EFFECT OF PRONUNCIATIONS ON OOV QUERIES IN SPOKEN TERM DETECTION

[1]*Dogan Can,*[2]*Erica Cooper,*[3]*Abhinav Sethy,*[3]*Bhuvana Ramabhadran ,*[1]*Murat Saraclar,*[4]*Christopher M. White*

[1] Bogazici University, [2] Massachusetts Institute of Technology,
[3] IBM, [4] HLT Center of Excellence, Johns Hopkins University

ABSTRACT

This paper focusses on the effect of pronunciations for Out-of-Vocabulary (OOV) query terms on the performance of a spoken term detection (STD) task. OOV terms, typically proper names or foreign language terms occur infrequently but are rich in information. The STD task returns relevant segments of speech that contain one or more of these OOV query terms. The STD system described in this paper indexes word-level and subword level lattices produced by an LVCSR system using Weighted Finite State Transducers (WFST). Experiments comparing pronunciations using n-best variations from letter-to-sound rules, morphing pronunciations using phone confusions for the OOV terms and indexing one-best transcripts, lattices and confusion networks are presented. The following observations are worth mentioning: phone indexes generated from subwords represented OOVs well, too many variants for the OOV terms degrades performance if pronunciations are not weighted.

Index Terms— Speech recognition, Speech indexing and retrieval, Weighted Finite State Transducers

1. INTRODUCTION

The rapidly increasing amount of spoken data calls for solutions to index and search this data. Spoken term detection (STD) is a key information retrieval technology which aims open vocabulary search over large collections of spoken documents. The major challenge faced by STD is the lack of reliable transcriptions, an issue that becomes even more pronounced with heterogeneous, multilingual archives. Considering the fact that most STD queries consist of rare named entities or foreign words, retrieval performance is highly dependent on the recognition errors. In this context, lattice indexing provides a means of reducing the effect of recognition errors by incorporating alternative transcriptions in a probabilistic framework.

The classical approach consists of converting the speech to word transcripts using large vocabulary continuous speech recognition (LVCSR) tools and extending classical Information Retrieval (IR) techniques to word transcripts. However, a significant drawback of such an approach is that search on queries containing out-of-vocabulary (OOV) terms will not return any result. These words are replaced in the output transcript by alternatives that are probable, given the acoustic and language models of the ASR. It has been experimentally observed that over 10% of user queries can contain OOV terms [1], as queries often relate to named entities that typically have a poor coverage in the ASR vocabulary. The effects of OOV query terms in spoken data retrieval are discussed in [2]. In many applications, the OOV rate may get worse over time unless the recognizer's vocabulary is periodically updated. An approach for solving the OOV issue consists of converting the speech to phonetic transcripts and representing the query as a sequence of phones. Such transcripts can be generated by expanding the word transcripts into phones using the pronunciation dictionary of the ASR system. Another way would be to use subword (phones, syllables, or word-fragments) based language models. The retrieval is based on searching the sequence of subwords representing the query in the subword transcripts. Some of these works were done in the framework of the NIST TREC Spoken Document Retrieval tracks in the 1990s and are described by [3]. Popular approaches are based on search on subword decoding [4, 5, 6, 7, 8] or search on the subword representation of word decoding enhanced with phone confusion probabilities and approximate similarity measures for search [9].

Other research works have tackled the OOV issue by using the IR technique of query expansion. In classical text IR, query expansion is based on expanding the query by adding additional words using techniques like relevance feedback, finding synonyms of query terms, finding all of the various morphological forms of the query terms and fixing spelling errors. Phonetic query expansion has been used by [Li00] for Chinese spoken document retrieval on syllable-based transcripts using syllable-syllable confusions from the ASR.

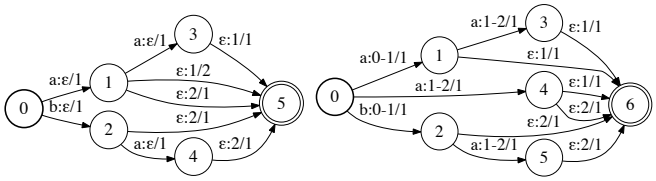
The rest of the paper is organized as follows. In Section 2 we explain the methods used for spoken term detection. These include the indexing and search framework based on WFSTs, formation of phonetic queries using letter to sound models, and expansion of queries to reflect phonetic confusions. In Section 3 we describe our experimental setup and present the results. Finally, in Section 4 we summarize our contributions.

