Confidence Measures for Large Vocabulary Continuous Speech Recognition

Frank Wessel, Ralf Schlüter, Klaus Macherey, and Hermann Ney, *Member, IEEE*

***Abstract*—In this paper, we present several confidence measures for large vocabulary continuous speech recognition. We propose to estimate the confidence of a hypothesized word directly as its posterior probability, given all acoustic observations of the utterance. These probabilities are computed on word graphs using a forward–backward algorithm. We also study the estimation of posterior probabilities on N-best lists instead of word graphs and compare both algorithms in detail. In addition, we compare the posterior probabilities with two alternative confidence measures, i.e., the acoustic stability and the hypothesis density. We present experimental results on five different corpora: the Dutch ARISE 1k evaluation corpus, the German Verbmobil '98 7k evaluation corpus, the English North American Business '94 20k and 64k development corpora, and the English Broadcast News '96 65k evaluation corpus. We show that the posterior probabilities computed on word graphs outperform all other confidence measures. The relative reduction in confidence error rate ranges between 19% and 35% compared to the baseline confidence error rate.**

***Index Terms*—Confidence measures, forward–backward algorithm, N-best lists, posterior probabilities, speech recognition, word graphs.**

I. Introduction

With the rising number of different application areas for speech recognition technology, the demand for the ability to spot erroneous words also increases. In this context confidence measures can be used to label individual words in the output of the speech recognition system with either *correct* or *incorrect*, thus enabling the system and subsequent modules to spot the position of possible errors in the output automatically.

This additional assessment of the word sequence produced by the speech recognition system has been and can be used in a variety of different applications.

1) In automatic inquiry systems, e.g., train timetable information system, confidence measures can be used to avoid unnecessary and very often annoying verification turns if the confidence for the relevant keywords in the speaker utterance is high enough. If no verification is needed, the dialogue duration can be shortened. Time and money can thus be saved and the overall acceptance of the service can be increased.
2) Confidence measures can be applied to unsupervised training and adaptation algorithms, e.g., vocal tract length normalization, maximum likelihood linear regression [16], and training of acoustic models on automatically generated transcriptions. In all of these cases confidence measures can be used to confine the algorithms to those speech segments whose transcription is most probably correct.
3) Yet another application is the decoding of the speech signal itself. In [10], confidence measures are used to dynamically change the weighting between the acoustic models and the language model during the recognition process depending on the confidence of the current language model history. In [24], the authors use confidence measures directly to improve the performance of the speech recognition system.

In the following, we will try to motivate our work by discussing why the computation of confidence measures in a speech recognition system is in fact a problem. The fundamental rule in all statistical speech recognition systems is Bayes' decision rule which is based on the posterior probability $p(w_1^M|x_1^T)$ of a word sequence $w_1^M = w_1, \cdots, w_M$, given a sequence of acoustic observations $x_1^T = x_1, \cdots, x_T$. That word sequence $\{w_1^M\}_{opt}$ which maximizes this posterior probability also minimizes the probability of an error in the recognized sentence

$$\begin{aligned}\{w_1^M\}_{opt} &= \underset{w_1^M}{\text{argmax}}\; p(w_1^M|x_1^T)\\ &= \underset{w_1^M}{\text{argmax}}\; \frac{p(x_1^T|w_1^M)\cdot p(w_1^M)}{p(x_1^T)}\\ &= \underset{w_1^M}{\text{argmax}}\; p(x_1^T|w_1^M)\cdot p(w_1^M)\end{aligned}$$

where

$p(w_1^M)$	denotes the language model probability;
$p(x_1^T|w_1^M)$	acoustic model probability;
$p(x_1^T)$	probability of the acoustic observations.

Strictly speaking, the maximization is also over all sentence lengths M.

If these posterior probabilities were known, the posterior probability $p(w_m|x_1^T)$ for a specific word w_m could easily be estimated by summing up the posterior probabilities of all sentences w_1^M containing this word at position m. This posterior word probability could directly be used as a measure of confidence.

Unfortunately, the probability of the sequence of acoustic observations $p(x_1^T)$ is normally omitted since it is invariant to the choice of a particular sequence of words. The decisions during the decoding phase are thus based on unnormalized

Manuscript received December 15, 1999; revised June 23, 2000. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Hsiao-Wuen Hon.

The authors are with the RWTH Aachen, Lehrstuhl für Informatik VI, 52056 Aachen, Germany.

Publisher Item Identifier S 1063-6676(01)01328-1.

scores. These scores can be used for a comparison of competing sequences of words, but not for an assessment of the probability that a recognized word is correct. This fact, and in other words the estimation of the probability of the acoustic observations, is the main problem for the computation of confidence measures.

Before presenting our solution for this problem, we give an overview over related work on confidence measures. Previous work can be categorized using at least two different, more or less orthogonal criteria.

1) The different approaches can be distinguished by regarding whether only information is used that is extracted directly from the original models in the recognizer. Alternatively, additional models can be used that are build solely for the purpose of computing confidence measures.
2) The second criterion which can be used for a distinction is whether the confidence measures are probabilistic in the broader sense or not.

Information that can be derived from the recognizer directly is used by several authors. In [4] and [18], a large number of heuristic features, e.g., the number of times a back-off in the language model occurs or the log of the number of phones in a word, are used to compute confidence measures. Other authors try to use more probabilistic methods to solve the normalization problem without additional models. In [7] and [22] word graphs are used to compute posterior probabilities. For the computation of these probabilities N-best lists are used in [17] and [21] instead.

An example for the use of additional models is the normalization of the word scores produced by a speech recognizer with the scores obtained with a phoneme recognizer [26] or filler models [21]. These models are often referred to as garbage models. These approaches are clearly probabilistic.

In many of the cases presented above, the authors use methods from all all of these categories. Very often, a large number of features, including normalized acoustic scores and heuristic features to name only a few, is gathered and is then combined to form a single confidence measure, e.g., [2], [3]. Gillick *et al.* use generalized linear models to relate several features directly to the probability of a word to be correct [6], whereas Weintraub *et al.* use artificial neural networks to model this interdependence, [21].

The confidence measures presented in this paper are all based on word graphs and N-best lists. Word graphs and N-best lists offer an important advantage for the computation of confidence measures which was the main reason for us to pursue this direction. As we will see later on, the posterior probabilities used in Bayes' decision rule can be approximated more or less directly on the basis of word graphs and also N-best lists. With these probabilities it is very easy to compute the posterior probability for individual word hypotheses.

The organization of this paper is as follows.[1] :

- First, we describe the computation of posterior probabilities for individual words on the basis of word graphs in detail. This quantity can directly be interpreted as the probability of a word to be correct. The well-known forward–backward algorithm can easily be adapted to this problem. In particular, we study the elimination of redundant silence edges in the word graph and the scaling of the different probabilities which are needed for the computation of the confidence measures.
- Second, we study the computation of posterior probabilities on N-best lists and discuss advantages and disadvantages of both methods, extending the work presented in [17].
- Third, we compare the posterior probabilities with two alternative features suggested previously by other authors, the acoustic stability [4] and the hypothesis density [3], [7] and show the superiority of the posterior probabilities.
- Finally, we present new experimental results on five different corpora.

Apart from the alternative criteria which are used for comparison, all of our confidence measures are probabilistic and exploit only information contained in the recognizer output.

