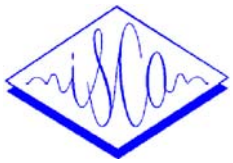
ISCA Archive
http://www.isca-speech.org/archive



INTERSPEECH 2004 - ICSLP
8th International Conference on Spoken Language Processing
ICC Jeju, Jeju Island, Korea
October 4-8, 2004

New Features based on Multiple Word Graphs for Utterance Verification

Alberto Sanchis, Alfons Juan and Enrique Vidal

Institut Tecnològic d'Informàtica
Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València, Valencia (Spain)
asanchis,ajuan,evidal@iti.upv.es

10.21437/Interspeech.2004-690

Abstract

The goal of Utterance Verification is to estimate a confidence measure which helps detecting words in the hypothesized sentence that are likely to have been missrecognized. Word graphs have been extensively employed for directly estimating the confidence measure and for extracting important predictor features. In all the cases, a single word graph which is obtained through the recognition process. In this paper we propose the use of multiple word graphs to compute new features. The experimental study shows that these proposed features outperform those computed on a single word graph and other well-known predictor features. Moreover, the combination of the proposed features along with other kind of features provides improvements in the verification accuracy.

1. Introduction

Current speech recognition systems are not error-free and, in consequence, it is desirable for many applications to predict the reliability of each word of the hypothesized sentence. The goal of Utterance Verification (UV) is to detect words that are likely to have been missrecognized. This implies the estimation of a confidence measure for each hypothesized word to classify it as either *correct* or *incorrect*.

The usefulness of word graphs in UV for different purposes is well known. In [7] the proposed features based on word graphs are the most important predictors. In [4] the confidence measure is estimated on word graphs directly by the posterior probability of a hypothesized word given all the acoustic observations of the utterance. The word posterior probability based on word graphs is used in [8] along with a large set of other predictive features to improve speech recognition accuracy. In all the cases, the authors use a single word graph which is obtained through the recognition process. An overview of the posterior probabilities based on a single word graph is given in section 2.

In section 3 we propose three new features which are based on word posterior probabilities estimated on multiple word graphs. For the experimental study, in section 4.1, we describe our smoothed naive Bayes model which allows to profitably combine different kinds of features under a sound statistical framework [2]. In section 4.2 we describe a set of well-known features for using them along with the proposed features. In section 4.3 we describe the experimental task. In section 4.4 we evaluate the performance of the proposed features comparing them with the alternative predictor features. We also compare the performance of using multiple or a single word graph in the computation of the features. Finally, we explore (naive Bayes) feature combinations in order to improve the classification accuracy.

2. Posterior probabilities on word graphs

A word graph G is a directed, acyclic, weighted graph. The nodes corresponds to discrete points in time. The edges are triplets $[w,\ s,\ e]$, where w is the hypothesized word from node s to node e. The weights are scores associated to the word graph edges. Any path from the initial to the final node forms a hypothesis h.

Given the acoustic observations $\mathbf{\Theta}_1^T$, the posterior probability for a specific word (edge) $[w,\ s,\ e]$ can be computed by summing up the posterior probabilities of all hypotheses of the word graph containing the edge $[w,\ s,\ e]$:

$$P([w,\ s,\ e] \mid \mathbf{\Theta}_1^T) = \frac{1}{P(\mathbf{\Theta}_1^T)} \sum_{\substack{h \in G\ : \\ \exists [w',s',e'] : \\ w'=w, s'=s, e'=e}} P(h,\ \mathbf{\Theta}_1^T) \qquad (1)$$

The probability of the sequence of acoustic observations $P(\mathbf{\Theta}_1^T)$ can be computed by summing up the posterior probabilities of all word graph hypotheses:

$$P(\mathbf{\Theta}_1^T) = \sum_h P(h\,, \mathbf{\Theta}_1^T) \qquad (2)$$

These posterior probabilities can be efficiently computed based on the well-known *forward-backward* algorithm [4].

