Comparison and combination of confidence measures

Georg Stemmer, Stefan Steidl, Elmar Nöth, Heinrich Niemann, Anton Batliner

Angaben zur Veröffentlichung / Publication details:

Stemmer, Georg, Stefan Steidl, Elmar Nöth, Heinrich Niemann, and Anton Batliner. 2002. "Comparison and combination of confidence measures." In *Text, speech and dialogue: 5th International Conference, TSD 2002 Brno, Czech Republic, September 9–12, 2002*, edited by Petr Sojka, Ivan Kopeček, and Karel Pala, 181–88. Berlin: Springer. https://doi.org/10.1007/3-540-46154-X_25.

Comparison and Combination of Confidence Measures

Georg Stemmer, Stefan Steidl, Elmar Nöth, Heinrich Niemann, and Anton Batliner

Universität Erlangen-Nürnberg, Lehrstuhl für Mustererkennung, Martensstrasse 3, D-91058 Erlangen, Germany
stemmer@informatik.uni-erlangen.de
http://www5.informatik.uni-erlangen.de

Abstract. A set of features for word-level confidence estimation is developed. The features should be easy to implement and should require no additional knowledge beyond the information which is available from the speech recognizer and the training data. We compare a number of features based on a common scoring method, the normalized cross entropy. We also study different ways to combine the features. An artificial neural network leads to the best performance, and a recognition rate of 76 % is achieved. The approach is extended not only to detect recognition errors but also to distinguish between insertion and substitution errors.

1 Introduction

Current speech recognizers are often extended by an additional module which computes a confidence measure for each recognized word. A confidence measure is an estimator of the correctness of the hypothesized word. If the confidence measure is accurate, it can be applied to various different tasks: For instance, a spoken dialogue system may ask the user for an additional confirmation if the confidence of a relevant word is very low. Other possible applications include unsupervised speaker adaption, where words with low confidence may be discarded for adaption, and the detection of out-of-vocabulary words. Confidence measures may also be used to repair speech recognition errors by an additional module. For certain applications, it may be important to extend the two class problem *correct* vs. *wrong* and to distinguish three different classes: *correct* (*cor*) *substitution* (*sub*) and *insertion* (*ins*). A spoken dialogue system may simply ignore all inserted words, but ask back if there has been a substitution.

A great amount of different confidence measures can be found in the literature [1]. In the following, we will concentrate on confidence measures providing a word-level annotation, which seems to be most useful for a majority of applications. In order to compute confidence measures for a speech recognizer, a feature vector has to be calculated for each word hypothesis. From the features, the confidence of the word can be estimated by using the score of a suitable classifier. Decision trees and artificial neural networks (ANN) are often applied for this purpose.

The decision for a certain feature set for the confidence measure is often guided by two requirements: firstly, the features should be simple to implement and fast to compute, and secondly, the set should provide as much information about the confidence of a word as possible. In this paper we compare a number of different features with respect to a common quality measure. We also evaluate the improvements which can be achieved by taking a combination of the features.

2 Quality of a Confidence Measure

For a fair comparison of different features we need to score the quality of the corresponding confidence measure. Several different methods have been described in the literature for this purpose. We decided to use two scoring methods. The first one is the Normalized Cross Entropy (NCE), which has been introduced by NIST. It is defined as the relative decrease in uncertainty brought by the confidence measure about the correctness of a word w

$$\mathrm{NCE} = \frac{H(X) + \frac{1}{N}\Big(\sum_{w\in\mathcal{C}_H} \log_2 P(c \mid w) + \sum_{w\in\mathcal{F}_H} \log_2(1 - P(c \mid w))\Big)}{H(X)} \quad (1)$$

where

$$H(X) = -\Big(p_c \log_2 p_c + (1 - p_c)\log_2(1 - p_c)\Big).$$

N is the number of words w which are taken into consideration, $P(c \mid w)$ is the confidence measure which estimates the probability that the word w is correct. p_c stands for the a-priori probability of the correctness of any word w. $\mathcal{F}_H$ is the set of words which are wrong and $\mathcal{C}_H$ contains the correct words. The NCE is always a value between zero and one. Only if the confidence measure performs worse than the a-priori classifier the result is negative. Please note that p_c in the normalization term is correlated to the performance of the recognizer. Several authors (e.g. [2]) mention that, despite of the normalization, the NCE depends on the error rate and therefore does not allow a fair comparison of confidence annotation across recognition systems. In the following, we will compute the NCE for confidence measures which estimate the probability of a word for being correct, being a substitution and being an insertion.