II. Word Probabilities on Word Graphs

For the following considerations, it is very useful to introduce explicit boundaries between the words in a word sequence w_1^M. Let τ denote the starting time and t the ending time of word w. With these definitions, $[w;\tau,t]$ is a specific hypothesis for this word. A sequence of M word hypotheses can thus be formulated as $[w;\tau,t]_1^M = [w_1;\tau_1,t_1],\cdots,[w_M;\tau_M,t_M]$, where $\tau_1 = 1$, $t_M = T$ and $t_{n-1} = \tau_n - 1$ for all $n = 2,\cdots,M$. In order to determine these word boundaries, we consider the following modified Bayes' decision rule. $p([w;\tau,t]_1^M|x_1^T)$ denotes the posterior probability for a sequence of word hypotheses, given the acoustic observations and $p(x_1^T|[w;\tau,t]_1^M)$ the acoustic model probability

$$\begin{aligned}\{[w;\tau,t]_1^M\}_{opt} &= \operatorname*{argmax}_{[w;\tau,t]_1^M} p([w;\tau,t]_1^M|x_1^T)\\ &= \operatorname*{argmax}_{[w;\tau,t]_1^M} \frac{p(x_1^T|[w;\tau,t]_1^M)\cdot p(w_1^M)}{p(x_1^T)}\\ &= \operatorname*{argmax}_{[w;\tau,t]_1^M} \frac{\prod_{m=1}^{M}\left[p(x_{\tau_m}^{t_m}|w_m)\cdot p(w_m|w_1^{m-1})\right]}{p(x_1^T)}.\end{aligned}$$

We assume that the generation of the acoustic observations $x_{\tau_m}^{t_m} = x_{\tau_m},\cdots,x_{t_m}$ depends on word w_m only. With these word boundaries, the posterior probability $p([w;\tau,t]|x_1^T)$ for a specific word hypothesis $[w;\tau,t]$ can be computed by summing up the posterior probabilities of all sentences which contain the hypothesis $[w;\tau,t]$

$$\begin{aligned}&p([w;\tau,t]|x_1^T)\\ &\quad= \sum_{\substack{[w;\tau,t]_1^M:\\ \exists\, n\in\{1,\cdots,M\}:\\ [w_n;\tau_n,t_n]=[w;\tau,t]}} \frac{\prod_{m=1}^{M}\left[p(x_{\tau_m}^{t_m}|w_m)\cdot p(w_m|w_1^{m-1})\right]}{p(x_1^T)}.\end{aligned} \tag{1}$$

[1] Some parts of this work were reported in [22] and [23]

It should be noted that the posterior probabilities of all parallel word graph edges hypothesized at a specific point in time t always sum up to one. Such a point in time can be interpreted as a cut through the word graph and it is evident that the total probability to intersect this cut must equal one

$$\sum_{\substack{[w;\tau',t']:\\ \tau'\leq t\leq t'}} p([w;\tau',t']|x_1^T) = 1 \quad \forall\, t \in \{1,\cdots,T\}.$$

Fig. 3 illustrates this property. In this example we assume that the language model probabilities and the acoustic model probabilities are uniform without loss of generality. As the illustration shows, the posterior probabilities sum up to one for any point in time.

In the following section, we discuss the construction of word graphs which can be used to approximate the posterior probabilities $p([w;\tau,t]|x_1^T)$ for a specific word hypothesis.

A. Construction of the Word Graphs

Let us first define the term *word graph*. In this paper, a word graph is a directed, acyclic, weighted graph. Its nodes $t \in \{1,\cdots,T\}$ represent discrete points in time, its edges word hypotheses $[w;\tau,t]$ for word w from node τ to node t and its weights the acoustic probabilities of the hypotheses. Any path through the word graph, i.e., any sequence of word hypotheses from the node corresponding to the first time frame of the utterance to the node corresponding to the last time frame, forms an alternative sentence hypothesis. As above, we use the notation $[w;\tau,t]_1^M = [w_1;\tau_1,t_1],\cdots,[w_M;\tau_M,t_M]$ for such a sequence of word hypotheses with given boundary times.

In our current speech recognition system [12], the word graphs are generated time-synchronously in one pass using the *word-conditioned lexical tree search method* [14]. During the recognition phase we store the most probable word end hypotheses $[w;\tau,t]$ (those hypotheses that survived the pruning process) for each time frame t. For each of these hypotheses we also store the acoustic probability $p(x_\tau^t|w)$ and the immediate predecessor word $v([w;\tau,t])$. This strategy results in a *word-conditioned word graph* [13], [25]. In a subsequent optimization step, illustrated in Fig. 1, the final word graph is constructed by merging all nodes with identical associated times into a single node. If there are parallel edges with the same word identity, only one of them is retained in the word graph. The immediate predecessor words $v([w;\tau,t])$ which were stored for each word hypothesis before are now subsumed in a list of predecessor words for each edge. The list of predecessors is needed to speed up the word boundary optimization and the pruning process using a sentence hypothesis tree [11] during the rescoring of the word graph.

During the computation of the confidence measures we do not use the additional information about the predecessors of a word graph edge. In other words, we regard all possible transitions between edges starting and ending in a specific word graph node. This strategy can easily be justified since the reduced number of possible transitions was only caused by the different pruning steps during the search process. In principal, all of these transitions are possible and should be considered.

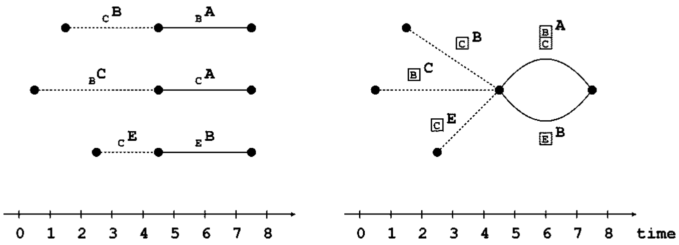

Fig. 1. The left side of the illustration shows a list of six hypotheses $[w;\tau,t]$ for a vocabulary $\mathcal{A} = \{A,B,C,D,E,F\}$. In this example, the acoustic probabilities $p(x_\tau^t|w)$ are omitted for readability. The lower index corresponds to the language model predecessor $v([w;\tau,t])$. On the right side of the illustration the resulting word graph is shown after hypotheses with the same word index, the same starting, and the same ending time were merged. Now, a list of the predecessors is attached to each edge. As already mentioned, we do not use this list to restrict the possible transitions during the computation of our confidence measures.

A word graph can thus be regarded as a limited representation of the infinite space of possible solutions for the maximization problem defined in Bayes' decision rule. If the word graph is constructed as described above, it contains the most probable sentence hypotheses and can thus be used to approximate the posterior probabilities defined in (1).

B. Computation of the Hypothesis Probabilities

In the following, we discuss how the posterior probability for a word hypothesis can be computed on the basis of word graphs. In the style of the forward-backward algorithm we compute the forward probability and the backward probability for a word hypothesis and combine both probabilities into the posterior probability of this hypothesis. In contrast to the forward-backward algorithm on a hidden Markov model state level the forward–backward algorithm is now based on a word hypothesis level. Let us assume that a word graph is given and that we use an m-gram language model and let $h_2^{m-1} = h_2,\cdots,h_{m-1}$ be the $m-2$ immediate predecessor words of word w (from now on referred to as the *history* of word w). We can now compute the forward probability $\Phi(h_2^{m-1};[w;\tau,t])$ that the last hypothesis of a sequence of n word hypotheses is $[w;\tau,t]$ and that its history is h_2^{m-1}

$$\begin{aligned}\Phi(h_2^{m-1};[w;\tau,t]) &\\ = \sum_{\substack{[w;\tau,t]_1^n:\\ [w_n;\tau_n,t_n]=[w;\tau,t]\wedge\\ w_{n-m+2}^{n-1}=h_2^{m-1}}} \prod_{i=1}^{n} p(x_{\tau_i}^{t_i}|w_i) &\\ \cdot\, p(w_i|w_{\max\{1,\,i-m+1\}}^{i-1}). &\qquad (2)\end{aligned}$$

The sum in the equation above is over all partial paths $[w;\tau,t]_1^n$ through the word graph which end in hypothesis $[w;\tau,t]$ and whose last $m-2$ language model predecessors are h_2^{m-1}. In order to simplify the notation we set $p(w_1|w_1^0) = p(w_1)$ and $p(w_2|w_1^1) = p(w_2|w_1)$. Equation (2) can be evaluated very efficiently in a recursive manner if the word hypotheses can be accessed directly using both, their starting and ending times. The forward probabilities are computed chronologically in an ascending order

$$\begin{aligned}\Phi(h_2^{m-1};[w;\tau,t]) \\ &= p(x_\tau^t|w)\cdot\sum_{h_1}\sum_{\tau'}\Phi(h_1^{m-2};[h_{m-1};\tau',\tau-1]) \\ &\quad\cdot p(w|h_1^{m-1}).\end{aligned}\tag{3}$$

Since τ is the starting time of word w, $\tau-1$ denotes the ending time of the preceding word h_{m-1}. Edges which represent segments of silence in the speech signal require special treatment. In order to keep track of the language model history, the forward probabilities for silence edges have to be computed separately for each preceding word.

