CONFIDENCE MEASURES FOR SPONTANEOUS SPEECH RECOGNITION

Thomas Schaaf *Thomas Kemp*
Interactive Systems Laboratories, ILKD
University of Karlsruhe
76128 Karlsruhe, Germany

ABSTRACT

For many practical applications of speech recognition systems, it is desirable to have an estimate of confidence for each hypothesized word, i.e. to have an estimate of which words of the output of the speech recognizer are likely to be correct and which are not reliable. We describe the development of the measure of confidence tagger JANKA, which is able to provide confidence information for the words in the output of the speech recognizer JANUS-3-SR. On a spontaneous german human-to-human database, JANKA achieves a tagging accuracy of 90% at a baseline word accuracy of 82%.

1. INTRODUCTION

Current speech recognition systems are far from perfect. Unfortunately, number and location of the errors in their output is usually unknown. This information, however, could be used in a number of applications. Examples for such applications are word selection for unsupervised adaptation schemes like MLLR [1], automatic weighting of additional, non-speech knowledge sources like lip-reading, or aiding a NLP system towards generating repair dialogs in case a semantically important word has a low confidence.

In this work, we introduce the measure-of-confidence (MOC) tagger JANKA, which is aimed at providing a word level estimate of confidence to the recognition result of our speech-to-speech translation system JANUS-3. Different knowledge sources are evaluated in terms of their ability to predict whether a particular decoded word is correct or wrong. The knowledge sources are combined into feature vectors and classified by a vector classifier into tags for each word of the hypothesis.

Consider the sentence "Mary loves her little child" and the corresponding speech recognizer output "Eight Mary loves her brittle child". Then, the desired output of a MOC tagger would be "0.0, 1.0, 1.0, 1.0, 0.0, 1.0" where "0.0" stands for a recognition error and "1.0" for a correctly recognized word.

The development of JANKA consisted of two parts: the selection of appropriate knowledge sources and their extraction into numerical feature vectors, and the classification of these feature vectors into the classes 'recognition error' and 'correct'.
The selection of useful features is described in section 3, the classifier design is outlined in section 5, and experimental results on spontaneous speech data are given in section 6.

2. DATABASE

For all described experiments we used the GSST database, which has been collected simultaneously at four different sites under the VERBMOBIL[2] project. It consists of human-to-human spontaneous German dialogs in the appointment scheduling domain, i.e. two persons try to schedule a meeting within the next month. The data is sampled with 16 kHz at a resolution of 16 bit in a quiet office environment using a close-speaking microphone.
The database contains about 33 hours of speech and has a bigram test set perplexity of around 54.

For the experiments described in this paper, 1251 additional utterances were collected at all four sites. None of the speakers of this additional data was included in the main database. Only the main database was used for the training of the acoustic models and the language models of the recognizer. The additional data was divided into a training set and a test set. Table 1 shows the composition of the additional database used for training and evaluation of the measure of confidence classificator.

set	speakers	utterances	words	duration (min)
Training	46	785	14906	101
Crossvalid.	6	134	3063	22
Test	20	332	5940	39
Total	72	1251	23909	162

Table 1. Database composition

3. FEATURE SELECTION FOR MEASURE OF CONFIDENCE

The selection of features for MOC tagging can be divided into two steps: 1) the search for a set of knowledge sources, which should be as large as possible, and 2) the selection of the relevant features out of this set.

3.1. Defining a set of candidate features

We have been investigating the following set of nineteen candidate features.

A-stabil, as proposed by Finke and Zeppenfeld [2] [3]. For this feature, a number (typically 100) of alternative hypotheses with different weighting between acoustic scores

[2]The VERBMOBIL project aims at the development of a large speech-to-speech translation system and is funded by the german ministry for science and technology (BMBF)

