論文 / 著書情報
Article / Book Information

Title	Recurrent Out-of-Vocabulary Word Detection Using Distribution of Features
Authors	Taichi Asami, Ryo Masumura, Yushi Aono, Koichi Shinoda
Citation	Proc. Interspeech, , , pp. 1320-1324, Interspeech.2016-562
Pub. date	2016, 9
Copyright	(c) 2016 International Speech Communication Association, ISCA
DOI	http://dx.doi.org/10.21437/Interspeech.2016-562

INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Recurrent Out-of-Vocabulary Word Detection Using Distribution of Features

Taichi Asami[1], *Ryo Masumura*[1], *Yushi Aono*[1], *Koichi Shinoda*[2]

[1]NTT Media Intelligence Laboratories, NTT Corporation, Japan
[2]Tokyo Institute of Technology, Japan

{asami.taichi, masumura.ryo, aono.yushi}@lab.ntt.co.jp, shinoda@cs.titech.ac.jp

Abstract

The repeated use of out-of-vocabulary (OOV) words in a spoken document seriously degrades a speech recognizer's performance. This paper provides a novel method for accurately detecting such recurrent OOV words. Standard OOV word detection methods classify each word segment into in-vocabulary (IV) or OOV. This word-by-word classification tends to be affected by sudden vocal irregularities in spontaneous speech, triggering false alarms. To avoid this sensitivity to the irregularities, our proposal focuses on consistency of the repeated occurrence of OOV words. The proposed method preliminarily detects recurrent segments, segments that contain the same word, in a spoken document by open vocabulary spoken term discovery using a phoneme recognizer. If the recurrent segments are OOV words, features for OOV detection in those segments should exhibit consistency. We capture this consistency by using the mean and variance (distribution) of features (DOF) derived from the recurrent segments, and use the DOF for IV/OOV classification. Experiments illustrate that the proposed method's use of the DOF significantly improves its performance in recurrent OOV word detection.

Index Terms: speech recognition, OOV word detection, recurrent OOV words, distribution of features

1. Introduction

Automatic speech recognition (ASR) systems normally use a pre-constructed lexicon that defines a word set, i.e. vocabulary, that can be uttered by users. Only in-vocabulary (IV) words can be recognized correctly. However, in practical use cases, out-of-vocabulary (OOV) words are likely to be input. For example, names of people/places/products or technical terms are likely to be OOV words since it is difficult to preliminarily define all of them in the lexicon.

Moreover, names or technical terms are likely to be important keywords in spoken documents such as conversations or lectures, and are likely repeatedly uttered. Our examination of academic lectures found 66% of OOV words were uttered 2 or more times (see Section 3.1). When OOV words are repeatedly uttered, speech recognizer's performance is seriously degraded since these keywords are never correctly recognized. In order to deal with this serious problem, detecting the recurrence of OOV words is important.

OOV word detection has been studied over the years, and methods based on the word/fragment hybrid ASR approach are widely used [1, 2, 3, 4, 5]. Hybrid ASR uses a hybrid lexicon consisting of not only words but also subword sequences (fragments) and a hybrid language model (LM) trained on texts in which low frequency words are replaced by fragment sequences. When an OOV word is uttered, fragments have high posterior probability in the confusion networks [6] derived from the hybrid ASR. Thus features based on the posterior probability are extracted from each slot (and its context) of the confusion networks and input to an IV/OOV classifier.

The conventional methods classify each slot of the confusion networks into IV or OOV and does not take into account the recurrence of OOV words. Features for classification are extracted in a slot-by-slot manner. However, the slot-by-slot features tend to be affected by the sudden vocal irregularities (or disfluencies) common in real utterances, such as hesitation, repairs, or sloppy pronunciations. Disfluencies are not OOV words, but cause high posterior probability of fragments in the confusion networks of the hybrid ASR. This sensitivity to disfluencies raises many false alarms in the OOV detector, especially with spontaneous speech.