2. METHODS

2.1. WFST-based Spoken Term Detection

General indexation of weighted automata provides an efficient means of indexing speech utterances based on the within utterance expected counts of substrings (factors) seen in the data [10, 6]. In the most basic form, mentioned algorithm leads to an index represented as a weighted finite state transducer (WFST) where each substring (factor) leads to a successful path over the input labels for each utterance that particular substring was observed. Output labels of these paths carry the utterance ids, while path weights give the within utterance expected counts. The index is optimized by weighted transducer determinization and minimization [11] so that the search complexity is linear in the sum of the query length and the number of indices the query appears. Figure 1.a illustrates the utterance index structure in the case of single-best transcriptions for a simple database consisting of two strings: "a a" and "b a". Explained construction is ideal for the task of utterance retrieval where the expected count of a query term within a particular utterance is of primary importance. In the case of STD, this construction is still

This work was partially done during the 2008 Johns Hopkins Summer Workshop. The authors would like to thank the rest of the workshop group, in particular Martin Jansche, Sanjeev Khudanpur, Michael Riley, and James Baker.



(a) Utterance Index (b) Modified Utterance Index

Fig. 1. Index structures

useful as the first step of a two stage retrieval mechanism [12] where the retrieved utterances are further searched or aligned to determine the exact locations of queries since the index provides the utterance information only. One complication of this setup is that each time a query term occurs within an utterance, it will contribute to the expected count within that particular utterance and the contribution of distinct instances will be lost. Here we should clarify what we refer to by an occurrence and an instance. In the context of lattices where arcs carry recognition unit labels, an occurrence corresponds to any path comprising of the query labels, an instance corresponds to all such paths with overlapping time-alignments. Since the index provides neither the individual contribution of each instance to the expected count nor the number of instances, both of these parameters have to be estimated in the second stage which in turn compromises the overall detection performance.

To overcome some of the drawbacks of the two-pass retrieval strategy, a modified utterance index which carries the time-alignment information of substrings in the output labels was created. Figure 1.b illustrates the modified utterance index structure derived from the time-aligned version of the same simple database: "a_{0-1} a_{1-2}" and "b_{0-1} a_{1-2}". In the new scheme, preprocessing of the time alignment information is crucial since every distinct alignment will lead to another index entry which means substrings with slightly off time-alignments will be separately indexed. Note that this is a concern only if we are indexing lattices, consensus networks or single-best transcriptions do not have such a problem by construction. Also note that no preprocessing was required for the utterance index, even in the case of lattices, since all occurrences in an utterance were identical from the indexing point of view (they were in the same utterance). To alleviate the time-alignment issue, the new setup clusters the occurrences of a substring within an utterance into distinct instances prior to indexing. Desired behavior is achieved via assigning the same time-alignment information to all occurrences of an instance.

Main advantage of the modified index is that it distributes the total expected count among instances, thus the hits can now be ranked based on their posterior probability scores. To be more precise, assume we have a path in the modified index with a particular substring on the input labels. Weight of this path corresponds to the posterior probability of that substring given the lattice and the time interval indicated by the path output labels. The modified utterance index provides posterior probabilities compared to expected counts provided by the utterance index. Furthermore, second stage of the previous setup is no longer required since the new index already provides all the information we need for an actual hit: the utterance id, begin time and duration. Eliminating second stage significantly improves the search time since time-alignment of utterances takes much more time compared to retrieving them. On the other hand, embedding time-alignment information leads to a much larger index since common paths among different utterances are largely reduced by the mismatch between time-alignments which in turn compromises the effectiveness of the weighted automata optimization. To smooth this effect out, time-alignments are quantized to a certain extent during preprocessing without altering the final performance of the STD system.

Searching for a user query is a simple weighted transducer composition operation [11] where the query is represented as a finite state acceptor and composed with the index from the input side. The query automaton may include multiple paths allowing for a more general search, i.e. searching for different pronunciations of a query word. The WFST obtained after composition is projected to its output labels and ranked by the shortest path algorithm to produce results [11]. In effect, we obtain results with decreasing posterior scores.

