EFFECT OF PRONUNCIATIONS ON OOV QUERIES IN SPOKEN TERM DETECTION

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

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

ABSTRACT

The spoken term detection (STD) task aims to return relevant segments from a spoken archive that contain the query terms whether or not they are in the system vocabulary. This paper focuses on pronunciation modeling for Out-of-Vocabulary (OOV) terms which frequently occur in STD queries. The STD system described in this paper indexes word-level and sub-word level lattices or confusion networks produced by an LVCSR system using Weighted Finite State Transducers (WFST). We investigate the inclusion of n-best pronunciation variants for OOV terms (obtained from letter-to-sound rules) into the search and present the results obtained by indexing confusion networks as well as lattices. The following observations are worth mentioning: phone indexes generated from sub-words represent OOVs well and too many variants for the OOV terms degrade performance if pronunciations are not weighted.

Index Terms— Speech Recognition, Speech Indexing and Retrieval, Spoken Term Detection, 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 many STD queries consist of rare named entities and 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 STD 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 is to use sub-word (phone, syllable, or word-fragment) based language models. The retrieval is based on searching the sequence of sub-words representing the query in the sub-word transcripts. During 1990s NIST TREC Spoken Document Retrieval tracks fostered speech retrieval research as described in [3]. Popular approaches are: search on sub-word decoding [4, 5, 6] or search on the sub-word representation of word decoding enhanced with phone confusion probabilities and approximate similarity measures for search [7].

OOV issue was also tackled by 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 for Chinese spoken document retrieval on syllable-based transcripts using syllable-syllable confusions from the ASR [8].

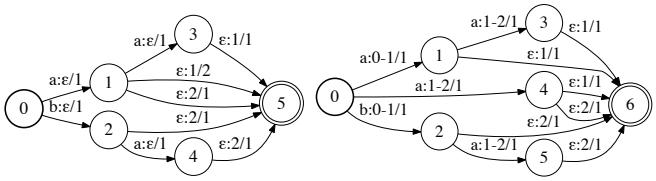
The rest of the paper is organized as follows. In Section 2 we explain the methods used for spoken term detection. These include the indexing and search framework based on WFSTs, formation of phonetic queries using letter to sound models, and expansion of queries to reflect phonetic confusions. In Section 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 [9, 4]. In the most basic form, this algorithm leads to an index represented as a weighted finite state transducer (WFST) where each substring 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 [10] 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". Utterance index 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 useful as the first step of

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. This work was supported in part by Bogazici University (BU) Research Fund and BU Foundation. Dogan Can was supported by TUBITAK BIDEB.



(a) Utterance Index (b) Modified Utterance Index

Fig. 1. Index structures

a two stage retrieval mechanism [11] where the retrieved utterances are further searched or aligned to determine the exact locations of queries since the index provides the utterance information only. Prominent complication of this setup is that each time a term occurs within an utterance, it will contribute to the overall 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 whose labels comprise of the query terms, 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 decreases the detection performance.

To overcome some 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 instead of 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 the second stage significantly improves the search time since time-alignment of utterances takes more time compared to their retrieval. On the other hand, embedding time-alignment information leads to a larger index since common paths among different utterances are largely reduced by the mismatch between time-alignments which in turn degrades 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 STD performance.

Modified utterance index structure can also be utilized to represent Position Specific Posterior Lattices (PSPL) [12] when the output labels carry position information instead of time. Also, when confusion networks are utilized instead of lattices, resulting index is similar to what is obtained with Time-based Merging for Indexing (TMI) [13] algorithm. In our setup confusion networks group alternative hypotheses that fall into the same time slot and lead to paths that do not exist in the original lattice, similar to the case of TMI.

Searching for a user query is a simple weighted transducer composition operation [10] 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 [10] to produce results. In effect, we obtain results with decreasing posterior scores.

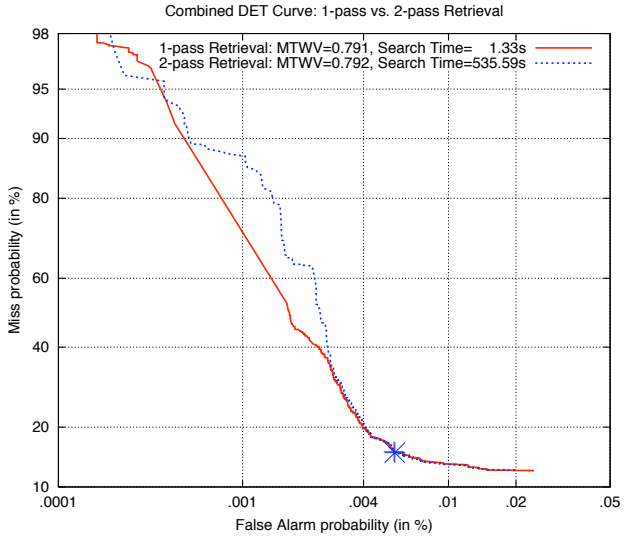


Fig. 2. Comparison of 1 & 2-pass strategies in terms of retrieval performance (Maximum Term Weighted Value - MTWV [14]) and runtime.