In this paper, we propose a novel method aiming at reducing the errors in detecting recurrent OOV words by utilizing their multiple appearance in spoken documents. When the same OOV word appears in multiple segments, the posterior probabilities of fragments in those segments become *consistently* high. Our key idea is that the sensitivity of the slot-by-slot features can be offset by focusing on this consistency. The segments in which the same word appears (recurrent segments) are detected by open vocabulary spoken term discovery using phoneme recognizers [7, 8, 9]. The mean and variance (distribution) of the slot-by-slot features of the recurrent segments capture the consistency, e.g. if the recurrent segments are OOV, the fragment posterior probabilities of the segments should have large mean and small variance. The proposed method uses the distribution of features (DOF) for OOV classification. Since DOF reflects the statistics of multiple samples, it should be robust to irregularities such as disfluencies.

One previous method detects recurrence of OOV words [4]. This method detects individual OOV words by conventional slot-by-slot classification and applies bottom-up clustering to detect OOV word clusters. This method is effective when individual OOV word detection is very accurate. Note that we assume a different situation where individual OOV word detection is made difficult due to the presence of many disfluencies.

This paper is organized as follows. Section 2 details our recurrent OOV word detection method; it uses the DOF derived from pre-detected recurrent segments. Conditions and results of OOV word detection experiments on spontaneous speech are presented in Section 3, and Section 4 concludes this paper.

2. Method for recurrent OOV word detection

The whole scheme of the procedure of our recurrent OOV word detector is illustrated in Figure 1.

An input spoken document, e.g. utterances in a lecture, is decoded by both a phoneme recognizer and a word/fragment

http://dx.doi.org/10.21437/Interspeech.2016-562

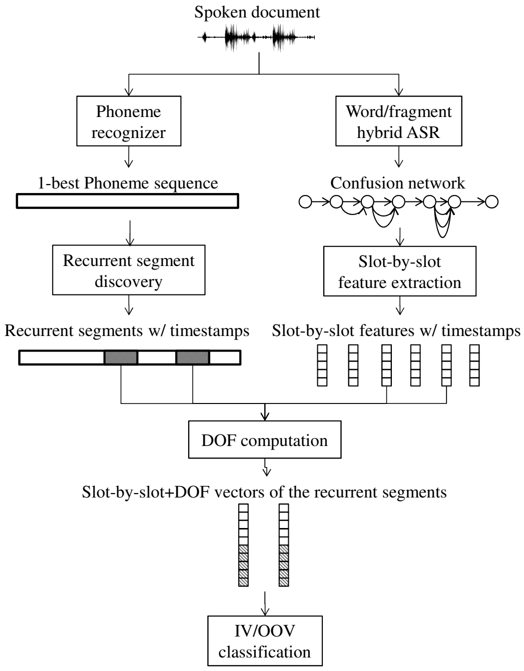
Spoken document
Phoneme recognizer
Word/fragment hybrid ASR
1-best Phoneme sequence
Confusion network
Recurrent segment discovery
Slot-by-slot feature extraction
Recurrent segments w/ timestamps
Slot-by-slot features w/ timestamps
DOF computation
Slot-by-slot+DOF vectors of the recurrent segments
IV/OOV classification

Figure 1: *Recurrent OOV word detection using distribution of features.*

hybrid recognizer. From the output of the phoneme recognizer, recurrent segments in which the same word is uttered, are detected by the recurrent segment discovery module. Standard slot-by-slot features are extracted from the confusion network yielded by the hybrid recognizer. DOFs are computed by using the slot-by-slot features that correspond to the recurrent segments. The slot-by-slot features and the DOF are concatenated and input to the IV/OOV classifier, and each recurrent segment is classified as either IV or OOV.

Note that Figure 1 shows the simplest case in which only one pair of recurrent segments are detected. Actually many recurrent segments (no overlaps) are detected and the DOF computation and IV/OOV classification are applied to each recurrent segment.