3. Features based on multiple word graphs

The posterior probabilities can be based on different kinds of knowledge depending on the weights associated to the word graph edges. In this work, we study three features:

- WgAC: the weights are acoustic scores.
- WgLM: the weights are language model probabilities.
- WgTOT: the weights are the combination of acoustic and language model scores.

The algorithm proposed to compute each of these features is described in Figure 1. It is a general algorithm in the sense that it can be used to compute any feature based on multiple word graphs.

Let W be the vocabulary of the task and let G_t be a word graph which contains the most probable word-end partial hypotheses at time t of the recognition process. Let WPost be an algorithm which, given a word graph G_t, computes the word posterior probabilities following the method described in section 2. We assume that the WPost algorithm returns a set $\mathcal{A}$ composed by the edges of G_t with the posterior probability estimation: $\mathcal{A} = \{([w,\, s,\, e], p) : [w,\, s,\, e] \in G_t\}$.

The algorithm uses a matrix C (of size $|W| \times T$) for storing the sum of the posterior probabilities that any word $w \in W$ is obtained at any time $t \in \{1, \ldots, T\}$ of the recognition. This matrix is initialized to zero at the beginning of the algorithm. The computation of the feature is carried out in two stages:

Stage 1: It is performed during all the recognition process. For each recognition time t do:

1. Compute the word posterior probabilities on the word graph G_t.
2. For each edge $[w,\, s,\, e] \in G_t$, the word posterior probability p is accumulated to the posterior probabilities that the word w have been obtained at each time t' within the interval time $[s,\, e]$.

Stage 2: When the recognition process is finished, the feature is computed for each word of the most probable hypothesis h_{best}. First, the values stored in the matrix C should be adequately normalized. Given a word $w \in W$ and a time $t \in \{1, \ldots, T\}$, the value $C(w, t)$ is restricted to the interval $[0, T - t + 1]$. The maximum value is due to this property [4]: the sum of the word posterior probabilities for a specific point in time must sum to one. Therefore, if a word w is the only one that appears at time t for all word graph $G_{t'} : t \leq t' \leq T$, the maximum value must be necessarily $T-t+1$. The matrix values can be then properly normalized dividing each value $C(w, t)$ by the maximum value $T-t+1$. Based on these normalized values, given a word w and its starting and ending times $[s,\, e]$, two different variants of the feature are computed:

1. The average of the normalized values that the word w is obtained in the interval time $[s,\, e]$.
2. The maximum normalized value that the word w obtains in the interval time $[s,\, e]$.

The motivation of using multiple word graphs is based on two aspects. First, the usefulness of word graphs depends directly on the word graph density [9]. Thus, the use of multiple word graphs should provide a better representation of the most probable hypotheses. Another advantage is the capability of extracting the features at any time of the recognition process. This can be very useful for some applications such as continuous discourse to predict the reliability of partial hypotheses.

For each word $w \in W$ **Do**
  For $t = 1$ **To** T **Do**
    $C(w, t) = 0$
  EndFor
EndFor
For $t = 1$ **To** T **Do**
  $\mathcal{A}$ =WPost(G_t)
  For each word $w : ([w,\, s,\, e], p) \in \mathcal{A}$ **Do**
    For $t' = s$ **To** e **Do**
      $C(w, t') = C(w, t') + p$
    EndFor
  EndFor
EndFor
h_{best} = the most probable hypothesis
For each word $(w,\, s,\, e) \in h_{\text{best}}$ **Do**

$$\text{Wg}_{\text{avg}}(w,\, s,\, e) = \frac{1}{e-s+1} \sum_{t=s}^{e} \frac{C(w,t)}{T-t+1}$$

$$\text{Wg}_{\max}(w,\, s,\, e) = \max_{s \leq t \leq e} \frac{C(w,t)}{T-t+1}$$

EndFor
Return $\text{Wg}_{\text{avg}}(w)$ and $\text{Wg}_{\max}(w)$
for each word $w \in h_{\text{best}}$

Figure 1: Algorithm to compute one feature based on multiple word graphs.