Analogously, let f_1^{m-2} denote the $m-2$ immediate successor words of word w (from now on referred to as the *future* of word w). With this definition we can compute the backward probability $\Psi([w;\tau,t];f_1^{m-2})$ that the first hypothesis of a sequence of n word hypotheses is $[w;\tau,t]$ and that its future is f_1^{m-2}

$$\begin{aligned}\Psi([w;\tau,t];f_1^{m-2}) \\ &= \sum_{\substack{[w;\tau,t]_1^n:\\ [w_1;\tau_1,t_1]=[w;\tau,t]\wedge\\ w_2^{m-1}=f_1^{m-2}}}\prod_{i=1}^{n}p(x_{\tau_i}^{t_i}|w_i) \\ &\quad\cdot\prod_{j=m}^{n}p(w_j|w_{j-m+1}^{j-1}).\end{aligned}\tag{4}$$

The language model probabilities for all words $w_1,\cdots,w_{m-1}$ are computed later in (6), because at this stage in the algorithm, the language model history for these words is not known. Silence edges are treated as described above. Equation (4) can be evaluated recursively as well. The backward probabilities are computed in a descending order

$$\begin{aligned}\Psi([w;\tau,t)];f_1^{m-2}) \\ &= p(x_\tau^t|w)\cdot\sum_{f_{m-1}}\sum_{t'}\Psi([f_1;t+1,t'];f_2^{m-1}) \\ &\quad\cdot p(f_{m-1}|wf_1^{m-2}).\end{aligned}\tag{5}$$

With the definitions in (1), (3) and (5), the posterior hypothesis probability can now be computed by summing over all histories and futures of the word hypothesis $[w;\tau,t]$

$$\begin{aligned}p([w;\tau,t]|x_1^T) \\ &= \sum_{h_2^{m-1}}\sum_{f_1^{m-2}}\frac{\Phi(h_2^{m-1};[w;\tau,t])\cdot\Psi([w;\tau,t];f_1^{m-2})}{p(x_1^T)\cdot p(x_\tau^t|w)} \\ &\quad\cdot\prod_{n=1}^{m-2}p(f_n|h_{n+1}^{m-1}wf_1^{n-1}).\end{aligned}\tag{6}$$

The last term in this equation represents the language model probabilities which are missing in (4), as mentioned above. The fraction above has to be divided by the acoustic probability for algorithmic reasons, because it was included twice, in (3) and (5). $p(x_1^T)$ in the denominator of (6) can be evaluated as follows:

$$\begin{aligned}p(x_1^T) &= \sum_{h_2^{m-1}}\sum_{w}\sum_{\tau}\Phi(h_2^{m-1};[w;\tau,T]) \\ &= \sum_{w}\sum_{t}\sum_{f_1^{m-2}}\Psi([w;1,t];f_1^{m-2}) \\ &\qquad\cdot\left[\prod_{n=1}^{m-2}p(f_n|w,f_1^{n-1})\right]\cdot p(w)\end{aligned}\tag{7}$$

where the last unigram probability has to be computed only if $m > 1$. As (7) reflects, the prior probability of the acoustic observations can be computed in two different ways. In our current implementation of the forward-backward algorithm this redundancy is used to assure that the forward and backward probabilities are computed correctly.

It is also interesting to note that the total number of distinct paths through a specific word graph edge and through the word graph in general can easily be computed with the forward-backward algorithm. By setting all language model probabilities and all acoustic probabilities to one, the product of the forward and the backward probabilities for each edge, i.e., (6) without renormalization with $p(x_1^T)$, corresponds directly to the number of paths through this edge and $p(x_1^T)$ directly to the total number of paths through the word graph.

The posterior hypothesis probability defined in (6) can now directly be used as a measure of confidence for each individual word hypothesis

$$\mathcal{C}([w;\tau,t]) = p([w;\tau,t]|x_1^T).\tag{8}$$

In our experiments, the confidence is later on compared with a tagging threshold optimized on a cross validation corpus beforehand. Words whose confidence exceeds this threshold are tagged as *correct*, all others as *false*.

C. Scaling of the Probabilities

In addition to the language model scaling factor β, we also use a scaling factor $\alpha < 1$ to scale the acoustic model probabilities. This scaling has a major impact on the computation of the posterior probabilities and their performance as a confidence measure. If the acoustic model probabilities are not scaled appropriately, the sums in all of the equations above are dominated by only a few word graph hypotheses because of the very large dynamic range of the acoustic scores (i.e., the negative logarithm of the unnormalized acoustic probabilities). The differences in the acoustic scores are mainly due to the variance of the acoustic features which is presumably underestimated. Since a re-estimation of these variances is very difficult, the acoustic probabilities have to be scaled in order to obtain useful results. This additional scaling can be regarded as a broadening of the variance of the acoustic model probabilities.

Both parameters α and β have to be estimated on a cross-validation corpus which must be distinct from the testing corpus in order to avoid over-adaptation. During the forward–backward algorithm, all language model probabilities are scaled with the factor β and all acoustic model probabilities with the factor α.

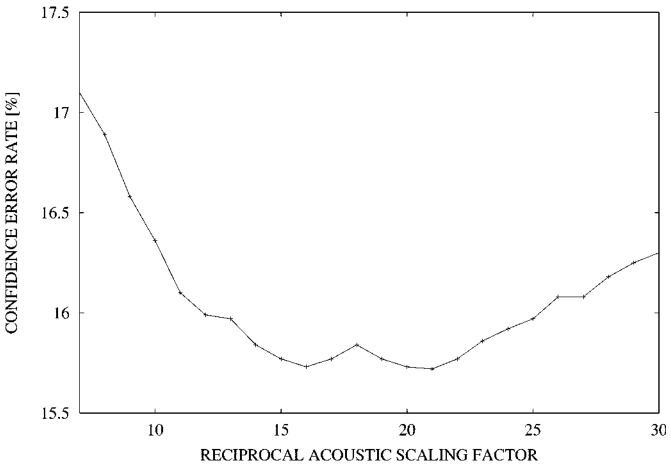

Fig. 2. Confidence error rate on the Verbmobil development corpus for different acoustic scaling factors α using the plain posterior hypothesis probabilities $p([w;\tau,t]|x_1^T)$ as a measure of confidence.

To specify how exactly the scaling factors are used, we take (3) as an example and rewrite it with all scaling factors

$$\Phi(h_2^{m-1};[w;\tau,t]) = p(x_\tau^t|w)^\alpha \cdot \sum_{h_1}\sum_{\tau'}\Phi(h_1^{m-2};[h_{m-1};\tau',\tau-1]) \cdot p(w|h_1^{m-1})^\beta.$$

All other equations are modified accordingly.

In order to illustrate the effect of the acoustic scaling factor α we ran a simple experiment on the Verbmobil development corpus, one of the corpora that we used for our experiments. We computed the hypothesis probabilities defined in (8) for each hypothesis in the word graph and tagged each word in the recognized sentence as either *correct* or *false* depending on whether the corresponding probability exceeded a certain threshold or not. Then we optimized this tagging threshold so as to minimize the confidence error rate[2] and plotted this error rate over the different acoustic scaling factors. As Fig. 2 clearly shows, the performance of the posterior hypothesis probability as a confidence measure strongly depends on the correct choice of the acoustic scaling factor α. We also observed that the optimal value for the language model scaling factor β, when optimizing α and β on the cross-validation corpus, is not identical but very close to 1.0. This is exactly what one would expect since the language model probabilities are normalized.

