interspeech 2008 incorporating SST 08

Towards Vocabulary-Independent Speech Indexing for Large-Scale Repositories

Jian Shao[1,2], *Roger Peng Yu*[1], *Qingwei Zhao*[2], *Yonghong Yan*[2], *Frank Seide*[1]

[1]Microsoft Research Asia, 5F Beijing Sigma Center, No. 49 Zhichun Rd., Beijing, P.R.C
[2]ThinkIT Speech Lab, Institute of Acoustics, Chinese Academy of Sciences, Beijing, P.R.C
{jshao, qzhao, yyan}@hccl.ioa.ac.cn, {rogeryu, fseide}@microsoft.com

10.21437/Interspeech.2008-563

Abstract

The Out-Of-Vocabulary problem remains a challenge for word-lattice-based speech indexing. Sub-word-based approaches address this problem effectively for small-scale tasks, but suffer from poor precisions on large-scale databases due to lack of strong language model constraints. We propose a method for searching OOV queries with large-scale databases in two steps. First, result candidates are extracted from a sub-word-based system, ensuring a high recall. The candidates are then refined by word-lattice rescoring aiming at a high precision. Experiments on a 160-hours lecture set show that the proposed approach achieves a relative improvement of 8.7% over the sub-word-based baseline, and 19.7% for only single-word queries.

Index Terms: Out-Of-Vocabulary, Keyword Spotting, Word Lattice, Phonetic Lattice, Large Scale

1. Introduction

The tremendous progress in audio compression and storage technologies and the pervasive adoption of the Intra/Internet has fostered a dramatic increase of the use of digital media, such as online lecture videos, archived meetings or conference calls, and voicemails. Search engines to deal with digital audio or video as well as text materials become important.

Typical audio and video for the Internet and enterprise scenario is still a challenge for today's speech-recognition technology, which achieves word accuracies of only 50-70% [1, 2, 3]. To maximize search accuracy, the probabilistic nature of speech recognition must be considered [4]. A significant improvement can be achieved through incorporating word confidence scores and alternative recognition candidates by searching *word lattices* instead of linear speech-to-text output [5, 6, 7, 8]. Word lattices are a compact representation of word candidates and their scores and time information.

However, these approaches do not address the problem of queries which are not in the recognizer's vocabulary. [9] reports that for the SpeechBot system, which indexes audio from public web sites, out-of-vocabulary (OOV) rates on the data are very low (<1.5%), but for the *queries* an OOV rate of 12% is observed.

To address this problem, researches on sub-word-based approaches [10, 11, 12] are reported. In these approaches, spoken contents are transcribed by sub-word-based recognizers. Queries are matched as combination of sub-words. By removing dependence on a pre-defined vocabulary, the OOV problem becomes a non-issue. On small-scale databases (less than 10 hours), where recall is a higher priority in most user scenarios, sub-word-based approaches have shown promising results. E.g., in our previous work [12], phonetic search has been shown to perform as well as word-based search on IN-Vocabulary (INV) queries, and nearly maintain the accuracy for OOV queries. However, the phonetic search system is observed to generate significantly more false alarms and the precision becomes unacceptable with large-scale databases. This is supported by the experiments in one of our preliminary studies below. Hence searching OOV queries with large-scale databases remains a challenge.

In this paper, we present a method for searching OOV queries in two steps. First phonetic search is used to generate initial result candidates. A word-lattice-based rescoring is then used to refine the confidence scores of the candidates. Our experiments show that the proposed method achieves a 8.7% improvement for OOV queries over a phonetic-search baseline, and the improvement is 19.7% with single-word queries.

The rest of this paper is structured as follows. In Section 2, we present preliminary studies to compare word-based and sub-word-bases systems with different database sizes, and to analyze the impact of query n-grams in speech recognition on the search performance. In Section 3, we introduce the proposed two-stage method for searching OOV queries. Section 4 reports experimental results and section 5 concludes.

