MAXIMUM ENTROPY BASED NORMALIZATION OF WORD POSTERIORS FOR PHONETIC AND LVCSR LATTICE SEARCH

Peng Yu, Duo Zhang, Frank Seide*

Microsoft Research Asia, 5F Beijing Sigma Center, 49 Zhichun Rd., 100080 Beijing, P.R.C.
{rogeryu,fseide}@microsoft.com
*Computer Science and Technology Department, Tsinghua University, 100084 Beijing, P.R.C.
zhang-d@mails.tsinghua.edu.cn

ABSTRACT

In many keyword-spotting systems, the *word posterior probability* is an elementary quantity. In theory, the posterior of a keyword match denotes the probability of the match being correct. However, posteriors estimated on lattices, in particular phoneme lattices, are often off by orders of magnitude. This paper investigates the problem of providing "correct" posteriors, in the context of lattice-based word spotting. Unlike other work on word posteriors that focusses on *relative* ranking of posteriors, we emphasize relevance of the *absolute* value of the posterior in our user scenario. We stipulate that the posteriors should approach empirical precisions in a limit sense. Using this as a constraint, we estimate a mapping function based on Maximum Entropy. We find that for posteriors generated from phonetic lattices, mapped posteriors are satisfyingly consistent with empirical precision. In a joint search task, where *different* words are ranked together by posterior, FOM (Figure Of Merit) improved from 11.2% to 57.8%, which demonstrated the effectiveness of the method. Applied to searching LVCSR-based word lattices, the improvement is neglectable, but it is still effective when combining phonetic and word-lattice search in a hybrid mode, yielding an improvement from 46.7% to 65.8%.

1. INTRODUCTION

In [1], we have developed a system for keyword search in recorded conversations. Instead of searching speech-to-text transcripts, the system searches alternate recognition candidates as well, yielding a dramatic improvement of recall and overall search accuracy. The alternate candidates are represented in a graph structure [2] called lattice. To allow open-vocabulary searches, the system uses both phoneme lattices as well as word lattices generated with large-vocabulary continuous-speech recognition (LVCSR). Keywords are searched both by pronunciation and by word identity. For each hit in a lattice, a posterior probability is calculated from the recognizer's acoustic and language-model probabilities recorded in the lattice, using the common forward-backward technique and proper log-linear weighting of acoustic scores [3, 4]. Hits are then ranked by the posteriors. The system achieves satisfying results.

However, the dynamic ranges of the posteriors, in particular those from phonetic matches, depend highly on the keyword, and the absolute values convey very little information on whether a match is correct. E.g., long words often have lower posteriors than short words, though their precision is commonly higher.

Our goal is to provide the user a precision predictor. E.g., a value of 0.8 should indicate that among 10 hits of this kind there will be 8 correct ones. We aim at normalizing the posteriors, particularly phoneme-lattice based posteriors, so that they correctly reflect the expected precision level, and can be compared between different keywords.

[3, 4] discuss the use of word posterior as confidence measure. There, posteriors were generated from LVCSR-based lattices. Posterior normalization is much less of an issue compared to phoneme systems. [5] proposed a modified word posteriors for confidence measuring, where additional parameters are introduced into posterior calculation, while those parameters are optimized on a development set with confidence error rate as the objective function. This work was also done for LVCSR systems.

In this paper, we propose a normalization mapping technique for posteriors. We will refer to posteriors calculated from lattice by *raw posteriors*, and to the output of the mapping function as *mapped posteriors*. We stipulate that the mapped posteriors should approach empirical precisions in a limit sense. Using this as a basic constraint, the mapping function is estimated by a Maximum Entropy (MAXENT) criterion.

The paper is organized as follows. Section 2 defines the problem of posterior normalization, section 3 introduces proposed method, section 4 presents the experimental results, and section 5 concludes this paper.

2. PROBLEM OF POSTERIOR NORMALIZATION

Our previous work [1] has shown that in a word spotting task, ranking by posteriors is in theory optimal if (1) a search hit is considered relevant iff the query keyword was indeed said there, and (2) the user expects a ranked list of results such that the accumulative relevance of the top-n entries of the list, averaged over a range of n, is maximized.