D. Elimination of Redundant Silence Edges

An important aspect is the elimination of redundant silence edges. As described in Section II-A the word graphs are optimized by merging all nodes with identical associated times into a single node and by retaining only one of parallel edges with the same word identity. In doing so, the dependence of a word edge on its predecessor is resolved and a directed acyclic graph in its classical sense is obtained.

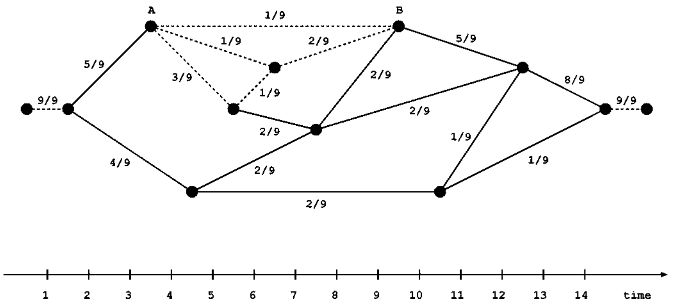

Fig. 3. Simple word graph. The solid edges represent word hypotheses, whereas the dashed edges represent silence hypotheses. In this example we assume that the language model probabilities and the acoustic model probabilities are uniform without loss of generality. For each edge $[w;\tau,t]$ the posterior hypothesis probability $p([w;\tau,t]|x_1^T)$ is specified. As the illustration shows, these probabilities sum up to one for any point in time.

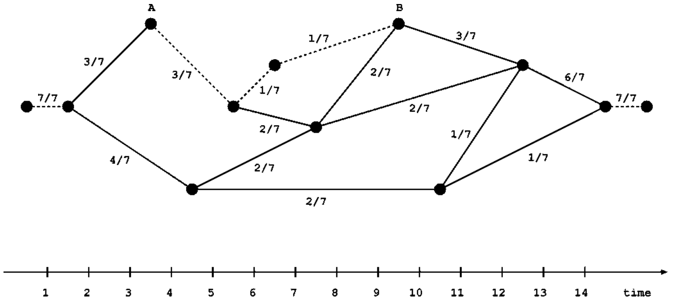

Fig. 4. Same word graph as in Fig. 3 after redundant silence edges have been removed. Again, we assume that the language model probabilities and the acoustic model probabilities are uniform. Note the different posterior probabilities for all edges $[w;\tau,t]$ in comparison with Fig. 3.

Sequences of silence edges do not exist in the original word graph before this optimization step is carried out. Omitting the dependence on the predecessor words for silence edges in particular, these sequences of silence edges can now come into existence. As Fig. 3 shows, there are three different paths consisting of silence edges (dashed lines) from node A to node B in the graph. Without sequences of silence, there would be only one path. From an algorithmic point of view, the two additional paths do not cause any problems. On the other hand, they do not contain any additional information. All three parallel paths represent silence in the same part of the speech signal. Since the posterior probabilities of all edges intersecting a given time frame sum up to unity, these parallel, more or less identical paths have a great influence on the posterior probabilities of all these edges. Therefore, two of the paths should be removed from the word graph. To maintain all other possible paths through the word graph, we remove all silence edges from the word graph which can be bridged by a sequence of shorter silence edges. Fig. 4 shows the impact of this additional step during the preprocessing of the word graphs on the posterior probabilities.

Apart from the fact that the redundant silence edges should be removed from a theoretical point of view, there is an algorithmic advantage of this processing step. The number of paths through the word graph can be reduced drastically if the word graphs contain a lot of silence edges. The computing time and

[2] The confidence error rate (CER) is simply defined as the number of incorrectly assigned tags, i.e., labels *correct* and *false*, divided by the total number of recognized words. The baseline confidence error rate is given by the number of insertions and substitutions, divided by the number of recognized words, i.e., tagging all words as *correct*.

the memory needed for the estimation of the forward and backward probabilities can thus be reduced.[3]

E. Computation of the Word Probabilities

In the experiments presented in Table II, the posterior hypothesis probabilities defined in (8) turned out to perform hardly better than the baseline confidence error rate[4]. In fact, this observation is not surprising since the fixed starting and ending time of a word hypothesis determine which paths in the word graph are considered during the computation of the forward-backward probabilities. Usually, several hypotheses with slightly different starting and ending times represent the same word and the probability mass of the word is split among them. The unsatisfactory performance of the confidence measure defined in (8) is indeed a strong indication that this problem, later on referred to as *segmentation of the word graph*, needs to be solved.

In the following, we describe several methods we implemented. In a first attempt we summed up the probabilities of all hypotheses with an identical word index for which the intersection of the time intervals defined by the starting and ending times of the considered hypotheses is not empty. From now on this confidence measure is referred to as $\mathcal{C}_{sec}([w;\tau,t])$

$$\mathcal{C}_{sec}([w;\tau,t]) = \sum_{\substack{[w;\tau',t']:\\ \{\tau,\cdots,t\}\cap\{\tau',\cdots,t'\}\neq\emptyset}} p([w;\tau',t']|x_1^T). \tag{9}$$

It is important to note that the accumulation for this and all following criteria is carried out over all different pronunciation variants of word w. As the results in Table II show, (9) performs significantly better than the posterior hypothesis probabilities defined in (8) on all testing corpora.

Unfortunately, the sum of the accumulated posterior probabilities over all different words for one specific time frame does no longer sum up to unity by definition. Although performing better in terms of confidence error rate, the question remains whether the missing normalization has an effect on the confidence measure. If we restrict the accumulation of the posterior hypothesis probabilities to a single time frame, common to all hypotheses for a specific word, the accumulated probabilities for all different words at a given point in time do sum up to unity. Thus, we accumulate only the posterior probabilities of those hypotheses for word w which intersect the median time frame of the hypothesis under consideration, later on referred to as $\mathcal{C}_{med}([w;\tau,t])$

$$\mathcal{C}_{med}([w;\tau,t]) = \sum_{\substack{[w;\tau',t']:\\ \tau'\leq\lceil\tau+t/2\rceil\leq t'}} p([w;\tau',t']|x_1^T). \tag{10}$$

As our results show, the performance is comparable with the confidence measure defined in (9) and the effect of the lacking normalization seems to be negligible.

In a final experiment, we investigated whether the choice of the time frame has an effect on the performance. We carried out the accumulation not only for the median time frame of the current hypothesis but for all of its time frames and chose the maximum of these values as a measure of confidence, from now on referred to as $\mathcal{C}_{\max}([w;\tau,t])$. The idea here is to determine the *best-case* probability for a given word to occur in a certain period of time

$$\mathcal{C}_{\max}([w;\tau,t]) \max_{t_{\max}\in\{\tau,\cdots,t\}} \sum_{\substack{[w;\tau',t']:\\ \tau'\leq t_{\max}\leq t'}} p([w;\tau',t']|x_1^T). \tag{11}$$

As the experiments presented in Table II indicate, this measure performs slightly better than the one defined in (10) and we therefore decided to use this quantity as our standard word graph confidence measure.

III. Word Probabilities on N-Best Lists

In [17], Rueber suggests to compute posterior probabilities for semantic items in a recognized sentence on N-best lists. This approach is very similar to the computation of posterior probabilities on word graphs and can easily be extended to the computation of posterior probabilities for individual words. Before describing this extension we first have to discuss two possible ways of defining what an N-best list contains the following.