0-8186-7919-0/97 $10.00 © 1997 IEEE

and language model scores is computed. Each of these hypotheses is aligned against the reference output of the recognizer, where the reference output is defined as the output with the (assumedly) best weighting between acoustics and language model. For each word of the reference output, the number of times the same word occurs in the set of alternative hypotheses, normalized by the number of alternative hypotheses, is taken as feature value. **A-stabil-before** is the same feature, computed on the hypothesis before vocal tract normalization.
LM-NGRAM (similar to [5][4]): the number of times a backoff in the language model occurs.
LogAWE-end is the logarithm of the number of active final word states in the search, averaged over a three-frame window around the last frame of the hypothesized word.
NScoreQ: the log-score of the word divided by the log *a-priori* probability of the time segment T_W.
N-active-leaf [4]: the average number of active final word states in the search during the time segment T_W, into which the word was aligned by the search.
NScore: a normalized score similar to [4]: the log-score of the word minus the log *a-priori* probability of the time segment T_W.
PronVar: 1 if the word is a pronounciation variant of the main dictionary entry, otherwise 0.
Score-per-frame: the non-normalized acoustic score per frame as given by the Viterbi decoder.
Duration: reciprocal of **SpkRate**
LogNPhones [5][4][3]: the log of the number of phones of the word.
LogAWE-beg is the logarithm of the number of active final word states in the search, averaged over a three-frame window around the first frame of the hypothesized word.
SNR: maximum SNR value within T_W.
NFrames [5][3]: the length of T_W in 10-millisecond-frames.
NDisfluent: the number of surrounding non-word entities (like breathing noise, coughing etc) to the left and to the right of word W.
A-Entropy: the acoustic frame-wise entropy $H = \sum_{y \in phoneSet} p_y log(p_y)$ of the acoustic models, averaged over the time segment T_W.
SpkRate: speaking rate computed by the quotient of the length of T_W and the expected word length. The expected word length is computed on the acoustic training set.
SNR-MinMax: the difference of the minimum and maximum signal-to-noise ratio (SNR) per frame within the interval T_W.
Log-train [5]: the log of the number of times the word was observed in the training material.

3.2. Feature Selection

For each of the 14906 words of the training set, a 20-dimensional feature vector is computed. The first 19 vector components are described in section 3. The 20th vector component is the correct / false tag as computed by aligning the hypothesis against the reference. It must be emphasized, that the term 'training set' is used with respect to the MOC tagger and that neither the training nor the crossvalidation data was included in the material used for the acoustic or language model training of the actual recognizer. The correlation matrix of this 20-dimensional feature space was computed. The last row of the correlation matrix gives the correlation of all features with the correct/false tag. If the absolute value of this correlation coefficient is high, the corresponding feature can be regarded as 'good'. Table 2 shows the correlation coefficients with the c/f tag for each of the 19 features under investigation.

Feature	correlation
A-stabil	0.481
A-stabil-before	0.431
LM-NGRAM	0.278
LogAWE-end	-0.213
NScoreQ	-0.173
N-active-leafs	-0.170
NScore	-0.161
PronVar	-0.113
Score-per-frame	-0.106
Duration	-0.102
LogNPhones	0.092
LogAWE-beg	-0.068
SNR	0.065
NFrames	0.047
NDisfluent	-0.043
A-Entropy	-0.029
SpkRate	-0.014
SNR-MinMax	0.006
Log-train	0.005

Table 2. Correlation coefficients to c/f tag

The correlation coefficient is only meaningful in the case of linear dependency between feature value and probability of error. Therefore, before discarding the features with a correlation coefficient below a given threshold, we checked all these features to test the assumption of linear dependency. The **SpkRate** and **A-Entropy** features exhibited a significant, but non-linear dependency and were therefore included into the final feature set, whereas all other features with a correlation coefficient lower than 0.05 were discarded. Figure 1 shows the error rate over **SpkRate** and over **T-active-Leaf**.

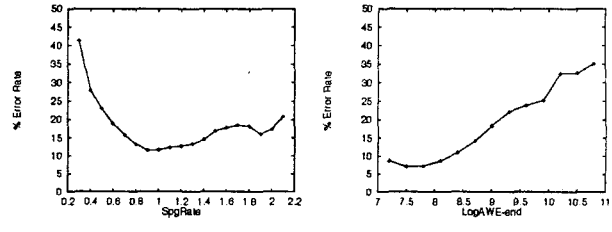

Figure 1. Error rate over feature value for feature **T-active-Leaf** (right) and feature **SpkRate** (left)

When normalizing the acoustic scores with the alignment of a phone recognizer (features **NScore** and **NScoreQ**, the correlation coefficient decreased by 10% relative, as compared to normalizing with the a-priori frame probability. This is in agreement with [4], who found that normalizing with a-priori probabilities outperformed the normalization with phone recognizer scores.

4. EVALUATING CONFIDENCE TAGGER

Different methods for the evaluation of confidence measuring systems have been proposed [4] [3] [9]. However, the best method for scoring depends on the application for the confidence tags. In this work, *confidence accuracy* CA, defined as

$$CA = \frac{\text{Number of correctly assigned tags}}{\text{total number of tags}} \quad (1)$$

is used.

Another measure, which can only be used for continuously valued confidence tags, is the plot of precision (PRC) and recall (RCL) over decision threshold. PRC and RCL are defined as

$$PRC_X = \frac{\text{Number of correctly assigned tags for class X}}{\text{Number of total tags for class X}} \quad (2)$$

$$RCL_X = \frac{\text{Number of correctly assigned tags for class X}}{\text{total number of elements in class X}} \quad (3)$$

where $X \in \{correct, false\}$.

