

# Utterance Verification based on the Likelihood Distance to Alternative Paths

Gies Bouwman and Lou Boves

Department of Language and Speech, P.O. Box 9103,  
6500 HD Nijmegen, The Netherlands  
{G.Bouwman, L.Boves}@let.kun.nl  
<http://lands.let.kun.nl>

**Abstract.** Utterance verification tries to reject incorrectly recognised utterances. For this purpose the probability of an error is often estimated by single confidence measure (CM). However, errors can have several different origins, and we argue that notion must be reflected in the design of the utterance verifier. In order to detect both in-vocabulary substitutions and out-of-vocabulary word errors, we compute CMs based on the log-likelihood distance to (1) the second best recognition result and to (2) the most likely free phone string.

Experiments in which different CMs were combined in different ways in the recognition of Dutch city names show that Confidence Error Rates [8] are reduced by 10% by combining the CMs using a classification and regression tree instead of a linear combination with a decision threshold.

## 1 Introduction

The need for confidence measures in ASR no longer requires an explicit motivation. At the moment, we are exploring their use within the scope of the SMADA project [1], that investigates the practical implications of developing a system for nationwide directory assistance (DA). One of the subtasks is to recognise Dutch city names. Due to the high perplexity of the task, we face error rates exceeding 10%. In order to enable a user-friendly dialogue in an automatic DA application, it is mandatory to automatically reject the least reliable recognition results. Another goal for which we need to identify utterances that were most probably misrecognised is automatic update of acoustic and language models based on speech that is recorded during the actual operation of the service. Several studies (among others [2]) have analysed the origin of speech recognition errors. It is necessary to distinguish at least two different types of errors:

- the input speech is (partly) not modelled, for instance because it contains OOV words or word sequences that were not foreseen in the grammar. As long as special measures, such as a garbage model, are not taken, "in domain" interpretation of those utterances is doomed to lead to one or more insertion errors.

- the input speech is modelled, but an alternative (incorrect) hypothesis happens to obtain a higher likelihood score. In this case a substitution error occurs.

Of course, deletion errors can also occur. However, for the task under analysis, deletion errors are less important.

Because the causes of recognition errors can be many and varied, it would be surprising if a single indicator of the reliability of an output is sufficient to flag (virtually) all errors. Therefore, some investigators have attempted to combine multiple confidence measures [2], [3], [4], [5]. Such combinations can take many different forms. In this paper we compare the power of a linear combination and a CART-like procedure.

In Section 2 we propose two measures to detect the two kinds of errors. We also describe our general system architecture and the material used to train and test the classifiers. In Section 3 we define the atomic confidence measures. We also outline two methods to combine these cues. Section 4 presents the results of our experiments, followed by a discussion of the results. Finally, Section 6 describes the main conclusions and perspectives for future work.

## 2 Method

### 2.1 Path distance

The idea of our approach is that we compute two likelihood ratios. The first concerns the likelihoods of the best candidate and the runner-up in the N-best list.

$$\frac{P(X|W_1)}{P(X|W_2)} \quad (1)$$

If the quotient is much greater than 1, a substitution error with  $W_2$  is unlikely, and the classification of  $x$  as  $W_1$  is probably correct. Thus, the likelihood ratio of the two top hypotheses can serve as a measure to detect substitution errors. However, incorrect classification may also be due to OOV speech. In that case the likelihood ratio between the first and second best hypotheses is not appropriate as confidence measure. For this reason, we also compute formula 2

$$\frac{P(X|W_1)}{P(X|W_{FPR})} \quad (2)$$

where  $W_{FPR}$  means the optimal phone string for the input speech, obtained with free phone recognition (FPR). Its likelihood score can serve as a normalisation coefficient for the likelihood of  $W_1$ . So when the word likelihood  $P(X | W_1)$  is higher than any other word, the free phoneme models may yield a much higher likelihood: relative to FPR,  $P(X | W_1)$  is small. In that case it is likely that an OOV word has been spoken.

In the following subsections we will elaborate on our general system architecture, our training and test material and some other elements of our design.

