

Finding Recurrent Out-of-Vocabulary Words

Long Qin, Alexander Rudnicky

Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, USA

{lqin, air}@cs.cmu.edu

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 estimating the pronunciation or the part-of-speech (POS) tag of the word. But in a conventional OOV word detection system, each OOV word is recognized and treated individually. We therefore investigated how to identify recurrent OOV words in speech recognition. Specifically, we propose to cluster multiple instances of the same OOV word using a bottom-up approach. Phonetic, acoustic and contextual features were collected to measure the distance between OOV candidates. The experimental results show that the bottom-up clustering approach is very effective at detecting the recurrence of OOV words. We also found that the phonetic feature is better than the acoustic and contextual features, and the best performance is achieved when combining all features.

Index Terms: OOV word detection, distributed evidence, bottom-up clustering

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 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 using 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 this paper, we describe how to find recurrent OOV words in a speech recognition system through unsupervised clustering. As we do not know the correct number of OOV words in the testing speech, and many OOV words only have one or two instances, we cannot apply the centroid-based or distribution-based clustering algorithms, such as the k-means algorithm. Therefore we propose to cluster multiple instances of the same OOV word using a bottom-up approach. We began with collecting the phonetic, acoustic and contextual features for OOV candidates in the hybrid system output. Then each OOV candidate was considered as one cluster and pairs of clusters were iteratively merged until the distance between two clusters exceeded a threshold. The proposed approach was tested on tasks 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 details of the bottom-up clustering approach and the definition of the phonetic, acoustic and contextual distances. Sections 3 and 4 discuss experiments and results. Concluding remarks are provided in Section 5.

2. Method

2.1. OOV word detection using a 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. And the hybrid LM was trained in a flat manner. First, pronunciations of all OOV words were estimated through the grapheme-to-phoneme (G2P) conversion [13], 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. Details of the hybrid system can be found in [12].

In the hybrid system output, sub-lexical sequences were considered as detected OOV candidates, where word boundary symbols were used to segment a sequence of sub-lexical units into multiple OOV candidates. 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 the words surrounding the OOV candidate. Note that since we worked on the hybrid system output, recognition errors might be incorporated in these 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.

Table 1: Examples of the phonetic, acoustic and contextual features of an OOV candidate.

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

2.2. Bottom-up clustering

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

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

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

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

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. In addition to averaging the pairwise distances between OOV candidates, we also experimented with calculating $D(C_m, C_n)$ as the maximum or minimum distance between OOV candidates in two clusters. However, we found that the clustering performance with different definitions of $D(C_m, C_n)$ was essentially the same, although the average one occasionally performed better.

2.2.1. Phonetic distance

The most direct way to determine whether two OOV candidates may correspond to the same OOV word 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 phone sequence p_s and $p_{s'}$:

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

where $|p_s|$ and $|p_{s'}|$ are the lengths of phone sequence p_s and $p_{s'}$. As shown previously 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 [14-17],

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

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

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

where $p(i, j)$ is the probability of mis-recognizing phone i and phone j , 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 [18-20]. Therefore, we used the posterior feature to model OOV candidates in our system. Precisely, each frame f_t in the OOV region was represented by a probability vector

$$v_t = [P(p_1|f_t), P(p_2|f_t), \dots, P(p_K|f_t)], \quad (6)$$

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

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

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 [20]. 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 [21, 22],

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

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

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

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:

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

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 an OOV candidate.

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 always 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

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

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 sentence containing s . Then the global contextual distance between OOV candidate s and s' was calculated as

$$d_C^g(s, s') = -\log\left(\frac{c_g \cdot c'_g}{\|c_g\| \|c'_g\|}\right), \quad (12)$$

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. Experiment setup

3.1. 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 was the F0 and F1 sets of the 1996 HUB4 Eval data.

Table 3: 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 3, we can find that the hybrid system performs very well in the WSJ and SWB tasks — more than 60% OOV words are detected and the precision is up to 75%. But in the BN task, utterances are usually much longer than that in the WSJ and SWB tasks and

Table 4: OOV instance count 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%
≥ 3 instances	5.2%	6.0%	11.7%

multiple OOV words can appear in one utterance or even in a sequence, which makes OOV word detection more difficult.

The number of instances each OOV word has is given in Table 4. 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.

3.2. Evaluation metrics

The Rand index (RI) is a common evaluation metric for clustering [23]. 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 [24], 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 4, 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 evaluation. We also tested the clustering result using the adjusted mutual information (AMI) score [25], 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.

4. Experiment results

4.1. The intra-cluster and inter-cluster distances

Before discussing the clustering performance, we first take a closer look at the testing data. Fig. 1 shows the comparison of the average distance between instances of the same OOV word (intra-cluster) with the average distance between instances of different OOV words (inter-cluster). It can be seen that for the phonetic, acoustic and contextual features, the intra-cluster distance is always smaller than the inter-cluster distance. Moreover, the difference between the phonetic intra-cluster and inter-cluster distances is greater than that of the other features. Furthermore, OOV candidates in the WSJ and SWB tasks seem to be more separable than those in the BN task.