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 the 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 that we are searching for a letter string l with the corresponding phone-strings $\{p\}$ belonging to the set $\Pi_n(l)$: the 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 the 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 query pronunciations can be extended 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, which is built by the same technique used for generating the *L2S* WFST. For the case of the *S2S* transducer both the input and the output alphabets are phones, and the parameters of the phone-phone pair model were trained using alignments between the reference and the decoded output of the RT-04 Eval set.

3. EXPERIMENTS

3.1. Experimental Setup

In the workshop, our goal was to address OOV pronunciation validation using speech 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 acoustic instances and 4 phones per word, and common English words were filtered out to obtain meaningful queries (e.g. NATALIE, PUTIN, QAEDA, HOLLOWAY). Once selected, these queries were removed from the recognizer's vocabulary and all speech utterances containing them were removed from training.

The LVCSR system was built using the IBM Speech Recognition Toolkit [15] 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 ASR 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 generating lattices used by the OpenFST [16] based STD system from Bogazici University.

WFST-based STD indexer used in this work encodes input and output labels [10] before WFST optimization since OpenFST Library supports the determinization of functional transducers only. Even though this situation compromises the search efficiency, it improves indexing time since few common paths exist before the final optimization step detailed in [9].

3.2. Results

The gold standard 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 best *L2S* pronunciation. However experiments with single-best transcriptions utilizing the composition of *S2S* and *L2S* did not yield any significant improvement over those utilizing *L2S*.

3.2.1. Gold Standard - Reflex

For the reflex experiments, we used the reference pronunciations to search for OOV queries in various indexes. The indexes were obtained from word and sub-word (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 (in terms of Actual Term Weighted Value - ATWV [14]) is obtained using sub-word 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

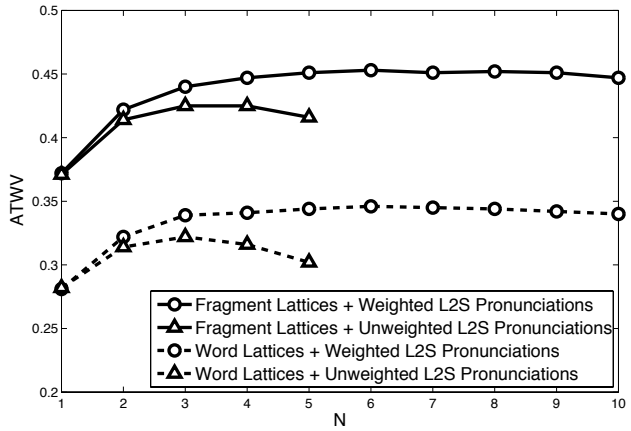


Fig. 3. ATWV vs N-best *L2S* Pronunciations

case. Furthermore, adding more pronunciations does not degrade the performance. Best results are comparable to the reflex results.

Table 2. Best Performing N-best *L2S* Pronunciations

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

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 [17]) points.

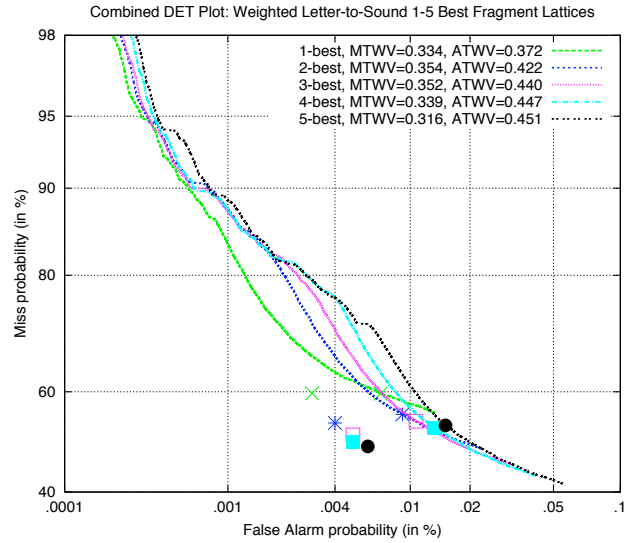


Fig. 4. Combined DET plot for weighted *L2S* pronunciations

4. CONCLUSION

Phone indexes generated from sub-words represent OOVs better than those generated from words. Using multiple pronunciations obtained from *L2S* system improves the performance, particularly when the alternatives are properly weighted. Modeling phonetic confusions does not yield significant improvements.

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

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

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

[7] 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.

[8] Y. C. Li, W. K. Lo, H. M. Meng, and P. C. Ching., "Query expansion using phonetic confusions for chinese spoken document retrieval," in *Proc. of IRAL*, 2000.

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

[10] 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.

[11] S. Parlak and M. Saraclar, "Spoken term detection for Turkish Broadcast News," in *Proc. ICASSP*, 2008.

[12] C. Chelba and A. Acero, "Position specific posterior lattices for indexing speech," in *Proc. of ACL*, 2005.

[13] Z. Y. Zhou, P. Yu, C. Chelba, and F. Seide, "Towards spoken-document retrieval for the internet: lattice indexing for large-scale web-search architectures," in *Proc. of HLT-NAACL*, 2006.

[14] NIST, "The spoken term detection (STD) 2006 evaluation plan. http://www.nist.gov/speech/tests/std/," 2006.

[15] 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.

[16] C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri, "OpenFst: a general and efficient weighted finite-state transducer library," in *Proc. of CIAA*, 2007.

[17] 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.