CALIBRATION OF WORD POSTERIOR ESTIMATION IN CONFUSION NETWORKS FOR KEYWORD SEARCH

Zhiqiang Lv*, Meng Cai*, Wei-Qiang Zhang*, Jia Liu*
* Tsinghua University, Beijing, China
E-mail: {lv-zq12,cai-m10}@mails.tsinghua.edu.cn {wqzhang,liuj}@tsinghua.edu.cn Tel/Fax: +86-010-62781847

***Abstract*—Word posterior probability has been widely used as the confidence estimation of automatic speech recognition (ASR) systems and has been proved to be quite effective in related applications such as keyword search. However, word posterior probability tends to overestimate the true probability of a hypothesis, as it is computed on a subset of the total hypothesis space. In this paper, we show that a more accurate estimation of posterior can be obtained by using a calibration method based on the conditional random field (CRF) model. By using calibrated posterior estimation for keyword search task, we obtain a maximum absolute gain of 1.15% for single-word keyword search on the maximum term-weighted value (MTWV) metric on the OpenKWS14 Tamil dataset.**

I. INTRODUCTION

A typical spoken term detection (STD) system is based on the transcriptions generated by the ASR system. Most ASR systems provide a confidence estimate for every word hypothesis. Word posterior probability is usually used as the confidence estimation, which is proved to be quite effective. However, word posterior probability is often overestimated due to the pruning of the recognizer when decoding. In the decoding process, hypotheses which get a relatively low likelihood are removed from the final transcriptions. When computing the posterior probability of a word hypothesis, we usually treat the remaining hypothesis space as the total hypothesis space and that leads to the overestimated probability. The goal of this work is to calibrate the word posterior estimation in confusion networks to improve the performance of STD systems. By using a CRF model to estimate the probability of whether a confusion network bin contains a correct hypothesis and normalizing the posterior probabilities of all the word hypotheses in the confusion network bin accordingly, the entire posterior estimation is calibrated.

In this paper, Section II mainly introduces the confusion networks used in our work. Prior work is discussed in Section III. In Section IV, we describe the keyword search task and the metrics. In Section V, we propose our posterior estimation calibration approach. In Section VI, the results are presented. Section VII is the summarization of our work.

This work is supported by National Natural Science Foundation of China under Grant No. 61273268, No. 61370034 and No. 61403224.

II. CONFUSION NETWORKS

Confusion networks [1, 2] are compact representations of lattices output by ASR systems. They are designed to minimize word error rates instead of sentence error rates and keep the property that all word hypotheses are totally ordered. Confusion networks are created by clustering edges which overlap in time in lattices, after which competing hypotheses are clustered into the same confusion network bin.

In lattices, each word hypothesis has a posterior probability which is the sum over the probabilities of all the paths which contain that word hypothesis. The formula is as below [3]:

$$p(W|X) = \frac{\sum_q p(q, W)}{p(X)} \tag{1}$$

where $p(W|X)$ is the probability of a word hypothesis given the observed signal X. The probability of the path q which contains the hypothesis W is denoted as $p(q, W)$. $p(X)$ is the prior probability of X and it is usually approximated by the sum over the probabilities of all the paths through the lattice. The approximation leads to the overestimation of posterior probabilities, because it treats the pruned hypothesis space as the total hypothesis space.

Confusion networks cluster competing hypotheses from different paths in lattices into one confusion network bin. In the clustering process, hypotheses which share the same word identification in a confusion network bin are merged into one. Their probabilities are added, while the other hypotheses remain unchanged. Therefore, probabilities in confusion networks are almost the original posterior probabilities in lattices and are overestimated. That means every confusion network bin is believed to contain a correct hypothesis. However, this is not always true for a considerable proportion of confusion network bins. And Fig. 1 shows an example of that: the fifth bin of the confusion network has no correct word hypothesis "DONE", but the probabilities of all the wrong hypotheses sum to 1, which is not expected.

In fact, the error (i.e., no correct hypothesis) rate of confusion network bins in our STD system built for Tamil is about 54%. Therefore, calibration of word posterior estimation is essential in confusion networks to obtain a more accurate confidence estimation.

