LEARNING BETTER LEXICAL PROPERTIES FOR RECURRENT OOV WORDS

Long Qin[*]

M*Modal Inc.
1710 Murray Avenue, Pittsburgh, PA 15217

Alexander Rudnicky

Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213

ABSTRACT

Out-of-vocabulary (OOV) words can appear more than once in a conversation or over a period of time. Such multiple instances of the same OOV word provide valuable information for learning the lexical properties of the word. Therefore, we investigated how to estimate better pronunciation, spelling and part-of-speech (POS) label for recurrent OOV words. We first identified recurrent OOV words from the output of a hybrid decoder by applying a bottom-up clustering approach. Then, multiple instances of the same OOV word were used simultaneously to learn properties of the OOV word. The experimental results showed that the bottom-up clustering approach is very effective at detecting the recurrence of OOV words. Furthermore, by using evidence from multiple instances of the same word, the pronunciation accuracy, recovery rate and POS label accuracy of recurrent OOV words can be substantially improved.

Index Terms— OOV word detection, recurrent OOV words, distributed evidence, OOV word learning

1. INTRODUCTION

Most speech recognition systems are closed-vocabulary recognizers and do not accommodate out-of-vocabulary (OOV) words. But in many applications, e.g., *voice search* or *spoken dialog systems*, OOV words are usually content words such as names and locations which contain information crucial to the success of these tasks. Speech recognition systems in which OOV words can be detected are therefore of great interest.

Hybrid speech recognition systems use a hybrid lexicon and hybrid language model (LM) during decoding to explicitly represent OOV words with smaller sub-lexical units [1-9]. In previous work, we have built hybrid systems using different types of sub-lexical units [10]. We also improved the hybrid system performance by applying system combination techniques [11, 12]. But in current OOV word detection systems, each OOV word is recognized and treated individually. We do not know whether two detected OOV words correspond to the same word or not.

In [13], we described how to find recurrent OOV words in a hybrid speech recognition system through bottom-up clustering. Specifically, we began with collecting the phonetic, acoustic and contextual features for OOV candidates in the hybrid system output. During clustering, each OOV candidate was initially treated as one cluster, then pairs of clusters were iteratively merged until the distance between two clusters exceeded a threshold. At the end, OOV candidates in the same cluster were considered as multiple instances of the same OOV word. In this paper, we extended our previous work to show that such multiple occurrences of an OOV word were very valuable in the OOV word learning task, where we could estimate a better pronunciation, spelling and part-of-speech (POS) label for the word. The proposed OOV word clustering and learning techniques were tested on data with different speaking styles and recording conditions including the Wall Street Journal (WSJ), Switchboard (SWB), and Broadcast News (BN) datasets.

The remainder of this paper is organized as follows. Section 2 describes the bottom-up clustering approach. Section 3 presents the details of estimating the pronunciation, spelling and POS label for recurrent OOV words. Sections 4 and 5 discuss experiments and results. Concluding remarks are provided in Section 6.

2. FINDING RECURRENT OOV WORDS

2.1. OOV word detection using the hybrid system

In our hybrid system, we applied a hybrid lexicon and hybrid LM during decoding to detect the presence of OOV words. The hybrid lexicon was obtained by integrating sub-lexical units and their pronunciations into the word lexicon. The hybrid LM was trained in a flat manner. First, the pronunciation of OOV words was estimated through the grapheme-to-phoneme (G2P) conversion [14], and then used to train the sub-lexical units. After that, OOV words in the training text were replaced by corresponding sub-lexical units to get a new hybrid text corpus. Finally, a hybrid LM was trained from this hybrid text data. When training the hybrid LM, sometimes two or more OOV words might appear consecutively in the training data. After representing OOV words using sub-lexical units, the word boundary between two OOV words was lost. To solve this problem, we added two more symbols into the sub-lexical sequence of each OOV word, which were the word start "∧" and word end "$". More details of our hybrid system can be found in [12].

