AN LSTM-CTC BASED VERIFICATION SYSTEM FOR PROXY-WORD BASED OOV KEYWORD SEARCH

Zhiqiang Lv, Jian Kang, Wei-Qiang Zhang, Jia Liu

Tsinghua National Laboratory for Information Science and Technology
Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
{lv-zq12,kangj13}@mails.tsinghua.edu.cn, {wqzhang,liuj}@tsinghua.edu.cn

ABSTRACT

Proxy-word based out of vocabulary (OOV) keyword search has been proven to be quite effective in keyword search. In proxy-word based OOV keyword search, each OOV keyword is assigned several proxies and detections of the proxies are regarded as detections of the OOV keywords. However, the confidence scores of these detections are still those of the proxies from lattices. To obtain a better confidence measure, we employ an LSTM-CTC verification method in this work and the confidence scores are regenerated. OOV keyword search results on the evalpart1 dataset of the OpenKWS16 Evaluation have shown consistent improvement and the maximum relative improvement can reach 21.06% for the MWTW metric.

Index Terms— verification, proxy keyword, OOV keyword, keyword search, CTC

1. INTRODUCTION

With the explosive increment of speech spreading on the Internet and through the telephone, keyword search has been more and more important nowadays. A typical keyword search system is based on large vocabulary continuous speech recognition (LVCSR) systems. First of all, LVCSR systems generate lattices for speech segments. Then indexes are created from lattices. We can search keywords directly in indexes. For LVCSR-based keyword search systems, lattices are generated according to a fixed lexicon and none OOV words will be recognized. Therefore, for LVCSR-based keyword search systems, OOV keywords could never be retrieved.

To solve the problem of OOV keyword search, two strategies can be adopted:

- Search OOV keywords via sub-word units such as phones, syllables or word-fragments. Sub-word lattices can be generated by using sub-word lexicons together with sub-word language models [1, 2, 3]. We can also convert word-level lattices into sub-word lattices directly [4]. After we have sub-word lattices, OOV keywords can be searched via sub-word sequences.
- Inspired by query expansion in text retrieval [5, 6, 7, 8], proxy words that are acoustically similar to OOV keywords are searched in lattices instead of OOV keywords. In [9], a phone confusion matrix has been employed for generating a list of likely-confusable proxy keywords from the fixed lexicon. Proxy keywords are searched in the entire lattices through a weighted finite state transducer (WFST) based framework and state-of-the-art performance has been achieved.

Comparing the two methods, proxy-word based keyword search could often lead to better performance because sub-word keyword search trends to result in more false alarms. For proxy-word based keyword search, stronger language constraint can be introduced in the decoding phase by a word-level lexicon and a word-level language model.

However, proxy-word based OOV keyword search doesn't generate a new confidence measure for detections of the proxies. The confidence scores of proxy-word detections are in fact the posterior probabilities of the proxies rather than the OOV keywords themselves. The language model probability for OOV words can not be estimated and that makes the posterior probabilities not accurate. Some efforts have been made to get more approximate probabilities for OOV keywords by removing the language model information. In [9], experiments removing language model scores from lattices have shown "negligible" improvement. In fact, both the language model scores and the acoustic model scores of the proxies for OOV words are not accurate. Therefore, to obtain a perfect confidence measure for OOV keywords, all we need to do is to estimate the acoustic posterior probability for each proxy-word detection.

In [10], we have proposed a novel LSTM-CTC keyword verification system for in vocabulary (IV) keyword search. Although the scores output by the verification system is not a good confidence measure as the posterior probability from lattices, it still can supply some complementary and useful acoustic information. In this work, we employed the verification system to assign every proxy-word detection a new acoustic score. Results on the evalpart1 dataset of the OpenKWS16 Evaluation have shown promising results.

In this paper, we briefly introduce the keyword search task and metrics in Section 2. Section 3 is about the LSTM-CTC based keyword verification method. Some basic environment setup is introduced in Section 4. Verification results for proxy-word based OOV keyword search and the discussions are presented in Section 5. Section 6 is about the conclusions.

2. TASK AND METRICS

The task of keyword search defined by NIST for the OpenKWS16 Evaluation is to find all the exact matches of given queries in a corpus of unsegmented speech data. A query, which can also be called "keyword", can be a sequence of one or more words. The result of this task is a list of all the detections of keywords found by keyword search systems. Each detection item in the list consists of the keyword id, the utterance in which the detection is found, the start and end time of the detection, and the confidence score.

