A HIDDEN-STATE MAXIMUM ENTROPY MODEL FOR WORD CONFIDENCE ESTIMATION

Peng Yu, Jie Xu[*]*, Guo-Liang Zhang, Yu-Chou Chang*[+]*, and Frank Seide*

Microsoft Research Asia, 5F Beijing Sigma Center, 49 Zhichun Rd., 100080 Beijing, P.R.C.
{rogeryu,leonzh,fseide}@microsoft.com
[*]Computer Science and Engineering College, Southeast University, 210096 Nanjing, P.R.C.
f-jiexu@msrchina.research.microsoft.com
[+]Electrical & Computer Engineering Department, Brigham Young University, Provo, Utah, USA, 84602
ycchang@et.byu.edu

ABSTRACT

We propose a probabilistic model for estimating word confidence by fusing predictor features. Starting from the Maximum Entropy (ME) method, we first prove that ME model is equivalent to the best model with certain form to the Minimum Expected Cross Entropy (MECE) criterion. Under the MECE criterion, We extend the form of ME model by introducing a hidden state. We call the new model Hidden-State Maximum Entropy (HSME) model. In a keyword-spotting task, we combine predictor features from both phonetic and word-level systems. Compared to lattice posterior alone, recall at 80% precision is improved from 38.1% to 49.5% on voicemail and from 37.1% to 51.9% on Switchboard. Compared with other fusion methods, HSME consistently outperforms decision tree, and most cases SVM.

Index Terms— confidence measure, keyword spotting

1. INTRODUCTION

Estimating word confidence is an important topic in speech recognition. Initially developed for utterance verification, confidence measures are found useful for many other applications, such as keyword spotting, spoken document indexing, and spoken language translation.

We define word confidence as the probability of a word being correctly recognized at a certain location in speech. Theoretically, it is equivalent to the *word posterior probability*. A common method is to compute word posteriors from speech recognition lattices [1]. This way, the confidence model is consistent with the model for speech recognition, thus having a strong theoretical basis.

There are two reasons why lattice-posterior based confidence measure may not be the best that we can do. First, the word posterior is theoretically optimum only under the ideal assumption that speech models are correct. Unfortunately, speech models used today have many approximations which are known to be incorrect. Most approximations have little impact on recognition accuracy, but some seriously degrade posteriors estimation. Second, lattice posteriors do not consider additional information that is difficult to be used in mainstream speech models like word duration and prosody, or external information like lip reading and OCR from video.

An alternative is to use predictor features that are informative to distinguish correctly recognized results from errors. These predictor features are combined in a certain way to generate a single score to indicate *correctness* of the recognition decisions, e.g., decision tree [2], Support Vector Machine (SVM) [3], boosting [4], neural network [5].

Most of those feature fusion methods target a verification task – hard correct/incorrect decision. However, soft word confidence is valuable because:

- in some applications, e.g., keyword spotting, user-required recall/precision level is not known in advance. The system must be prepared for all possible requirements (instead of being optimized for a single condition);
- in some applications, e.g., spoken document indexing, speech recognition results are used as input to succeeding modules. Probabilistic confidence is important for overall optimization.

In [6], a probabilistic model for feature fusion, named generalized linear model, was proposed for confidence. In [7], we proposed to build the probabilistic model under Maximum Entropy (ME) criterion, and resulted in the same form as generalized linear model. On a relative simple phonetic posterior normalization task, improvement was observed with the ME model. However, the linear combination form is too simplified to model word correctness off predictor features.

In this paper, we first re-motivate the use of the ME model by showing that it is the optimum solution under Minimum Expected Cross Entropy (MECE) criterion with a certain model form. We extend the form of ME model by introducing a hidden state behind each sample, and optimize under the MECE criterion. We call the new model Hidden-State Maximum Entropy model (HSME). In a keyword-spotting task, we combine predictor features from both a phonetic and a Large-Vocabulary Continuous Speech Recognition (LVCSR) system. Compared to LVCSR lattice posterior alone, recall at

80% precision is improved from 38.1% to 49.5% on voicemail and from 37.1% to 51.9% on Switchboard. Compared with other fusion methods, HSME consistently outperforms decision tree, and most cases Support Vector Machine (SVM).