4.2. The bottom-up clustering results

The performance of bottom-up clustering using one feature is given in Fig. 2. 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 be because that measur-

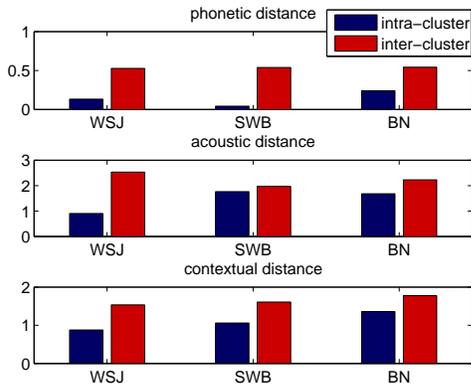


Figure 1: Comparison of the average distance between instances of the same OOV word (intra-cluster) with the average distance between instances of different OOV words (inter-cluster).

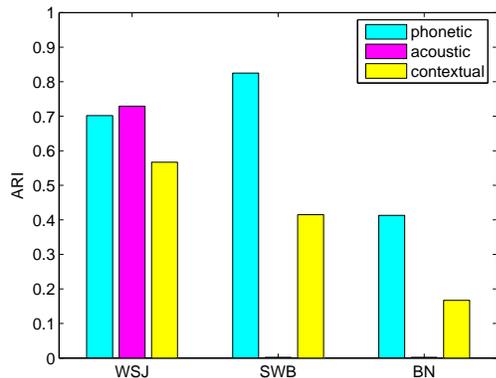


Figure 2: The performance of bottom-up clustering using one feature.

ing the distance between acoustic signals in the spontaneous or noisy speech is less reliable than in clean speech. Although the contextual feature is not as good as the phonetic one, it does produce positive results across different tasks. By comparing Fig. 2 with Fig. 1, we can also learn that the clustering performance is highly correlated with the difference between the intra-cluster and inter-cluster distances of one feature. For instance, the difference between the phonetic intra-cluster and inter-cluster distances is great in all tasks, and the clustering performance using the phonetic feature is always good. On the other hand, the difference between the acoustic intra-cluster and inter-cluster distances is only noticeable in the WSJ task, and the bottom-up clustering using the acoustic feature performs badly in the SWB and BN tasks. The best performance is obtained when using the acoustic feature in the WSJ task and using the phonetic feature in the SWB and BN tasks.

In addition to using only one feature to measure the distance between OOV candidates during clustering, we also applied the combined feature defined in Eq. 2. Fig. 3 shows the performance of bottom-up clustering using the combined feature, in which the red bar is the best clustering performance using one feature, the green bar is the performance when using both the

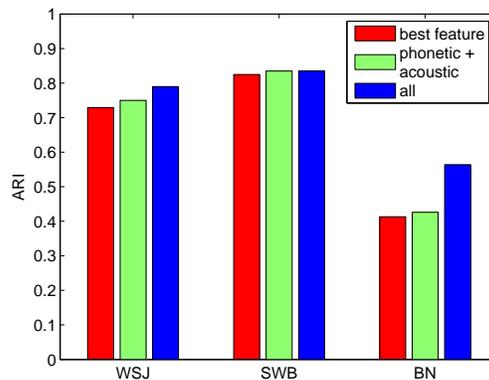


Figure 3: The performance of bottom-up clustering using the combined feature.

phonetic and acoustic features, and the blue bar is the performance when combining all features. 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 of the recurrent OOV words using the proposed bottom-up clustering approach. In fact, in the clustering result, most clusters only contain instances of the same OOV word. When calculating ARI only from those clusters, the ARI score is up to 0.9 in all tasks. Therefore, the clustering result is good enough for further process, such as learning the pronunciation of POS tag of recurrent OOV words.

5. Conclusions and future work

In this paper, we studied a bottom-up clustering approach to find recurrent OOV words in speech recognition. We collected phonetic, acoustic and contextual features to measure the distance between OOV candidates. From our experimental results, we found that the phonetic feature is more effective than the acoustic and contextual features for detecting the recurrence of OOV words, but the best performance is achieved when combining all features. In the future, we would like to investigate how to build a better phonetic representation from multiple instances of the same OOV word. We are also interested in learning the POS tag and language model scores of recurrent OOV words.

6. 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.

7. 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, "Sub-word 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] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434-451, 2008.
- [14] R. Wagner and M. Fischer, "The string-to-string correction problem," *J. ACM*, vol. 21, no. 1, pp. 168-173, 1974.
- [15] K. Audhkhasi and A. Verma, "Keyword search using modified minimum edit distance measure," *Proc. ICASSP-2007*, vol. 4, pp. 929-932, 2007.
- [16] 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.
- [17] H. Printz and P. Olsen, "Theory and practice of acoustic confusability," *Computer Speech & Language*, vol. 16, pp. 131-164, 2002.
- [18] G. Aradilla, J. Vepa and H. Bourlard, "Using posterior-based features in template matching for speech recognition," *Proc. Interspeech-2006*, 2006.
- [19] T. Hazen, W. Shen and C. White, "Query-by-example spoken term detection using phonetic posteriorgram templates," *Proc. ASRU-2009*, pp. 421-426, 2009.
- [20] Y. Zhang and J. Glass, "Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams," *Proc. ASRU-2009*, pp. 398-403, 2009.
- [21] T.K. Vintsyuk, "Speech discrimination by dynamic programming," *Kibernetika*, vol. 4, pp. 81-88, 1968.
- [22] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. ASSP*, vol. 26, no. 1, pp. 43-49, 1978.
- [23] 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.
- [24] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, no. 1, pp. 193-218, 1985.
- [25] 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.