A single metric for confidence scores, which can be viewed as normalized cross entropy, has been proposed by NIST as

$$S = \frac{H(C) + p \sum_{correct} \log(P_c) + (1-p) \sum_{incorrect} \log(1-P_c)}{H(C)} \quad (4)$$

where P_c is the output of the MOC tagger for the a-posteriori probability that word c has been correctly recognized. $H(C)$ is the base entropy $H(C) = -(p \log p + (1-p) \log(1-p))$ and p the a-priori probability that a hypothesis word is correct.

5. COMPUTATION OF MEASURE OF CONFIDENCE

The analysis of the properties of the input features showed for most of the features a good linear correlation to the probability of error. Therefore, we designed two classifiers: a linear classifier based on a covariance matrix optimality criterion, and a multilayer perceptron classifier.

5.1. Linear classifier approach

A one-dimensional linear classifier was built in the following way. First, all training patterns were divided into two sets C and F, the first one containing all correctly recognized words and the second one all incorrectly hypothesized words. The total scatter matrix $\mathbf{S}_T$ and the average within-class scatter matrix $\mathbf{S}_W$ were computed with

$$\mathbf{S}_{W,c} = \frac{1}{N} \sum_{i \in c} (\vec{x}_i - \vec{\mu}_c)(\vec{x}_i - \vec{\mu}_c)^T \quad (5)$$

$$\mathbf{S}_{W,f} = \frac{1}{N} \sum_{i \in f} (\vec{x}_i - \vec{\mu}_f)(\vec{x}_i - \vec{\mu}_f)^T \quad (6)$$

$$\mathbf{S}_W = p(c)\mathbf{S}_{W,c} + p(f)\mathbf{S}_{W,f} \quad (7)$$

$$\mathbf{S}_T = \sum_i (\vec{x}_i - \vec{\mu}_{all})(\vec{x}_i - \vec{\mu}_{all})^T \quad (8)$$

With a linear transformation $\vec{y} = \mathbf{A}\vec{x}$ the input feature space X is transformed into a target feature space Y. This transformation is chosen such that $tr(\mathbf{S}_T^{-1}\mathbf{S}_W)$ in Y-space is minimized, e.g. that while keeping the total scatter unchanged, the within class scatter is minimized and hence the class separability is increased. This technique is well known as linear discriminant analysis [6].

For the two-class problem, the rank of the resulting transformation matrix $\mathbf{A}$ is one, and the feature space Y is one-dimensional. Therefore, the classification problem in the target feature space can be easily solved by choosing a threshold T and deciding "Error" for $Y < T$ and "Correct" otherwise.

5.2. Neural net classifier

The transformation based approach described in the previous section works well for linearly separable classes. However, on many data sets it does not yield satisfying results. Therefore, a 3-layer neural network classifier with sigmoidal activation function units was trained, using a mean square error function and standard backpropagation. Experiments on the held-out data showed rapid convergence after about 100 iterations when using an update step after each training sample. Several different topologies and layer sizes have been evaluated. However, an extremely simple classifier using shortcut connections and one single unit in the hidden layer could not be significantly outperformed by more complex topologies. Therefore, we used this simple classifier in all our experiments.

6. EXPERIMENTAL

6.1. The JANUS-3 system

The speech-to-speech translation system JANUS-3 [7] is a joint effort of the Interactive Systems Labs at Carnegie Mellon University, Pittsburgh, and at the University of Karlsruhe, Germany.

The baseline speech recognition component of JANUS-3 uses mixture-gaussian densities with a scalable amount of parameter tying. For the experiments described, we used 10000 decision-tree clustered context-dependent subquinphones which shared 2500 codebooks. In the preprocessing stage 13 mel-scale cepstral coefficients were computed with a frame rate of 10 ms. The cepstral coefficients along with their first and second order derivatives were merged into a 39-dimensional input feature vector. This 39-dimensional input vector was reduced by linear discriminant analysis (LDA [6]) to the final 32-dimensional input stream. Training was done with Viterbi alignment. To capture some of the effects of spontaneous speech, specialized noise nodels were included [10]. The decoder computes word lattices with a multi-pass strategy. After the first recognition pass, vocal tract normalization parameters [11] are computed. The second recognition pass is then performed with the vocal tract normalization estimated on the first pass.

The JANUS-3 decoder achieved a word error rate of 13.2% in the 1996 VERBMOBIL evaluation. This was the lowest error rate of the five participating institutions.

In the experiments described, the system that was used for the required test of the 1996 VERBMOBIL evaluation was evaluated.

The baseline confidence accuracy on the MOC test set, when tagging all words with 'correct', was 85.3%.

6.2. Results

The most useful features, judging by the correlation to the error rate, appear to be **A-stabil** and **A-stabil-before**. To exploit the performance of this two features and to compare them against the other features, we built three different linear classifiers. The results are summarized in table 3.