Details of each module are given below.

2.1. Phoneme recognition and recurrent segment discovery

The input spoken document is converted into a 1-best phoneme sequence by a phoneme recognizer using a deep neural network-based triphone HMM (DNN-HMM) acoustic model and a phoneme 3-gram LM. In our experiments on 2701 lectures in the Corpus of Spontaneous Japanese (CSJ) dataset [10], the phoneme error rate was 13.7%.

The objective of recurrent segment discovery is detecting segments where the same word is uttered. We borrow the idea of subword-based open vocabulary spoken term detection [7, 8, 9], and assume that similar sub-sequences appearing in the 1-best phoneme sequence can be treated as the same word. In the proposed method, similar sub-sequences are extracted by two steps: 1) Detecting sub-sequences whose frequency is at least N and length (number of phonemes) is at least L, and 2) clustering similar sub-sequences.

All sub-sequences that have at least frequency N and length L can be efficiently extracted by the PrefixSpan algorithm [11] which is widely used for frequent sequential pattern mining. We set L to 5 since most OOV words have at least 5 phonemes (approximately 3 Japanese moras), and N to 2 for extracting as many as possible recurrent segments (i.e. OOV word candidates). Sub-sequences are extracted with timestamps in the spoken document, and if detected sub-sequences overlap, they are merged into one longer sub-sequence.

Even if the same word is uttered, the decoded phoneme sequences are likely to be slightly different because of ambiguity in pronunciation or phoneme recognition errors. In order to overcome these small differences, we collect similar sub-sequences based on the edit distance between sub-sequences. The distance between two sub-sequences, s_1 and s_2, is calculated as normalized edit distance:

$$D(s_1, s_2) = \frac{\text{edit}(s_1, s_2)}{\max(|s_1|, |s_2|)}, \qquad (1)$$

where edit(s_1, s_2) is the edit distance between s_1 and s_2, and $|s_1|$ and $|s_2|$ are the number of phonemes in s_1 and s_2, respectively. $D(s_1, s_2)$ becomes 0 when s_1 and s_2 are the same, and 1 when s_1 and s_2 are completely different.

Since the number of unique words in the spoken document is unknown, the number of clusters cannot be pre-determined. Thus, we employ a graph-based clustering method that detects the appropriate number of clusters automatically. A similarity graph of sub-sequences is constructed based on the normalized edit distance (similarity is $1 - D(s_1, s_2)$), and input to the graph-based clustering algorithm. In our experiments, the Chinese Whispers algorithm [12] is used as the graph-based clustering method, as it is parameter-free and good performance was reported in [13]. Sub-sequences in the same cluster are treated as recurrent segments.

2.2. Hybrid ASR and slot-by-slot feature extraction

The input spoken document is also processed by the word/fragment hybrid ASR to extract slot-by-slot features.

Fragments (phoneme sequences) used in the hybrid ASR system are selected by the strategy described in [1]. The LM training texts are converted into phoneme sequences by the grapheme-to-phoneme converter. Then a phoneme 5-gram LM is trained using the converted texts, and entropy-based pruning [14] is applied to select important fragments. In experiments, we adjusted the pruning parameter so as to select 10K fragments. The hybrid lexicon and 3-gram LM are constructed on the LM training texts in which words with frequency 1 are replaced by their fragment sequences.

As the slot-by-slot features, we use word/fragment posteriors and LM-related scores obtained from the confusion network of the hybrid ASR. The effectiveness of these values was reported in previous studies [1, 2]. Specifically, we extract the following values from each slot of the confusion network:

- **Fragment posterior**: Sum of posterior probabilities of fragments in the target slot:

$$\text{FragmentPosterior} = \sum_{f \in S} P(f|S), \qquad (2)$$

where f denotes a fragment in the hybrid lexicon and S denotes a set of words/fragments in the target slot of the confusion network.