The paper is organized as follows. In section 2 we will recapitulate the ME model for confidence measure [7], and will re-motivate it with MECE criterion. Section 3 introduces the new HSME model. Section 4 shows the results, and section 5 concludes the paper.

2. MAXIMUM ENTROPY MODEL FOR CONFIDENCE MEASURE

In our previous work [7], we found Maximum Entropy criterion is useful for posterior refinement in keyword spotting. In this section, we recapitulate the method.

Let $x = (O, t_s, t_e, w)$ denote a test sample, with observation O being an audio document, t_s, t_e two time points in O, and w being a word. Let C be the event "t_s and t_e is the boundary of one word in O, and the word is w". The *confidence measure* is defined as the conditional probability $p(C|x)$.

Now, assume a large number of training samples (x_i, C_i), $i = 1, \cdots, N$. An empirical probability distribution $\tilde{p}(x, C)$ can be estimated from the sample set.

The Maximum Entropy criterion is widely used for model estimation [8]. With our problem, the optimum $p(C|x)$ under Maximum Entropy criterion is found by:[1]

$$\hat{p} = \arg\max_{p}\{-\sum_{x,C} \tilde{p}(x,C)p(C|x)\log p(C|x)\}$$

$$\text{s.t.} \sum_{x} \tilde{p}(x)p(C=T|x)\vec{f}(x) = \sum_{x} \tilde{p}(x, C=T)\vec{f}(x). \quad (1)$$

Here $\vec{f}(x) = (f_1(x), \cdots, f_K(x))$ are so-called *predictor features* extracted from x.

It is known [8] that the optimum distribution p under ME can be found by the following equivalent optimization problem

$$\hat{\alpha} = \arg\max_{\alpha}\{\sum_{x,C} \tilde{p}(x,C)\log P_{\mathrm{ME}}(C_i|x_i,\alpha)\}, \quad (2)$$

where α is a parameter vector and

$$p(C|x) = P_{\mathrm{ME}}(C|x,\alpha) = \frac{\exp(\alpha \cdot \vec{f}(x))}{\exp(\alpha \cdot \vec{f}(x)) + 1}, \quad (3)$$

which we call Maximum Entropy (ME) model.

An alternative way of writing the objective function in Eq. 2 is,

$$\begin{aligned} Q(\alpha) &= \sum_{x,C} \tilde{p}(x,C)\log P_{\mathrm{ME}}(C_i|x_i,\alpha) \\ &= \sum_{x} \tilde{p}(x) \sum_{C} \tilde{p}(C|x)\log P_{\mathrm{ME}}(C_i|x_i,\alpha) \\ &= -\sum_{x} \tilde{p}(x) H(\tilde{p}(C|x), P_{\mathrm{ME}}(C_i|x_i,\alpha)). \end{aligned}$$

Here $H(p, q)$ stands for the *cross entropy* of two probability distribution p and q. This shows, the ME model is equivalent to the Minimum Expected Cross Entropy (MECE) model with the form in Eq. 3.

Eq. 3 also shows, the ME model predicts C by a linear combination of all features. Experiments show that this is too simplified to model the dependency between C and $\vec{f}$. To compensate for this, we will introduce an extended version of ME model in the next section.