1) An N-best list can be defined as the list of the N best sentence hypotheses $[w;\tau,t]_1^{M_1},\cdots,[w;\tau,t]_1^{M_N}$. With this definition, several of the hypotheses in the list can of course be identical when comparing only on a word level. In this case, only the starting and ending times of one or several words are different. It is obvious that using such an N-best list, the posterior probability defined in (1) can easily be computed by summing over all sentence hypotheses containing the specified word hypothesis $[w;\tau,t]$. If the N-best list contains exactly the same sentence hypotheses as the word graph, the posterior probabilities computed on the word graph and the N-best list are identical. On the other hand, the advantage of the word graph becomes very obvious. Using the word graph, the posterior probabilities can be computed very efficiently with the forward–backward algorithm in a dynamic programming fashion. When using the N-best list, all of the sentence hypothesis probabilities have to be added explicitly. In other words, there is no need to use this type of N-best lists since a word graph is a more compact and efficient representation of the set of different sentence hypotheses.
2) Alternatively, N-best lists can be defined as the N best sentence hypotheses, each of which contains only the sequence of words, but no information about the starting and ending times of the words. This definition is more straightforward than the first and bears an interesting advantage over word graphs. As described in the previous sections, the relaxation of the word graph segmentation is crucial for a reliable estimation of the confidence. N-best

[3] The effect on the performance of the confidence measure is very small on all of our testing corpora. The main advantage is, in fact, the reduced number of forward and backward probabilities that have to be computed.

[4] As before, the baseline confidence error rate is defined as the number of insertions and substitutions divided by the number of recognized words, i.e., tagging all words as *correct*.

lists that are based on word positions only and that contain no information about starting and ending times do not suffer from this segmentation problem and might thus be used to compute confidence measures without having to accumulate posterior probabilities of several word hypotheses.

Since there is no advantage of N-best lists over word graphs when using the first definition, we decided to focus on the second type of N-best lists. In our speech recognition system, the N-best lists are constructed on the basis of word graphs. The algorithm is comparable to the normal word graph rescoring algorithm [11], the only difference being that instead of keeping only the best out of several hypotheses for each language model history at a specific point in time, we keep the N best hypotheses for each language model history. It is important to note that there are no approximations or pruning steps during the generation of the N-best lists.

As we already noted, the second type of N-best lists is based only on word positions. Unfortunately, the concept of a *word position* is somehow ill-defined. First, the sentences $v_1^{M_1}, \cdots, v_1^{M_N}$ in the N-best list may have different lengths. Second, even if all lengths were identical, word positions could not be compared directly due to possible deletion and insertion errors. Thus, the different sentence hypotheses have to be aligned using dynamic programming. In [8], Levensthein suggests an algorithm to compute that alignment which minimizes the sum of insertions, deletions, and substitutions when comparing two different sentences. The important feature of this algorithm is the following: for word w_m at position m in the reference sentence w_1^M we are able to assign the corresponding word v in any of the other sentences $v_1^{M_1}, \cdots, v_1^{M_N}$. We denote this by $v = \mathcal{L}_m(w_1^M, v_1^{M_n})$. Using this definition, the posterior word probability for each word w_m in the reference sentence w_1^M can easily be computed

$$p(w_m|x_1^T, w_1^M, \mathcal{L}) = \frac{\sum_{n=1}^{N} p(x_1^T|v_1^{M_n})^\alpha \cdot p(v_1^{M_n})^\beta \cdot \delta(w_m, \mathcal{L}_m(w_1^M, v_1^{M_n}))}{\sum_{n=1}^{N} p(x_1^T|v_1^{M_n})^\alpha \cdot p(v_1^{M_n})^\beta}$$

where the Kronecker function δ returns 1 if both arguments are identical and 0 if they are different. As the equation above shows, we used a scaling factor α again to scale the acoustic model probabilities. As before, β is the language model scaling factor. The scaling factor α greatly influences the performance of this confidence measure just like in our experiments on word graphs. As shown later, the confidence measure computed on N-best lists performs quite well on most of the testing corpora. We will discuss the differences between the computation on N-best lists and word graphs in detail later on in Section V.

IV. Alternative Criteria Measures

In order to compare the posterior probabilities with alternative criteria we implemented the acoustic stability criterion [4] and the hypothesis density criterion [7].

A. Computation of the Acoustic Stability

The motivation for the acoustic stability is that a word is most probably correct if it is contained at the same position, specified by the Levensthein alignment, in the majority of sentences generated with different weighting between the acoustic and the language model scores. We implemented the acoustic stability criterion as follows. In a first step, we rescore the word graph with the standard language model scaling factor β_{ref} in order to obtain the first-best sentence w_1^M. Second, we rescore the word graph with N different language model scaling factors and obtain N alternative first-best sentences $v_1^{M_1}, \cdots, v_1^{M_N}$. The N language model scales $\beta_1, \cdots, \beta_N$ are equidistant values taken from the interval $[(1-\gamma) \cdot \beta_{ref} \cdots (1+\gamma) \cdot \beta_{ref}]$. All of these sentences $v_1^{M_1}, \cdots, v_1^{M_N}$ are then aligned with the reference sentence w_1^M using the Levensthein algorithm again. Pronunciation variants of the same word w are again treated as equal, since they only represent different hypotheses for the same word. The relative frequency of any word taken from the reference sentence occurring at the same position in all of the sentences $v_1^{M_1}, \cdots, v_1^{M_N}$ is a direct measure for the acoustic stability

$$\mathcal{C}_{acu}(w_n) = \frac{1}{N} \cdot \sum_{n=1}^{N} \delta(w_n, \mathcal{L}_n(w_1^M, v_1^{M_n})).$$

As the experiments presented in Table II show, this criterion performs well on all corpora, except for the Broadcast News evaluation corpus.

B. Computation of the Hypothesis Density

Another criterion suggested previously is the hypothesis density [7]. In order to reduce the computational complexity during the decoding process, unlikely hypotheses are usually pruned, using a variety of different pruning strategies. If a large number of hypotheses have similar scores at the same point in time, no effective pruning will take place and the number of hypotheses which is stored in the word graph will be above average. Since a word is usually hypothesized several times with different starting and ending times, we count each word only once while computing the hypothesis density for a given time frame. Let WG denote the set of all hypotheses contained in the word graph. The hypothesis density for time frame t can then be computed as follows:[5]

$$D(t') = |\{w : [w; \tau, t] \in WG \ \wedge \ \tau \le t' \le t\}|.$$

To capture the time dependence of this quantity we used the average hypothesis density in the time interval $\{\tau, \cdots, t\}$ as our measure of confidence

$$\mathcal{C}_{den}([w; \tau, t]) = \frac{1}{t - \tau + 1} \cdot \sum_{t'=\tau}^{t} D(t').$$

As Table II clearly shows, the performance of this criterion is disappointing on most of the evaluation sets. We will try to explain possible reasons in the next section.

[5] The notation $|X|$ indicates the number of elements in (or the size of) a set X.

TABLE I
SUMMARY OF THE EXPERIMENTAL SETUP AND SPECIFICATION OF THE WORD GRAPHS USED IN THE EXPERIMENTS. WGD DENOTES THE WORD GRAPH DENSITY, NGD THE NODE GRAPH DENSITY, BGD THE BOUNDARY GRAPH DENSITY AND GER THE WORD GRAPH ERROR RATE

	size of	WGD	NGD	BGD	GER	trigram	del - ins - **WER**
corpus	voc.				[%]	perpl.	[%]
ARISE	985	218.8	86.0	24.4	7.4	12.6	2.1 - 3.2 - **15.8**
Verbmobil	7128	209.2	73.1	18.3	8.7	56.1	6.1 - 6.9 - **33.6**
NAB 20k	19987	98.4	47.5	10.9	4.1	124.5	1.9 - 2.1 - **13.2**
NAB 64k	64736	87.1	43.9	10.0	1.8	145.9	2.0 - 1.5 - **11.1**
Broadcast News	65491	105.5	39.1	10.1	10.6	213.7	6.0 - 4.3 - **33.3**

V. EVALUATION OF CONFIDENCE MEASURES

Once the confidence has been computed, each word of the recognized sentence is simply tagged as either *correct* or *false*, depending on whether its confidence exceeds a certain threshold or not. Here, two different types of errors can occur. The first is a *false acceptance*, i.e., a false word is tagged as *correct*, and the second is a *false rejection*, i.e., a correct word is tagged as *false*. Obviously, there is a trade-off between the two types of errors, depending on the choice of the tagging threshold.