- **Word entropy**: Entropy of posterior probabilities of words in the target slot:

$$\text{WordEntropy} = -\sum_{w \in S} P(w|S) \log P(w|S), \qquad (3)$$

where w denotes a word in the hybrid lexicon.

- **1-best posterior probability**: Maximum posterior probability in the target slot.
- **LM score**: LM score of the word/fragment that has the largest posterior probability in the target slot.
- **LM back-off order**: The back-off order of the 3-gram of word/fragment with the largest posterior probability in the previous 2 slots and the target slot.

These five values are computed for each slot, and the values of surrounding slots are concatenated as context features. We set the context window size to 2, i.e. previous 2 and post 2 slots are used as the context, and a concatenated 25 dimensional vector is used as a slot-by-slot feature of the target slot. We don't use the word itself as a feature since the raw lexical information is highly dependent on the domain (topic) of the LM training texts.

2.3. DOF computation

In order to capture the consistency of the slot-by-slot features from the multiple appearances of the same word, distribution of features (DOF) are computed using the sub-sequence cluster (i.e. recurrent segments) obtained in Section 2.1.

The DOF consists of the means and variances of slot-by-slot features. If recurrent segments in a cluster are recurrent OOV words, the segments are likely to have consistently OOV-like features, e.g. large fragment posteriors. This consistency is captured by taking the means and variances in the cluster, e.g. large mean and small variance of fragment posteriors strongly indicate that the recurrent segments in the cluster are recurrent OOV words. These statistics should be a more robust indicator of OOV than individual slot-by-slot features.

A DOF is computed for each cluster as follows:

1. Slot-by-slot features corresponding to the cluster are selected based on timestamps. For each recurrent segment in the cluster, a slot-by-slot feature that has the longest overlap is selected as the corresponding feature.
2. The DOF of the cluster, $\boldsymbol{d}$, is computed as the element-wise means and variances of the selected slot-by-slot features:

$$\boldsymbol{d} = [\boldsymbol{\mu}^T \boldsymbol{\sigma}^T]^T, \tag{4}$$

$$\boldsymbol{\mu} = \frac{1}{M}\sum_{m=1}^{M} \boldsymbol{v}_m, \tag{5}$$

$$\boldsymbol{\sigma} = \mathrm{diag}\left\{ \frac{1}{M}\sum_{m=1}^{M} (\boldsymbol{\mu} - \boldsymbol{v}_m)(\boldsymbol{\mu} - \boldsymbol{v}_m)^T \right\} \tag{6}$$

where M denotes the number of recurrent segments in the cluster, and $\boldsymbol{v}_m$ denotes the corresponding slot-by-slot feature of the m-th recurrent segment. T denotes vector transposition and diag represents the vector consisting of the diagonal elements of the matrix.

Finally, the m-th recurrent segment in the cluster has 75 (25 slot-by-slot and 50 DOF) dimensional feature vector, $[\boldsymbol{v}_m^T \boldsymbol{d}^T]^T$, and this vector is used for IV/OOV classification. Note that recurrent segments in a cluster share the same DOF. By applying the above procedure to all clusters, all recurrent segments are assigned their own 75 dimensional feature vector with DOF.

Table 1: *Amounts of data sets.*

Group	#lectures	Time length	Vocab. size
A	1351	266h	62741
B	1350	265h	62887

Table 2: *Data used in the experiments.*

Group	Test 1	Test 2	Test 3	Test 4
A-1	ASRtrain	ASRtrain	OOVtrain	Test
A-2	ASRtrain	ASRtrain	Test	OOVtrain
B-1	OOVtrain	Test	ASRtrain	ASRtrain
B-2	Test	OOVtrain	ASRtrain	ASRtrain

2.4. IV/OOV classification

IV/OOV classification is based on the standard supervised training framework. A training set, a set of spoken documents in which true OOV segments are known, is used for training a classifier. The timestamps of the true OOV segments can be obtained by forced alignment using manual transcriptions. The trained classifier is used for labeling recurrent segments in the test spoken documents with either IV or OOV. Feature vectors with DOF described in Section 2.3 are used for classification.