The goal of this paper is to use posteriors, in addition to relative ranking of hits, also as an absolute predictor of correctness of a hit. In this sense, what is a user's expectation of posteriors? To answer this question, we introduce the concepts of "empirical precision" and "predicted precision."

2.1. Empirical and Predicted Precision

Assume h is a hit generated by the system. Let $h.A$ denote the corresponding audio fragment, $h.W$ the keyword, $h.P$ the raw posterior, and $h.C$ the event that h is correct, i.e., $h.C = T$ (T for true) means $h.W$ was indeed said in $h.A$.

Consider a user of the system a "tester" who will select a set of hits, $\mathcal{H} = h_1, \cdots, h_n$, assess the correctness of each hit, and measure the overall precision of all the hits,

$$\tilde{P}(\mathcal{H}) = \frac{1}{n} \cdot \sum_{i=1}^{n} I_C(h_i),$$

where $I_C(h) = \begin{cases} 1 & h.C = T \\ 0 & h.C = F \end{cases}$ is the indicator function of event $h.C$. We call this observed precision "empirical precision."

Before the user takes the test, he can also get an prediction of the precision by posteriors,

$$\hat{P}(\mathcal{H}) = \frac{1}{n} \cdot \sum_{i=1}^{n} h_i.P$$

which we call the "predicted precision."

The user's expectation will be that for large enough n, the predicted precision should approach the empirical precision.

$$\lim_{n\to\infty} \hat{P}(\mathcal{H}) = \lim_{n\to\infty} \tilde{P}(\mathcal{H}). \tag{1}$$

In our view, posteriors should satisfy this basic constraint.

2.2. Problem with Raw Lattice Posteriors

According to the law of large numbers [6], if each hit is independent, Eq. 1 always holds given the posteriors are calculated correctly.

However, speech models used in today's speech recognizers are still rather crude approximations of real speech, and calculated posteriors are not precise. Besides, speech recognizers are commonly designed for finding the single best path, rather than for providing accurate posteriors. Some of the model approximations that have only small impact on best path selections can have significant impact on posteriors. E.g., the widely used beam-pruning technique, which assigns zero probabilities to low-scoring hypotheses, affects the word error rate only slightly but leads to unnormalized posteriors.

In the case of computing posteriors from phoneme lattices, the key issue is the very weak language model used in phonetic recognition, which has little or no capability of modeling lexical constraints. Fig 1 shows an extremely simplified example for illustration. Assume a language has only two phonemes – A and B, and only two words – AAAAA and BBBBB – with the pronunciation same as the spelling. We further assume the two words are equally probable. In

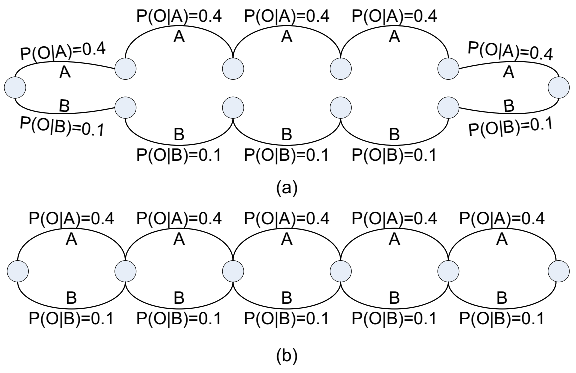


Fig. 1. Extremely simplified example to illustrate the problem with phonetic posterior calculation. Probabilities shown are acoustic likelihoods. Both words AAAAA and BBBBB shall be equally probable. (a) Using a language model with explicit knowledge of which phoneme sequences constitute valid words, $P(\text{AAAAA}|O) = 0.8$. (b) Using a phonetic unigram LM, $P(\text{AAAAA}|O) = 0.8^5 \approx 0.3$.

an isolated word utterance, a lattice generated by LVCSR – i.e. a model with explicit knowledge of which phoneme sequences constitute words – can be represented phonetically as in Fig 1a. With the acoustic likelihoods as shown in the figure, the posterior of word AAAAA would be $P(\text{AAAAA}|O) = 0.8$). Fig 1b shows a phoneme lattice as it might be generated by a phoneme recognizer using a unigram model. The posterior of word AAAAA in this case are $P(\text{AAAAA}|O) = 0.8^5 \approx 0.3$.