In the hybrid system output, we considered the recognized sub-lexical units as detected OOV candidates. And we segmented a sequence of sub-lexical units into multiple OOV candidates using the word start and word end symbols. Then, we collected the phonetic, acoustic and contextual features for each OOV candidate. As given in Table 1, the phonetic feature is simply the decoded phone sequence of an OOV candidate, the acoustic feature is posterior probability vectors extracted from the OOV region in the testing speech, while the contextual feature is obtained from words surrounding the OOV candidate in the hybrid decoding result. Since we collected evidence from the hybrid system output, recognition errors might be incorporated in those features. For example, in the contextual feature of OOV candidate s_1, the word "major" is a mis-recognition of "mayor"; and the correct pronunciation of OOV candidate s_2 is actually "B AO R AO F". Depending on the hybrid system performance, the collected features could be very noisy, which thus could cause a poor clustering performance.

---

[*] This research was conducted while I was a PhD student at CMU.

Table 1. Examples of the phonetic, acoustic and contextual features of OOV candidates.

OOV	Phonetic	Acoustic	Contextual
s_1	S EH L T S	[0.00 ... 0.17]	... major join crowd wall street ...
s_2	M AO R AO F	[0.01 ... 0.24]	... pakistani minister campaign ...
s_3	W AO L IY	[0.02 ... 0.01]	... play ball court rule gym schedule ...

2.2. Bottom-up clustering

As we normally do not know the number of OOV words in the testing speech and many OOV words only appear once, we cannot apply the centroid-based or distribution-based clustering algorithms, such as the k-means algorithm. Therefore, after collecting features from the hybrid system output, we performed the bottom-up clustering to iteratively find multiple instances of the same OOV word. Initially, each OOV candidate was considered as a single cluster. Then, in each iteration, two clusters with the smallest distance were merged. This clustering procedure ended when the distance between clusters was larger than a threshold. And the threshold was tuned on the development data to achieve the best clustering performance.

We defined the distance between two clusters as the average of pairwise distances between OOV candidates in two clusters. Formally, the distance between cluster C_m and C_n is

$$D(C_m, C_n) = \frac{1}{|C_m||C_n|} \sum_{s \in C_m} \sum_{s' \in C_n} d(s, s'), \tag{1}$$

where $|C_m|$ and $|C_n|$ are the number of candidates in cluster C_m and C_n, and

$$d(s, s') = \omega_P d_P(s, s') + \omega_A d_A(s, s') + \omega_C d_C(s, s'), \tag{2}$$

is the distance between two OOV candidates. Here, $d_P(s, s')$, $d_A(s, s')$ and $d_C(s, s')$ are the phonetic, acoustic and contextual distances between OOV candidate s and s', while ω_P, ω_A, ω_C are their weights respectively.

2.2.1. *Phonetic distance*

The most direct way to determine whether two OOV candidates correspond to the same OOV word or not is to examine whether they have the same pronunciation. To do that, we measured the phonetic similarity between OOV candidates by computing the distance between their decoded phone sequences. Specifically, the phonetic distance $d_P(s, s')$ between OOV candidate s and s' was formulated as the normalized edit distance between their decoded phone sequence p_s and $p_{s'}$:

$$d_P(s, s') = \frac{edit(p_s, p_{s'})}{|p_s| + |p_{s'}|} \tag{3}$$

where $|p_s|$ and $|p_{s'}|$ are the lengths of phone sequence p_s and $p_{s'}$. As shown in Table 1, the decoded phone sequences of OOV candidates may incorporate recognition errors. Particularly, similar phones, such as “AA” and “AO”, are more often to mis-recognize than the other phones. Therefore, we adopted a modified edit distance that compensates for the acoustic confusability between phones [15-18],

$$\begin{aligned} edit(0,0) &= 0 \\ edit(i,0) &= i \\ edit(0,j) &= j \\ edit(i,j) &= min \begin{cases} edit(i-1,j)+1 \\ edit(i,j-1)+1 \\ edit(i-1,i-1)+c(i,j). \end{cases} \end{aligned} \tag{4}$$

In Eq. 4, $c(i, j)$ is the confusability between phones at positions i and j

$$c(i,j) = \begin{cases} 0 & \text{if } i = j \\ 1 - p(i,j) & \text{if } i \neq j, \end{cases} \tag{5}$$

where $p(i, j)$ is the probability of mis-recognizing two phones, which was estimated from the recognition result of the training speech.

