Web Derived Pronunciations for Spoken Term Detection

Dogan Can* Boğaziçi University
Erica Cooper MIT
Arnab Ghoshal Johns Hopkins University
Martin Jansche Google, Inc.
Sanjeev Khudanpur Johns Hopkins University
Bhuvana Ramabhadran IBM T.J. Watson Research
Michael Riley Google, Inc.
Murat Saraclar Boğaziçi University
Abhinav Sethy IBM T.J. Watson Research
Morgan Ulinski Cornell University
Christopher White Johns Hopkins University

ABSTRACT

Indexing and retrieval of speech content in various forms such as broadcast news, customer care data and on-line media has gained a lot of interest for a wide range of applications, from customer analytics to on-line media search. For most retrieval applications, the speech content is typically first converted to a lexical or phonetic representation using automatic speech recognition (ASR). The first step in searching through indexes built on these representations is the generation of pronunciations for named entities and foreign language query terms. This paper summarizes the results of the work conducted during the 2008 JHU Summer Workshop by the Multilingual Spoken Term Detection team, on mining the web for pronunciations and analyzing their impact on spoken term detection. We will first present methods to use the vast amount of pronunciation information available on the Web, in the form of IPA and ad-hoc transcriptions. We describe techniques for extracting candidate pronunciations from Web pages and associating them with orthographic words, filtering out poorly extracted pronunciations, normalizing IPA pronunciations to better conform to a common transcription standard, and generating phonemic representations from ad-hoc transcriptions. We then present an analysis of the effectiveness of using these pronunciations to represent Out-Of-Vocabulary (OOV) query terms on the performance of a spoken term detection (STD) system. We will provide comparisons of Web pronunciations against automated techniques for pronunciation generation as well as pronunciations generated by human experts. Our results cover a range of speech indexes based on lattices, confusion networks and one-best transcriptions at both word and word fragments levels.

*Authors listed in alphabetical order

SIGIR'09, July 19–23, 2009, Boston, Massachusetts, USA.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms: Algorithms, Experimentation, Performance

Keywords: Web Pronunciation extraction, Pronunciation normalization, Spoken Term Detection, Open vocabulary

1. INTRODUCTION

The rapidly increasing amount of spoken data generated in a wide range of applications, such as market intelligence gathering, customer analytics, and on-line media search, calls for solutions to index and search this data. Spoken term detection (STD) is a key information retrieval technology which aims for open vocabulary search over large collections of spoken documents.

The classical approach in STD consists of converting speech to word transcripts using ASR and extending classical Information Retrieval (IR) techniques to word transcripts. However, 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 [22]. An approach for solving the OOV issue consists of converting the speech content to phonetic transcripts and representing queries as sequence of phones. Such transcripts can be generated by expanding the word transcripts using the pronunciation dictionary of the ASR system or by the use of subword based ASR systems. 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 1990's and are described by [10]. Popular approaches are based on search on subword decoding [6, 18, 17, 19, 12] or search on the subword representation of word decoding enhanced with phone confusion probabilities and approximate similarity measures for search [5].

The ability to correctly identify the pronunciation of a word is critical for the performance of STD systems Previous approaches have either employed trained people to manually generate pronunciations, or have used letter-to-phoneme (L2P) rules, which were either hand-crafted or machine-learned from a manually transcribed corpus [9, 8]. The first approach is expensive; the second can be of variable quality, depending on the skill of the experts or size and quality of the transcribed data. We investigate a novel strategy of mining the huge quantities of pronunciation information on the Web and thus use the web as a vast but noisy corpus for pronunciations.

Two kinds of pronunciations are common on the Web: The first is expressed in the International Phonetic Alphabet (IPA), for example 'Lorraine Albright /ɔl braɪt/'. IPA pronunciations use special symbols, such as 'ɔ', which can unambiguously denote a particular English phoneme. However, there are no universally accepted conventions for transcribing pronunciations in IPA, and the use of IPA requires some skill. It is then not surprising that we find considerable variation in IPA strings captured on the Web and there is a need to normalize them to follow a common set of conventions.