3. HIDDEN-STATE MAXIMUM ENTROPY MODEL

We introduce the following assumptions:

- Each sample (x, C) has a hidden state attached, $s \in \{S_1, S_2, \cdots, S_T\}$. The state can be understood as a group of samples sharing some common properties;
- Samples in one state can be modeled by an individual exponential model as Eq. 3,

$$P(C|x, S_l) = P(C|x, \alpha_l) = \frac{\exp(\alpha_l \cdot \vec{f}(x))}{\exp(\alpha_l \cdot \vec{f}(x)) + 1};$$

- The hidden state s can be predicted by x with another exponential model. It has a similar but slightly different form with Eq. 3, as it is used for multiple classes[2]:

$$P(S_l|x, \beta_1^T) = \frac{\exp(\beta_l \cdot \vec{f}(x))}{\sum_{k=1}^{T} \exp(\beta_k \cdot \vec{f}(x))}.$$

Combining all of the above, we define the Hidden-State Maximum Entropy (HSME) model:

$$\begin{aligned} P_{\mathrm{HSME}}(C|x,\Delta) &= \sum_{l=1}^{T} P(C, S_l|x,\Delta) \\ &= \sum_{l=1}^{T} P(C|x,S_l,\Delta)P(S_l|x,\Delta) \\ &= \sum_{l=1}^{T} P(C|x,\alpha_l)P(S_l|x,\beta_1^T), \end{aligned}$$

where $\Delta = (\alpha_1^T, \beta_1^T) = (\alpha_1, \cdots, \alpha_T, \beta_1, \cdots, \beta_T)$.

Again, the MECE criterion is used to find best model:

$$\begin{aligned} \hat{\Delta} &= \arg\min_{\Delta} \sum_{x} \tilde{p}(x) H(\tilde{p}(C|x), P_{\mathrm{HSME}}(C_i|x_i,\Delta)) \\ &= \arg\max_{\Delta} \sum_{x,C} \tilde{p}(x,C)\log(P_{\mathrm{HSME}}(C_i|x_i,\Delta)). \quad (4) \end{aligned}$$

Eq. 4 can be solved by an Expectation-Maximization (EM) algorithm [9], where the objective function in each *maximiza-*

[1] Eq. 1, 2, and 3 has already been adapted to the specific two-classes ($C = T/F$) problem for confidence measure. In its raw form, ME can be used to estimate $p(y|x)$ with y being any random variable.

[2] Actually one of β_l can be fixed to 0 without changing the definition.

tion step is

$$\begin{aligned} & Q(\Delta|\Delta') \\ = & \sum_{x,C} \tilde{p}(x,C) E_{s|C,x,\Delta'}[\log P(C,s|x,\Delta)] \\ = & \sum_{x,C} \tilde{p}(x,C) \sum_{l=1}^{T} P(S_l|C,x,\Delta') \log P(C,S_l|x,\Delta) \\ = & \sum_{l=1}^{T} \sum_{x,C} \tilde{p}(x,C) P(S_l|C,x,\Delta') \log P(C|x,\alpha_l) \\ + & \sum_{x,C} \tilde{p}(x,C) \sum_{l=1}^{T} P(S_l|C,x,\Delta') \log P(S_l|x,\beta_1^T). \quad (5) \end{aligned}$$

And the *expectation* step calculates

$$\begin{aligned} P(S_l|C,x,\Delta') &= \frac{P(S_l,C|x,\Delta')}{\sum_{k=1}^{T} P(S_k,C|x,\Delta')} \\ &= \frac{P(C|S_l,x,\alpha_l')P(S_l|x,{\beta'}_1^T)}{\sum_{k=1}^{T} P(C|S_k,x,{\alpha'}_k)P(S_k|x,{\beta'}_1^T)}. \end{aligned}$$

Optimization for Eq. 5 can be split into the following sub problems, each solved by Quasi-Newton methods individually.

$$\hat{\alpha}_l = \arg\max_{\alpha} \sum_{x,C} \tilde{p}(x,C) P(S_l|C,x,\Delta') \log P(C|x,\alpha_l)$$

for $l = 1, \cdots, K$, and

$$\hat{\beta}_1^T = \arg\max_{\beta_1^T} \sum_{x,C} \tilde{p}(x,C) \sum_{l=1}^{T} P(S_l|C,x,\Delta') \log P(S_l|x,\beta_1^T).$$

4. RESULTS

4.1. Setup

We evaluate the above method on a keyword-spotting task. Predictor features are extracted from both a phonetic recognition and a LVCSR system. We report results by calculating recall at four precision levels (80%, 60%, 40%, and 20%). For each precision level, hits for all keywords are put together and cut off with a *shared* threshold.