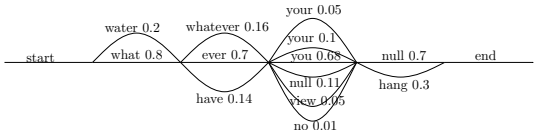

Fig. 1. Confusion network of the sentence "what have you done"

III. PRIOR WORK

There has been similar work for calibrating word posterior estimation in confusion networks. Hillard et al. [4] trained two Support Vector Machine classifiers to compensate for posterior estimation in confusion networks, and improved the 1-best confidence estimate significantly. However, whether a confusion network bin contains a correct hypothesis is heavily affected by its surrounding bins. And a wrong bin in which there is no correct hypothesis will probably lead to an error of the following bin considering the decoding process. So the problem of calibrating the posterior estimation in confusion networks is more similar to the one of sequence labeling instead of a classification problem as is described in [4]. Tur et al. [5] employed the CRF model for semantic parsing based on confusion networks for the task of spoken language understanding (SLU), in which the mission is to label every confusion network bin with a semantic related label. And this method outperforms other classification methods for semantic parsing. Similarly, our work uses the CRF model to label confusion network bins with "True" or "False". Accordingly, posterior probabilities of word hypotheses in a confusion network bin are normalized and a better confidence estimation is obtained for keyword search task. The calibrated confusion networks lower the posterior probabilities of word hypotheses in wrong bins, which will directly lower the probabilities of false alarms in keyword search results and will surely improve the performance of STD systems.

IV. DATA AND METRICS

The data for our experiments is conversational speech provided by the Intelligence Advanced Research Projects Activity (IARPA) Babel project from a low-resource language: Tamil. All data for this project is disseminated by the National Institute of Standards and Technology (NIST) on behalf of the IARPA. And the data is divided into training, development (dev), and evaluation (eval) partitions. The training partition with 60 hours of transcribed conversational speech and 20 hours untranscribed speech is used to train the ASR system. The dev partition consists of 10 hours of transcribed speech, which is used to train the CRF model. The evaluation data used in our work is a subset of the eval partition, which is released by NIST for local tests and is designated Eval Part One (evalPart1). The evalPart1 partition consists of 15 hours of transcribed speech and is used to test if our approach works.

The speech recognizer used in our experiments is the HDecode in the HTK [6] and a convolutional maxout neural networks (CMNN) [7] acoustic model is trained for building a start-of-the-art ASR system. The result of keyword search task is a list of all hits found for every keyword in the keyword list. Every hit in the list is labeled with the audio file in which the keyword is found, the start and end time of the hit in the audio and a confidence score for the hit. The keyword search result is evaluated using term-weighted value (TWV) [8]:

$$TWV(\theta) = 1 - [P_{miss}(\theta) + \beta P_{FA}(\theta)] \tag{2}$$

The parameter θ is a confidence measure of a hit in the hit list. TWV is a function of θ, and actual term-weighted value (ATWV) is the TWV value at a specific θ. MTWV is the maximum of TWV over all possible values of θ.

V. CALIBRATING POSTERIOR ESTIMATION

A. CRF

Linear-chained CRF [9] is a discriminative model framework which is widely used for segmenting and labeling sequence data.

Calibrating posterior estimation in confusion networks is in fact a problem of labeling the sequence bins with "True" or "False". Given a confusion network of N bins, the problem is like this:

$$\hat{Y} = \arg\max_{Y} P(Y|X) \tag{3}$$

where $X = x_1, x_2..., x_N$ is the input confusion network bin sequence, and $Y = y_1, y_2, ..., y_N$ is the output label sequence. $P(Y|X)$ is defined as:

$$P(Y|X) = \frac{1}{Z(X)} \exp(\sum_{k} \lambda_k f_k(y_{t-1}, y_t, x_t)) \tag{4}$$

where $f_k(y_{t-1}, y_t, x_t)$ represents a feature function, and λ_k is its associated weight learned on training dataset. $Z(X)$ is the normalization term [9]. After the linear-chain CRF model has been trained, the output label sequence $\hat{Y}$ is generated using the Viterbi algorithm.

In our work, we use four labels for labeling confusion network bins: BEGIN stands for the start of an utterance, END stands for the end; TRUE indicates there is a correct hypothesis in current bin, while FALSE indicates there is none.

B. Feature extraction and feature selection

We use a series of features extracted from the confusion networks, including score features of original posterior probabilities, structure features of confusion network bins and position specific features extracted from the whole utterance. Details are listed below:

1. Non-null probabilities: the sum probabilities over all the word hypotheses except null hypotheses in current confusion network bin.

2. Top hypotheses features: this set of features are related to the word hypothesis W_{top} with the highest posterior probability in the confusion network bin. The probability of W_{top} and its associated R_0 and R_1 [10] are calculated. R_0 and R_1 are defined as below:

$$R_0(K) = \frac{\sum_i c_f(d_i^K)}{T} \tag{5}$$

$$R_1(K) = \frac{\sum_i (1 - c_f(d_i^K))}{T} \quad (6)$$

where $c_f(d_i^K)$ is the probability of the i-detection of word K in the utterance and T is the duration of the utterance. Also, top probabilities in the adjacent bins which come before and after current bin are included.