Several binary classifiers can be used for IV/OOV classification. We use the multi-layer perceptron (MLP) as a classifier since the proposed DOFs are real values and MLP can use real values as input without any quantization. Note that sequence classifiers such as the conditional random field or the recurrent neural network are not suitable since the classification targets (recurrent segments) do not necessarily form a sequence as shown in Figure 1.

3. Experiments

3.1. Data

Corpus of Spontaneous Japanese [10] was used for OOV word detection experiments. It consists of 2701 academic lectures (531 hours, 7M words) with manual transcriptions. Each lecture was treated as one spoken document.

The lectures were randomly split into two groups to make ASR training sets so that the amounts of the two groups were balanced. Table 1 shows the size of the groups. The DNN-HMM acoustic model, the hybrid lexicon and the hybrid 3-gram LM trained on Group A were used for recognizing Group B, and vice versa. The DNN of the acoustic model had 8 hidden layers with 2048 sigmoid units and a softmax output layer with 3072 units, which was initialized by discriminative pre-training [15] and fine-tuned by SGD with momentum. 11 consecutive frames (center, previous 5 and post 5 frames) of 38 dimensional acoustic features (12MFCC, 12ΔMFCC, 12$\Delta\Delta$MFCC, Δpower and $\Delta\Delta$power) were concatenated and input to the DNN. JTAG [16] was used as the grapheme-to-phoneme converter in training of the hybrid LM. Decoding was performed by the WFST-based decoder VoiceRex [17, 18]. Word error rates of Group A and B were 22.9% and 23.0%, respectively. In this setting, total number of OOV words in Group A and B was 79826, and the OOV rate was 1.1%. Figure 2 shows the histogram of the number of OOV word repetitions per lecture. According to the histogram, 66% of OOV words in a lecture appeared 2 or more times.

To make a training set for the OOV classifier separately from the ASR training set, we conducted two-fold cross vali-

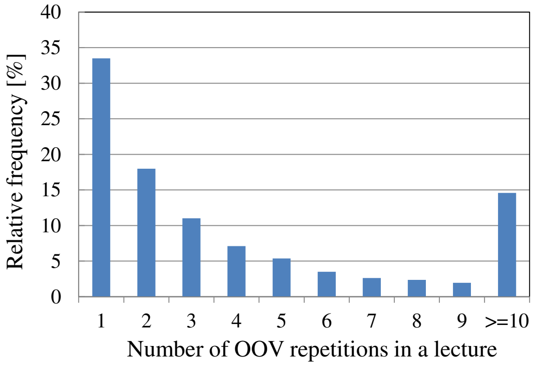

Figure 2: *Histogram of number of OOV repetitions in a lecture.*

dation. Table 2 shows the data used in our experiments. All 2701 lectures were used as a test set through the four tests, and the overall results are reported in Section 3.3.

3.2. Experimental conditions

The parameters of recurrent segment discovery, hybrid ASR and slot-by-slot feature extraction are described in Section 2.1 and 2.2. The MLP for IV/OOV classification has 2 hidden layers with 64 sigmoid units and a softmax output layer with 2 (IV or OOV) units. It was trained by standard stochastic gradient descent (SGD) with momentum.

The true OOV segments in the lectures were labeled by forced alignment using manual transcriptions. Recurrent segments and their feature vectors were extracted by the method described in Section 2. In training, recurrent segments that overlapped the true OOV segments were treated as positive samples, and those did not overlap the true OOV segments were treated as negative samples. In testing, the MLP gave OOV probabilities to recurrent segments, and the segments that have higher OOV probability than a decision threshold were classified as OOV. Note that the segments that were not extracted by recurrent segment discovery were not classified as OOV. The segments that were misclassified into OOV were counted as false alarms, and the true OOV segments that did not overlap segments classified as OOV were counted as misses. The performance was evaluated by the detection error tradeoff (DET) curve, contour of false alarm probabilities and miss probabilities formed when the threshold is varied.