The second, and more frequent, kind of pronunciation use an ad-hoc transcription based on a simpler or less ambiguous spelling than standard English orthography. For example, when we see 'bruschetta (pronounced broo-SKET-uh)', the intended pronunciation is more intuitively represented by the letters 'SKET' than it is by 'schet'. Ad-hoc transcriptions follow the rules of English orthography and do not require any specialized skills. However, they do not provide a phonemic transcription, so one of our tasks is to predict phonemes from a combination of the standard orthography and ad-hoc transcription of a word.

Processing IPA and ad-hoc transcriptions proceeds in three major phases. In the extraction phase (Sec. 2) we find a candidate pronunciation and its corresponding orthographic form on a web page. In the second phase, extraction validation (Sec. 3), we determine if an orthography/pronunciation pair was correctly extracted. For example, most instances of 'pronounced dead' do not correspond pronunciations that we would like to keep. In the final normalization phase (Sec. 4), we canonicalize irregularities in the IPA pronunciations and map the ad-hoc pronunciations to their phonemic form.

To measure the effectiveness of the web as a pronunciation corpus, we will present extensive experiments comparing web pronunciations with automatically generated pronunciations as well as pronunciations generated by human experts. We also provide comparisons of the performance of these pronunciations on STD systems built on a range of index types including lattices, confusion networks and one-best transcriptions at both word and word fragments levels.

2. PRONUNCIATION EXTRACTION

The extraction, validation and normalization steps used in this paper require letter-to-phoneme, letter-to-letter, or phoneme-to-phoneme models. Methods for constructing such models include those based on decision trees [4], pronunciation-by-analogy [13], and hidden Markov models [21]. We chose to use n-gram models over pairs [7, 3].

Our pronunciations are extracted from Google's web and news page repositories. The pages are restricted to those that Google has classified as in English and from non-EU countries. The extraction of IPA and of ad-hoc pronunciations uses different techniques.

2.1 IPA Pronunciation Extraction

The Unicode representation of most English words in IPA requires characters outside the ASCII range. For instance, only 3.8% to 8.6% (depending on transcription conventions) of the words in the 100K Pronlex dictionary[1] have completely ASCII-representable IPA pronunciations (e.g., 'beet' `/bit/`). Most of the non-ASCII characters are drawn from the Unicode IPA extension range (0250–02AF), which are easily identified on web pages. Our candidate IPA pronunciations consist of web terms[2] that are composed entirely of legal English IPA characters, that have at least one non-ASCII character[3], and that are delimited by a pair of forward slashes ('/ . . . /'), back slashes ('\ . . . \'), or square brackets ('[. . .]').

Once these candidate IPA pronunciations are identified, the corresponding orthographic terms are next sought. To do so, an English phoneme-to-letter (L2P) model, $\Pr[\lambda|\pi]$, is used to estimate the probability that an orthographic string λ corresponds to the given phonemic string π.

L2P models are bootstrapped from a seed pronunciation lexicon as follows. Each orthographic-phonemic training pair is first aligned, e.g. (w, w) (i, i) (m, m) (b, b) (–, ə) (l, l) (e, –) (d, d) (o, ə) (n, n). Alignments are derived by training a unigram model of (letter, phoneme) pairs (including letter deletions and phoneme insertions) using EM from a flat start and subsequently finding the most likely sequence of pairs under the model. Each aligned (letter, phoneme) pair is then treated as a single token for a Kneser-Ney n-gram model [11]. Once built, the n-gram model is represented as a weighted finite-state transducer (FST), mapping letters to phonemes, using the OpenFst Library [1], which allows easy implementation of the operations that follow. Note that this results in a *joint* model $\Pr_u[\lambda, \pi]$.

For pronunciation extraction, a unigram letter-phoneme model $\Pr_u[\lambda, \pi]$ is trained on the Pronlex dictionary using the method described above. We use $n = 1$ both to ensure wide generalization and to make it likely that the subsequent results do not depend greatly on the bootstrap English dictionary. With this model in hand, we extract that contiguous sequence of terms λ, among the nearby[4] terms preceding each candidate pronunciation π, which maximizes $\Pr[\lambda|\pi] = \Pr_u[\lambda, \pi]/\Sigma_\lambda \Pr_u[\lambda, \pi]$. We found 2.53M candidate orthographic and phonemic string pairs (309K unique pairs) in this way. These are then passed to extraction validation in Sec. 3.