2.2.2. *Acoustic distance*

Besides measuring the phonetic distance between OOV candidates, we can also compare their acoustic features extracted from the OOV region in the testing speech. Acoustic features, such as the mel-scale frequency cepstral coefficients (MFCCs), are highly sensitive to speaker and channel variations. On the other hand, posterior-based features, such as the phonetic posteriorgram, are more robust and also widely used in speech recognition [19-21]. Therefore, we used the posterior feature to measure the acoustic distance between OOV candidates. Precisely, each frame f_t in the OOV region was represented by a probability vector

$$v_t = [P(p_1|f_t), P(p_2|f_t), ..., P(p_K|f_t)], \tag{6}$$

where $P(p_k|f_t)$ is the posterior probability of f_t belonging to phone p_k and K is the number of phones. To estimate $P(p_k|f_t)$, we trained a Gaussian mixture model (GMM) with 256 Gaussian components for each phone. Then the posterior probability $P(p_k|f_t)$ can be calculated as

$$P(p_k|f_t) = \frac{P(f_t|p_k)}{\sum_{k \in K} P(f_t|p_k)}, \tag{7}$$

where $P(f_t|p_k)$ is the likelihood of observing f_t from the GMM of p_k. In our experiments, we found that the probability mass was usually absorbed by only a few GMMs. Most phones had a posterior probability close to zero. Because of that, we performed a discounting-based smoothing on the posterior probability vector v_t in a way similar to [21]. Specifically, each zero element in v_t was assigned a small posterior probability λ, and each non-zero element was discounted by $(1 - N\lambda)$, where N is the number of zero elements in v_t.

After constructing the posterior features, we calculated the acoustic distance between OOV candidates using the dynamic time warping (DTW) algorithm [22, 23],

$$d_A(s, s') = DTW(s, s'). \tag{8}$$

In DTW, the distance between two posterior vectors v_i and v_j was defined as the negative log cosine similarity between v_i and v_j

$$d(v_i, v_j) = -log(\frac{v_i \cdot v_j}{\|v_i\|\|v_j\|}). \tag{9}$$

Moreover, similar to the phonetic distance, we also normalized the acoustic distance by the lengths of OOV regions.

2.2.3. Contextual distance

OOV words are usually content words such as names or locations and the same OOV word may appear in similar contexts or environments. If two OOV candidates are surrounded by the same words or used in the same topic, they may actually be the same OOV word. As presented in Eq. 2, besides the phonetic and acoustic distances, we also measured the contextual distance between OOV candidates during clustering. To take the position of surrounding words into account, the contextual distance has two elements:

$$d_C(s, s') = \omega^l d_C^l(s, s') + \omega^g d_C^g(s, s'). \tag{10}$$

Here, $d_C^l(s, s')$ is the local contextual distance that measures the similarity between the adjacent words of OOV candidates, which works like an N-gram LM. And $d_C^g(s, s')$ is the global contextual distance, which resembles a topic model.

Table 2. Examples of the local and global contextual features of OOV candidates.

OOV	s_1	s_2
Text	i am going to watch tonight because s_1 ryan is going to pitch	i love s_2 ryan i alway like to watch him pitch
Local Context	tonight because s_1 ryan is	i love s_2 ryan i
Global Context	watch:0.33 pitch:0.33 ryan:0.33	watch:0.25 pitch:0.25 ryan:0.25 love:0.25

To calculate the local contextual distance, just like the trigram LM, we compared the left two and right two words of OOV candidates

$$d_C^l(s, s') = 1 - \frac{M}{4}, \tag{11}$$

where M is the number of matched words. For instance, as shown in Table 2, there is one match between the local context of OOV candidate s_1 and s_2, hence $d_C^l(s, s')$ equals to 0.75.