4. Experimental Study

4.1. Naive Bayes model

We have recently proposed a *smoothed naive Bayes* classification model [2] to profitably combine different features.

We denote the class variable by c; $c = 0$ for correct and $c = 1$ for incorrect. Given a hypothesized word w and a D-dimensional vector of (discrete) features $\boldsymbol{x}$, the class posteriors can be calculated via the Bayes' rule as

$$P(c|\boldsymbol{x}, w) = \frac{P(c|w)\, P(\boldsymbol{x}|c, w)}{\sum_{c'} P(c'|w)\, P(\boldsymbol{x}|c', w)} \tag{3}$$

We make the naive Bayes assumption that the features are mutually independent given a class-word pair. Unknown probabilities are estimated by direct relative frequencies. For robustness, this *word-dependent* (specific) model is smoothed using a *word-independent* (generalized) naive Bayes model [2].

UV is performed by classifying a word as incorrect if $P(c = 1 \mid \boldsymbol{x}, w)$ is greater that a certain threshold τ.

4.2. Alternative predictor features

To compare the proposed features with alternative predictor features, a set of well-known features has been selected:

Acoustic stability (AS): Number of times that a hypothesized word appears at the same position (as computed by Levenshtein alignment) in K alternative outputs of the speech recognizer obtained using different values

of the *Grammar Scale Factor* (GSF), i.e. a weighting between acoustic and language model scores [6].

LMPr: Language model probability.

Hypothesis density (HD): The average number of the active hypotheses within the word boundaries [7].

PercPh: The percentage of word phones that match the phones obtained in a "phone-only" decoding.

Duration (Dur): The word duration in frames divided by its number of phones.

AcScore: The acoustic log-score of the word divided by its number of phones.

Word Trellis Stability (WTS): We have recently introduced this feature [1]. Given a word w and its starting and ending times $[s,\ e]$, two variants of the WTS are computed as:

$$WTS_{\text{avg}}(w) = \frac{1}{e-s+1}\sum_{t'=s}^{e}\frac{C(w,t')}{\sum_{w'}C(w',t')}$$

$$WTS_{\max}(w) = \max_{s\le t'\le e}\frac{C(w,t')}{\sum_{w'}C(w',t')}$$

$$C(w,t') = \sum_{t=t'}^{T}\sum_{h\in\mathcal{H}_t(w,t')}(\alpha_f-\alpha_i)$$

where T is the number of frames of the given utterance, $\mathcal{H}_t$ is a set of word-boundary partial hypotheses that are most probable at time t for a certain range of GSF values [α_i,α_f]. In addition, in each hypothesis of $\mathcal{H}_t(w,t')$ the word w must be active at time frame t'.

4.3. Experimental setup

We carried out experiments using the *FUB task*, an Italian speech corpus of phone calls to the front desk of a hotel, acquired in the context of the EUTRANS project [5]. The *FUB* corpus involves highly spontaneous speech data and contains many non-speech artifacts. Basic statistics of the (disjoint) training and test sets are summarized in table 1.

Table 1: FUB speech corpus

	training	test
speakers	276	24
running words	52,511	5,381
vocabulary size	2,459	–
bigram perplexity	–	31

The training set was used to train Italian context-dependent phone models. The acoustic models were left-to-right continuous density HMMs, trained using Linear Discriminant Analysis (LDA) and a Viterbi approximation [3]. Decision-tree clustered generalized triphones (CART with 1,500 tied states plus silence) were used as phone-units. A smoothed trigram language model was estimated using the transcriptions of the training utterances. The test-set Word Error Rate was 27.5 %.