3. Entropy features: the entropies of current bin and its adjacent bins.

4. Statistics features: the means, variances of probabilities of current bin and its adjacent bins.

5. Null hypothesis features: Boolean features indicating whether the current bin and its adjacent bins have the null hypothesis as the most probable hypothesis [4].

6. Position specific features: the length-normalized position of current bin, the log length and duration of the utterance [4].

7. Syntactic features: the unigram probability of the top word hypothesis from the recognizer language model in current bin, the bigram probability which sums over all the bigram probabilities of word hypotheses in current bin with its adjacent bins, the maximum bigram probability of word hypotheses in current bin with its adjacent bins.

In fact, two sets of confusion networks have been used for feature extraction in our approach, of which one removes the language model likelihoods compared to the other one. This is because that the language model sometimes leads to worse estimation of a hypothesis, especially on low-resource conditions like on the Tamil dataset. The set of confusion networks without language model likelihoods is denoted by CNs-AM, the other set is denoted by CNs. Features mentioned above are extracted in both CNs-AM and CNs.

For comparison, we also train an SVM model just like what has been done in [4]. As for the CRF model, there is no need to select features because the CRF model is capable of dealing with highly correlated and complex features. However, as for the SVM model, feature selection is necessary.

In this work ,we use the quadratic programming feature selection (QPFS) [11] for feature selection. The QPFS optimizes the relevance of the selected features to the class labels and minimizes redundancy among the selected feature set. Using the QPFS, we find out that the 10 best-ranked features are: the unigram probability of the top word hypothesis in current bin from CNs, the mean of probabilities in current bin from CNs-AM, the mean of probabilities in current bin from CNs, the Boolean feature of null hypothesis in current bin from CNs-AM, the non-null probability of current bin from CNs-AM, the top hypothesis probability in the following bin from CNs, the Boolean feature of null hypothesis in the following bin from CNs-AM, the variance of probabilities in current bin from CNs-AM, the top hypothesis probability in current bin from CNs and the non-null probability of current bin from CNs.

C. Experiments

We use the confusion networks of the dev partition to train our CRF model and the SVM model. And the training data consists of 10566 utterances segmented by our ASR system. Features described above are extracted from those utterances.

After models have been trained, we apply them to calibrate the confusion networks of the eavl partition according to the probabilities of the label TRUE. That is, in the original confusion networks, every bin has a probability of 1 for being labeled as TRUE; while in calibrated confusion networks, the probability is generated from either the CRF model or the SVM model. The best SVM model is trained with a Gaussian kernel, which is the same with the approach described in [4].

To evaluate the performance of our approach, the MTWV metric is calculated on 3 different keyword lists. The first keyword list is the one for OpenKWS14 evaluation. There are 5576 keywords in this list, of which 1272 are single-word and 4304 are multi-word. We denote this keyword list by LIST1. The second keyword list is one consists of 1931 single-word keywords and is denoted by LIST2. LIST2 is mainly used for development because of its simplicity. The third keyword list consists of 5213 single-word keywords and is denoted by LIST3. LIST3 is used for verifying the effctiveness of our method on single-word keyword search.

VI. RESULTS

As is described in section 5.3, a CRF model and an SVM model are trained on the dev partition. And the classification performance of the two models is evaluated, which has been shown in Table I. Obviously, the CRF model outperforms the SVM model on classification performance, just as expected. Although the F1 is not very high, we can still use it for calibrating scores in confusion networks. Then the confusion networks are calibrated and keyword search task is conducted. The results of MTWV on the dev partition itself using LIST2 are in Table II. We can see that the calibration shows significant improvement over the baseline for the two models and the CRF model achieves more gain than the SVM model, which shows correlation with the better classification performance. For further exploration, calibrated confusion networks are used for keyword search using three different keyword lists on the evalPart1 partition. The results are in Table III.

As we can see in Table III, the CRF model still outperforms the SVM model on different keyword lists. Consistent improvement has been observed over the baseline using the

TABLE I
CLASSIFICATION PERFORMANCE OF BOTH MODELS

dev/eval	Model	recall	precision	F1
dev	SVM	64.6%	68.54%	66.5%
	CRF	65.0%	70.29%	67.54%
evalPart1	SVM	60.0%	39.51%	47.65%
	CRF	63.5%	39.71%	48.86%

TABLE II
MTWV RESULTS ON THE DEV PARTITION USING LIST2

Model	Baseline	Calibrate	Absolute Gain
SVM	21.58%	21.95%	0.37%
CRF		22.22%	0.64%

CRF model, while the results of the SVM model on LIST1 and LIST2 do not show improvement. However, the results on LIST3 of both models show significant improvement over the baseline. The results indicate that the gain may be mainly obtained from single-word keywords, and further analysis of the results of both models using LIST1 is shown in Table IV.