The global contextual distance was calculated in the same manner as measuring the similarity between two documents in information retrieval. However here, we focused on words in the same sentence and we only used content words. Particularly, for an OOV candidate s, its global context was represented by a term frequency vector c_g which was built from the content words of the recognition hypothesis containing s. Then the global contextual distance between OOV candidate s and s' was calculated as

$$d_C^g(s, s') = -log(\frac{c_g \cdot c_g'}{||c_g|| ||c_g'||}), \tag{12}$$

which is the negative log cosine similarity between the global context of s and s'. Examples of the global context are also provided in Table 2.

3. LEARNING RECURRENT OOV WORDS

After finding recurrent OOV words from the hybrid system output, we worked on learning the lexical properties for those OOV words. Specifically, we considered clusters with more than one OOV candidates as recurrent OOV words which appeared more than once in the testing speech. Then, we estimated the pronunciation, spelling and POS label for recurrent OOV words by combining evidence from their multiple instances.

3.1. Learning the pronunciation and spelling

To learn a better pronunciation for a recurrent OOV word, we combined the pronunciations of its multiple instances. Specifically, we implemented an algorithm similar to the recognizer output voting error reduction (ROVER) system to produce a composite pronunciation from multiple OOV candidates' decoded pronunciations [24]. Here, the multiple pronunciations were first combined into a single phone transition network. Then this phone transition network was re-scored and searched to find the optimal pronunciation for the OOV word. When re-scoring the phone transition network, we calculated both the phone frequency and phone posterior probability,

$$Score(p_i) = \alpha \cdot \frac{N(p_i)}{\sum_p N(p_i)} + (1 - \alpha) \cdot P(p_i), \tag{13}$$

where $N(p_i)$ is the count of phone p at the i-th alignment in the phone transition network, $P(p_i)$ is the posterior probability calculated from 256 GMM components, and α is the weight used to balance the phone frequency and posterior probability.

For example, as shown in Table 3, our system found three instances of the OOV word "PASHOVSKI" in the testing speech. And the decoded pronunciation of each instance is different from each other. According to the reference lexicon, the correct pronunciation for "PASHOVSK" should be "P AH SH AA V S K IY". We can find that none of the three pronunciations is correct. But by correctly combining those pronunciations, we may be able to estimate the correct pronunciation for the OOV word "PASHOVSK".

Table 3. Examples of the decoded pronunciations of recurrent OOV words.

OOV Candidates	**Decoded Pronunciations**
s_1	K R AH SH N AA V S K IY
s_2	P AH S EH V S K IY
s_3	P AE SH AA F S K IY

After learning the pronunciation for recurrent OOV words, we applied the phoneme-to-grapheme (P2G) conversion to estimate the spelling of those words. To achieve the best P2G conversion performance, we trained a 6-gram joint-sequence model with short graphone units as suggested in [25].

3.2. Learning the POS label

After representing decoded OOV words with estimated spellings in the hybrid system output, we performed the POS tagging to estimate the POS label for recurrent OOV words, where the Stanford MaxEnt POS tagger was used [26]. we adopted all 35 labels from the Penn Treebank POS tag set [27]. In our system, words like "I'VE" or "TEAM'S" were processed as a single unit. However, in the POS tagger, those words were predicted with separate labels. For instance, the POS tagger output of "TEAM'S" was "TEAM NN" and "'S POS". To solve this problem, we combined the separate labels of a word to form a compound label, such as "TEAM'S NN+POS". Therefore, besides the 35 base labels, there were also many compound POS labels in our system. As multiple instances of the same OOV word may be incorrectly tagged with different POS labels if they appeared in different context, we applied majority voting to combine the multiple POS labels of a recurrent OOV word. During vote, if there was a tie between different labels, we randomly selected one label for that OOV word.

