our ASR as this derived word. Because this word differs in just one phone from the original recording, the recognition output has to contain an error. We supposed that the error has to be in a place of the phone mismatch. In such way the decoder is forced to make recognition errors at the demanded places. After that, for each recording and its derived word we obtained an output of the recognizer. Since the outputs of the recognizer are matrices containing scores of all phone-states in all time frames, it is not hard to find boundaries between individual phones by Viterbi search. The right and the left boundary of the misrecognized phone determine the part of the recognizer output from which the features of the rejection model are taken as its training data. The training data for the acceptation model are obtained in similar way from matrix places corresponding to phones of correctly recognized words.

3.3. Features

For the description of acceptation and rejection model we used 2 sets of features listed below (these 2 sets together represent the feature vector). The first feature set (features...
1...8) is based on acoustic scores produced by the RN and MN. The acoustic score alone is a good indicator of correctness of recognition - the higher recognition error the higher the decoding score - but as a standalone feature it has not sufficient performance. The second feature set (features 9...14) is based mainly on using additional information from the MN, e.g. we determine several best phones and their scores and transcriptions for each given time frame.

For the feature set description we use the following notation: 
- **RNS** - RN score; 
- **MMS** - MN score; 
- **LS** - labeller score, the value of L matrix for a given state and a given time frame; 
- **t** - the beginning of the analyzed phone; 
- **t2** - the end of the analyzed phone.

1. **AGNetDiff** – The difference between RNS and MMS scores in a given time \( t_2 \) minus the difference between RNS a MMS in a given time \( t_1 \), divided by the difference between times \( t_2 \) and \( t_1 \). This feature also represents a differential curve used in [2].

\[
AGNetDiff = \frac{(RNS(t_2) - MMS(t_2)) - (RNS(t_1) - MMS(t_1))}{t_2 - t_1} \tag{2}
\]

2. **MinRNSDiff** – The minimal increase of RNS during one time frame in a time interval \( <t_1, t_2> \).

\[
MinRNSDiff = \min_{t_1 < t < t_2} (RNS(t + 1) - RNS(t)) \tag{3}
\]

3. **MAXRNSDiff** – The maximal increase of RNS during one time frame in a time interval \( <t_1, t_2> \).

\[
MAXRNSDiff = \max_{t_1 < t < t_2} (RNS(t + 1) - RNS(t)) \tag{4}
\]

4. **RNSDiff** – The difference between RNS at the end and at the start of a time interval \( <t_1, t_2> \) for a given phone.

5. **ARNSDiff** – The value of RNSDiff divided by the length of the time interval \( <t_1, t_2> \).

6. **MinLS** – The value of the minimal labeller score for a time interval \( <t_1, t_2> \) and a given phone.

7. **MaxLS** – The value of the maximal labeller score for a time interval \( <t_1, t_2> \) and a given phone.

8. **AvgLS** – The sum of labeller scores for a given phone and a given time interval \( <t_1, t_2> \), divided by the length of the time interval \( <t_1, t_2> \).

9. **IPRNSLS1** – The number of matches between the recognized phone and the first best phone determined by the labeller, divided by the length of the time interval \( <t_1, t_2> \).

10. **IPRNSLS3** – The number of matches between the recognized phone and one of the three first best phones determined by the labeller, divided by the length of the time interval \( <t_1, t_2> \).

11. **IPRNSMMS1, IPRNSMMS3** – similar as feature 9, MN is used instead of the labeller.

12. **IPRNSRN1, IPRNSRN3** – similar as feature 9, RN is used instead of the labeller.

### 3.4. Classifiers

To test the CM introduced above we used two classifiers: the Bayes classifier used already in our previous work [1], [2], and an artificial neural network (ANN) as an alternative classifier.

### 3.5. Bayes classifier (BC)

We expressed the \( Ac \) and \( Ar \) models as mixtures of 3 Gaussians. To decorrelate and reduce the feature vector dimensionality we employed the Principal Component Analysis (PCA) known also as the Karhunen-Loeve Expansion (KLE). Particularly, the method of Kittler-Young [4] was used. All model parameters were estimated through the EM algorithm.

### 3.6. Artificial neural network

In addition, we also tested the CM with the ANN based classifier. We realized that for classification into two classes it is sufficient to construct a feed-forward backpropagation neural network.