3. PROPOSED METHOD FOR POSTERIOR NORMALIZATION

Our goal is to find a mapping function $f(h)$ from a hit to its normalized posterior. Since this normalization is done after lattice generation, only the word identity and the raw posterior are available, i.e., $f(h) = f(h.W, h.P)$.

3.1. Maximum Entropy Criterion

Ideally, the mapped posterior satisfies the constraint defined in Eq. 1. With this constraint, the Maximum Entropy Criterion [7] can be used to find the optimal mapping function:

$$\begin{aligned} f^* &= \arg\max_f H(f) \\ &= \arg\max_f \{-\sum_{W,P} \tilde{p}(W,P) \cdot f(W,P) \cdot \log f(W,P)\} \end{aligned} \tag{2}$$

such that Eq. 1 holds for mapped posterior $f(W,P)$, with $\tilde{p}(W,P)$ being the empirical probability of $\{h.W = W, h.P = P\}$.

3.2. Weakening the Constraint

Since for practical realization only a limited training set is available, denoted as $R = r_1, \cdots, r_N$, we replace Eq. 1 by a weaker constraint:

$$\sum_{i=1}^{N} I_C(r_i) \cdot g^k(r_i) = \sum_{i=1}^{N} f(h_i.W, h_i.P) \cdot g^k(r_i) \qquad k = 1, \cdots, K \tag{3}$$

where $g^k(r_i)$ is some functions defined on r_i. In MAXENT terminology, this function is called a "feature"[1].

Eq. 2 and Eq. 3 completely define our optimization problem, which has a solution of the form [7]:

$$\begin{aligned} f_{\alpha,\beta} &= \frac{1}{Z_{\alpha,\beta}} \cdot \exp(\sum_{k=1}^{K} \alpha_k \cdot g^k), \\ Z_{\alpha,\beta} &= \exp(\sum_{k=1}^{K} \alpha_k \cdot g^k) + \exp(\sum_{j=1}^{K} \beta_k \cdot g^k). \end{aligned} \quad (4)$$

where the coefficients α, β are computed as

$$\begin{aligned} (\alpha,\beta)^* &= \arg\max_{\alpha,\beta}\{ \sum_{i:r_i.C=T} \log f_{\alpha,\beta}(r_i) \\ &+ \sum_{i:r_i.C=F} \log(1 - f_{\alpha,\beta}(r_i))\}. \end{aligned} \quad (5)$$

3.3. Selecting the Feature Functions

In our implementation, following feature functions are chosen.

- $g(W,P) = I_{\text{CLASS}^\ell}(W) = \begin{cases} 1 & W \in \text{CLASS}^\ell \\ 0 & W \notin \text{CLASS}^\ell \end{cases}$, where CLASS^ℓ are pre-defined word classes.
- $g(W,P) = P \cdot I_{\text{CLASS}^\ell}(W)$.
- $g(W,P) = P^{PH}(W) \cdot I_{\text{CLASS}^\ell}(W)$, where $P^{PH}(W)$ is the phoneme-unigram probability of the pronunciation of word W.

3.4. Training Set and Weighting of Training Samples

The training set R is generated by selecting a large number of query terms, performing search on a development set, and labelling the resulting hit list as being correct or incorrect based on a ground-truth annotation. For phonetic search, the vast majority of these training hits are low-scoring false positives. As a consequence, the resulting mapping function would be biased towards accurately modeling poorly scoring false positives, instead of good hits that we care about. To compensate for this, we introduce a weight term w_i for each hit in Eq. 5:

$$\begin{aligned} (\alpha,\beta)^* &= \arg\max_{\alpha,\beta}\{ \sum_{i:r_i.C=T} w_i \cdot \log f_{\alpha,\beta}(r_i) \\ &+ \sum_{i:r_i.C=F} w_i \cdot \log(1 - f_{\alpha,\beta}(r_i))\}. \end{aligned}$$