The second quality measure we used is simply the recognition rate of the confidence measure for the two class problem *correct* vs. *wrong* and also for the three class problem *cor*, *sub* or *ins*. We also compute the class-wise average of the recognition rates. Of course, the recognition rate is highly dependent on the error rate of the speech recognizer. Nevertheless, the recognition rate can give a good idea of the performance of a confidence measure in a real application. Closely related to the recognition rate are the values of precision and recall. If $\mathcal{C}_H$ contains the correct words, and $\mathcal{M}_c$ contains the words, which have been marked as *correct* by the confidence measure, then the precision of the class *correct* is defined as

$$prc_c = \frac{|\mathcal{C}_H \cap \mathcal{M}_c|}{|\mathcal{M}_c|} \quad (2)$$

where $|\cdot|$ counts the number of elements in a set. Recall of the class *correct* is equivalent to the recognition rate:

$$rec_c = \frac{|\mathcal{C}_H \cap \mathcal{M}_c|}{|\mathcal{C}_H|} \tag{3}$$

3 Features for Confidence Measure Computation

3.1 Word-Based Features

Category of the word. Category-based confidence features are given by $P(c|cat(w))$, which is the probability that a word w which belongs to a category $cat(w)$ is correct. The probability is estimated from the training sample. We evaluate two different category systems, HANDCAT and POS, the corresponding features are WCHAND and WCPOS. HANDCAT contains about 160 categories, which have been derived manually. POS assigns one of 15 part-of-speech labels to each word of the vocabulary. The feature WCPOS can be extended by the part-of-speech labels of the left and right neighbors of w (feature WCPOS ± 1 and WCPOS ± 2 for two respective four neighbors). We also try to use simply the name of the part-of-speech label $cat(w)$ for confidence computation (feature WCPOSNAME).

Language model. In [3] it has been noted that a word w is more likely to be correct, if the score $P(w|\boldsymbol{v})$ of the language model for the word and its context $\boldsymbol{v}$ is high. In the following, $-\log P(w|\boldsymbol{v})$ is used as feature LSCORE for confidence estimation.

Word length. As long words are usually recognized better than short words [4], the length of a word may be a useful confidence feature. The length of a word hypothesis may be computed from the number of phones in the word (feature LPHONE) or its duration in frames (feature LFRAME).

Word frequency. If a word appears less frequently in the acoustic training data, the corresponding HMM may be trained worse. Therefore, we evaluate the feature WFREQ, which is the logarithm of the absolute frequency of the word in the data.

3.2 Features Based on the Acoustic Score

Similar to the feature LSCORE, the acoustic score $-\log P(O_{t_s}, .., O_{t_e}|w)$ for a word w and an observation sequence $O_{t_s}, .., O_{t_e}$ can be used as the feature ASCORE; t_s and t_e denote the start and end time of the word. A low acoustic score indicates that the word is misrecognized. We try to improve the results by applying some normalization. The feature MASCORE is the acoustic score divided by the duration of the word hypothesis:

$$\text{MAScore}(w) = \frac{\text{AScore}(w)}{t_e - t_s + 1} \tag{4}$$

Under adverse acoustic conditions, the overall score of all output densities $\mathcal{N}(O_t|\mu_k\Sigma_k)$ of the recognizer may be low. As a consequence an additional normalization factor PSCORE is introduced:

$$\text{PScore}(t_s,..,t_e) = \sum_{t=t_s}^{t_e} -\log \sum_k p_k \; \mathcal{N}(O_t|\mu_k\Sigma_k) \tag{5}$$

p_k stands for the a-priori probability of output density k and has to be estimated from the training data. The feature NASCORE takes PSCORE into account:

$$\text{NAScore}(w) = \text{AScore}(w) - \text{PScore}(t_s,..,t_e) \tag{6}$$

It is also possible to use PSCORE as a feature or to combine the two normalization methods (feature MNASCORE). The sum of the features PSCORE, ASCORE, MASCORE and NASCORE gives the new feature GSCORE.