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

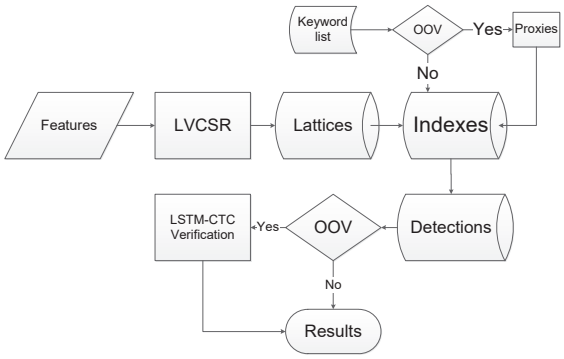

Fig. 1. *The proxy-based OOV keyword verification system pipeline.*

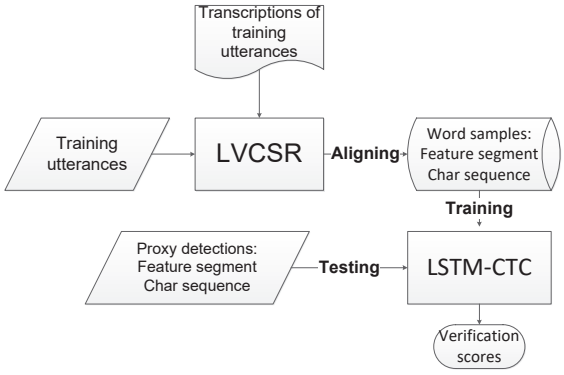

Fig. 2. *The LSTM-CTC verification: training and testing phases.*

To evaluate the performance, term-weighted value (TWV) [11] is adopted:

$$TWV(\theta) = 1 - \frac{1}{K}\sum_{w=1}^{K}(\frac{\#miss(w,\theta)}{\#ref(w)} + \beta\frac{\#fa(w,\theta)}{T - \#ref(w)}) \quad (1)$$

where θ is the decision threshold, K is the number of keywords. $\#miss(w,\theta)$ is the number of true tokens of keyword w that are missed at threshold θ, $\#fa(w,\theta)$ is the number of false detections of keyword w at threshold θ, $\#ref(w)$ is the number of reference tokens of w, T is the total amount of the evaluated speech, β is a constant set at 999.9.

As we can see, TWV is a function of the decision threshold θ. A global threshold θ is used to make the hard decision whether a detected keyword is correct. The TWV at the specified global threshold is the actual TWV (ATWV). The optimal threshold results in the maximum term-weighted value (MTWV).

3. LSTM-CTC BASED OOV KEYWORD VERIFICATION

After having the OOV detection list, we need to generate a new confidence measure for proxy detections. In this work, we adopted the LSTM-CTC verification system proposed in [10]. The verification system is in fact a universal acoustic model. The verification procedure is to align feature segments to the corresponding character sequences and the posterior probability is taken as the verification score. The verification system pipeline is illustrated in Fig. 1.

Connectionist temporal classification (CTC) [12] has been proven to be very suitable for labeling such unsegmented sequence data. Given a input feature sequence X of length T and its character sequence W, CTC is an objective function defined below:

$$L_{CTC}(X,W) = \sum_{C_W} p(C|X) = \sum_{C_W}\prod_{t=1}^{T} p(c_t|X), \quad (2)$$

where C_W is any label sequence of length T corresponding to the correct character sequence W. By summing up over all sets of label locations that yield the same label sequence W, CTC determines a probability distribution over possible labelings, conditioned on the input sequence X.

The long short-term memory (LSTM) neural network [13] has been demonstrated to be very effective to deal with sequence labeling problems. For our verification system, we put a CTC layer on top of an LSTM neural network trained directly using the original speech feature.

In the training phase, word samples are fed into a multi-layer LSTM neural network using the CTC criterion. To prepare word samples from utterances, the LVCSR system is employed to align utterance-level transcriptions to word-level transcriptions. In the testing phase, proxy detections for OOV keywords are fed into the LSTM neural network and the verification scores are output. The LSTM-CTC verification system is illustrated in Fig. 2.

In fact, the CTC verification scores are also acoustic scores and are similar to acoustic likelihoods in LVCSR systems. The difference between the two scores lies in :

- The CTC score is trained on character level and no pronunciation lexicon is needed. The acoustic likelihood is often trained on phoneme level or more precisely on state level. Because of the nature of the CTC criterion, no alignment for every frame is needed in the training phase for CTC scores.
- The acoustic likelihood is only the likelihood of the most possible label sequence, while the CTC score is the posterior probability summation over all possible label sequences. That makes the CTC score more approximate to the true acoustic posterior probability.
- The aim of our verification system is to verify the correctness of the detection segment, not the whole utterance. That makes it more focused on the internal properties in individual words and less impacted by the context.