Results in Table IV show that the improvement is indeed obtained from single-word keywords. Calibration of the SVM model harms the performance of multi-word keywords a lot, while the CRF model does little.

From the results, we can see that calibrating confusion networks achieves much better MTWV than the baseline when dealing with single-word keyword search. And our approach outperforms the method using the SVM model. It improves the performance of single-word keyword search without harming that of multi-word keyword search. The reason why our approach does not help improve the performance of multi-word keyword search is that our approach aims to calibrate single confusion network bins, which are associated with single-word keywords directly. We do not pay that much attention to the correctness of confusion network bin sequences, which are associated with multi-word keywords. The calibration of single bins is averaged when processing multi-word keyword search. So the calibration does not contribute to the gain of multi-word keyword search.

VII. CONCLUSIONS

We propose a calibration method for confusion networks to obtain a more accurate posterior estimation for keyword search task. The method employs the CRF model for calibrating confusion networks and achieves consistent improvement for single-word keyword search. However, the classification performance of this method still needs further improvement to obtain better results. Also, how to calibrate the probabilities of confusion network bin sequences, which are associated with multi-word keyword search, is to be explored.

TABLE III
MTWV RESULTS ON THE EVALPART1 PARTITION

Keyword List	Model	Baseline	Calibrate	Absolute Gain
LIST1	SVM	35.20%	34.96%	-0.24%
	CRF		35.40%	0.20%
LIST2	SVM	24.56%	24.59%	0.03%
	CRF		25.40%	0.84%
LIST3	SVM	19.32%	20.01%	0.69%
	CRF		20.47%	1.15%

TABLE IV
MTWV RESULTS OF SINGLE-WORD KEYWORDS AND MULTI-WORD KEYWORDS SEPARATELY ON EVALPART1 USING LIST1

Single/Multi	Model	Baseline	Calibrate	Absolute Gain
Single(1272)	SVM	28.32%	28.41%	0.09%
	CRF		29.15%	0.83%
Multi(4304)	SVM	37.50%	37.18%	-0.32%
	CRF		37.49%	-0.01%

REFERENCES

[1] L. Mangu, E. Brill, and A. Stolcke,"Finding consensus in speech recognition: word error minimization and other applications of confusion networks," *Computer Speech & Language* vol. 14, no. 4, pp. 373400, 2000.

[2] L. Mangu, B. Kingsbury, H. Soltau, H.K. Kuo and M. Picheny, "Efficient spoken term detection using confusion networks," in *Peoc. ICASSP* 2014.

[3] G. Evermann and P. Woodland, "Large vocabulary decoding and confidence estimation using word posterior probabilities,' in *Proc. ICASSP,* 2000, pp. 16551658.

[4] D. Hillard and M. Ostendorf, "Compensating for word posterior estimation bias in confusion networks," in *Proc. ICASSP,* 2006, pp. 11531156. J. Clerk Maxwell, *A Treatise on Electricity and Magnetism,* 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[5] G. Tür, A. Deoras, and D. Hakkani-Tür, "Semantic parsing using word confusion networks with conditional random fields," in *Proc. INTERSPEECH,* 2013, pp. 25792583.

[6] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, et al., *The HTK book,* vol. 2, Entropic Cambridge Research Laboratory Cambridge, 1997.

[7] M. Cai, Y. Shi, J. Kang, J. Liu, and T. Su, "Convolutional maxout neural networks for low-resource speech recogition," in *Proc. ISCSLP,* 2014, pp. 133137.

[8] "DRAFT KWS14 KEYWORD SEARCH EVALUATION PLAN," http://www.nist.gov/itl/iad/mig/upload/ KWS14-evalplan-v11.pdf, 2014.

[9] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. ICML,* 2001.

[10] F. Wessel, K. Macherey, and R. Schluter, "Using word probabilities as confidence measures," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing 1998, Seattle, WA, May, 1998,* pp. 225228.

[11] I. Rodriguez-Lujan, R. Huerta, C. Elkan, and C. S. Cruz, "Quadratic programming feature selection," *Journal of Machine Learning Research,* vol. 11, pp. 14911516, August,2010.