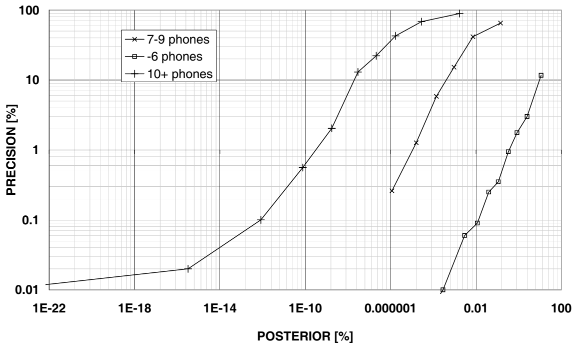
The weights are calculated by grouping all hits with the same word identity W, ranking them by raw posteriors, and assigning a weight of $w_i = \rho^i$ to the i^{th} hit (where ρ is a constant).

4. RESULTS

4.1. Setup

We have evaluated our system on two in-house sets of interview recordings, one recorded over the telephone, and one using a single microphone mounted on the interviewee's lapel. Each set is about one hour long. For each data set, a keyword list was selected by an automatic procedure [8]. Example keywords are *olympics* and *"automated accounting system"*.

[1]Actually the feature functions here are $I_C(r_i) \cdot g^k(r_i)$



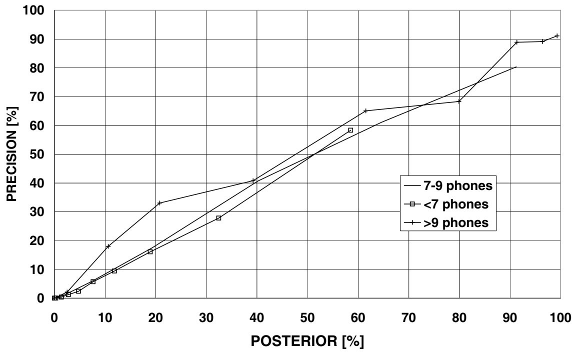


Fig. 2. Empirical precision with (a)raw and (b)mapped posteriors.

A two-hour meeting set (ICSI subset of the NIST RT04S dev set [9]) was used as development set to train the mapping functions.

The acoustic model we used was trained on 309h of the Switchboard corpus (SWBD-1). The phonetic language model was trained on the phonetic version of the transcriptions of SWBD-1 and Broadcast News 96 plus about 87000 background dictionary entries, a total of 11.8 million phoneme tokens. The LVCSR language model was trained on the transcriptions of the Switchboard training set, the ICSI-meeting training set, and the LDC Broadcast News 96 and 97 training sets. The recognition dictionary has 51388 words.

For mapping function training, five word classes depending on the number of phonemes in each word are used as constraint functions in section 3.3 for training phonetic posterior mapping function. The weighting coefficient ρ in section 3.4 is 0.9.

4.2. Empirical Precision

Fig. 2 shows empirical precision plotted over raw (Fig. 2a) and mapped (Fig. 2b) posteriors, for the phonetic-search system. Posteriors are binned, and for each bin, the precision of all hits falling into the bin is calculated. The horizontal axis is the posterior value, and the vertical axis is the corresponding precision. To have enough data for stating empirical precision for each posterior level, we group all words into three classes depending on the number of phonemes in each word.

In Fig. 2a, the mismatch between raw posterior and empirical precision is so extreme that we have to plot it on loga-

Table 1. Joint ranking using raw and mapped posteriors (phonetic lattices, FOMs in percent).

	Joint FOM			Reg.
test set	raw	+ map	+ wgt.	FOM
interviews (phone)	11.3	51.3	63.0	68.5
interviews (lapel)	11.0	44.7	52.5	68.3
average	11.2	48.0	57.8	68.4

Table 2. Joint ranking, comparing phonetic, word-based, and hybrid lattice search.

	Joint FOM		Reg.
setup	raw	+map+wgt.	FOM
phonetic lattice	11.2	57.8	68.4
LVCSR lattice	51.5	52.9	54.6
hybrid lattice	46.7	65.8	72.3

rithmic scale. The plot show the huge difference in dynamic range across word classes. On the other hand, the mapped posterior in Fig. 2b is satisfyingly consistent with the empirical precision, and the difference across word classes is properly compensated.