In one word, the verification system focuses on verifying whether a feature segment is corresponding to the specified label sequence. The verification system is quite easy to build and very flexible.

4. EXPERIMENTS SETUP

4.1. The LVCSR system setup

All the OOV keyword search experiments were conducted on the evalpart1 dataset of the OpenKWS16 Georgia Evaluation. The

acoustic model was a subspace GMM (SGMM) [14] based model trained using the full language pack (FullLP). The FullLP consists of 80 hours of conversational speech, of which 50% is transcribed. Only the 40 hours' transcribed speech from the FullLP was used. Besides, we used multilingual bottleneck (MBN) features [15, 16] for building the LVCSR system. The MBN features were trained using 8 Babel languages including 102 the Assamese, 201 the Creole, 202 the Swahili, 206 the Zulu, 305 the Guarani, 306 the Igbo, 307 the Amharic and 403 the Dholou.

A trigram language model was trained using transcriptions of the FullLP together with some selected web text data. To obtain better performance, the acoustic model was enhanced using the boosted maximum mutual information (BMMI) criterion [14]. There is also a tuning dataset consisting of 3 hours of transcribed speech for the OpenKWS16 Evaluation. We used the tuning dataset for parameter tuning.

4.2. OOV keyword search setup

We used the babel404b-v1.0a_convevalpart1.annot.kwlist3.xml, the keyword list provided by NIST for the OpenKWS16 Evaluation. There are 4469 keywords in the keyword list, of which 554 are OOV keywords in our LVCSR system. To handle the OOV keywords, we employed the proxy-word based OOV keyword search in [9]. First of all, the phone confusion matrix was trained using the tuning dataset. Then for each OOV keyword, 500 proxy words in the vocabulary were generated. For shorter keywords often resulting in a lot of false alarms, we ignored the OOV keywords with less than 5 phones. This set of parameters of proxies is denoted as "Proxy_baseline". The experiment removing the language model scores in lattices is denoted as "Proxy_no_LM" .

Besides, to obtain state-of-the-art keyword search performance, all the TWV results reported in this work have been normalized using the keyword specific threshold (KST) normalization method [17].

4.3. The LSTM-CTC verification system setup

To train the LSTM-CTC verification system, we aligned the utterance-level transcriptions of the FullLP to word-level transcriptions using our LVCSR system first. Then we split every utterance into individual words and 252535 samples can be obtained. We adopted a character-level LSTM-CTC verification system using the 252535 samples. In total, there are 33 different characters for Georgia. The verification system was trained to align the MBN features of every word sample to their corresponding character sequences using the CTC objective function. The LSTM neural network was a bi-directional neural network, which consisted of 4 hidden layers and each layer had 320 forward memory cells and 320 backward memory cells. The tuning dataset was also split into individual words and 24891 samples were obtained for validation.

In [10], we have also proposed three normalization methods for normalizing the verification score. We denote the original verification score output by the verification system as s, then the three normalization methods can be written as below:

$$s_{word} = s^{(1/\#words)}, \tag{3}$$

$$s_{char} = s^{(1/\#characters)}, \tag{4}$$

$$s_{frame} = s^{(1/\#frames)}. \tag{5}$$

$\#words$ is the number of words in the keyword. $\#characters$ is the number of characters in the keyword. $\#frames$ is the number of frames where the detection lasts. For longer keywords trending to get relatively low scores in the verification system, the three normalization methods try to eliminate the effect of different lengths of keywords.

In our experiments, we denote the original verification experiment as "Ver", the word-level normalization as "Ver_word", the character-level normalization as "Ver_char" and the frame-level normalization as "Ver_frame".

In addition, we have also trained a contrastive LSTM-CTC system using utterances as training samples. The same normalization methods have also been adopted. We denote the verification system trained using utterance samples as "Ver_utt", the word-level normalization version as "Ver_utt_word", the character-level normalization version as "Ver_utt_char", the frame-level normalization version as "Ver_utt_frame".

5. EXPERIMENT RESULTS AND DISCUSSIONS

5.1. Results for the verification system

The proxy-word based OOV keyword search was conducted on the OpenKWS16 Evaluation evalpart1 dataset. Then the verification system was employed to verify every detection of the proxy keywords. Results of our verification system is listed in Table 1.