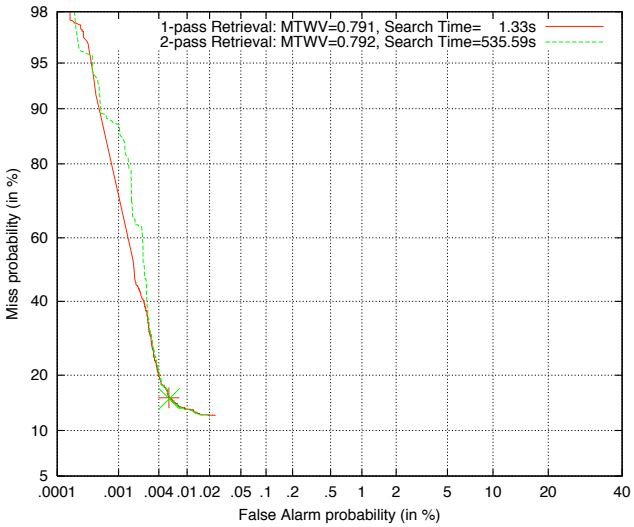


Fig. 2. Comparison of 1-pass & 2-pass strategies in terms of retrieval performance and runtime

Figure 2 compares the proposed system with the 2-pass retrieval system on the `stddev06` data-set in a setup where `dryrun06` query-set, word-level ASR lattices and word-level indexes are utilized. As far as Detection Error Tradeoff (DET) curves are concerned, there is no significant difference between the two methods. However, proposed method has a much shorter search time, a natural result of eliminating time-costly second pass.

2.2. Query Forming and Expansion for Phonetic Search

When using a phonetic index, the textual representation of a query needs to be converted into a phone sequence or more generally a WFST representing the pronunciation of the query. For OOV queries, this conversion is achieved using a letter-to-sound (L2S) system. In this study, we use n-gram models over (letter, phone) pairs as the L2S system, where the pairs are obtained after an alignment step. Instead of simply taking the most likely output of the L2S system, we investigate using multiple pronunciations for each query. Assume we are searching for a letter string l with the corresponding phone-strings set $\Pi_n(l)$: n-best L2S pronunciations. Then the posterior probability of finding l in lattice L within time interval T can be written as

$$P(l|L,T) = \sum_{p \in \Pi_n(l)} \tilde{P}(l|p) P(p|L,T)$$

where $P(p|L,T)$ is the posterior score supplied by the modified utterance index and $\tilde{P}(l|p)$ is the posterior probability derived from L2S scores.

Composing an OOV query term with the L2S model returns a huge number of pronunciations of which unlikely ones are removed prior to search to prevent them from boosting the false alarm rates. To obtain the conditional probabilities $\tilde{P}(l|p)$, we perform a normalization operation on the retained pronunciations which can be expressed as

$$\tilde{P}(l|p) = \frac{P^{\alpha}(l,p)}{\sum_{\pi \in \Pi_n(l)} P^{\alpha}(l,\pi)}$$

where $P(l,p)$ is the joint score supplied by the L2S model and α is a scaling parameter. Most of the time, retained pronunciations are such that a few dominate the rest in terms of likelihood scores, a situation which becomes even more pronounced as the query length increases. Thus, selecting $\alpha = 1$ to use raw L2S scores leads to problems since most of the time best pronunciation takes almost all of the posterior probability leaving the rest out of the picture. The quick and dirty solution is to remove pronunciation scores instead of scaling them. This corresponds to selecting $\alpha = 0$ which assigns the same posterior probability $\tilde{P}(l|p)$ to all pronunciations: $\tilde{P}(l|p) = 1/|\Pi_n(l)|$, for each $p \in \Pi_n(l)$. Although simple, this method is likely to boost false alarm rates since it does not make any distinction among pronunciations. The challenge is to find a good query-adaptive scaling parameter which will dampen the large scale difference among L2S scores. In our experiments we selected $\alpha = 1/|l|$ which scales the log likelihood scores by dividing them with the "length of the letter string". This way, pronunciations for longer queries are effected more than those for shorter ones. Another possibility is to select $\alpha = 1/|p|$, which does the same with the "length of the phone string". Section 3.2.2 presents a comparison between removing pronunciation scores and scaling them with our method.

Similar to obtaining multiple pronunciations from the L2S system, the queries can be extended to similar sounding ones by taking phone confusion statistics into account. In this approach, the output of the L2S system is mapped to confusable phone sequences using a sound-to-sound (S2S) WFST. The S2S WFST is built using the same technique which was used for generating the L2S WFST. For the case of the S2S transducer both the input and output alphabet are phones and the parameters of the phone-phone pair model were trained using alignments between the reference and decoded output of the RT-04 Eval set.