Before we present experimental results for all of the confidence measures described in the sections before, we discuss several of the evaluation metrics for confidence measures which were suggested previously.

The confidence error rate (CER), already introduced before, is is simply defined as the number of incorrectly assigned tags divided by the total number of recognized words. The baseline CER is given by the number of insertions and substitutions, divided by the number of recognized words. The CER does of course strongly depend on the choice of the tagging threshold. Therefore, the threshold should not be adjusted to minimize the CER on the testing corpus, but on a clearly distinct cross-validation corpus. With this threshold the confidence measure can then be evaluated on the testing corpus. The main drawback of the CER is that it depends on the prior probability of the two classes *correct* and *false*.

Another criterion is the equal-error-rate (EER). The EER can be computed by adjusting the tagging threshold so that the false acceptance and the false rejection error rate are equal. Just like the CER, the EER strongly depends on the prior probability of the two classes *correct* and *false*.

Yet another evaluation metric is the detection-error-tradeoff (DET) curve which comprises the equal-error-rate as one of many possible operating points. The DET curve simply contains a plot of the false acceptance rate over the false rejection rate.

The final criterion discussed here is the normalized cross entropy. For a definition, the reader is referred to [19]. In our opinion, this quantity cannot be used to evaluate the confidence measures presented in this paper. Although it can easily be interpreted, it approaches infinity as soon as the posterior probability of a word equals one, despite the fact that this word was not recognized correctly. Two different ways to elude this problem are to remove all words from the test corpus whose posterior probabilities are one although they were recognized incorrectly or to limit the posterior probability to a value below 1.0. Instead, we confine ourselves to the use of the confidence error rate because of its simplicity and to the DET curve because it contains a high amount of information for different operating points of the system.

A. Experimental Setup

We carried out experiments on five different corpora. The English NAB'94 20k dev corpus [15] consists of read newspaper articles, recorded under high-quality conditions. The NAB'94 64k dev task uses the same evaluation corpus with a larger vocabulary. The Broadcast News '96 evaluation corpus [5] consists of broadcast television and radio news. The German Verbmobil '98 evaluation corpus [1] consists of spontaneous human-to-human dialogues, also recorded under high-quality conditions. The Dutch ARISE corpus [9] is composed of human-to-machine dialogues, recorded over the telephone with an automatic train timetable information system in The Netherlands. Table I summarizes the experimental setup and specifies the word graphs generated with our speech recognition system [12]. The word graph density (WGD) is defined as total number of word graph edges divided by the number of spoken words, the node graph density (NGD) as the total number of different words ending at each time frame divided by the number of spoken words, and the boundary graph density (BGD) as the number of different word boundaries, i.e., different starting and ending times, per spoken word. The graph error rate (GER) is computed by determining that sequence of word hypotheses through the word graph which best matches the spoken sentence. This measure provides a lower bound for the word error rate which can be achieved with a given word graph. For details on these quantities the reader is referred to [13].

For all of the following experiments we optimized all model parameters, i.e., the acoustic scaling factors, the language model scaling factors and the tagging thresholds on separate cross-validation sets beforehand in order to avoid over-adaptation on the testing data.

B. Confidence Error Rates

Table II comprises the baseline confidence error rates and the confidence error rates achieved with the different confidence measures on the five testing corpora.

Before comparing the different confidence measures presented in this paper, we focus on the different relaxation strategies which are necessary to compute confidence measures

TABLE II
CONFIDENCE ERROR RATES FOR THE DIFFERENT CONFIDENCE MEASURES PRESENTED IN THIS PAPER. ALL ERROR RATES ARE GIVEN IN [%]. THE BASELINE CONFIDENCE ERROR RATE IS DEFINED AS THE NUMBER OF INSERTIONS AND SUBSTITUTIONS DIVIDED BY THE NUMBER OF RECOGNIZED WORDS

corpus	baseline	$\mathcal{C}_{den}$	$\mathcal{C}_{acu}$	N-best				word graph			
				100	200	300	1000	$\mathcal{C}$	$\mathcal{C}_{sec}$	$\mathcal{C}_{med}$	$\mathcal{C}_{max}$
ARISE	13.6	11.7	8.2	8.9	8.6	9.0	9.1	11.5	8.9	8.8	**8.9**
Verbmobil	27.3	26.0	22.1	21.7	21.2	21.1	21.6	23.3	19.0	20.0	**18.9**
NAB 20k	11.3	11.3	9.9	9.4	9.2	9.1	9.2	10.3	9.2	9.2	**9.2**
NAB 64k	9.2	9.1	8.0	7.5	7.5	7.6	7.6	8.4	7.2	7.2	**7.2**
Broadcast News	27.7	27.7	25.8	25.3	25.3	25.2	25.4	23.7	20.6	20.4	**20.6**

using the forward-backward algorithm on word graphs. Table II shows the effect of the different strategies. Obviously, the relaxation of the word graph segmentation is essential for the computation of the confidence measure using the criteria $\mathcal{C}_{sec}([w;\tau,t])$, $\mathcal{C}_{med}([w;\tau,t])$, and $\mathcal{C}_{\max}([w;\tau,t])$. Compared to the posterior hypothesis probability $\mathcal{C}([w;\tau,t])$ defined in (8) all of the accumulated probabilities perform significantly better. The missing probability normalization for $\mathcal{C}_{sec}([w;\tau,t])$ only has a negligible effect on the performance. Since $\mathcal{C}_{\max}([w;\tau,t])$ yields the best results we chose this criterion as our standard confidence measure for all further comparisons with the other methods. It should also be mentioned that there is almost no degradation in performance when using less dense word graphs. In order to study this effect, we pruned the five word graphs using a forward–backward pruning algorithm [20] and repeated the experiments. We observed no loss in performance as long as the WGD remained above 2.5 for the ARISE and the NAB 20k corpus, above 5.0 for the NAB 64k corpus, and above 25.0 for the Verbmobil and the Broadcast News corpus.

Table II shows that there is no significant difference between the computation of posterior probabilities on word graphs and the computation on N-best lists for the ARISE, the NAB 20k, and the NAB 64k task. On the Verbmobil and the Broadcast News task on the other hand, the word graph posterior probabilities perform significantly better than those computed on N-best lists. Apparently, the information contained in the N best alternative sentence hypotheses is only sufficient for the first three corpora. For the ARISE corpus we attribute this effect to very short average length of the utterances which is 3.4 words per sentence. It is more difficult to explain the good performance of the N-best list criterion on the NAB tasks especially when comparing the results with the performance on the Broadcast News evaluation corpus. The NAB 64k and the Broadcast News tasks are both defined for a vocabulary with more than 64k words and both corpora contain very long sentences. The size of the N-best list alone, which might in fact be too small, can thus not be the only explanation. The main difference between both corpora is that the first was recorded under high-quality conditions and that it contains read speech. Due to the higher quality of the acoustic models and the language model, the probability distributions discriminate better between different hypotheses. As a result, fewer sentence hypotheses contribute to the posterior probabilities and a smaller size of the N-best list is thus sufficient.

In fact, the combination of the size of the N-best lists and the quality of the acoustic models and the language model seems to be the explanation for the rather poor performance on the Verbmobil and the Broadcast News tasks. One could argue that by simply increasing the size of the N-best lists this disadvantage can easily be compensated for. Unfortunately, we observed an increase in CER for $N = 1000$. A detailed analysis of the N-best lists showed that words are occasionally aligned which do not represent the same segment in time. Obviously, the Levensthein algorithm does not always lead to reasonable alignments and causes additional problems. Another aspect is that it is more difficult to handle very large N-best lists efficiently.