3.3 Word Graph-Based Features

Beam width. The beam search algorithm increases the number of active states if the best path through the search space has a poor score. As the beam width bw_t depends on the current time frame t, we have to combine the values of bw_t for $t_s \leq t \leq t_e$ in order to derive a confidence feature for the word w. The feature MBEAM is the mean of bw_t in the interval $t_s \leq t \leq t_e$, SBEAM its standard deviation. MINBEAM is the minimum value of bw_t, while MAXBEAM corresponds to its maximum. The positions of the minimum and maximum, measured in percent of the interval length $t_e - t_s + 1$, give two additional features POSMINBEAM and POSMAXBEAM.

A-posteriori probability. The confidence in a word w, which covers the frames $t_s,..,t_e$ of an utterance $\boldsymbol{O}$ can be associated directly with its a-posteriori probability $P(w,t_s,t_e|\boldsymbol{O})$. The a-posteriori probability can be estimated from a word graph. As described in [5], all preceeding and succeeding contexts $\boldsymbol{w}_p, \boldsymbol{w}_s$ of the word w which can be found in the word graph have to be taken into consideration:

$$\begin{aligned} P(w,t_s,t_e \mid \boldsymbol{O}) &= \sum_{\boldsymbol{w}_p} \sum_{\boldsymbol{w}_s} P(\boldsymbol{w}_p, w, \boldsymbol{w}_s \mid \boldsymbol{O}) \\ &= \frac{\sum_{\boldsymbol{w}_p} \sum_{\boldsymbol{w}_s} P(\boldsymbol{O} \mid \boldsymbol{w}_p, w, \boldsymbol{w}_s) \cdot P(\boldsymbol{w}_p, w, \boldsymbol{w}_s)}{P(\boldsymbol{O})} \end{aligned} \tag{7}$$

where

$$P(\boldsymbol{O}) = \sum_w \sum_{\boldsymbol{w}_p} \sum_{\boldsymbol{w}_s} P(\boldsymbol{O} \mid \boldsymbol{w}_p, w, \boldsymbol{w}_s) \cdot P(\boldsymbol{w}_p, w, \boldsymbol{w}_s) \tag{8}$$

As a word graph usually contains several instances w_i of the word w which differ in $t_s(i)$ and $t_e(i)$ the confidence measure can be improved by summing up $P(w_i,t_s(i),t_e(i)|\boldsymbol{O})$ of all word hypotheses w_i which overlap in the time domain. The resulting feature will be denoted as APOSTERIORI.

4 Data

For our experiments we use a set of spontaneous dialogues between humans, which have been collected in the VERBMOBIL project [6]. In Tab. 1 the three subsets which are used to train the speech recognizer and to evaluate the confidence measures are shown.

Table 1. Subsets of the VERBMOBIL data which are used to evaluate the confidence measures.

subset		utterances	words
training of the speech recognizer		15647	358505
confidence measure evaluation		4938	103855
	training of the classifier	3704	78125
	test of the classifier	1234	25730

5 Short Description of the Speech Recognizer

The speech recognizer used for the experiments is a speaker independent continuous speech recognizer. The recognition process is done in two steps. First, a beam search is applied, which generates a word graph. The beam search uses a bigram language model. In the second phase, the best matching word chain is determined from the word graph by an $A^\star$-search, which rescores the graph with a 4-gram language model. Please refer to [7] for a more detailed description. The a-priori probabilities of the classes *cor*, *sub* and *ins* in the recognition result of the speech recognizer on the VERBMOBIL data are 0.646, 0.269 and 0.085. A confidence measure which would simply label every word as *correct* would have a recognition rate for the class *correct* of 64.6 % and a NCE of zero. The class-wise average of the recognition rates for *correct* and *wrong* would be 50 %.