2.2 Ad-hoc Pronunciation Extraction

Ad-hoc pronunciations are identified by matches to the regular expressions indicated in Table 1. To find the corresponding conventionally-spelled terms, we use an English letter-to-letter (L2L) model $\Pr[\lambda_2|\lambda_1]$ to estimate the probability that the conventionally-spelled string λ_2 corresponds to a given ad-hoc pronunciation string λ_1. Assuming that λ_1 and λ_2 are independent given their true underlying phone-

[1]CALLHOME American English Lexicon, LDC97L20.

[2]By *terms* we mean tokens exclusive of punctuation and HTML markup.

[3]This, unfortunately, excludes pronunciations in SAMPA, ARPABet, etc.

[4]We used an empirically-determined twenty term window.

Type	Pattern	Count
paren	`\(pronounced (as |like )?([^)]+)\)`	3415K
quote	`pronounced (as |like )?"([^"]+)"`	835K
comma	`, pronounced (as |like )?([^,]+),`	267K

Table 1: Ad-hoc pronunciation extraction patterns and counts.

mic pronunciation π, $\Pr[\lambda_2 \,|\, \lambda_1] = \sum_\pi \Pr[\lambda_2 \,|\, \pi] \Pr[\pi \,|\, \lambda_1]$ (implemented by weighted FST composition). The unigram model $\Pr_u[\lambda, \pi]$ of Sec. 2.1 is used to derive the estimates

$$\Pr[\lambda_2|\pi] = \Pr_u[\lambda_2, \pi] / \Sigma_\lambda \Pr_u[\lambda, \pi]$$
$$\Pr[\pi|\lambda_1] = \Pr_u[\lambda_1, \pi] / \Sigma_\pi \Pr_u[\lambda_1, \pi]$$

We then extract that contiguous sequence of terms λ_2, among the nearby[5] terms preceding each candidate pronunciation λ_1, which maximizes $\Pr[\lambda_2|\lambda_1]$. We found 4.52M candidate orthographic and "pronunciation" pairs (568K unique pairs) with pair counts for specific patterns indicated in Table 1. These pairs are then passed to extraction validation described in the next section.

3. PRONUNCIATION EXTRACTION VALIDATION

Once extraction has taken place, a validation step is applied to check whether the items extracted are correct in the sense that they find each orthographic term and the corresponding pronunciation provided word-for-word. We manually labeled 667 randomly selected (orthography, IPA pronunciation) pairs and 1000 (orthography, ad-hoc pronunciation) pairs for correctness of extraction. We used this data to build and test classifiers for winnowing the extracted pronuniciations.

Sixteen features of the IPA pronunciations and 57 features of the ad-hoc pronunciations are computed for this data. Features shared among both types of pronunciations include the string length of the extracted orthography and pronunciation, the distance between them, the presence of certain substrings (e.g. spaces, function words, non-alphabetic characters), and the log probabilities assigned by the L2P/L2L alignment models used during extraction. In the IPA case, the extracted IPA pronunciation is aligned with a predicted pronunciation – predicted from the extracted orthography by a 5-gram model trained on Pronlex – and per-phoneme alignment features are computed. These include the fraction of mismatched consonants and vowels, since we noticed that vowel mismatches are common in good extractions but consonant mismatches are highly indicative of bad extractions. In the ad-hoc case, additional features include letter-to-letter log probabilities of unigram, bigram and trigram letter-pair models, counts of insertions and deletions in the best alignment, and capitalization styles, which often signal bad extractions.

SVM classifiers were constructed separately for the IPA and ad-hoc pronunciation data using these features. Five-fold cross validation on the 667/1000 labeled examples was used to produce the precision-recall curves in Fig. 1, parameterized by the SVM-margin. In particular, the IPA extraction classifier has a precision of 96.2% when the recall was 88.2%, while the ad-hoc classifier has a precision of 98.1% when the recall was 87.5% (indicated by dots in Fig. 1).

[5]We used an empirically-determined eight term window.