4.4. Experimental Results

We have used different metrics for the evaluation of the classification accuracy. In evaluating verification systems, two measures are of interest: the *True Rejection Rate* (TRR, the number of incorrect words that are classified as incorrect divided by the number of incorrect words) and the *False Rejection Rate* (FRR, the number of correct words that are classified as incorrect divided by the number of correct words). The trade-off between TRR and FRR values depends on a decision threshold τ. A *Receiver Operating Characteristic* (ROC) curve represents TRR against FRR for different values of τ. The area under a ROC curve divided by the area of a worst-case diagonal ROC curve, provides an adequate overall estimation of the classification accuracy. We denote this area ratio as AROC. Note that an AROC value of 2.0 would indicate that all words can be correctly classified. Another criterion is the *Confidence Error Rate* (CER) defined as the number of classification errors divided by the total number of recognized words. A baseline CER is obtained assuming that all recognized words are classified as correct.

The training and the test data included $N = 51.609$ and $N = 5.131$ samples $\{(\boldsymbol{x_n}, c_n, w_n)\}_{n=1}^{N}$, respectively, where for each word w the class c and a vector of 20 features were obtained: 12 features correspond to the two variants of the three proposed features (WgAC,WgLM,WgTOT) computed on a single and multiple word graphs. The other 8 features are those described in section 4.2.

Table 2 shows the AROC, CER and the relative reduction in baseline CER using the (single-feature) smoothed model. As can be seen the WgLM feature gets the best AROC value and outperforms significantly all the other features. Only AS which has proved to be very useful [7, 6] achieves similar results. The WgTOT performance is slightly better than WgAC. Although these two features are not better than AS and WTS features, they outperform all the other features. Thus, the use of the language model probabilities in the word posterior estimations produce better results than the use of acoustic scores. The large range of the acoustic scores is the cause of this effect [4]. The variant used in the calculation of the features seems to be negligible in the performance.

The naive Bayes model was employed to explore the performance of feature combinations. All the possible combinations among the proposed features along with AS and WTS features were tested. The other features were added to the best combinations in order to achieve possible improvements. Table 3 shows the AROC, CER and the relative reduction in baseline CER for the best feature combinations. The combination of the two best single features: WgLM+AS produces a significant improvement over the single feature performance. The incremental addition of new features to this combination produces further improvements, finally achieving a relative reduction in baseline CER of 37.6%. Figure 2 shows the ROC curves for the best single feature and the best feature combination. It shows the gain by exploting the (naive Bayes) combination of features.

Table 2: AROC, CER and relative reduction in baseline CER for each individual feature.

Feature	AROC	CER(%)	rel. red. (%)
$WgLM_{avg}$	1.73	16.4	21.9
$WgLM_{max}$	1.72	16.4	21.9
AS	1.70	16.3	22.4
WTS_{max}	1.68	17.9	14.8
WTS_{avg}	1.66	18.2	13.3
$WgTOT_{max}$	1.65	18.0	14.3
$WgAC_{max}$	1.63	18.7	11.0
$WgTOT_{avg}$	1.61	18.4	12.4
$WgAC_{avg}$	1.61	19.0	9.5
LMPr	1.59	18.8	10.5
HD	1.59	18.9	10.0
PercPh	1.51	19.4	7.6
Dur	1.48	19.3	8.1
AcScore	1.48	19.6	6.7
Baseline	-	21.0	-

Table 3: AROC, CER and relative reduction in baseline CER baseline for the best feature combinations.

Features	AROC	CER(%)	red.(%)
AcScore+$WgTOT_{max}$+ WTS_{max}+Dur+AS+$WgLM_{avg}$	1.81	13.1	37.6
WTS_{max}+Dur+AS+$WgLM_{avg}$	1.81	13.6	35.2
Dur+AS+$WgLM_{max}$	1.79	14.4	31.4
AS+$WgLM_{max}$	1.78	14.5	31.0
$WgLM_{avg}$	1.73	16.4	21.9
Baseline	-	21.0	-

Table 4 shows the AROC and CER of using a single or multiple word graphs in the computation of the features. The use of multiple word graphs produce better AROC values for all the features. The improvements are most significant for the WgTOT and WgAC features. This fact suggests that the use of multiple word graphs helps reducing the negative impact of the the large range of acoustic scores. Nevertheless, although AROC values indicates that the use of multiple word graphs achieves better overall verification performance, the differences between CER values are not significant.