The ANN used in our experiments contained: 14 input neurons with hyperbolic tangent sigmoid transfer function, 25 neurons in hidden layer also with hyperbolic tangent sigmoid function, and 1 output neuron with a linear transfer function.

### 3.7. Word-level CM

The CM described so far is intended as a phone-level CM. For the word level, the CM can be derived from CMs of all word phones as follows:

\[
CM_{\text{WORD}} = \frac{1}{N} \sum_{n=1}^{N} CM_{\text{Phone}}(n). \tag{5}
\]

where \( CM_{\text{Phone}}(n) \) is CM of \( n \)-th phone, \( N \) is a number of word phones. If the recognized word has \( CM_{\text{WORD}} \) lower than some predefined threshold, it is rejected, otherwise accepted.

### 4. Experimental results

The described rejection technique has been evaluated with a telephone yellow-page database. The speech corpus comprised 2 sets of data. The set A contained 70 utterances from 70 speakers, in total 140 words. Each utterance was one Czech person name (first name and surname) and was spoken by different people. The set B was used exclusively for word rejection testing. It comprised additional 100 word recordings (60 different words from the economic area) from other 30 speakers.

For each word from the set A the artificial vocabulary was constructed and the word was recognized with this vocabulary by the decoder (see Section 3.2). Through this procedure the training and test data (both containing the correct and the incorrect recognized phones) were obtained. One part of this data (90%) was used for acceptance/rejection models estimation and the second part (10%) was used for phone rejection technique evaluation (test data set TDI).
4.1. Test data sets

- TD1 – test data of the set A (10%) for the phone rejection technique; each test data entry contains a feature vector of one phone; phone rejection test.
- TD2 – test data of the set A used for recognition with artificial vocabularies (i.e. words which differ from the recognized word in only one phone); the recognition output was classified by the described rejection technique; word recognition test.
- TD3 – test data from the corpus B used for recognition with ASR vocabulary containing 60 words from corpus B (the vocabulary words differs in an arbitrary number of phones); word rejection test.

4.2. Evaluation

To evaluate the performance of the proposed confidence measure the value of the confidence measure of each phone/word was compared with a predefined threshold and subsequently the phone/word was either accepted or rejected. The threshold can be tuned to control the trade-off between false acceptance and false rejection.

By denoting the number of false rejected phone/word as $FR$, the number of correctly accepted phone/word as $FA$, the number of correctly rejected phone/word as $TR$, and the number of correctly accepted phone/word as $TA$, we can define the false rejection error $fr$ and the false acceptance error $fa$ as:

$$fr = \frac{FR}{(FR + TA)} \times 100\% \quad (6)$$

$$fa = \frac{FA}{(FA + TR)} \times 100\% \quad (7)$$

The results are given in Figure 1 and in Table 1. Figure 1 shows the DET (Detection Error Trade-off) curves for the three test data sets.

5. Conclusion

This paper introduced our recently developed technique for detection of recognition errors. The main advantage of this technique over the previous one is the usage of artificial training data for estimation of acceptation/rejection model parameters. This leads to a more robust technique for detection of recognition errors. The presented technique achieved 5% EER in incorrectly recognized phone rejection task, and 6-16% EER in incorrectly recognized word rejection task. The 16% of EER corresponds to the worst situation where the recognized words differ in only one phone from each other. The 6% of ERR can be achieved in applications that use a common vocabulary.

The both classifiers (BC and ANN) yield almost identical results, but the BC classifier seems to be more stable on a wider range of CM threshold settings and has smaller memory requirements, however the ANN is faster.

In the future more accurate results can be expected by using a phone confusion matrix, which contains the information about “interchange ability” of phones. It could be beneficial to consider it for training, because then only the data where the phone recognition errors are well apparent could be taken for rejection model parameters estimation. In the rejection technique just presented in this paper an error can occurred for example when the phone “a” is confused with a very similar phone “á” (i.e. long variant of “a”). In such cases the recognition network output seems to be errorless. Therefore, if these data are used for training the rejection model, then this model can be insufficiently discriminative from the acceptation one.

Further improvements can also be done by using a more sophisticated mumble model, e.g. triphone and/or n-gram based model.

6. Acknowledgment

This work was supported by Grant Agency of the Academy of Sciences of the Czech Republic, p. no. 1ET101470416.

7. References