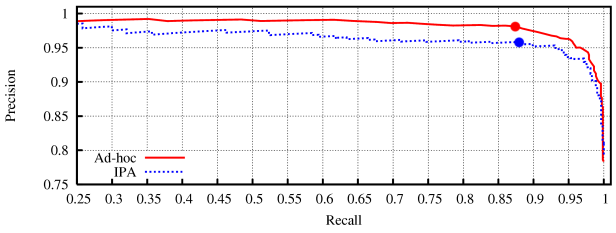


Figure 1: Precision vs. recall in pronunciation extraction validation.

To summarize, our extraction consists of a simple first-pass extraction step, suitable for efficiently analyzing a large number of web pages, followed by a more comprehensive validation step that has high precision with good recall. Given this high recall and the fact that most extraction errors, in our error analysis of a subsample, have no correct alternatives on the given page, we feel confident about this two-step approach.

4. PRONUNCIATION NORMALIZATION

Up to this point, we have extracted millions of candidate IPA and ad-hoc pronunciations from the Web with high precision. We refer to the collection of extracted and validated data as the *Web-IPA lexicon* and the *ad-hoc lexicon*. The Web-IPA lexicon is based on extractions from websites that use idiosyncratic conventions (see below), while the ad-hoc pronunciations are still in an orthographic form. In both cases, they need to be *normalized* to a standard phonemic form to be useful for many applications.

The training and test data used for the pronunciation normalization experiments are based on the subset of words common to both the web-derived data and Pronlex.We use a 97K word subset of the Web-IPA lexicon for these experiments, which has 30K words in common with Pronlex, with an average of 1.07 Pronlex and 1.87 Web-IPA pronunciations per word. In Sec. 4.1 we consider subsets of this dataset where only data from a single website was used. The training data for ad-hoc normalization was augmented by words whose pronunciations could be assembled from hyphenated portions of the ad-hoc transcription (e.g. if the ad-hoc transcription of 'Circe' is 'Sir-see', we look up the Pronlex phonemic transcriptions of 'sir' and 'see').

We evaluate pronunciations by aligning a predicted phoneme string with a reference and computing the phoneme error rate (PhER) – analogous to word error rate in automatic speech recognition – as the number of insertions, deletions, and substitutions divided by the number of phonemes in the reference (times 100%). In cases of multiple predicted[6] or reference pronunciations, the pair with the lowest PhER is chosen.

4.1 IPA Pronunciation Normalization

We first compare the quality of the Web-IPA lexicon derived from a website with Pronlex, by performing 5-fold cross-validation experiments on their orthographic intersection. For each cross-validation run, two L2P models are trained on the same 80% subset of the intersection – one using the Pronlex pronunciations, and the other using the

[6]Multiple predicted pronunciations for an orthographic form result only if it was *extracted* with multiple distinct IPA (or ad-hoc) pronunciations.

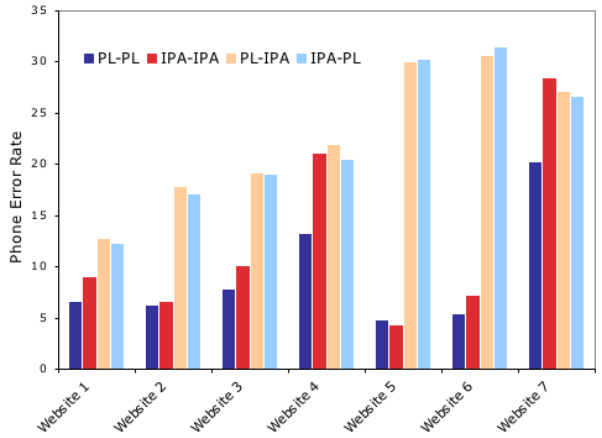


Figure 2: Cross-validation results by website.

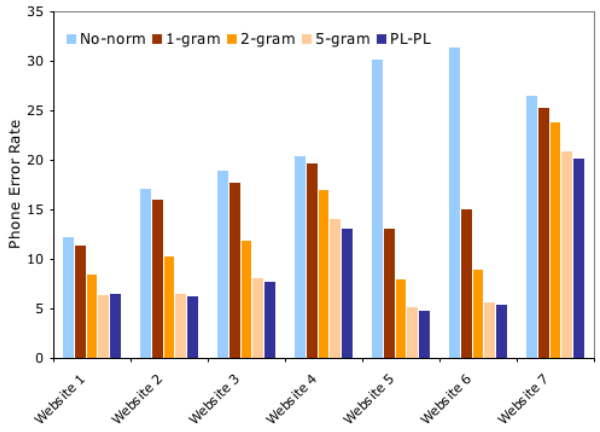


Figure 3: L2P models trained on normalized data are better at predicting ref. pronunciations.