6 Experimental Results

6.1 Comparison of the Features

For each of the individual features we train a decision tree classifier which has to assign the label *correct* or *wrong* to each word which has been hypothesized by the speech recognizer from the test data. We also measure the performance for the three classes *cor*, *sub*, *ins*. In Tab. 2 the NCE for all the features introduced in the previous section can be found. Please note, that the NCE in Eq. 1 is defined only for a two class problem. For the column *NCE(all)* the definition is extended for three class labels. In order to compute the NCE for only one of the classes *cor*, *sub* or *ins*, the other two classes are merged. The NCE of *cor*, *sub* or *ins*

Table 2. Normalized Cross Entropy (NCE) for the individual features. Results are given for the two and the three class problem.

feature	*correct* vs. *wrong*	*cor* vs. *sub* vs. *ins*			
	NCE	NCE (all)	NCE (*cor*)	NCE (*sub*)	NCE (*ins*)
WCHand	0.018	0.081	0.018	0.047	0.171
WCPosName	0.019	0.082	0.018	0.038	0.181
WCPos	0.022	0.084	0.021	0.040	0.182
WCPos ±1	0.057	0.121	0.054	0.061	0.246
WCPos ±2	0.064	0.123	0.059	0.064	0.243
WFreq	0.039	0.094	0.036	0.060	0.173
LScore	0.042	0.086	0.039	0.068	0.133
LPhone	0.012	0.079	0.012	0.037	0.182
LFrame	0.030	0.058	0.029	0.006	0.161
AScore	0.022	0.046	0.021	0.009	0.121
MAScore	0.052	0.070	0.056	0.029	0.139
NAScore	0.030	0.067	0.029	0.005	0.193
MNAScore	0.003	0.039	0.005	0.016	0.097
PScore	0.029	0.063	0.028	0.004	0.183
GScore	0.089	0.123	0.095	0.073	0.211
MBeam	0.024	0.039	0.025	0.040	0.043
SBeam	0.012	0.043	0.013	0.007	0.123
MinBeam	0.046	0.046	0.046	0.534	0.023
PosMinBeam	0.020	0.034	0.019	0.005	0.085
MaxBeam	0.017	0.049	0.016	0.023	0.112
PosMaxBeam	0.008	0.007	0.007	0.002	0.012
all Beam features	0.069	0.093	0.072	0.066	0.145
APosteriori	0.126	0.100	0.113	0.135	0.001

cannot be compared with each other, because the three classes differ w.r.t. the a-priori probability.

The results in Tab. 2 give a rank ordering of the confidence features. Despite some exceptions, like MNAScore, SBeam or PosMaxBeam nearly every feature seems to contain useful information about the confidence of a word. The a-posteriori probability performs better than all other single features. The results indicate that feature combinations, like in GScore, improve results significantly. For WCPos we were able to get a better confidence annotation by incorporating the part-of-speech labels of the neighboring words.

6.2 Feature Combination

We combine all features into one feature vector for each word and classify it with a decision tree or an ANN. WCPosName is not included because we did not want to code the name with several binary features. The ANN is a multi-layer perceptron with two hidden layers and is trained with backpropagation. Each feature gets one input node, the number of output nodes corresponds to

the number of classes (two or three). In Tab. 3, the NCE for all features is computed. In Tab. 4 the corresponding recognition rates are given. The ANN shows

Table 3. Normalized Cross Entropy (NCE) for all confidence features.

classifier	*correct* vs. *wrong*	*cor* vs. *sub* vs. *ins*			
	NCE	NCE (all)	NCE (*cor*)	NCE (*sub*)	NCE (*ins*)
decision tree	0.236	0.248	0.237	0.211	0.275
neural network	0.241	0.253	0.239	0.217	0.293

Table 4. Recognition rates (RR) for the class *correct* and class-wise averaged recognition rates (avg) for two and three classes with all confidence features.

classifier		*correct* vs. *wrong*	*cor* vs. *sub* vs. *ins*
decision tree	RR	74.6 %	72.1 %
	avg.	71.4 %	59.9 %
neural network	RR	75.4 %	72.5 %
	avg.	72.9 %	62.1 %

a slightly better recognition performance than the decision tree. We could further improve the results by taking the context of the current word into account: First, the confidence values are computed for the neighboring words. Next, these two numbers are used as two additional features for the confidence computation of the current word. The results can be found in Tab. 5. In Fig. 1 the relation