Table 1. *Verification results for proxy-word based OOV keyword search*

	ATWV	MTWV	Imp.(MTWV)
$Proxy\_baseline$	0.3005	0.3048	-
$Proxy\_no\_LM$	0.3033	0.3113	2.13%
Ver	0.3146	0.3254	6.76%
Ver_word	0.3146	0.3254	6.76%
Ver_char	**0.3368**	**0.3448**	**13.12%**
Ver_frame	0.3100	0.3282	7.68%
Ver_utt	0.2350	0.2386	-21.72%
Ver_utt_word	0.2346	0.2350	-22.90%
Ver_utt_char	0.2691	0.2828	-7.22%
Ver_utt_frame	0.2703	0.2843	-6.73%

From the results, we can see that proxy-word based keyword search has shown excellent performance. Removing language model scores from lattices brings a little improvement for both the ATWV and MTWV metric. It proves that wrong language model information has negative influence on calculating the confidence measure for proxies of OOV keywords.

Even by employing the original verification scores, the relative improvement of MTWV can reach 6.76%, much bigger than that of removing language model scores (2.13%). Normalization according to the keyword length has brought consistent improvement over the plain CTC score. Normalizing the verification score using the character length shows the biggest improvement of 13.12%.

In fact, removing the language model scores from lattices is not a very reasonable method for calibrating the confidence measure of OOV keyword detections. Because for IV words, the language model information is still very necessary for calculating the posterior probability in lattices. Therefore, assigning every proxy detection a new acoustic confidence measure is a better choice.

The reason why our verification system works is that the verification system generates discriminative scores of aligning feature segments to the corresponding character sequences directly. From

the results, we can also conclude that normalizing is essential for the verification procedure. Among the three normalization methods, the character-level normalization worked best because our verification system was trained on character sequences. The character-level normalization tries to evaluate the average correctness for each character component in keywords.

The verification system trained using utterance samples instead of individual word samples has led to much performance decrease. The reason is that training using utterances has to dealing with much longer char sequences, which consists of transitions across words and makes the training procedure more difficult.

5.2. Verification results with different numbers of proxies

Some more experiments have been done to explore the potential of proxy-based OOV keyword search with different numbers of proxies for each OOV keyword. The result are presented in Table 2.

Table 2. *OOV keyword search performance with different numbers of proxies*

Number of proxies	ATWV	MTWV
100	0.2871	0.2922
200	0.2795	0.2813
300	0.2877	0.2904
400	0.3070	0.3087
500	0.3005	0.3048
1000	0.2855	0.2950
1500	0.2861	0.2944
2000	0.2886	0.2981
2500	0.3020	0.3120
3000	0.3073	0.3147
4000	**0.3113**	**0.3188**

From the results in Table 2, we can see that better performance of OOV keyword search can be obtained by increasing the number of proxies. However, generating more than 400 proxies doesn't bring much improvement over that of 400 proxies. The reason is that false alarms increase rather quickly as the number of proxy words increases. For proxy-based OOV keyword search with different numbers of proxies, verification experiments have also been conducted and the results are in Fig. 3.

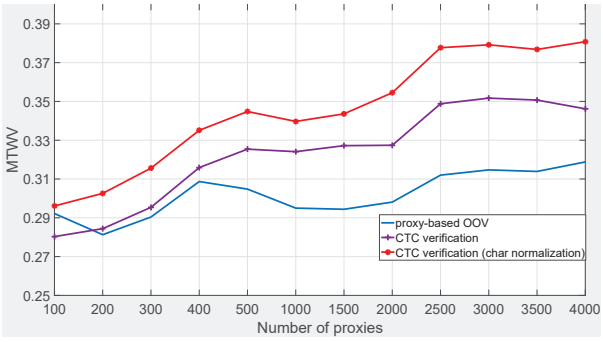

Fig. 3. Verification results with different numbers of proxies.

The "char normalization" in Fig. 3 refers to the character-level normalization. The results indicate that our verification system can improve the performance of proxy-based OOV keyword search for different numbers of proxies consistently, especially when using the character-level normalized CTC verification scores. The biggest relative improvement of MTWV can reach **21.06%** where the number of proxy words is 2500.

Besides, much more absolute MTWV improvement has been observed as the number of proxies increases. For example, the absolute MTWV gain is 0.062 where the number of proxies is 4000, while it is only 0.0039 where the number of proxies is 100. That means our verification system has provided much better confidence measure and the confidence measure helps a lot to distinguish true hits from false alarm.

5.3. Verification results for proxies with different cutoff lengths

In the previous proxy-based keyword search, proxies for OOV keywords with less than 5 phones were ignored to avoid a number of false alarms. Experiment results with "proxy cutoff" at different lengths of phones in keywords are presented in Table 3.