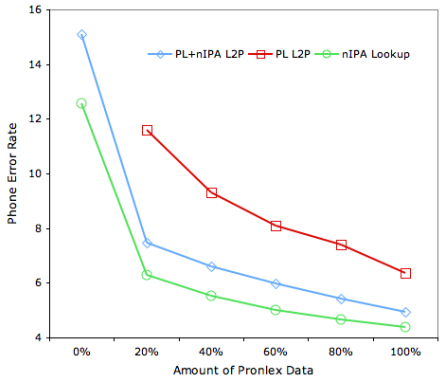


Figure 4: Performance on rare words – normalized web pronunciations help a lot on rare words.

Web-IPA pronunciations. Each model is used to generate candidate pronunciations for the words in the 20% subset left out of training, which are then scored against their pronunciations from both lexica, giving us four PhER numbers.

Figure 2 shows these four PhER numbers for 7 of the 10 websites with the most extracted pronunciations. We notice that for several websites, L2P models trained using the Web-IPA are almost as good at predicting the Web-IPA pronunciations (red bars) as a model trained on Pronlex is at predicting Pronlex pronunciations (dark blue bars). However, in all cases, models trained on web data are poor predictors of Pronlex data (light blue bars), and vice versa (light tan bars).

These experiments demonstrate that websites vary in the quality of pronunciations available from them. Also, the Web-IPA pronunciations are different from those found in Pronlex. The differences can be caused both by improper use of IPA symbols, as well as other site-specific conventions. For instance, 'graduate' is pronounced as either /gɹædʒuɪt/ or /gɹædʒueɪt/ in Pronlex, but appears as /gɹadjʊeɪt/, /gɹædjʊɪt/, /gɹædjuət/, /gɹædjuɪt/, and /gɹædʒʊət/ among the ten most frequent websites.

Site-specific normalization The considerable variability of pronunciations across websites strongly motivates the need for a site-specific normalization of the pronunciations to a more site-neutral target form. Here we use Pronlex as our target. Using the Pronlex and Web-IPA pronunciation pairs in the orthographic intersection, we train a website-specific phoneme-to-phoneme (P2P) transduction model, which takes a pronunciation obtained from the website and converts it to a Pronlex-like form.[7]

The L2P models are then trained on these normalized pronunciations. The PhER results of these L2P models are presented in Fig. 3 for P2P transducers of varying n-gram orders. As one can clearly see, normalization helps to improve the quality of the pronunciations obtained from the web. In fact, normalized pronunciations generated by a 5-gram model (light tan bars) have a PhER that's comparable to the pronunciations predicted by a model trained on Pronlex (dark blue bars). Based on this, we conclude that *L2P models trained on normalized Web-IPA pronunciations are as good as models trained on comparable amounts of Pronlex.*

Performance on rare words To test performance on rare words, we remove from Pronlex any word with a frequency of less than 2 in the Broadcast News (BN) corpus. Among these rare words, the ones that are found in the extracted Web-IPA lexicon form our test set (about 3.8K words). To estimate the size of hand-crafted lexicon (Pronlex in the present case) needed for normalization, we subdivide the frequent (BN frequency > 2) Pronlex words, into 5 subsets based on decreasing frequency, using respectively 20%, 40%, 60%, 80%, and 100% of the most frequent words.

For each of the subsets of Pronlex, we generate candidate pronunciations for the words in our rare-word test set using each of the following three methods: (1) an L2P model trained on the subset of Pronlex; (2) pronunciations looked up from the normalized Web-IPA lexicon; and (3) an L2P model trained on the normalized Web-IPA and the Pronlex subset combined together.

Fig. 4 shows the PhER on the rare words using these three methods. The normalized Web-IPA data clearly produces better pronunciations for rare words. Note that the Web-IPA, normalized using only 20% of the hand-crafted Pronlex dictionary (roughly 10K most frequent words), produces pronunciations that are as good as those generated by an L2P model trained on the whole of Pronlex.