For the acoustic stability criterion we used $\gamma = 0.9$ and $N = 100$. We noticed only a negligible change in performance for larger values of N and different values of γ. As Table II indicates, the acoustic stability achieves good results on all corpora, except for the Broadcast News evaluation corpus. Nevertheless, the acoustic stability is clearly not able to outperform the posterior probability on word graphs. Only on the ARISE corpus it performs extraordinarily well. As before, we attribute this effect to the very short average length of the utterances.

The hypothesis density criterion is also not able to outperform the performance of the accumulated posterior probability $\mathcal{C}_{\max}([w;\tau,t])$. The number of parallel hypotheses for a given time frame is obviously not sufficient as a confidence measure.

C. Detection-Error Tradeoff Curves

The DET curves in Figs. 5–9 support the analysis presented above. For all of the five testing corpora the word graph based confidence measure yields the best results.

Unfortunately, we were not able to plot the DET curves for the acoustic stability criterion for all possible operating points. Here, the problem is that a rather large number of incorrect words occur at the same position in all of the M different sentences. As soon as the tagging threshold is smaller than 1.0, all of these incorrectly recognized words are automatically tagged as correct. There is no way to compute the DET curve between the 0.0% *false acceptance rate*/100.0% *false rejection rate* point in the plot and the point where the DET curves for the acoustic stability start. Therefore we did not draw a connecting line between these two operating points.

The same problem occurs for the N-best list criterion. Even for $N = 1000$ we were not able to plot the curves for all operating points. The explanation is the same as for the acoustic

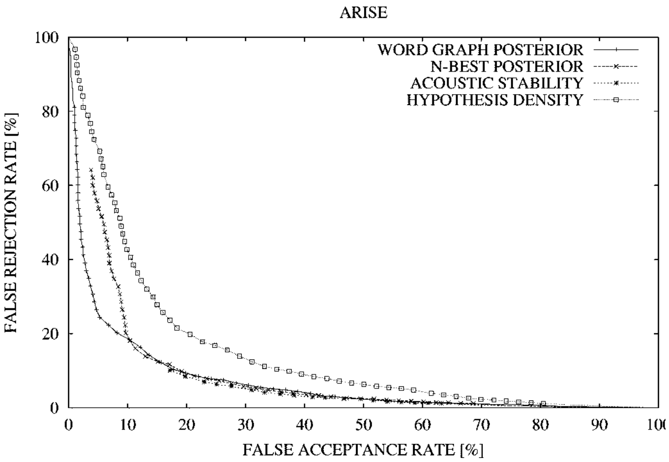

Fig. 5. DET curves for the ARISE evaluation corpus.

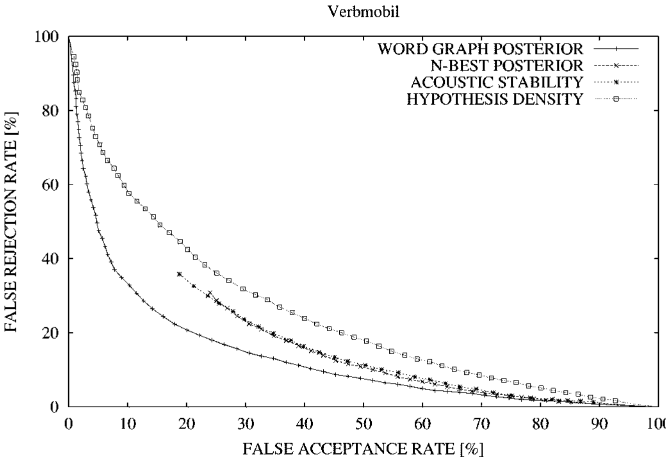

Fig. 6. DET curves for the Verbmobil evaluation corpus.

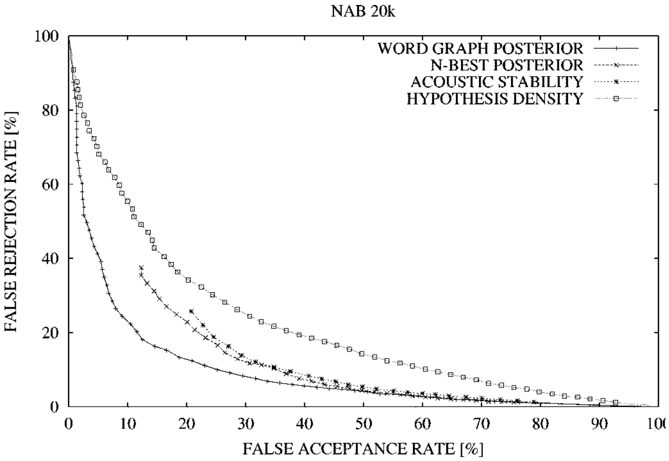

Fig. 7. DET curves for the NAB 20k evaluation corpus.

stability criterion. If all of the N different sentences contain the same word at a specific position, the N-best posterior probability is 1.0, no matter whether the word is correct or not. We believe that values of $N > 1000$ would not help to solve this problem since we already observed a deterioration of the performance of the N-best list posterior probabilities for $N = 1000$. As already discussed, we attribute this effect to the Levenshtein alignment.

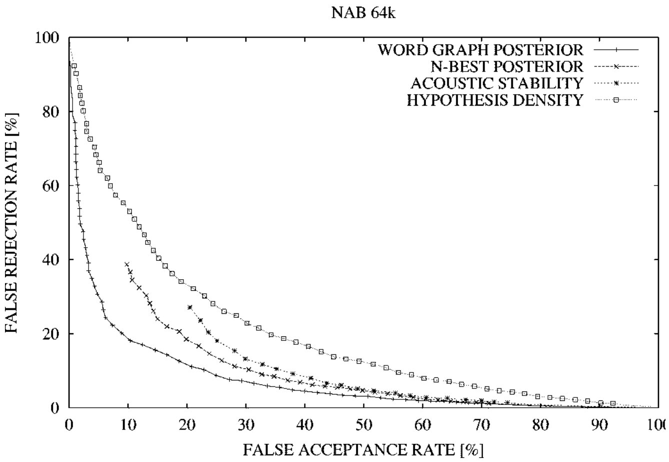

Fig. 8. DET curves for the NAB 64k evaluation corpus.

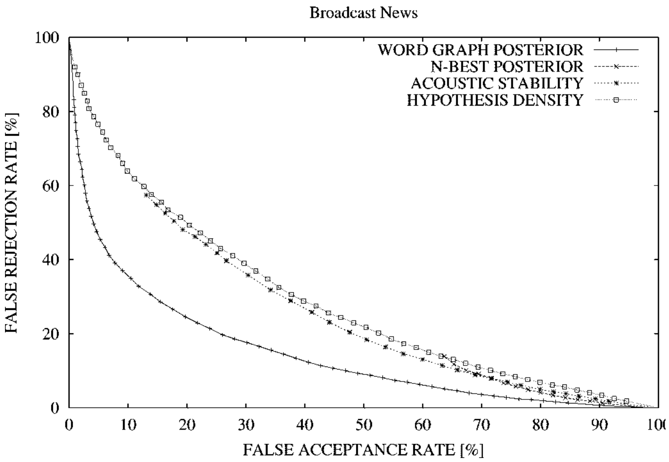

Fig. 9. DET curves for the Broadcast News 96 evaluation corpus.

VI. Conclusions

In this paper, several confidence measures based on word graphs and N-best lists are presented and compared. Experimental evidence clearly shows that posterior word probabilities outperform alternative confidence measures, i.e., the acoustic stability and the hypothesis density. Additional experiments prove that the estimation of posterior word probabilities on word graphs yields better results than their estimation on N-best lists. The relative reduction in confidence error rate ranges between 19% and 35% on different corpora using a trigram language model and the best posterior probability based confidence measure, defined in (11). The relative reduction was highest for corpora which are commonly regarded as difficult, consisting of spontaneous speech. For these corpora, the advantage of the confidence measures based on word graph posterior probabilities was also highest compared to the other confidence measures. It is interesting to note that this improvement is achieved with a single confidence measure and not with a vector of numerous features which can be extracted from a word graph. A combination of the different confidence measures presented in this paper using a linear discriminant analysis have not improved the results presented in this work.