2. Preliminary Studies

2.1. Phonetic Search vs. Word-Based Search with Different Database Scales

Figure 1 compares phonetic search with word-based search on different database scales. Both systems are tested on three databases of 1.6 hours, 16 hours, and 160 hours respectively. On each database, the same set of (INV) queries are used for both systems (see the detailed setup in section 4.1). When database size grows, both systems show a descending performance. This is because with the Figure Of Merit (FOM) metric, the number of allowed false alarms is fixed (which we believe reflects the user requirement as the number of false alarms users can tolerate does not grow with larger databases). So at the same level of recall, a higher precision is required for larger databases. Figure 1 also shows that phonetic search's performance decreases much more rapidly than word-based search's. It can be expected that, with even larger database, the performance gap between word-based search and phonetic search will become larger. The conclusion we draw from this study is that, to deal with large-scale databases, we really need a well-tuned Large Vocabulary Continuous Speech Recognition (LVCSR) system, which also means that, we need to find a solution for searching

Work performed during the first author's internship at MSR Asia.

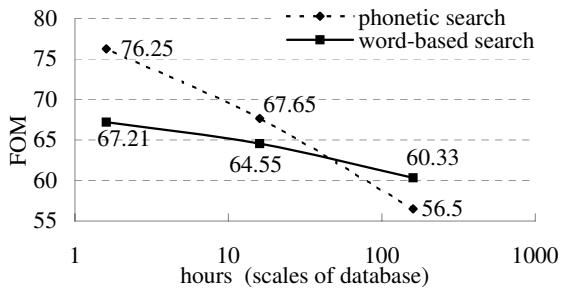

Figure 1: *Phonetic search vs. word-based search with different database scales. Reported are Figure Of Merit (FOM).*

OOV queries with large-scale databases.

2.2. Impact of Query n-grams on Search Accuracy

Suppose that in some cases it is possible to re-run speech recognition when we get an OOV query from users. Though we can easily include the OOV word in the recognition vocabulary, it is difficult to get well-trained n-grams for the OOV word in the language model[1]. In this study, we want to answer the question "how important is the query n-grams for search?" Table 1 shows the experimental results. The baseline LVCSR system uses a trigram language model. We select a set of INV queries from the vocabulary so that they all have well-trained n-grams. Then we keep removing high-order n-grams for the selected queries (n-grams for other words are not touched) step by step.

The results shows that, from 3-gram to 0-gram (which means that the query unigrams are replaced with a constant, calculated as the average unigrams of all words in the vocabulary), the search accuracy degrades from 62.4% to 60.3%, which is still acceptable. The conclusion we draw here is, we do not need well-trained query n-grams to benefit from INV search. It is certainly not possible to re-run speech recognition for each OOV query, but the conclusion is helpful for the method we will present below.

Table 1: *Search accuracy vs. query n-grams. Query words are forced to have only 2-grams/1-grams/0-grams, while all others words still have 3-grams. Reported are FOM. Recall (REC) listed for reference.*

n-gram	FOM	REC
3-gram	62.4	68.1
2-gram	62.5	68.1
1-gram	61.5	67.0
0-gram	60.3	64.8

3. A Two-stage Approach for Searching OOV Queries

Though phonetic search suffers from a poor precision on large-scale databases, it still provides a high recall for OOV queries. At the same time, table 1 shows, the LVCSR-based approach provides a satisfying precision even without well-trained language model scores for query words. Based on these two observations, we propose a two-stage approach: first generate a candidate result list with phonetic search, then "simulate" LVCSR lattices as if running speech recognition with the OOV word in the vocabulary.

[1]By query n-grams we mean all n-grams with the query either in the history or as the predicted word.

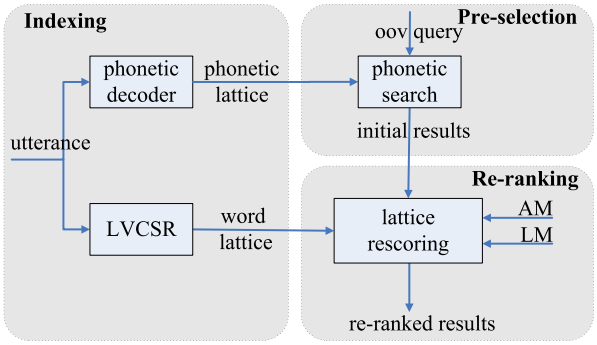

Figure 2: *System diagram.*

3.1. System Diagram

Figure 2 shows the system diagram, which contains three major blocks:

- Indexing: both word-based and phonetic lattices are generated;
- Pre-selection: the OOV query is searched against phonetic lattices to get initial candidates. Each candidate is represented as a 4-tuple (w, t_s, t_e, P), with w being the OOV query, t_s and t_e being the start and end time, and P being the confidence score calculated as the posterior from the phonetic lattice. Only *ntop* candidates are kept for the next step;
- Re-ranking: initial candidates are re-ranked by lattice rescoring as detailed in the next section.