[7]The P2P transducer is identical to the L2P models described earlier, but trained on aligned (phoneme, phoneme) pairs, instead of (letter, phoneme) pairs.

4.2 Ad-hoc Pronunciation Normalization

For ad-hoc normalization our task is to predict phonemic transcriptions from the extracted ad-hoc transcriptions, which are in orthographic form, but presumably reveal the intended pronunciation of a word more easily than the standard orthography. We investigated four ways of predicting phonemes from the extracted (orthography, ad-hoc transcription) pairs: (1) Apply a letter-to-phoneme model to the standard orthography (a competitive baseline). (2) Apply a letter-to-phoneme model to the ad-hoc transcription. (3) Model the phonemes as the latent source in a noisy channel model with independent channels for the orthography and ad-hoc transcription. (4) Train a language model on aligned (orthography, ad-hoc, phoneme) triples and apply it to the orthography and ad-hoc transcription.

We evaluate the predicted phoneme strings on a test set with 256 words that are associated with extracted ad-hoc and phonemic pronunciations manually transcribed to correspond both to the orthographic and ad-hoc forms. This yielded a total of 1181 phonemes.

For (1) we trained a 5-gram L2P model on a subset of Pronlex from which the vocabulary was removed, achieving 29.5% PhER. Next (2) we trained a 5-gram L2P model on the 43K word training dictionary described earlier. This ignores the orthography and predicts phonemes directly from ad-hoc transcription, giving 20.5% PhER.

By contrast, the remaining two models use both the orthography and ad-hoc transcription to predict the phonemes. Model (3) is the noisy channel model given by

$$\Pr[\lambda_1, \lambda_2, \pi] = \Pr[\lambda_1 \mid \pi] \Pr[\lambda_2 \mid \pi] \Pr[\pi]$$

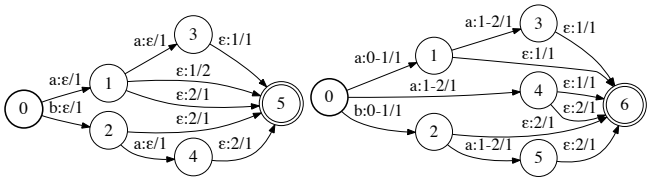
which generates a latent pronunciation π and, conditional on π, generates the orthography λ_1 and ad-hoc transcription λ_2 independently. It can be implemented straightforwardly in terms of the joint and conditional transducer models discussed in Sec. 2. This achieves 19.4% PhER. The last model (4) drops the independence assumption. It is a 5-gram language model on (orthography, ad-hoc, phoneme) triples, trained on the 43K lexicon by first aligning ad-hoc transcriptions with phonemes and then aligning the orthography with the already aligned (ad-hoc, phoneme) pairs. During testing the model is first combined with the orthography to predict (ad-hoc, phoneme) pairs, and those are further combined with the observed ad-hoc transcription to predict the phonemes. This model achieves 18.8% PhER – a 36% relative error rate reduction over the baseline model (1). We conclude that ad-hoc pronunciations – alone or in combination with the standard orthography – are extremely useful for predicting the pronunciations of unseen rare words.

5. SPOKEN TERM DETECTION

In the previous sections of the paper, we described a strategy for mining the web for pronunciations expressed either as IPA or ad-hoc transcriptions. We discussed various methods to validate the extraction and normalize the extracted pronunciations. The remainder of the paper will provide a comparative analysis of these pronunciations against automatically generated pronunciations and human generated pronunciations for the task of detecting OOV query terms in a spoken term detection system. We will first provide a brief description of our FST based spoken term detection system and the experimental setup.

General indexation of weighted automata [2] provides an efficient means of speech indexing used for spoken utterance retrieval (SUR) [17] or spoken term detection (STD) [16]. In STD, the goal is to determine the time interval containing the query. Here, each occurrence of the query has to be detected separately. Therefore, retrieval is based on the posterior probability of substrings (factors) in a given time interval. In this work, the weighted automata to be indexed are the preprocessed lattice outputs of the ASR system. The input labels are phones, the output labels are quantized time-intervals and the weights are normalized negative log probabilities. The index is represented as a weighted finite state transducer where each substring (factor) leads to a successful path over the input labels whenever that particular substring was observed. Output labels of these paths carry the time interval information followed by the utterance IDs. The path weights give the probability of each factor occurring in the specific time interval of that utterance.