In order to evaluate the effectiveness of DOF, we compared the following two conditions:

- **Baseline**: Classify recurrent segments using only slot-by-slot features described in Section 2.2.
- **Baseline+DOF**: Classify recurrent segments using features with DOF described in Section 2.3.

Moreover, the performance of DOF may be dependent on the number of OOV word repetitions since DOF represents the statistics of multiple features. Thus we compared the detection performance of OOV words repeated 2 or more times and that of OOV words repeated 5 or more times in a lecture. The true OOV segments appearing once in a lecture are ignored (i.e. not classified as OOV and not counted as misses) in "freq $\geq$ 2" condition, and those appearing 4 or less times in a lecture are ignored in "freq $\geq$ 5" condition.

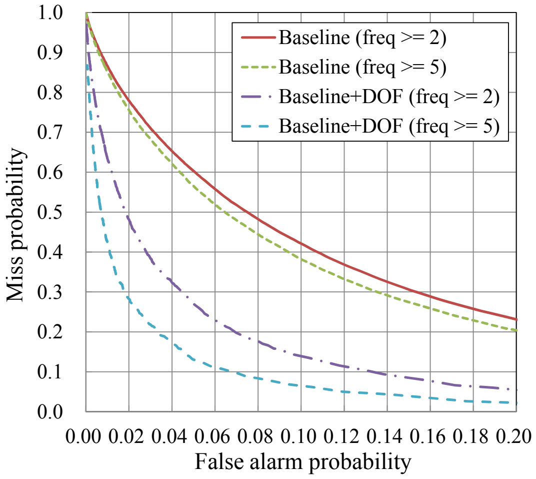

Figure 3: *Detection error tradeoff curves of recurrent OOV word detection with/without DOF.*

3.3. Results

The DET curves of all conditions are shown in Figure 3. The two curves yielded with DOF were below the curves created using only slot-by-slot features. This result confirms that the DOF extracted by the proposed framework dramatically reduces detection errors of recurrent OOV words.

In both "Baseline" and "Baseline+DOF" conditions, detection error rate in "freq. $\geq$ 5" was lower than "freq. $\geq$ 2" condition, but larger improvement was yielded when the DOF was used. This means that our framework effectively utilizes the repeated appearance of OOV words. Although the DOF is still effective to detect OOV words repeated 2 times, it becomes more powerful as the number of OOV word repetitions increases.

When the decision threshold was set to yield the false alarm probability of 5%, the ratio of the number of false alarms in disfluency (filler) segments to the total number of false alarms in "Baseline (freq $\geq$ 2)" and "Baseline+DOF (freq $\geq$ 2)" conditions were 17.2% and 9.4%, respectively. This confirms that the DOF can effectively reduce false alarms due to the vocal irregularities as expected.

4. Conclusion

In this paper we proposed a novel framework to extract effective features for detecting recurrent OOV words in a spoken document, which would normally degrade speech recognizer performance significantly. In order to improve the robustness of OOV word detection by utilizing recurrent OOV words, the proposed method discovers recurrent segments wherein the same word is uttered by using a phoneme recognizer, and uses the means and variances of slot-by-slot features corresponding to the recurrent segments as DOF for IV/OOV classification.

Experiments on CSJ 2701 academic lectures showed that the use of DOF dramatically reduces the detection errors of recurrent OOV words. We also confirmed that our framework effectively reduces false alarms due to disfluencies by utilizing recurrent appearance of OOV words, and the DOF becomes more effective as the number of recurrences of OOV words increases.