3. EXPERIMENTS

3.1. Experimental Setup

Our goal was to address pronunciation validation using speech for OOVs in a variety of applications (recognition, retrieval, synthesis) for a variety of types of OOVs (names, places, rare/foreign words). To this end we selected speech from English broadcast news (BN) and 1290 OOVs. The OOVs were selected with a minimum of 5 of acoustic instances per word, and common English words were filtered out to obtain meaningful OOVs (e.g. NATALIE, PUTIN, QAEDA, HOLLOWAY), excluding short (less than 4 phones) queries. Once selected, these were removed from the recognizer's vocabulary and all speech utterances containing these words were removed from training.

The LVCSR system was built using the IBM Speech Recognition Toolkit [13] with acoustic models trained on 300 hours of HUB4 data with utterances containing OOV words excluded. The excluded utterances (around 100 hours) were used as the test set for WER and STD experiments. The language model for the LVCSR system was trained on 400M words from various text sources. The LVCSR system's WER on a standard BN test set RT04 was 19.4%. This system was also used for lattice generation for indexing for OOV queries in the STD task.

3.2. Results

The baseline experiments were conducted using the reference pronunciations for the query terms, which we refer to as *reflex*. The L2S system was trained using the reference pronunciations of the words in the vocabulary of the LVCSR system. This system was then used to generate multiple pronunciations for the OOV query words. Further variations on the query term pronunciations were obtained by applying a phone confusion *S2S* transducer to the *L2S* pronunciations.

3.2.1. Baseline - Reflex

For the baseline experiments, we used the reference pronunciations to search for the queries in various indexes. The indexes were obtained from word and subword (fragment) based LVCSR systems. The output of the LVCSR systems were in the form of 1-best transcripts, consensus networks, and lattices. The results are presented in Table 1. Best performance is obtained using subword lattices converted into a phonetic index.

Table 1. Reflex Results

Data	P(FA)	P(Miss)	ATWV
Word 1-best	.00001	.770	.215
Word Consensus Nets	.00002	.687	.294
Word Lattices	.00002	.657	.322
Fragment 1-best	.00001	.680	.306
Fragment Consensus Nets	.00003	.584	.390
Fragment Lattices	.00003	.485	.484

3.2.2. L2S

For the L2S experiments, we investigated varying the number of pronunciations for each query for two scenarios and different indexes. The first scenario considered each pronunciation equally likely (unweighted queries) whereas the second made use of the L2S probabilities properly normalized (weighted queries). The results are presented in Figure 3 and summarized in Table 2. For the unweighted case the performance peaks at 3 pronunciations per query. Using weighted queries improves the performance over the unweighted case. Furthermore, adding more pronunciations does not degrade the performance. Best results are comparable to the reflex results.

The DET plot for weighted L2S pronunciations using indexes obtained from fragment lattices is presented in Figure 4. The single dots indicate MTWV (using a single global threshold) and ATWV (using term specific thresholds [14]) points.

3.2.3. S2S

For the S2S experiments, we investigated expanding the 1-best output of the L2S system. In order to mimic common usage we used indexes obtained from 1-best word and subword hypotheses converted to phonetic transcripts. As shown in Table 3 a slight improvement was obtained when using a trigram S2S system representing the

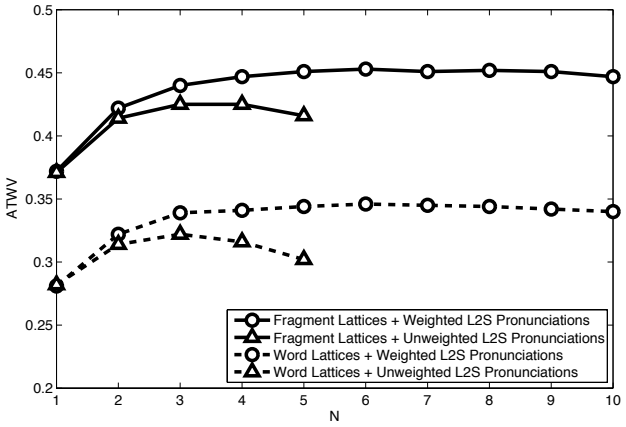


Fig. 3. ATWV vs N-best L2S Pronunciations

Table 2. Best Performing N-best L2S Pronunciations

Data	L2S Model	# Best	P(FA)	P(Miss)	ATWV
Word 1-best	Baseline	1	.00001	.796	.190
	Weighted	6	.00004	.730	.233
Word Lattices	Baseline	1	.00002	.698	.281
	Unweighted	3	.00005	.625	.322
	Weighted	6	.00005	.606	.346
Fragment 1-best	Baseline	1	.00001	.757	.229
	Weighted	10	.00005	.662	.286
Fragment Lattices	Baseline	1	.00003	.597	.372
	Unweighted	3	.00006	.512	.425
	Weighted	6	.00006	.487	.453