Figure 5.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". Figure 5.b illustrates the time interval 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}".



(a) Utterance Index (b) Time Interval Index

Figure 5: Index structures

Prior to indexing, each ASR output lattice is preprocessed to obtain the weighted automata. 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. First, distinct clusters are formed using non overlapping occurrences of each word (or sub-word). Then, the other occurrences are assigned to the cluster with which they maximally overlap. Each cluster has a time interval that is the union of all its members. Finally, the time intervals are adaptively quantized to an acceptable resolution. The input labels are also preprocessed by converting the words (or subwords) into phone sequences. The weights are normalized to form a probability distribution over the paths in the lattice.

Similar to the FST index construction algorithm presented in [2], each FST in the database is processed separately and the union of the resulting FSTs is optimized to yield the index FST. For each FST representing an utterance, the factors to be indexed are selected, utterance IDs are introduced as a final output label and the resulting FST is optimized for efficiency. Factor selection introduces a new start state, a new final state, new transitions from the new start state to each state, and new transitions from each state to the new final state. The weights of these new transitions are computed according to the probability distribution defined by the lattice. The optimization step makes use of weighted transducer determinization and minimization [14]. The OpenFst Library [1] is used for the construction of the index FST as well as for retrieval.

Searching for a user query is a simple weighted transducer composition operation [14] where the query is represented as a finite state acceptor and composed with the index from the input side. For a string query and a deterministic index FST, the search complexity is linear in the sum of the query length and the number of times the query appears in the archive. The query automaton may include multiple paths allowing for a more general search, i.e. searching for different pronunciations of a query word. The FST obtained after composition is projected to its output labels and ranked by the shortest path algorithm to produce results [14]. In the end, we obtain results with decreasing posterior scores.

5.1 Data and Experimental Setup

The objective of this section is to validate and analyze the impact of web extracted pronunciations on retrieval of OOVs in a spoken-term detection system. A variety of OOV types, comprising of names, places, and rare/foreign words adding to a total of 1290 OOV terms were carefully selected. We also selected speech data from English broadcast news (BN) comprising of broadcasts from channels such as, CNN, ABC, and NPR [8] [20]. These OOVs were selected with a minimum of 5 acoustic instances per word to ensure that metrics used to evaluate the performance of the system were not biased. In addition, common, frequently occuring English words were filtered out to obtain meaningful OOV terms such as, NATALIE, PUTIN, QAEDA, HOLLOWAY. Short terms, defined as words with pronunciations that contain less than four phonetic units were excluded. Once the OOV terms and speech data were selected, these terms were removed from the recognizer's vocabulary and all speech utterances containing these terms were also removed from the training data used to train the Large Vocabulary Speech Recognition System (LVCSR).

The LVCSR system was built using the IBM Speech Recognition Toolkit [20]. The acoustic models were trained on 300 hours of BN data. The excluded utterances containing the OOV terms (100 hours) were used as the test set for spoken-term detection. The language models for the LVCSR system was trained on 400M words from various text sources [20]. The performance of this LVCSR system measured in terms of its WER on a standard BN test set (RT04) was 19.4%. This performance although not the very best, renders the system state-of-the-art for the purpose of these experiments. This LVCSR system was used to produce lattices and consensus networks on the test set at word and sub-word levels, subsequently indexed using the weighted finite state transducer WFST-based spoken-term detection system described in Section 5. The LVCSR decoding dictionary was used for the transduction of ASR output symbols (words and sub-word units) to phonemes. The next section presents results when different pronunciations for the OOV terms were used during retrieval.