3.2. Candidates Re-ranking By Lattice Rescoring

A word lattice is a representation of the search space of a LVCSR decoding process. Theoretically, there can be an arc between any two time points with any word. The fact that an arc is not present in the lattice normally means such an arc has a bad (acoustic or language model) score and gets pruned by the decoder.

By inserting OOV candidates into word lattices, we want to simulate word lattices generated by running speech recognition with a vocabulary containing the OOV word. The hope is, if the arc will be present in the target lattice, it will get a reasonable score. Otherwise, it is supposed to be pruned and should have a bad score.

In our system, word lattices are generated with cross-word triphones and a trigram language model. A node is denoted as $n = (u[n], v[n], r[n], t[n])$, where $t[n]$ is the time, $(u[n], v[n])$ is the language model history, and $r[n]$ is the right-context phone. An arc is denoted as $a = (S[a], E[a], w[a], p_{ac}[a], p_{lm}[a])$, where $S[a]$ and $E[a]$ are start and end nodes, $w[a]$ is the word hypothesis, $p_{ac}[a]$ and $p_{lm}[a]$) are acoustic and language model scores.

The algorithm to insert a candidate (w, t_s, t_e, P) to a lattice is described below:

- **Arc Insertion**:
  - collect start nodes set N_s (see fig. 3):

* find both left and right closest node to t_s: n_{sl} and n_{sr}. If $t_s - t[n_{sl}] > span$ or $t[n_{sr}] - t_s > span$, remove the candidate;
* collect all nodes in the time interval $(t[n_{sl}] - margin, t[n_{sr}] + margin)$;
* remove nodes whose right-context phone $r[n]$ does not match the first phone of w[2];

– collect end nodes set N_e in similar steps as N_s;

– $\forall n_s \in N_s, \forall n_e \in N_e$:

* calculate acoustic score with corresponding left and right context phones, time period, and the word w;
* add a new arc from n_s to n_e, with the calculated acoustic score and w (language model score will be re-calculated later);
* if the language model history of n_e is not maintained, put n_e to the set N_{expand};

- **Node Expansion**:

– $\forall n \in N_{expand}$, duplicate n with each language model history;

– re-calculate language model scores for all arcs, assuming w being a word in the vocabulary with a constant unigram.

The candidate confidence score is then updated with the word posterior of w against the result lattice. All candidates are re-ranked finally with the updated confidence scores.

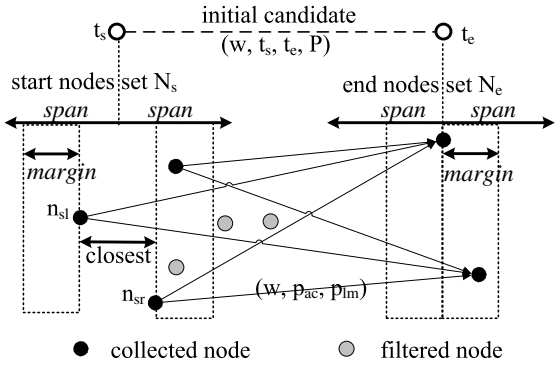

Figure 3: *Parameters in the arc insertion process.*

4. Experiments

4.1. Setup

We evaluate the proposed method on the 160-hour MIT iCampus lectures set [3]. A 18.3-hour subset is selected as our development set for parameter tuning.

Word lattices were generated with a speaker-independent LVCSR system [8]. Its acoustic model was trained on the 1700-hour Switchboard "Fisher" telephony-speech set [2]. Due to limited LM data for lectures, we partitioned the test set into 10 parts, and recognized each part with an LM trained on the transcripts of the remaining 9 parts, keeping training and test disjunct. Word error rate is 45.7%. Phonetic lattices were generated with a phonetic decoder as described in [12] with the same acoustic model. Phoneme error rate is 58.4%.

[2]The right-context constraint is observed to kill too many candidates. In our real implementation, we took the approximation to always keep the node with the largest forward probability, and to keep nodes with the same right-context with this node.