5. References

[1] A. Rastrow, A. Sethy and B. Ramabhadran, “A new method for OOV detection using hybrid word/fragment system,” *Proc. ICASSP*, pp. 3953–3956, 2009.

[2] C. Parada, M. Dredze, D. Filimonov and F. Jelinek, “Contextual information improves OOV detection in speech,” *Proc. NAACL*, pp. 216–224, 2010.

[3] A. Marin, T. Kwiatkowski, M. Ostendorf and L. Zettlemoyer, “Using syntactic and confusion network structure for out-of-vocabulary word detection,” *Proc. SLT*, pp. 159–164, 2012.

[4] L. Qin and A. Rudnicky, “Finding recurrent out-of-vocabulary words,” *Proc. INTERSPEECH*, pp. 2242–2246, 2013.

[5] H.K. Kuo, E.E. Kislal, L. Mangu, H. Soltau and T. Beran, “Out-of-vocabulary word detection in a speech-to-speech translation system,” *Proc. ICASSP*, pp. 7158–7162, 2014.

[6] L. Mangu, E. Brill and A. Stolcke, “Finding consensus in speech recognition: word error minimization and other applications of confusion networks,” *Computer Speech and Language*, vol. 14, no. 4, pp. 373–400, 2000.

[7] J. Pinto, I. Szoke, S. Prasanna and H. Hermansky, “Fast approximate spoken term detection from sequence of phonemes,” *Proc. SSCS 2008: Speech search workshop at SIGIR*, pp. 28-33, 2008.

[8] K. Katsurada, S. Sawada, S. Teshima, Y. Iribe and T. Nitta, “Evaluation of fast spoken term detection using a suffix array,” *Proc. INTERSPEECH*, pp. 909–912, 2011.

[9] H. Nishizaki, H. Furuya, S. Natori and Y. Sekiguchi, “Spoken term detection using multiple speech recognizers’ outputs at NTCIR-9 SpokenDoc STD subtask,” *Proc. NTCIR-9 Workshop Meeting*, pp. 236–241, 2011.

[10] K. Maekawa, H. Koiso, S. Furui and H. Isahara, “Spontaneous speech corpus of Japanese,” *Proc. LREC*, pp. 947–952, 2000.

[11] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal and M.C. Hsu, ”PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth,” *Proc. ICDE*, pp. 215–224, 2001.

[12] C. Biemann, “Chinese whispers: An efficient graph clustering algorithm and its application to natural language processing problems,” *Proc. the first workshop on graph based methods for natural language processing*, pp. 73–80, 2006.

[13] A.D. Marcoa and R. Navigli, “Clustering and diversifying web search results with graph-based word sense induction,” *Computational Linguistics*, vol. 39, no. 3, pp. 709–754, 2013.

[14] A. Stolcke, “Entropy-based pruning of backoff language models,” *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, pp. 270–274, 1998.

[15] F. Seide, G. Li, X. Chen and D. Yu, “Feature engineering in context-dependent deep neural networks for conversational speech transcription,” *Proc. ASRU*, pp. 24–29, 2011.

[16] T. Fuchi and S. Takagi, “Japanese morphological analyzer using word co-occurrence -JTAG-,” *Proc. COLING-ACL*, pp. 409–413, 1998.

[17] H. Masataki, D. Shibata, Y. Nakazawa, S. Kobashikawa, A. Ogawa and K. Ohtsuki, “VoiceRex – Spontaneous speech recognition technology for contact-center conversations,” *NTT Technical Review*, vol. 5, no. 1, pp. 22–27, 2007.

[18] T. Hori, C. Hori, Y. Minami and A. Nakamura, “Efficient WFST based one-pass decoding with on-the-fly hypothesis rescoring in extremely large vocabulary continuous speech recognition,” *IEEE Transaction on Audio, Speech and Language Processing*, vol. 15, no. 4, pp. 1352–1365, 2007.