## 2.2 Architecture of the utterance verifier

We implemented our utterance verification method as a two-pass procedure. Figure 1 shows that in parallel to the recogniser generating an N-best city name

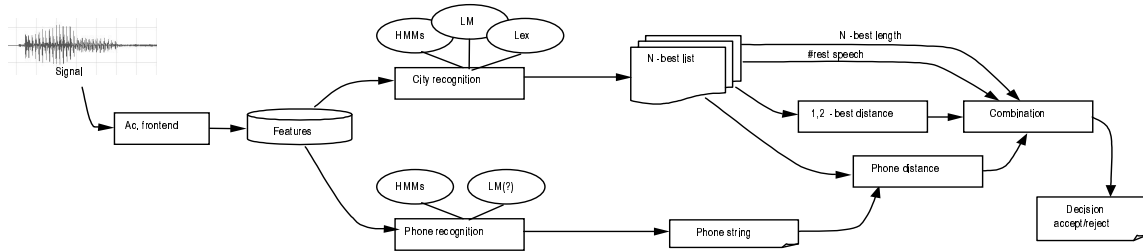


Fig. 1. System architecture

list, the speech input is also decoded in terms of the optimal phone path. In the following steps, we compute four cues to decide whether the best solution is to be rejected or accepted:

1. the length of the original N-best list ( $> 0, \leq N$ );
2. the number of frames assigned to other speech models than those of the recognised city name, like the garbage model or alternative city names;
3. the distance between the paths of the first and second best city name hypotheses;
4. the distance between the path of the best hypothesis and the path of phoneme recognition.

In the final step, we use either a linear combination (LC) or a classification and regression tree (CART) [6] to come to a decision. The linear combination combines the four measures into a one-dimensional confidence measure that allows one to reject (or accept) utterances based on some threshold value. The CART approach keeps the four dimensions separate; a rejection decision of a test case depends on the leaf node where that case is classified. We will return to this issue in Section 3.

## 2.3 Material and models

Our material consists of the city name utterances of the DDAC2000 corpus [7]. Callers were prompted to say for which city they wanted a directory listing. The recordings were divided in a train, development and test set of 25k, 11k and 11.5k utterances.

For the signal representation and ASR details we refer to [5]. The main characteristics of our ASR are that it is an HMM based system, with 2,369 city names in the lexicon and a bigram language model. The test set perplexity was 204.5.

## 2.4 Content words

In our recognition experiments, the city names (or the expression ‘I don’t know’) are the content words. These words convey the relevant information for the dialogue manager in this part of the interaction. Therefore, we evaluate recognition and verification only at that (semantic) level.

An implication of using N-grams (with discounting and backing-off strategies) is that recognition hypotheses may contain zero or multiple content words. The latter occur especially when lengthy OOV utterances are produced. In these cases, the first city name determines the value of the whole utterance and is passed on to the verification component; all other city names in the output are ignored.

## 2.5 Evaluation

The performance of our utterance verifier will be optimised and evaluated using the Confidence Error Rate (CER) [8]. This is the total number of false accepts (#FA) and false rejects (#FR) divided by the total number of all cases: correct (#COR) and incorrect (#INC). In order to find the corresponding points in the ROC curves, we also compute false accept rate ( $FAR = \#FA/\#INC$ ) and false reject rate ( $FRR = \#FR/\#COR$ ).

## 3 Implementation and experiments

This section first describes the way the distance measures are computed, and the experiments to compare the different measures. Next, the implementation of the CART procedure is explained, and the way in which these implementations are compared.

### 3.1 Path distance

First, we assume that the phone alignment of both paths is known. In other words, for each feature vector we have the information about which HMM unit was aligned against it. Doing this by forced segmentation, we are also able to obtain the corresponding acoustic log-likelihood scores. The absolute difference between two scores for the same time frame is a log-likelihood ratio (LLR) score at frame level, as displayed by formula 3.

$$LLR(x_t|S_t^b, S_t^a) = LL(x_t|S_t^b) - LL(x_t|S_t^a) \quad (3)$$

where  $S_t^b$  and  $S_t^a$  are the states of the best and alternative hypotheses aligned against  $x_t$ , the feature vector at time  $t$ .  $LL(x|S)$  is the log likelihood score of vector  $x$  as computed with  $S$ 's pdf.

Next we combine the frame scores into a single score per content word. Formula 4

shows how we first take the average absolute LLR on phone level and next on word level.

$$\frac{1}{|W|} \sum_{\psi \in W} \left[ \frac{1}{(\psi_e - \psi_s)} \sum_{t=\psi_s}^{\psi_e} \text{abs} ( \text{LLR}(x_t | S_t^b, S_t^a) ) \right] \quad (4)$$

where  $W$  is the content word we compute confidence for.  $|W|$  denotes the number of phones in  $W$ . The index  $\psi \in W$  runs over all phones of  $W$  with  $\psi_s$  and  $\psi_e$  being their respective start and end times.

In our experiments we used formula 4 to compute the distance between the recognised content word, i.e. the top candidate of the N-best list, and the runner up, if available. We refer to this distance as  $D_{rup}$ . At the same time, we compute the distance between the best content word and the optimal phone string resulting from free phone recognition, which we call  $D_{fpr}$  from now on.