We used two test sets, LDC Voicemail [10], and Switchboard [11]. Table 1 summaries the setup. The first block lists acoustic model training set, acoustic model adaptation, and language model training set for speech recognizer. The SWBD training set "SWBD+ICSI+BN" includes transcriptions of SWBD-I, LDC ICSI-meeting training set, and LDC Broadcast News 96 and 97 training sets. The phonetic recognizer uses a phonetic word-fragment language model as detailed in [12].

Table 1. Corpora summary.

test set	LDC Voicemail	Switchboard
AM train	SWBD-I (309h)	
AM adapt	VM-I	-
LM train	VM-I	SWBD+ICSI+BN
conf. model train	13.5h from LDC VM-II	
dev set	1.5h from LDC VM-II	
eval set	vmtest (1.5h)	RT03S (6.3h)
eval WER	44.4%	55.2%
eval #keywords	3223	2611

The second block (next 3 lines) lists distinct training/dev/eval sets for the confidence models. vmtest is a also subset of LDC Voicemail defined in [13].

A list of keywords are extracted from reference transcripts for each set by an automatic algorithm [12], example keywords are *pentium*, *federal-express package*, and *internet workstation address request*. The number of keywords for eval sets, together with word error rate for the LVCSR system are listed in the third block.

4.2. Fusion of Phonetic and LVCSR Features

Our previous work [12], [14], [7] has shown that for the keyword-spotting task, combining a phonetic and a LVCSR system by simply summing up the phonetic and LVCSR posteriors results in improvement over each single system. In the present paper, we use both posteriors as predictor features, and estimate word confidence with fusion methods.

[7] has also shown phonetic posteriors have a strong dependency on the keyword. Although the unnormalized posteriors is still useful for ranking hits for same keyword, they are not suitable for a task where a shared threshold is used across keywords. Also it was found that the phonetic language model scores are useful for normalizing the phonetic posteriors. Thus, we use phonetic language model score as a separate feature as well.

Table 2 compares keyword-spotting results with variant features and fusion methods. Recall at different cutoff threshold are given. Sometimes the recall is not available.

The first three columns list setup no., method and features. Features used here are phonetic posterior ($\mathrm{PP_{ph}}$), phonetic LM ($\mathrm{LM_{ph}}$), and LVCSR posterior ($\mathrm{PP_{wd}}$).

Line 1 is LVCSR baseline using word posteriors alone. Results with phonetic posteriors alone are not shown as unnormalized phonetic posteriors are not suitable for this task.

Line 2 and 3 combines $\mathrm{PP_{ph}}$ and $\mathrm{LM_{ph}}$ by the ME model and the HSME model respectively. By this, a somewhat workable system can already be built with pure phonetic features. The ME result was published in [7], and the new HSME model has a significantly better performance.

Line 4 to 8 uses different methods to combine all three features: $\mathrm{PP_{ph}}$, $\mathrm{LM_{ph}}$, $\mathrm{PP_{wd}}$. Line 4 is the heuristic we used in [7], which sums up normalized phonetic posteriors (line 3 here) and $\mathrm{PP_{wd}}$. It shows improvement over $\mathrm{PP_{wd}}$ at 20% precision, while being worse with higher precisions.