Table 5. Recognition rates (RR) for the class *correct* and class-wise averaged recognition rates (avg) when the confidence of left and right neighbor is used.

	correct vs. *wrong*	*cor* vs. *sub* vs. *ins*
RR	76.1 %	72.8 %
avg.	73.0 %	63.7 %

between precision and recall of the class *correct* is shown. Fig. 1 also shows how precision and recall depend on the value of a threshold Θ, when all words w with $P(c|w) > \Theta$ are assigned to the class c.

7 Conclusion and Outlook

We compared a large number of features for confidence scoring. From all features the a-posteriori probability of a word achieves by far the best performance. However, the a-posteriori probability alone leads to a NCE of 0.126, which is

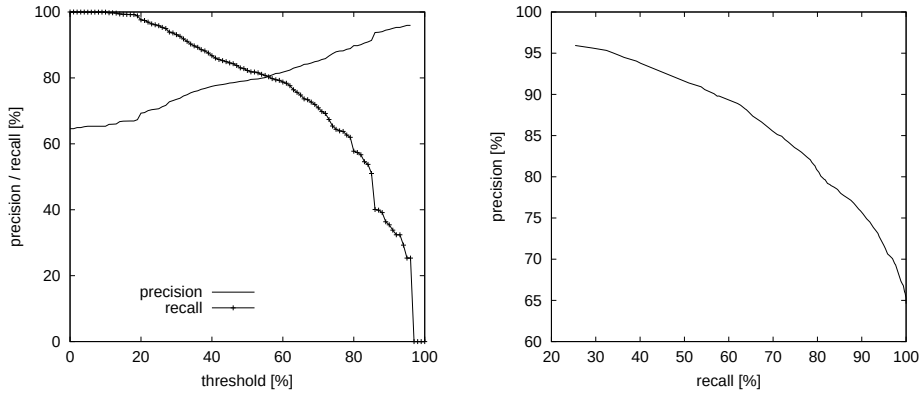

Fig. 1. Precision and recall for the class *correct* depending on a threshold Θ (left) and relation between precision and recall (right).

only one half of the NCE that can be reached by the full set of features. Please note, that in order to determine the a-posteriori probability the complete word graph must be processed in forward and backward direction, which makes the feature computationally more time consuming than all other features. We tend to the conclusion that the decision for a certain classifier should not be overrated: The difference in recognition rate between the neural network and the decision tree could become even smaller if we put more effort in the initialization and optimization of the tree. We have shown that even for a low speech recognition accuracy, confidence measures can distinguish correct from wrong words with a reasonable performance. Even the three class problem seems to be solvable. In the future we want to investigate into the integration of the confidence scores into a spoken dialogue system and plan to evaluate whether the reliability of the confidence measure may be higher for semantically important words.

References

1. L. Chase: Error-Responsive Feedback Mechanisms for Speech Recognition. Ph.D. Thesis, Carnegie Mellon University (1997)
2. B. Mison and R. Gopinath: Robust Confidence Annotation and Rejection for Continuous Speech Recognition Proc. IEEE ICASSP (2001) vol. 1
3. E. Eide and H. Gish and P. Jeanrenaud and A. Mielke: Understanding and Improving Speech Recognition Performance through the Use of Diagnostic Tools. Proc. IEEE ICASSP (1995) vol. 1, 221-224
4. S. Cox and R. C. Rose: Confidence Measures for the SWITCHBOARD Database. Proc. IEEE ICASSP (1996) vol. 1, 511–514
5. F. Wessel and K. Macherey and R. Schlüter: Using Word Probabilities as Confidence Measures. Proc. IEEE ICASSP (1998) vol. 1, 225–228
6. W. Wahlster: Verbmobil: Foundations of Speech-to-Speech Translation. Springer (2000)
7. S. Steidl: Konfidenzbewertung von Worthypothesen. Student Thesis (in German), Chair for Pattern Recognition, University of Erlangen-Nuremberg (2001)