### 3.2 The role of Language Models

One of the assumptions we made is that the distance between first best and best FPR path says something about the credibility of the acoustic score. At this point the question arises whether to use a phone language model (LM) in the FPR. Using an LM can help to minimise phone error rate, but one may wonder if this is a goal to aim for. By ignoring prior knowledge we will truly maximise acoustic likelihood. For  $D_{rup}$  however, the language model scores are important and should probably not be ignored. After all, if two candidates have equal acoustic scores, but one is enforced by the language model, the a-posteriori probability of an error will be minimised by selecting the hypothesis with the best score including the LM. Summarising, in the distance ratio  $D_{fpr}$  the acoustic score ought to be sufficient, but in  $D_{rup}$  we should use both.

Experiment 1 In experiment 1 we investigate the role of LMs in the computation of the likelihood scores. There are four combinations ( $D_{fpr}$  with and without LM) x ( $D_{rup}$  with and without LM) and for each combination we computed the distance scores on a development and a test set. The distance between two paths is computed according to combination formula 4. The development set was used to train the coefficients of a linear combination function using LDA (for each of the four systems separately). With these functions we combined the distances of the test set and thus obtained a single confidence score for every utterance.

### 3.3 LC versus CART

Although LCs have the advantage of yielding a one-dimensional score, a linear separation may not be optimal in the face of multiple and unrelated causes of recognition errors. CART procedures have proven to be a very powerful alternative for LC in many speech recognition tasks. Therefore, we compare CART and LC for their power in distinguishing between correct and erroneous hypotheses at the output of our ASR system.

When training a CART, it is necessary to define an optimisation criterion for splitting a data set. Correctly recognised city names can thus be optimally separated from the incorrect ones. In our situation we shift a threshold over each of the four numerical confidence cues to classify the ASR output. The best threshold value according to either formula 5 or 6 is stored as the binary separator.

$$\hat{T} = \operatorname{argmax}_T \left[ 1 - FAR(T) - FRR(T)^{1/\psi} \right] \quad (5)$$

$$\hat{T} = \operatorname{argmax}_T \left[ 1 - FRR(T) - FAR(T)^{1/\psi} \right] \quad (6)$$

In words, these formulas express that the optimal threshold is at a value where false accept rate is optimised while false reject rate is close to its minimum or vice versa. The strictness of ‘close’ is controlled with exponent  $\psi$ , with typical values around 2.0.

Experiment 2 We estimate the parameters of the tree on the same development set as used in Experiment 1. The evaluation is of course on the test set. Since the CART approach yields only one optimal separation scheme, there is no straightforward way to generate an ROC curve. Therefore, we compare the performance of CART and LDA in terms of Confidence Error Rate.

## 4 Results

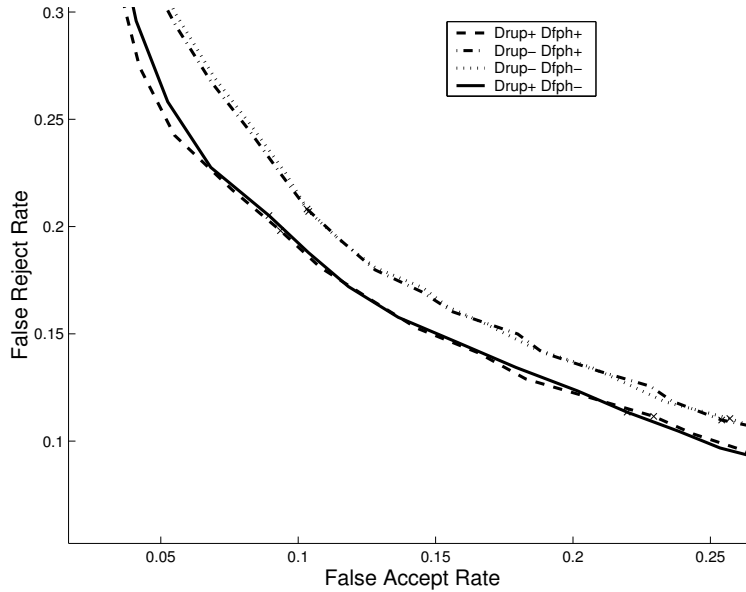
All verification results are based on a single run of our recogniser on the test set. Since previous reports, like [5], we have improved performance and currently 14.4% of the utterances are incorrectly recognised. This percentage can be split up in different types of errors. 8.4% of all recognition results contain a substitution error, while 4.2% can be ascribed to the presence of OOV words (insertion errors). The remaining 1.8% are deletion errors, that we consider as relatively harmless.