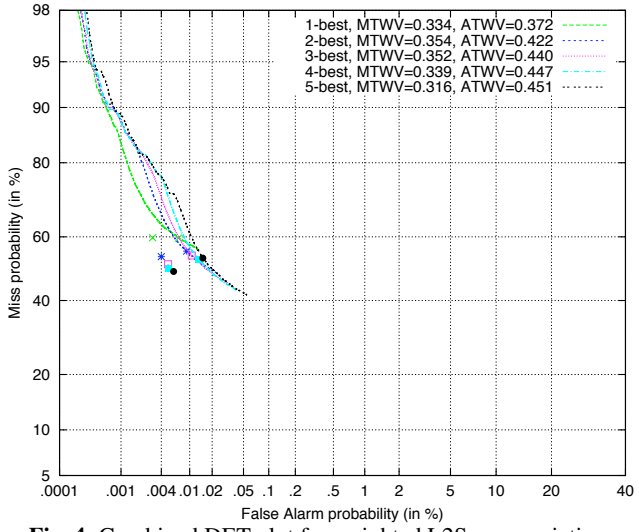


Fig. 4. Combined DET plot for weighted L2S pronunciations

phonetic confusions. These results were obtained using unweighted queries and using weighted queries may improve the results.

4. CONCLUSION

Phone indexes generated from subwords represent OOVs better than phone indexes generated from words. Modeling phonetic confusions yields slight improvements. Using multiple pronunciations obtained from L2S system improves the performance, particularly when the alternatives are properly weighted.

Table 3. S2S N-best Pronunciations expanding L2S output

Lattices	# Best	P(FA)	P(Miss)	ATWV
Words	1	.00002	.795	.190
	2	.00002	.785	.192
	3	.00003	.778	.193
	4	.00004	.775	.189
	5	.00004	.771	.185
Fragments	1	.00002	.757	.228
	2	.00002	.748	.230
	3	.00003	.742	.229
	4	.00004	.738	.227
	5	.00004	.736	.221

5. REFERENCES

[1] B. Logan, P. Moreno, J. V. Thong, and E. Whittaker, "Confusion-based query expansion for oov words in spoken document retreival," in *Proc. ICSLP*, 2002.

[2] P. Woodland, S. Johnson, P. Jourlin, and K. S. Jones, "Effects of out of vocabulary words in spoken document retreival," in *Proc. of ACM SIGIR*, 2000.

[3] J. S. Garofolo, C. G. P. Auzanne, and E. M. Voorhees, "The trec spoken document retrieval track: A success story," in *Proc. of TREC-9*, 2000.

[4] M. Clements, S. Robertson, and M. S. Miller, "Phonetic searching applied to on-line distance learning modules," in *Proc. of IEEE Digital Signal Processing Workshop*, 2002.

[5] F. Seide, P. Yu, C. Ma, and E. Chang, "Vocabulary-independent search in spontaneous speech," in *Proc. of ICASSP*, 2004.

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

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

[8] J. Mamou, B. Ramabhadran, and O. Siohan, "Vocabulary independent spoken term detection," in *Proc. of ACM SIGIR*, 2007.

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

[10] C. Allauzen, M. Mohri, and M. Saraclar, "General-indexation of weighted automata-application to spoken utterance retrieval," in *Proc. HLT-NAACL*, 2004.

[11] M. Mohri, F. C. N. Pereira, and M. Riley, "Weighted automata in text and speech processing," in *Proc. ECAI, Workshop on Extended Finite State Models of Language*, 1996.

[12] S. Parlak and M. Saraclar, "Spoken term detection for turkish broadcast news," in *Proc. ICASSP*, 2008.

[13] H. Soltau, B. Kingsbury, L. Mangu, D. Povey, G. Saon, and G. Zweig, "The ibm 2004 conversational telephony system for rich transcription," in *Proc. ICASSP*, 2005.

[14] D. R. H. Miller, M. Kleber, C. Kao, O. Kimball, T. Colthurst, S. A. Lowe, R. M. Schwartz, and H. Gish, "Rapid and Accurate Spoken Term Detection," in *Proc. Interspeech*, 2007.