4. EXPERIMENT SETUP

4.1. The hybrid system

We built hybrid systems from the the Wall Street Journal (WSJ), Switchboard (SWB) and Broadcast News (BN) corpora, respectively. The WSJ and BN system had a 20k-word vocabulary, while the SWB system had a 10k-word vocabulary. For WSJ, the evaluation data included the WSJ '92 20k-word and '93 64k-word Eval sets. For SWB, a subset of the SWB2 data was selected for evaluation. And for BN, the evaluation data were the F0 and F1 sets of the 1996 HUB4 Eval data.

Table 4. The OOV word detection performance.

Task	WSJ	SWB	BN
OOV Rate	2.2%	1.7%	2.0%
Precision	63.8%	67.2%	49.8%
Recall	74.0%	74.6%	62.4%

From the OOV word detection performance in Table 4, we can find that the hybrid system performs very well in the WSJ and SWB tasks — up to 75% OOV words are detected and the precision is more than 60%. But in the BN task, utterances are usually much longer than those in the WSJ and SWB tasks and multiple OOV words can appear in one utterance or even in a sequence, which make OOV word detection more difficult.

Table 5. The number of instances an OOV word has in the hybrid system output.

OOV Word Has	WSJ	SWB	BN
1 instance	70.8%	77.5%	68.8%
2 instances	24.0%	16.5%	19.5%
$\geq$ 3 instances	5.2%	6.0%	11.7%

The number of instances an OOV word has is given in Table 5. It can be seen that about 70% OOV words only have one instance and less than 10% OOV words have more than two instances. On average, one OOV word has 1.2 instances.

4.2. Evaluation metrics

The Rand index (RI) is a common evaluation metric for clustering [28]. It involves counting pairs of items on which the hypothesis and reference clusterings agree or disagree. In practice however, RI does not take on a constant value for random clustering. Especially, when the number of classes is large and the number of candidates is small, a random clustering result can have a very good RI score. Contrarily, the adjusted Rand index (ARI) is another widely used clustering evaluation metric [29], which adjusts for the chance of a clustering result. The ARI score is bounded between -1 to 1. Independent clusterings has a negative ARI score, similar clusterings has a positive ARI score and an ARI score of 1 indicates a perfect match between the hypothesis and reference clusterings. As shown in Table 5, in our experiment, the majority of clusters only contain one candidate and the candidate to cluster ratio is as low as 1.2. If without clustering but simply consider each candidate as one OOV word, the RI score will be almost 1, but the ARI score will be a small value close to 0. For that reason, we chose to use ARI for clustering evaluation. We also tested the clustering result using the adjusted mutual information (AMI) score [30], which calculates the mutual information between the hypothesis and reference clusterings and is also normalized against chance. In our experiment, we found ARI and AMI had very similar observations. Therefore, only the ARI score was reported.

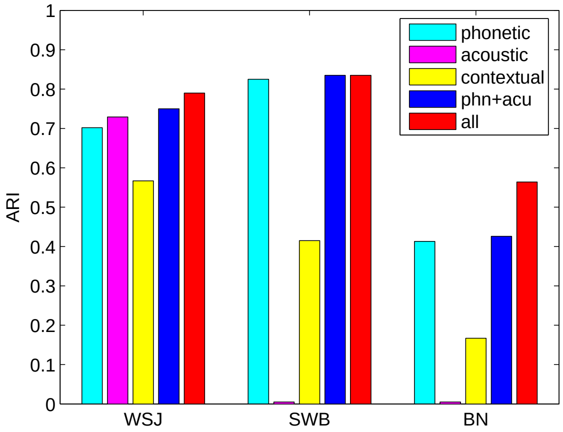

Fig. 1. The bottom-up clustering performance.

To evaluate the OOV word learning performance, we calculated the pronunciation accuracy (PA), recovery rate (RA) and POS label accuracy. PA measures how many detected OOV words are decoded with the correct pronunciation, while RA measures how many detected OOV words' spelling is correct, and the POS label accuracy computes the percentage of OOV words with the correct POS label.