Table 2: *Statistics for selected keywords on development and whole set, with single-word and phrase queries.*

test set	keywords	all	sing.	phra.
whole set	sim. OOV	938	427	511
	real OOV	282	152	130
dev. set	sim. OOV	163	68	95

Table 3: *FOM results on development set for simulated OOV queries. "Queries as INV" used as upper-bound estimation. span=150, marge=20, ntop=40.*

queries	all		single-word		phrase	
methods	FOM	REC	FOM	REC	FOM	REC
queries as INV	71.1	74.0	62.4	68.1	77.8	78.6
phn. search	58.3	70.2	42.7	60.4	70.3	77.8
+ re-ranking	63.5	70.2	50.3	60.4	73.8	77.8
rel. impr.	8.9	-	17.8	-	5.0	-

An automatic procedure as described in [12] is used to select query keywords, this process generates both INV and OOV queries[3]. Besides the real OOV queries, we "simulate" a set of OOV queries from the INV queries by excluding those queries from the vocabulary. Single-word queries are directly removed from the vocabulary, while for phrase queries, the word constitute with lowest unigram is removed. The purpose of using the simulated OOV queries is that the INV query performance provides an upper-bound estimation. Table 2 lists statistics for the selected keywords.

Search accuracies are reported in Figure of Merit (FOM), which is defined by National Institute of Science and Technology (NIST) as the detection/false-alarm curve averaged over range of [0...10] false alarms per keyword per h hour. Instead of the original h=1, we use h=data set duration. Lattice recalls (recall of all query matches within lattice, which in an upper bound of FOM) are listed as well for analysis purpose.

4.2. Results on Development Set

Table 3 shows experiments on the development set for simulated OOV queries. The first line is evaluated when the queries are in the vocabulary, as an upper-bound estimation. The second line is the phonetic search baseline, and the third line shows the proposed lattice rescoring. The proposed method gains a relative improvement of 8.9% over the phonetic search baseline. The improvement mainly comes from single-word queries. The reason is that phrase queries are typically longer, and precisions suffer less with the phonetic search baseline.

Figure 4 and 5 show the impact of *span* and *margin*. FOM keeps improving by raising *span* and *margin*, as the tolerance increased between time boundaries from phonetic lattices and word lattices. Although larger *span*s and *margin*s may have better performance, we stop doing that as the time cost of rescoring arcs becomes too expansive.

Figure 6 shows the effect of *ntop* in candidate selection. Raising *ntop* increases recall, which improves FOM in the ini-

[3]A phrase is OOV if one of its constitute is OOV.

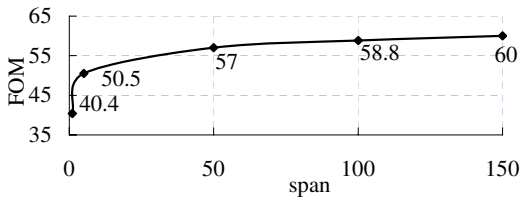

Figure 4: *FOM vs. span on dev. set. margin=1, ntop=40.*

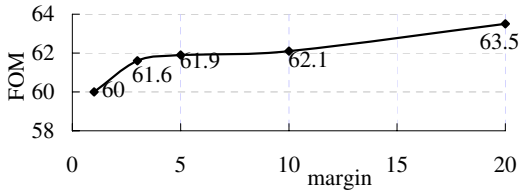

Figure 5: *FOM vs. margin on dev. set. span=150, ntop=40.*

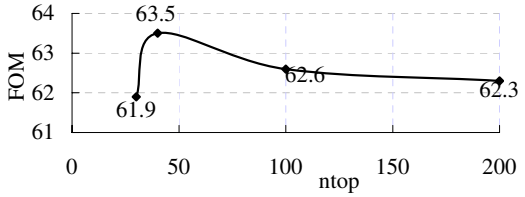

Figure 6: *FOM vs. ntop on dev. set. span=150, margin=20.*

tial stage. After *ntop*=40, there are too many false alarms included in the candidate list, which makes FOM go down slowly.

4.3. Results on the Whole 160-hour Set

Table 4 shows results on the whole 160-hour set. The first block is for simulated OOV queries. The proposed method improves FOM from a phonetic baseline at 43.3% to 50.4% with a 16.4% relative improvement. The improvement is larger than on the development set. This is because the phonetic search suffer more on a larger database from the low precision. As on the development set, single-word queries are observed to have larger improvement (26.2%). Compared with the INV upper-bound (FOM 57.7), we have reduced the performance gap by half.