4.3. Joint Ranking

For a quantitative evaluation of search accuracy, we use the "Figure Of Merit" (FOM) metric defined by NIST as the average of detection/false-alarm curve taken over the range [0..10] false alarms per hour per keyword. FOMs of all keywords are commonly averaged to get an overall measurement.

This "Regular FOM" metric is invariant to our score normalization, as posteriors of different keywords are not directly compared. To evaluate effectiveness of normalization, we propose a *joint* ranking instead, where hits for all keywords are collected and ranked together, and a "Joint FOM" is calculated. Obviously, for this Joint FOM the relative ranking between different keywords is critical, and the Joint FOM shows how well posteriors from different word classes are normalized w.r.t. each other – the closer the Joint FOM approaches the Regular FOM, the more effective the normalization (technically the Regular FOM is not an upper bound for the Joint FOM, but the gap between the two is still a good yardstick).

An important motivation behind this evaluation are advanced search scenarios that allow logical combination of different keywords such as "sport AND car," and extensions to information-retrieval ranking such as TF.IDF. For this, posteriors need to be comparable across different keywords.

Table 1 shows the joint ranking result for raw and mapped posteriors. For raw posteriors, the very poor result of 11% is expected (first column). The second column shows that the mapping, if trained without using the training weight (section 3.4), significantly improves the Joint FOM, to 48.0%. Using the weight further improves the Joint FOM, and the final gap to the Regular FOM (right column) is about 11% points.

4.4. Normalization for LVCSR and Hybrid Search

So far our evaluation focused on phonetic search. In [1], we have shown that a hybrid search combining phonetic lattice and LVCSR search achieves the best performance. Table 2 compares the joint ranking result for phoneme, LVCSR, and hybrid systems.

The second row shows the effect of our method applied to word-based LVCSR lattices. The effect is small: a 1.4-point improvement is achieved (51.5 to 52.9%), and the gap to the Regular FOM is only 2% points. Posteriors derived from word lattices are already well normalized.

The third row shows the result for the hybrid system. We trained no specific mappings for this. Instead, mapped posteriors from phoneme and LVCSR system are summed up to get hybrid posteriors. The normalized hybrid system has the best Joint FOM of 66%, and the gap to the Regular FOM is smaller (7% points) than for the pure phonetic system.

5. CONCLUSION

We have investigated the problem of providing a "correct" posterior for the word-spotting task. Unlike other work on word posteriors, which focused on the *relative* ranking of posteriors, our target is the *absolute* value of posteriors as predictor of precision and applicable across different keywords. We stipulated that the posteriors should approach empirical precisions in a limit sense. Using this as a constraint, we estimated a mapping function based on a Maximum Entropy criterion. For posteriors generated from phonetic lattices, the mapped posteriors are satisfyingly consistent with empirical precision. In a joint ranking task ranking hits for different keywords together, Joint FOM was improved from 11.2% to 57.8%, demonstrating the effectiveness of the method. As opposed to phonetic search, searching in LVCSR-lattices did not benefit so much from posterior normalization (from 51.5% to 52.9%), but in a hybrid mode which combines the phonetic lattice and LVCSR-lattice search, FOM was improved from 46.7% to 65.8%.

6. ACKNOWLEDGEMENTS

The authors wish to thank our colleagues Asela Gunawardana, Patrick Nguyen, Yu Shi, and Ye Tian for sharing their Switchboard setup and models with us.

7. REFERENCES

[1] P. Yu, K.J. Chen, L. Lu, and F. Seide, Searching the Audio Notebook: Keyword Search in recorded Conversations. *Proc. HLT'05*, Vancouver, 2005

[2] M. Oerder and H. Ney. Word graphs: An efficient interface between continuous-speech recognition and language understanding. In *Proc. ICASSP93*, Vol. II, pp. 119–122, Minneapolis, 1993.

[3] 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.

[4] G. Evermann, P.C. Woodland, Posterior Probability Decoding, Confidence Estimation and System Combination, Proc. Speech Transcription Workshop, 2000

[5] F. Song, W.K. Lo, S. Nakamura, Generalized word posterior probability (GWPP) for measuring reliability of recognized words, Proc. of SWIM2004

[6] A.N. Shiryayev, Probability Theory

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

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

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