Figure 2 shows the ability to separate correct utterances from incorrect ones in terms of % False Accept and % False Reject. The +/- signs in the legend indicate whether LM scores are taken into account for the respective path distance. Table 1 shows the optimal CER values that correspond with these curves.

The second experiment comprised building a CART. Its character cannot be

**Table 1.** CER, FAR and FRR when using the LM score (+lm) or disregarding it (-lm) in  $D_{rup}$  and  $D_{fpr}$ .

$D_{rup}$	$D_{fpr}$	%CER	%FAR	%FRR
-lm	+lm	11.2	51.3	4.4
-lm	-lm	11.4	53.0	4.4
+lm	+lm	10.5	54.2	3.1
+lm	-lm	10.5	54.1	3.2



**Fig. 2.** ROC curves with and without language model contribution in the path distances

illustrated by an ROC curve, because there is no threshold involved. After optimising the decision tree on our development set and classifying the test set, we measured the CER being 9.4%. The corresponding FAR and FRR amounted to 31.3% and 5.6% respectively.

## 5 Discussion

When comparing the curves in Figure 2, we see that the two systems that take the language score distance into account for  $D_{rup}$ , have an ROC that is closer to the origin. This means that their ability to separate correct utterances from incorrect utterances is better. The CER values in rows 3 and 4 of Table 1 are significantly better than those in the first two rows (95% confidence). At the same time we see that the use of a language model to optimise the free phone recognition makes no significant difference. Here we see a confirmation of our idea that minimising phone error rate is not a goal to aim for. However, this is no evidence that omitting the LM is beneficial.

When comparing the two confidence measure combination methods, it appears immediately that the classification tree method is better than the linear combinations. The relative improvement of CART when compared with the best result obtained with LC is about 10%, which is significant. Although we did not perform a case-by-case analysis, this gives reason to believe that the problem of utterance verification is not optimally served by a one-dimensional confidence

measure. Errors have diverse causes that can well be reflected in a multidimensional vector. The CART verifier has access to the vector components until the final decision moment.

We also examined the split parameters in our D-tree. It appeared that  $D_{rup}$  was in fact the most informative variable to split the population at the root node. The next split, down either of the branches, was based on  $D_{fpr}$ . The total number of nodes in the tree amounted to 8 and all four cues were used.

## 6 Conclusions

In this paper we proposed confidence scores on the basis of distances between the best and second best recognition hypotheses and between the best word and phone recognition results. In this way, we aimed to detect substitution and OOV errors. Reasoning that these two source processes are unrelated, we tested the hypothesis that a CART is a better combiner than a Linear Combination; the relative confidence error rate improvement of 10% confirmed this assumption.

In future work we would like to include syllable-based measures that take lexical stress into account, like in the linear combinations we tested in [5]. Encouraged by the results of the present study, we believe that a CART may help to disclose more valuable information in specific parts of the complex space.

## References

1. L. Boves, D. Jouviet, J. Siemel, R. de Mori, F. Béchet, L. Fissore, P. Laface ASR for Automatic Directory Assistance: the SMADA Project Proc. of ASR2000 Paris (2000), pp unavailable
2. D. Charlet, G. Mercier, G. Jouviet: On Combining Confidence Measures for Improved Rejection of Incorrect Data. Proc. of Eurospeech '01. Aalborg (2001), pp. 2113–2116
3. S. Kamppari, T. Hazen: Word and Phone Level Acoustic Confidence Scoring. Proc. of ICASSP '00, vol III. Istanbul (2000), pp. 1799–1802
4. T. Hazen, I. Bazzi: A Comparison and Combination of Methods for OOV Detection and Word Confidence Scoring. Proc. of ICASSP '01, vol I. Salt Lake City (2001), pp. 397–400
5. G. Bouwman, L. Boves: Using Information on Lexical Stress for Utterance Verification. Proc. of ITRW on Prosody in ASRU. Red Bank (2001), pp. 29–34
6. L. Breiman(ed) et al.: Classification and Regression Trees. Chapman & Hall. 1998
7. J. Sturm, H. Kamperman, L. Boves, E. den Os: Impact of Speaking Style and Speaking Task on Acoustic Models Proc. of ICSLP '00, vol I. Beijing (2000), pp. 361–364
8. F. Wessel, K. Macherey, R. Schlüter: Using Word Probabilities as Confidence Measures. Proc. of ICASSP '98, vol I. Seattle (1998), pp. 225–228