We compared the linear classifier and the neural net classifier using shortcut connections and one single hidden unit. To exploit the usefulness of contextual information for the detection of errors, we added the feature vectors of the neighbouring words in an additional experiment to the input of the neural net, thereby increasing the dimensionality

Features	CA_BIN	error reduction
baseline	85.3%	-
AStabil+AStabil-before	88.3%	20.4%
all others	87.3%	13.6%
combined	89.3%	27.2%

Table 3. Performance of different feature sets

of the feature space to 39. The results are summarized in table 4.

Classifier	CA_BIN	error reduction	S
baseline	85.3%	-	0
linear	89.3%	27.2%	-
neural net (NN)	89.7%	29.9%	0.377
NN with context	90.0%	32.0%	0.381

Table 4. Result of different classifiers

The result in terms of PRC and RCL are shown in figure 2. For a recall rate of 84%, i.e. 84% of the correctly recognized words are spotted as such, a remarkable precision of more than 95% can be achieved. With such a high precision, unsupervised adaptation can be used in a very efficient way.

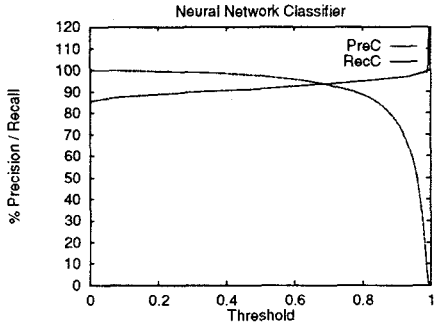

Figure 2. 'Correct' precision and recall over threshold

7. CONCLUSIONS

We have introduced the confidence measure tagger JANKA, which is based on a vector classifier approach. With a set of thirteen input features, use of contextual information and a neural net classificator, JANKA achieves a tagging accuracy of 90% on a difficult human-to-human spontaneous database.

8. ACKNOWLEDGEMENTS

This research was partly funded by grant 413-4001-01IV101S3 from the German Ministry of Science and Technology (BMBF) as a part of the VERBMOBIL project. The JANUS project was supported in part by the Advanced Research Project Agency and the US Department of Defense. The views and conclusions contained in this document are those of the authors.
The authors wish to thank all members of the Interactive Systems Labs, especially Michael Finke, for useful discussions and active support.

REFERENCES

[1] C.J. Legetter, P.C. Woodland: *Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models*, Computer Speech and Language **9** (1995), 171-185

[2] M. Finke, T. Zeppenfeld, M. Maier, L. Mayfield, K. Ries, P. Zhan, J. Lafferty, A. Waibel: *Switchboard April 1996 Evaluation Report*, DARPA, April 1996

[3] Haitao Qiu: *Confidence Measure for Speech Recognition Systems*, Masters Thesis, Carnegie Mellon University Computational Linguistics Philosophy Department, Pittsburgh, PA, April 1996

[4] S. Cox, R. Rose: *Confidence Measures for the Switchboard Database*, in Proc. ICASSP-96, pp 511 ff, Atlanta, Mai 1996, ISBN 0-7803-3192-3

[5] E. Eide, H. Gish, P. Jeanrenaud, A. Mielke: *Understanding and improving speech recognition performance through the use of diagnostic tools*, in Proc. ICASSP-95, pp. 221 ff., vol 1, Detroit, Michigan, May 1995

[6] K. Fukunaga: *Introduction to statistical pattern recognition*, Academic Press Inc., San Diego, CA 92101, ISBN 0-12-269851-7, San Diego, 1990

[7] M. Woszczyna, M.Finke, D.Gates, M.Gavalda, T.Kemp, A.Lavie, A.McNair, L.Mayfield, M.Maier, I.Rogina, K.Shima, T.Sloboda, A.Waibel, P.Zhan, T.Zeppenfeld: *Janus II - advances in spontaneous speech translation*, in Proc. ICASSP-96, pp 409 ff, Atlanta, May 1996, ISBN 0-7803-3192-3

[8] T.Kemp, A.Jusek: 'Modelling unknown words in spontaneous speech', in Proc. ICASSP-96, pp 530 ff, Atlanta, Mai 1996, ISBN 0-7803-3192-3

[9] Sheryl Young: *Detecting misrecognitions and out-of-vocabulary words*, in Proc. ICASSP-94, pp. II-21 ff., Adelaide, Australia, April 1994

[10] T. Schultz and I. Rogina, *Acoustic and Language Modeling of Human and Nonhuman Noises for Human-to-Human Spontaneous Speech Recognition*, Proc. ICASSP 1995, vol 1, pp 293-296

[11] M. Finke, P. Geutner, H. Hild, T. Kemp, K. Ries and M. Westphal, *The Karlsruhe Verbmobil speech recognition engine*, elsewhere in this proceedings