References

[1] T. Bub and J. Schwinn, "VERBMOBIL: The evolution of a complex large speech-to-speech translation system," in *Proc. Int. Conf. Spoken Language Processing 1996*, Philadelphia, PA, Oct. 1996, pp. 2371–2374.

[2] L. Chase, "Word and acoustic confidence annotation for large vocabulary speech recognition," in *Proc. 5th Eur. Conf. Speech Communication Technology*, Rhodes, Greece, Sept. 1997, pp. 815–818.

[3] S. Cox and R. Rose, "Confidence measures for the switchboard database," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing 1996*, Atlanta, GA, May 1996, pp. 511–514.

[4] "Tech. Rep. Interactive Systems Labs.," ILKD, Apr. 1996.

[5] J. S. Garofolo, J. G. Fiscus, and W. M. Fisher, "Design and preparation of the 1996 HUB-4 broadcast news benchmark test corpora," in *Proc. DARPA Speech Recognition Workshop*, Chantilly, VA, Feb. 1997, pp. 15–21.

[6] L. Gillick, Y. Ito, and J. Young, "A probabilistic approach to confidence measure estimation and evaluation," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Munich, Germany, Apr. 1997, pp. 879–882.

[7] T. Kemp and T. Schaaf, "Estimating confidence using word latices," in *Proc. 5th Eur. Conf. Speech, Communication, Technology 1997*, Rhodes, Greece, Sept. 1997, pp. 827–830.

[8] V. I. Levensthein, "Binary codes capable of correcting deletions, insertions and reversals," *Sov. Phys.-Dokl.*, vol. 10, pp. 707–710, Feb. 1966.

[9] J. Mariani and L. Lamel, "An overview of EU programs related to conversational/interactive systems," in *Proc. 1998 Broadcast News Transcription Understanding Workshop*, Lansdowne, TX, Feb. 1998, pp. 247–253.

[10] C. Neti, S. Roukos, and E. Eide, "Word-based confidence measures as a guide for stack search in speech recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing 1997*, Munich, Germany, Apr. 1997, pp. 883–886.

[11] H. Ney, S. Ortmanns, and I. Lindam, "Extensions to the word graph method for large vocabulary continuous speech recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing 1997*, Munich, Germany, Apr. 1997, pp. 1787–1790.

[12] H. Ney, L. Welling, S. Ortmanns, K. Beulen, and F. Wessel, "The RWTH large vocabulary continuous speech recognition system," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Seattle, WA, May 1998, pp. 853–856.

[13] S. Ortmanns, H. Ney, and X. Aubert, "A word graph algorithm for large vocabulary continuous speech recognition," *Comput. Speech Lang.*, vol. 11, pp. 43–72, Jan. 1997.

[14] S. Ortmanns, H. Ney, F. Seide, and I. Lindam, "A comparison of time conditioned and word conditioned search techniques for large vocabulary speech recognition," in *Proc. IEEE Int. Conf. Spoken Language Processing 1996*, Philadelphia, PA, Oct. 1996, pp. 2091–2094.

[15] D. S. Pallet, J. G. Fiscus, W. M. Fisher, J. S. Garofolo, B. A. Lund, A. Martin, and M. A. Przybocki, "1994 Benchmark test for the ARPA spoken language program," in *Proc. ARPA Spoken Language Technol. Workshop*, Austin, TX, Jan. 1995, pp. 5–36.

[16] M. Pitz, F. Wessel, and H. Ney, "Improved MLLR speaker adaptation using confidence measures for conversational speech recognition," in *Proc. Int. Conf. Spoken Language Processing*, Beijing, China, Oct. 2000.

[17] B. Rueber, "Obtaining confidence measures from sentence probabilities," in *Proc. 5th Eur. Conf. Speech Communication Technology 1997*, Rhodes, Greece, Sept. 1997, pp. 739–742.

[18] T. Schaaf and T. Kemp, "Confidence measures for spontaneous speech recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing 1997*, Munich, Germany, Apr. 1997, pp. 875–878.

[19] M. Siu, H. Gish, and F. Richardson, "Improved estimation, evaluation and applications of confidence measures for speech recognition," in *Proc. 5th Eur. Conf. Speech Communication Technology 1997*, Rhodes, Greece, Sept. 1997, pp. 831–834.

[20] A. Sixtus and S. Ortmanns, "High quality word graphs using forward-backward pruning," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Phoenix, AZ, Mar. 1999, pp. 593–596.

[21] M. Weintraub, F. Beaufays, Z. Rivlin, Y. Konig, and A. Stolcke, "Neural-network based measures of confidence for word recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing 1997*, Munich, Germany, Apr. 1997, pp. 887–890.

[22] F. Wessel, K. Macherey, and R. Schlüter, "Using word probabilities as confidence measures," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing 1998*, Seattle, WA, May 1998, pp. 225–228.

[23] F. Wessel, K. Macherey, and H. Ney, "A comparison of word graph and N-best list based confidence measures," in *Proc. 6th Eur. Conf. Speech Communication Technology 1999*, Budapest, Hungary, Sept. 1999, pp. 315–318.

[24] F. Wessel, R. Schlüter, and H. Ney, "Using posterior probabilities for improved speech recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing 2000*, Istanbul, Turkey, June 2000, pp. 1587–1590.

[25] P. C. Woodland, C. J. Leggetter, J. J. Odell, V. Valtchev, and S. J. Young, "The 1994 HTK large vocabulary speech recognition system," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing 1995*, Detroit, MI, May 1995, pp. 573–576.

[26] S. Young, "Detecting misrecognitions and out-of-vocabulary words," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing 1994*, Adelaide, Australia, Apr. 1994, pp. 21–24.

Frank Wessel received the Diploma degree with honors in computer science from the University of Technology, Aachen, Germany (RWTH) in 1997.

In 1997, he joined the Department of Computer Science, RWTH. Since then, he has been with the Lehrstuhl für Informatik VI and a member of the Speech Recognition Group. His scientific interests cover all aspects of automatic speech recognition, language modeling, confidence measures, and discriminative training.

Ralf Schlüter studied physics at the University of Technology, Aachen, Germany (RWTH) and Edinburgh University, Edinburgh, U.K. He received the Diploma degree with honors in physics from RWTH in 1995.

From November 1995 to April 1996, he was with the Institute for Theoretical Physics B, RWTH, where he worked on statistical physics and stochastic simulation techniques. Since May 1996, he has been with the Lehrstuhl für Informatik VI, RWTH. From May 1996 to April 1999, he was with Siemens AG, Munich, Germany. His research interests cover automatic speech recognition, acoustic and stochastic modeling, discriminative training, confidence measures, and signal analysis.

Klaus Macherey received the Diploma degree in computer science from the University of Technology, Aachen, Germany (RWTH), in 1999.

Since then, he has been a Research Assistant with the Lehrstuhl für Informatik VI, RWTH. His primary research interests cover speech recognition and understanding, dialogue systems, and reenforcement learning.

Herman Ney (M'86) received the Diploma degree in physics from the University of Goettingen, Goettingen, Germany, in 1977 and the Dr.-Ing. degree in electrical engineering from the Technical University of Braunschweig, Braunschweig, Germany, in 1982.

In 1977, he joined Philips Research Laboratories where he worked on various aspects of speaker verification, isolated and connected word recognition, and large-vocabulary continuous-speech recognition. In 1985, he was appointed Head of the Speech and Pattern Recognition Group. In 1988–1989, he was a Visiting Scientist with AT&T Bell Laboratories, Murray Hill, NJ. In July 1993, he joined the University of Technology, Aachen, Germany, as a Professor of computer science. His current interests cover all aspects of pattern and speech recognition, such as signal processing, search strategies, language modeling, automatic learning, and language translation.