The ME model results (line 5) were poor. This reflects the fact that a linear combination of features is not enough

Table 2. Keyword-spotting results with different features and different fusion methods. (R*xx* for recall in % at *xx* precision. Important results boldfaced.)

		set:	LDC Voicemail				Switchboard ("RT03S")			
no.	method	features	R80	R60	R40	R20	R80	R60	R40	R20
1		$\mathrm{PP}_{\mathrm{wd}}$	**38.1**	49.7	56.7	62.0	**37.1**	48.3	57.1	66.2
2	ME [7]	$\mathrm{PP}_{\mathrm{ph}}$, $\mathrm{LM}_{\mathrm{ph}}$	n/a	8.2	12.6	27.2	1.8	4.9	10.2	21.3
3	HSME	$\mathrm{PP}_{\mathrm{ph}}$, $\mathrm{LM}_{\mathrm{ph}}$	18.3	36.0	46.6	63.3	21.4	31.5	41.9	57.5
4	posterior sum	$\mathrm{PP}_{\mathrm{ph}}$, $\mathrm{LM}_{\mathrm{ph}}$, $\mathrm{PP}_{\mathrm{wd}}$	17.5	38.0	52.1	73.6	22.1	33.3	44.9	64.9
5	ME	$\mathrm{PP}_{\mathrm{ph}}$, $\mathrm{LM}_{\mathrm{ph}}$, $\mathrm{PP}_{\mathrm{wd}}$	n/a	33.2	44.4	62.6	17.7	31.9	46.3	65.5
6	decision tree	$\mathrm{PP}_{\mathrm{ph}}$, $\mathrm{LM}_{\mathrm{ph}}$, $\mathrm{PP}_{\mathrm{wd}}$	46.7	61.4	70.0	81.1	46.4	58.5	66.2	76.9
7	SVM	$\mathrm{PP}_{\mathrm{ph}}$, $\mathrm{LM}_{\mathrm{ph}}$, $\mathrm{PP}_{\mathrm{wd}}$	47.7	64.7	74.9	85.4	51.8	64.1	71.6	80.8
8	HSME	$\mathrm{PP}_{\mathrm{ph}}$, $\mathrm{LM}_{\mathrm{ph}}$, $\mathrm{PP}_{\mathrm{wd}}$	**49.5**	66.1	75.0	84.7	**51.9**	63.0	70.8	79.8

for modeling the dependency between word correctness and predictor features.

The next three lines show results for decision tree, SVM and HSME respectively. For HSME, 32 states are used (The number of states has little effect beyond 8 states, as shown in Fig. 1). HSME is consistently better than the decision tree and in most cases, especially with high precisions which are more interesting, better than SVM. Note that, SVM requires substantial parameter tuning effort and takes significant longer time for testing.

To investigate the physical meaning of HSME states, we did a HSME training with another phonetic feature $\mathrm{NM}_{\mathrm{ph}}$: the number of phonemes in the keyword, comparing with the HSME model without $\mathrm{NM}_{\mathrm{ph}}$. Fig. 1 shows recall at 80% precision w.r.t. number of states used in HSME. Both setups converge to the same recall with 8 or more states, but with $\mathrm{NM}_{\mathrm{ph}}$ converges notably faster. This indicates that for no-$\mathrm{NM}_{\mathrm{ph}}$ setup, states actually learn the information of $\mathrm{NM}_{\mathrm{ph}}$, and finally catch up with the system that knows $\mathrm{NM}_{\mathrm{ph}}$ explicitly.

5. CONCLUSION

In this paper, we proposed a novel model-based fusion method for confidence measure. We first proved that the best model for confidence estimation under Maximum Entropy criterion is equivalent to the best model with given form under Minimum Expected Cross Entropy (MECE) criterion. Then under the MECE criterion, we extended the form of ME model to a new model which we call Hidden-State Maximum Entropy (HSME) model.

We evaluated the new method by a keyword-spotting task combining features from phonetic and LVCSR systems. Compared with LVCSR lattice posteriors alone, at 80% precision, recall was improved from 38.1% to 49.5% on LDC Voicemail set and from 37.1% to 51.9% on Switchboard ("RT03S"). Compared with other fusion methods, HSME performs consistently better than decision tree, and in most precision levels better than SVM.