5. EXPERIMENT RESULTS

5.1. The bottom-up clustering results

The bottom-up clustering performance is given in Fig. 1. Let us first compare the clustering performance when using one feature to measure the distance between OOV candidates. We can find that the phonetic feature is very effective in all tasks. The acoustic feature works well in the WSJ task but shows the same ARI score as random clustering in the SWB and BN tasks. This may because that measuring the distance between acoustic signals in the spontaneous or noisy speech is less reliable than in the clean speech. Also note that the WSJ data consists of read speech with extended recordings from each speaker. Although the contextual feature is not as good as the phonetic one, it does produce positive results across different tasks. In addition to using only one feature during clustering, we also applied the combined feature as defined in Eq. 2. It can be seen that the ARI score gradually increases when using more features during clustering. Even for the SWB and BN tasks, where the acoustic feature does not work at all, combining the phonetic and acoustic features can still yield some improvement. And the best performance is achieved when combining all features. Overall, the ARI score is up to 0.8 in the WSJ and SWB tasks and about 0.6 in the BN task, which indicates that we can successfully find most recurrent OOV words using the proposed bottom-up clustering approach.

The goal of finding recurrent OOV word is to combine the evidence from its multiple instances, so that we can estimate better lexical properties for the word. Therefore, during clustering, we prefer not having different OOV words in the same cluster than trying

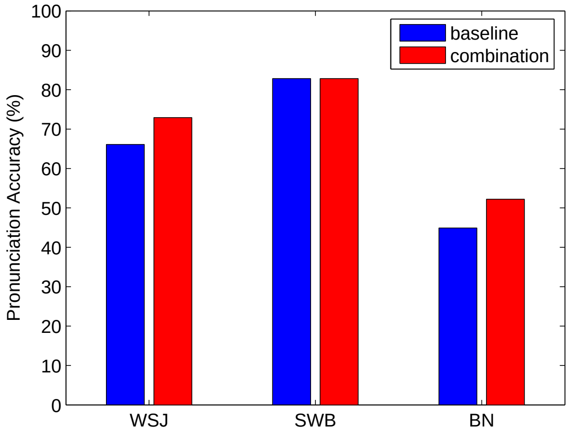

Fig. 2. The pronunciation accuracy of recurrent OOV words with and without combination.

to find all instances of one word. For example, there are four instances of the OOV word "CIBA" in the testing speech. We prefer finding two or three instances than grouping all four "CIBA" with some other OOV words into the same cluster. From the clustering result, for clusters with more than one OOV candidates, we counted how many clusters are pure — only contain instances of the same OOV word. From Table 6, we can find that most detected recurrent OOV words are correct, which only consist of instances of the same word. As a result, we were able to collect evidence from multiple instances of recurrent OOV words for OOV word learning.

Table 6. The percentage of detected recurrent OOV words which only contain instances of the same OOV word.

Task	**WSJ**	**SWB**	**BN**
Correct Clusters	97%	81%	86%

5.2. The OOV word learning results

To estimate a better pronunciation for recurrent OOV word, we combined the pronunciations of its multiple instances. The pronunciation accuracy (PA) of recurrent OOV words with and without combination is presented in Fig. 2. It can be seen that PA increases substantially after combining the multiple pronunciations of an OOV word in the WSJ and BN tasks. In the SWB task, however, we found that multiple instances of the same OOV words were usually decoded with the same pronunciation. As a result, we did not get any improvement from combination.

We also evaluated how many recurrent OOV words had the correct spelling after the P2G conversion. The recovery rate (RR) of recurrent OOV words with and without combination is given in Fig. 3. We can see that RR also increases when performing the P2G conversion on the combined pronunciation of a recurrent OOV word in the WSJ and BN tasks. As there is no improvement on PA in the SWB task, RR also does not change. By comparing Fig. 2 with Fig. 3, we can find that RR is usually lower than PA, as the P2G conversion failed to estimate the spelling of many OOV words, although their pronunciations are correct.