The second block shows results for real OOV queries. The proposed method improves FOM from a phonetic baseline at 40.2% to 43.7% with an 8.7% relative improvement, while on single-word queries, the improvement is 19.7%.

5. Conclusions

We presented a two-stage approach for search OOV queries with large-scale databases. Phonetic search is used to pre-select result candidates for OOV queries, and the candidates are then inserted to LVCSR lattices and rescored to get updated confidence estimation. Experiments on a 160-hour lecture set reduced the gap between OOV and INV queries by half on a set of simulated OOV queries. With real OOV queries, a relative 8.7% improvement is achieved over phonetic search baseline, and with single-word queries, the improvement becomes 19.7%.

The work is still an initial study towards solving the OOV problem. The performance gap between INV and OOV queries is still not completely reduced, and further work needs to be done on speed up the search process to make it realistic for a large-scale search scenario.

Table 4: *FOM results on the whole 160-hour set for both simulated and real OOV queries. "Queries as INV" used as upper-bound estimation. span=150, marge=20, ntop=40.*

queries	all		single-word		phrase	
methods	FOM	REC	FOM	REC	FOM	REC
simulated OOV queries						
queries as INV	57.7	69.9	54.2	72.4	63.7	65.5
phn. search	43.3	59.1	31.7	50.3	63.2	74.3
+re-ranking	50.4	58.2	40.0	49.7	68.3	72.8
rel. impr.	16.4	-	26.2	-	8.1	-
real OOV queries						
phn. search	40.2	52.8	30.4	45.8	51.6	61.0
+re-ranking	43.7	53.0	36.4	45.8	52.3	61.0
rel. impr.	8.7	-	19.7	-	1.4	-

6. References

[1] M. Padmanabhan, G. Saon, J. Huang, B. Kingsbury, and L. Mangu, Automatic Speech Recognition Performance on a Voicemail Transcription Task, IEEE Trans. on Speech and Audio Processing, Vol. 10, No. 7, 2002.

[2] G. Evermann, H. Y. Chan, M. J. F. Gales, B. Jia, X. Liu, D. Mrva, K. C. Sim, L. Wang, P. C. Woodland, K. Yu, Development of the 2004 CU-HTK English CTS Systems Using More Than Two Thousand Hours of Data, *Proc. Fall 2004 Rich Transcription Workshop (RT-04), 2004*.

[3] J. Glass, T. J. Hazen, L. Hetherington, C. Wang, Analysis and Processing of Lecture Audio Data: Preliminary Investigation, *Proc. HLTNAACL'2004 Wordkshop: Interdisciplinary Approches to Speech Indexing and Retrieval*, Boston, 2004.

[4] P. Yu, K. J. Chen, C. Y. Ma, F. Seide, Vocabulary-Indexpendent Indexing of Spontaneous Speech, IEEE Trans. on Speech and Audio Processing, Vol. 13, No. 5, 2005.

[5] M. Saraclar, R. Sproat, Lattice-based search for spoken utterance retrieval, *Proc. HLT'2004*, Boston, 2004.

[6] C. Chelba and A. Acero, Positin Specific Posterior Lattices for Indexing Speech, *Proc. ACL'2005*, Ann Arbor, 2005.

[7] Z. Y. Zhou, P. Yu, C. Chelba, F. Seide, Towards Spoken-Document Retrieval for the Internet: Lattice Indexing for Large-Scale Web-Search Architecture, *Proc. HLT'06*, New York, 2006.

[8] F. Seide, P. Yu and Y. Shi, Towards Spoken-Document Retrieval for the Enterprise: Approximate Word-Lattice Indexing with Text Indexers",*Proc. ASRU'2007*, Kyoto, Japan, 2007.

[9] B. Logan, P. Moreno, J. M. Van Thong, and E. Whittacker, An Experimental Study of an Audio Indexing System for the Web, *Proc. ICSLP'2000*, Beijing, China, 2000.

[10] K. Ng, Subword-Based Approaches for Spoken Document Retrieval, Ph. D thesis, MIT, 2000.

[11] B. Logan, P. Moreno, and O. Deshmukh, Word and Sub-word Indexing Approaches for Reducing the Effects of OOV Queries on Spoken Audio, *Proc. HLT'2002*, 2002.

[12] F. Seide, P. Yu, et al., Vocabulary-Independent Search in Spontaneous Speech, *Proc. ICASSP'04*, Montreal, 2004.