6. ACKNOWLEDGEMENT

The authors wish to thank our colleagues Lie Lu and Kit Thambiratnam for helpful discussion and suggestion. We also thank Dr. Mark Gales of Cambridge University for sharing his `vmtest` set definition.

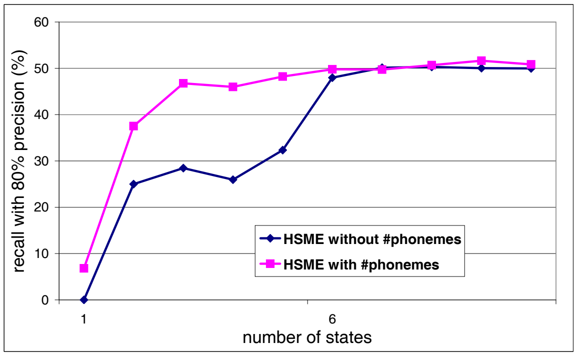

Fig. 1. Convergence of recall with number of states. Comparing HSME with and without $\mathrm{NM}_{\mathrm{ph}}$. Recall are calculated on voicemail test set at 80% precision.

7. REFERENCES

[1] F. Wessel, R. Schlüter, K. Macherey, H. Ney, Confidence Measures for Large Vocabulary Continuous Speech Recognition, IEEE Transactions on Speech and Audio Processing, Vol. 9, No. 3, 2001.

[2] C. V. Neti, S. Roukos, E. Eide, Word-Based Confidence Measures as a Guide for Stack Search in Speech Recognition, *Proc. ICASSP'97*, Munich, 1997.

[3] R. Zhang, A.I. Rudnicky, Word Level Confidence Annotation using Combinations of Features, *EuroSpeech'01*, Aalborg, 2001.

[4] P. J. Moreno, B. Logan, B. Raj, A Boosting Approach for Confidence Scoring, *EuroSpeech'01*, Aalborg, 2001.

[5] M. Weintraub, F. Beaufays, Z. Rivlin, Y. Konig, A. Stolcke, Neural-Network Based Measures of Confidence for Word Recognition, *Proc. ICASSP'97*, Munich, 1997.

[6] L. Gillick, Y. Ito, J. Young, A Probabilistic Approach to Confidence Estimation and Evaluation, *Proc. ICASSP'97*, Munich, 1997.

[7] P. Yu, D. Zhang, F. Seide, Maximum Entropy Based Normalization of Word Posteriors for Phonetic and LVCSR Lattice Search, *Proc. ICASSP'2006*, Toulouse, 2006.

[8] A. Bergers, S.D. Pietra, V. D. Pietra, A Maximum Entropy Approach to Natural Language Processing, Computational Linguistics, 1996.

[9] T. K. Moon, The Expectation-Maximization Algorithm, IEEE Signal Proceeding Magazine, Nov. 1996.

[10] M. Padmanabhan et al. Voicemail Corpus Part I (LDC98S77) and Part II (LDC2002S35). Linguistic Data Consortium, `http://www.ldc.upenn.edu`.

[11] NIST Spoken Language Technology Evaluations, `http://www.nist.gov/speech/tests/`.

[12] F. Seide, P. Yu, *et al*, Vocabulary-Independent Search in Spontaneous Speech. *Proc. ICASSP'04*, Montreal, 2004.

[13] M. Gales et al. Porting: Switchboard to the Voicemail task. *Proc. ICASSP'03*, Hongkong, 2003.

[14] Z. Y. Zhou, P. Yu, C. Chelba, F. Seide, Towards Spoken-Document Retrieval for the Internet: Lattice Indexing For Large-Scale Web-Search Architectures. *Proc. HLT'06*, New York, 2006.