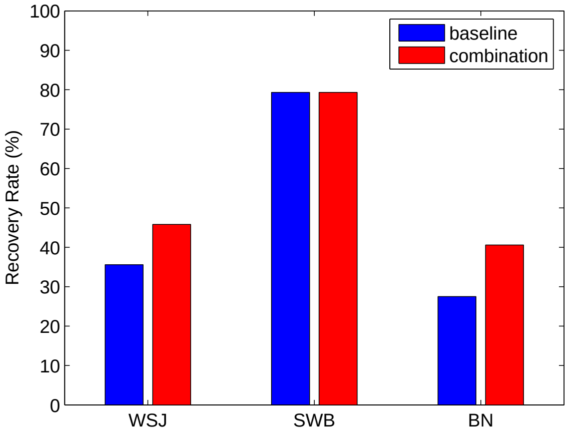

Fig. 3. The recovery rate of recurrent OOV words with and without combination.

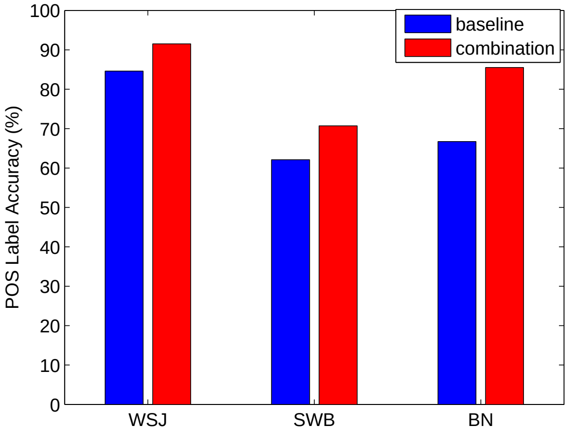

Fig. 4. The POS label accuracy of recurrent OOV words with and without combination.

Besides estimating better pronunciation and spelling for recurrent OOV word, we also combined POS labels of multiple instances of the same word to produce a more accurate label. Fig. 4 presents the POS label accuracy of recurrent OOV words with and without combination. It can be seen that the POS label accuracy is improved after combining labels from multiple instances of an OOV word. The POS label accuracy in the SWB task is much lower than that in the WSJ and BN tasks, this may because many conversational utterances are not grammatical in the SWB task, which makes POS tagging much harder.

6. CONCLUSIONS

In this paper, we studied learning better lexical properties for recurrent OOV words. Specifically, we first identified recurrent OOV words through bottom-up clustering. We then estimated better pronunciation, spelling and POS label for recurrent OOV words by utilizing their multiple instances. From the experimental results, we found that the bottom-up clustering approach correctly identified most recurrent OOV words. Furthermore, the OOV word learning performance was substantially improved by combining evidence from multiple instances of the same word. Next, we would like to investigate how to integrate detected OOV words into the recognizer's lexicon and language model, so that the recognizer can recognize those OOV words as IV words when encountered in the future.

7. ACKNOWLEDGEMENTS

This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense U.S. Army Research Laboratory (DoD/ARL) contract number W911NF-12-C-0015. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government.

8. REFERENCES

[1] D. Klakow, G. Rose, and X. Aubert, "OOV-detection in large vocabulary system using automatically defined word-fragments as fillers," *Proc. Eurospeech-1999*, pp. 49-52, 1999.

[2] I. Bazzi and J. Glass, "Modeling out-of-vocabulary words for robust speech recognition," *Proc. ICSLP-2000*, vol. 1, pp. 401-404, 2000.

[3] T. Schaaf, "Detection of OOV words using generalized word models and a semantic class language model," *Proc. Eurospeech-2001*, pp. 2581-2584, 2001.

[4] L. Galescu, "Recognition of out-of-vocabulary words with sub-lexical language models," *Proc. Eurospeech-2003*, pp. 249 252, 2003.

[5] M. Bisani and H. Ney, "Open vocabulary speech recognition with flat hybrid models," *Proc. Interspeech-2005*, pp. 725-728, 2005.