Table 3. *Verification results of MTWV with different cutoff lengths.*

Length	Baseline	Verification
4	0.3184	**0.3808**
5	0.3188	**0.3808**
6	**0.3204**	0.3766
7	0.2940	0.3241

"Verification" in Table 3 refers to the LSTM-CTC verification results with character-level normalization, while "Baseline" refers to the plain proxy-based keyword search. The minimum number of phones in keywords is 4 (only one keyword has the length of 3 phones). So the cutoff length is set varying from 4 to 7. With bigger cutoff length, a little improvement is obtained from 4 to 6 for "Baseline". Using the verification system, we can see that a smaller cutoff can get better MTWV. This phenomenon indicates that our verification system generates a better confidence measure, even for short OOV keyword proxies.

Considering that the minimum number of phones is 4, the verification performance for OOV keywords with less phones (e.g. 2 or 3) should be checked.

6. CONCLUSIONS

In this article, we have proposed an LSTM-CTC based verification system for proxy-based OOV keyword search. The verification method trained a character-level LSTM-CTC neural network and tried to align feature segments to their corresponding character sequences. Using the verification scores normalized by the character length, we can obtain quite a lot of improvement consistently over the plain proxy-based OOV keyword search. For proxy conditions resulting in much more false alarms, the verification system can achieve even bigger gains and shows strong robustness.

7. REFERENCES

[1] M. Saraclar and R. Sproat, "Lattice-based search for spoken utterance retrieval," in *Proc. HLTNAACL*, 2004.

[2] O. Siohan and M. Bacchiani, "Fast vocabulary-independent audio search using path-based graph indexing," in *Proc. Interspeech*, 2005.

[3] J. Mamou, B. Ramabhadran, and O. Siohan, "Vocabulary independent spoken term detection," in *Proc.ACM SIGIR*, 2007, pp. 615–622.

[4] U. V. Chaudhari and M. Picheny, "Improvements in phone based audio search via constrained match with high order confusion estimates," in *Proc. ASRU*, 2007, pp. 665–670.

[5] Y.-C. Li, W.-K. Lo, H. Meng, and P. Ching, "Query expansion using phonetic confusions for chinese spoken document retrieval," in *Proc. IRAL*, 2000.

[6] B. Logan and J.-M. V. Thong, "Confusion-based query expansion for oov words in spoken document retrieval," in *Proc. Interspeech*, 2002.

[7] B. Logan, J.-M. V. Thong, and P. J. Moreno, "Approaches to reduce the effects of oov queries on indexed spoken audio," vol. 7, no. 5, pp. 899–906, 2005.

[8] P. Karanasou, L. Burget, D. Vergyri, M. Akbacak, and A. Mandal, "Discriminatively trained phoneme confusion model for keyword spotting," in *Proc. Interspeech*, 2012.

[9] G. Chen, O. Yilmaz, J. Trmal, D. Povey, and S. Khudanpur, "Using proxies for oov keywords in the keyword search task," in *Proc. ASRU*, 2013.

[10] Z. Lv, M. Cai, W.-Q. Zhang, and J. Liu, "A novel discriminative score calibration method for keyword search," in *Proc. Interspeech*, 2016.

[11] "Draft kws16 keyword search evaluation plan," http://nist.gov/itl/iad/mig/upload/KWS16-evalplan-v04.pdf, 2016.

[12] A. Graves, S. Fernandez, F. Gomez, and S. J., "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proc. ACM*, 2006.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[14] D. Povey, L. Burget, M. Agarwal, P. Akyazi, F. Kai, A. Ghoshal, O. Glembek, N. Goel, K. M., A. Rastrow, R. C. Rose, S. P., and T. S., "The subspace Gaussian mixture modela structured model for speech recognition," *Computer Speech & Language*, vol. 25, pp. 404–439, 2011.

[15] K. M. Knill, M. J. F. Gales, S. P. Rath, P. C. Woodland, C. Zhang, and S.-X. Zhang, "Investigation of multilingual deep neural networks for spoken term detection," in *Proc. ASRU*, 2013, pp. 138–143.

[16] J. T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Crosslanguage knowledge transfer using multilingual deep neural network with shared hidden layers," in *Proc. ICASSP*, 2013, pp. 7304–7308.

[17] D. R. H. Miller, M. Kleber, C. L. Kao, O. Kimball, T. Colthurst, S. A. Lowe, R. M. Schwartz, and H. Gish, "Rapid and accurate spoken term detection," in *Proc. Interspeech*, 2007.