5. Conclusions

We have proposed three new features for utterance verification. These features are based on word posterior probabilities estimated on multiple word graphs. The results show that the proposed features outperform those computed on a single word graph and other well-known predictor features. The single feature performance is improved through the (naive Bayes) combination of the proposed features along with other kind of features.

6. Acknowledgements

This work was partially supported by the AVCiT Ayuda para Grupos I+D+I (GRUPOS03/031) and the EU project "TT2" (IST-2001-32091).

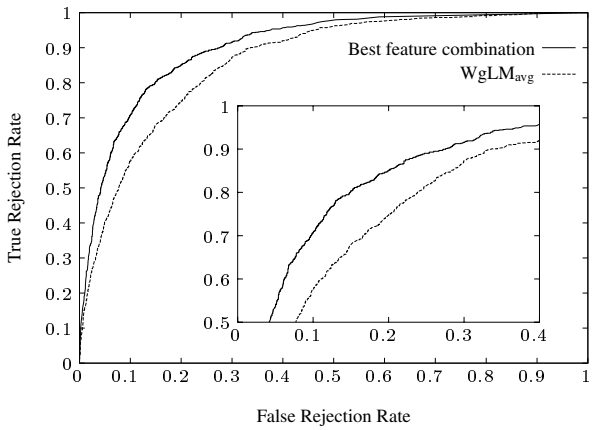


Figure 2: Comparative ROC curves for the best single feature versus the best feature combination.

Table 4: Comparative AROC and CER values for each feature computed using a single or multiple word graphs.

	Multiple WG		Single WG	
Feature	AROC	CER(%)	AROC	CER(%)
$WgLM_{avg}$	1.73	16.4	1.69	16.6
$WgLM_{max}$	1.72	16.4	1.70	16.5
$WgTOT_{max}$	1.65	18.0	1.51	17.8
$WgTOT_{avg}$	1.61	18.4	1.49	19.0
$WgAC_{max}$	1.63	18.7	1.56	19.1
$WgAC_{avg}$	1.61	19.0	1.59	18.8

7. References

[1] A. Sanchis, A. Juan, and E. Vidal. "Estimating confidence measures for speech recognition verification using a smoothed naive bayes model". *IbPRIA* 2003.

[2] A. Sanchis, A. Juan, and E. Vidal. "Improving utterance verification using a smoothed naive bayes model". Proc. of *ICASSP* 2003.

[3] H. Ney et al. "The RWTH large vocabulary continuous speech recognition system". Proc. of *ICASSP* 1998.

[4] F. Wessel et al. "Confidence measures for large vocabulary continuous speech recognition". *IEEE Trans. on Speech and Audio Processing*, 9(3):288–298, 2001.

[5] F. Casacuberta et al. "Some approaches to statistical and finite-state speech-to-speech Translation". *Computer Speech and Language*, vol. 18, pp. 25-47, 2004.

[6] T. Zepenfeld et al. "Recognition of conversational telephone speech using the JANUS speech engine". Proc. of *ICASSP*, 1997.

[7] T. Kemp and T. Schaaf. "Estimating confidence using word lattices". Proc. of *EUROSPEECH*, 1997.

[8] D. Vergyri. "Use of word level side information to improve speech recognition". Proc. of *ICASSP*, 2000.

[9] T. Fabian et al. "Impact of word graph density on the quality of posterior probability based confidence measures". Proc. of *EUROSPEECH*, 2003.