[6] A. Rastrow, A. Sethy, B. Ramabhadran, and F. Jelinek, "Towards using hybrid, word and fragment units for vocabulary independent LVCSR systems," *Proc. Interspeech-2009*, pp. 1931-1934, 2009.

[7] C. Parada, M. Dredze, D. Filimonov, and F. Jelinek, "Contextual information improves OOV detection in speech," *Proc. HLT-NAACL-2010*, pp. 216-224, 2010.

[8] M. Shaik, A. El-Desoky, R. Schluter, and H. Ney, "Hybrid language model using mixed types of sub-lexical units for open vocabulary German LVCSR," *Proc. Interspeech-2011*, pp. 1441-1444, 2011.

[9] I. Bulyko, J. Herrero, C. Mihelich, and O. Kimball, "Subword speech recognition for detection of unseen words," *Proc. Interspeech-2012*, 2012.

[10] L. Qin, M. Sun, and A. Rudnicky, "OOV detection and recovery using hybrid models with different fragments," *Proc. Interspeech-2011*, pp. 1913-1916, 2011.

[11] L. Qin, M. Sun, and A. Rudnicky, "System combination for out-of-vocabulary word detection," *Proc. ICASSP-2012*, pp. 4817-4820, 2012.

[12] L. Qin and A. Rudnicky, "OOV word detection using hybrid models with mixed types of fragments," *Proc. Interspeech-2012*, 2012.

[13] L. Qin and A. Rudnicky, "Finding recurrent out-of-vocabulary words," *to appear in Proc. Interspeech-2013*, 2013.

[14] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434-451, 2008.

[15] R. Wagner and M. Fischer, " The string-to-string correction problem," *J. ACM*, vol. 21, no. 1, pp. 168-173, 1974.

[16] K. Audhkhasi and A. Verma, "Keyword search using modified minimum edit distance measure," *Proc. ICASSP-2007*, vol. 4, pp. 929-932, 2007.

[17] M. Pucher, A. Turk, J. Ajmera and N. Fecher, "Phonetic distance measures for speech recognition vocabulary and grammar optimization," *Proc. the 3rd Congress of the Alps Adria Acoustics Association*, 2007.

[18] H. Printz and P. Olsen, "Theory and practice of acoustic confusability," *Computer Speech & Language*, vol. 16, pp. 131-164, 2002.

[19] G. Aradilla, J. Vepa and H. Bourlard, "Using posterior-based features in template matching for speech recognition," *Proc. Interspeech-2006*, 2006.

[20] T. Hazen, W. Shen and C. White, "Query-by-example spoken term detection using phonetic posteriorgram templates," *Proc. ASRU-2009*, pp. 421-426, 2009.

[21] Y. Zhang and J. Glass, "Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams," *Proc. ASRU-2009*, pp. 398-403, 2009.

[22] T.K. Vintsyuk, "Speech discrimination by dynamic programming," *Kibernetika*, vol. 4, pp. 81-88, 1968.

[23] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. ASSP*, vol. 26, no. 1, pp. 43-49, 1978.

[24] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)," *Proc. ASRU-1997*, pp. 347-354, 1997.

[25] S. F. Chen, "Conditional and joint models of grapheme-to-phoneme conversion," *Proc. Eurospeech-2003*, pp. 2033-2036, 2003.

[26] K. Toutanova, D. Klein, C. Manning and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," *Proc. HLT-NAACL-2003*, pp. 252-259, 2003.

[27] M. P. Marcus, M. Marcinkiewicz and B. Santorini, "Building a large annotated corpus of English: the penn treebank," *Computational Linguistics*, vol. 19, no. 2, pp. 313-330, 1993.

[28] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical Association*, vol. 66, no. 336, pp. 846-850, 1971.

[29] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, no. 1, pp. 193-218, 1985.

[30] N. X. Vinh and J. Epps, "Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance," *Journal of Machine Learning Research*, vol. 11, pp. 2837-2854, 2010.