5.2 Results

To set an expectation of what the best performance could approach, we used the reference pronunciations (*reflex*) for the OOV terms to search through the 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 2. Three different metrics were used to evaluate the performance, False Alarms, Misses and Actual Term-Weighted Value (ATWV) defined below. The ATWV metric was used to evaluate STD systems in the NIST 2006 STD Evaluation.

$$\mathrm{ATWV}(\theta) = 1 - \frac{1}{Q}\sum_{k=1}^{Q}\{P_{miss}(q_k,\theta) + \beta.P_{FA}(q_k,\theta)\} \quad (1)$$

$$\mathrm{P_{miss}}(q) = 1 - \frac{C(q)}{R(q)} \qquad \mathrm{P_{FA}(q)} = \frac{A(q) - C(q)}{T_{speech} - C(q)} \quad (2)$$

where:
$R(q_k)$: Number of occurences of query q_k,
$A(q_k)$: Total no. of retrieved documents for query q_k,
$C(q_k)$: No. of correctly retrieved documents for query q_k,
β: Weight assigned to false alarms that is proportional to the prior probability of occurence of a specific term and its cost-value ratio.

The best performance that trades false alarms and misses is obtained using subword (fragment) lattices that were subsequently converted to phonetic lattices and indexed. The richness of the lattices, coupled with the fragment representation of the audio contribute to this performance.

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

Table 2: Baseline Results

The next set of experiments uses pronunciations derived from the web with the methods described in Sections 2, 3 and 4.2. One can derive a good sense of the coverage of these OOV terms in the web-derived prounciations from Table 5.

Table 5 illustrates the number of OOV queries that could be represented by pronunciations from the web after various filtering and cleaning processes. We could recover adhoc pronunciations for around 25% more words than IPA based pronunciations. We manually converted 50% of the adhoc pronunciations to a phonetic representation to get an upper bound on the impact of the adhoc pronunciation conversion process described in Section 2.2. For query terms that did not have pronunciations available from any of the web extraction methods, we backed off to the ones generated by a letter-to-phone system [7, 3]. This letter-to-sound system was trained using the *reflex* pronuniations.

The various pronunciations that we explored are:

- Pronunciations obtained from a letter-to-sound system trained on the gold standard (reflex) and n-best variations (L2P 00)
- Ad-hoc pronunciations from the web (Webpron 03)
- Cleaned and normalized IPA pronunciations obtained by mining the web (Webpron 04, 05)
- Hand-cleaned ad-hoc pronunciations (Webpron 06)
- Best Pronunciation used by an LVCSR system
- Union of pronunciations (can have multiple pronunciations for any word from different methods)

The baseline pronunciations for the OOV terms referred to as L2P in the figures is derived from a letter-to-phone

[8]This is a commonly used data set in evaluations that benchmark technologies.

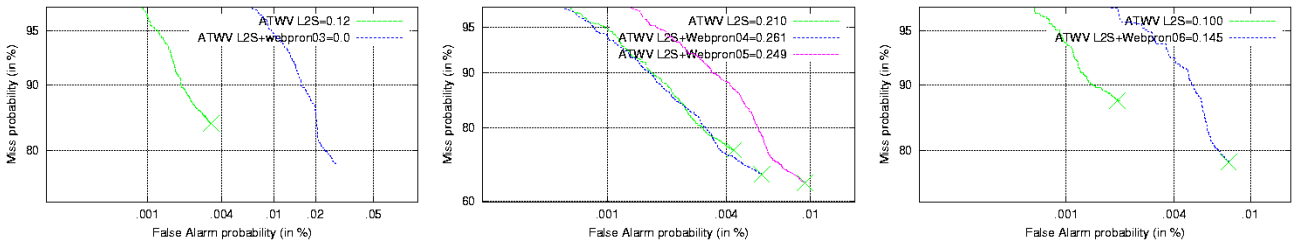


Figure 6: Results on pairwise union of pronunciations

Example	Pronunciations		Ref/Corr/FA/Miss	
	L2P	Web Based	L2P	Web Based
ALBRIGHT	ae l b r ay t	ao l b r ay t	276/0/1/276	276/254/20/22
GREENSPAN	g r iy n s p aa n	g r iy n s p ae n	157/0/0/157	157/85/0/72
SHIMON	sh ih m ax n	sh ih m ow n	109/0/0/109	109/98/12/11

Table 3: Examples of L2P v/s Web-based pronunciations with positive impact

system. Figure 6 plots the DET curves for different pairwise union of pronunciations from the set of OOVs. The subplots in the figure are not directly comparable with each other as each method of deriving pronunciations from the web results in pronunciations for a subset of the total number of OOV terms. Therefore there is a different baseline for each of the subplots in Figure 6. In general, the use of web-based pronunciations seems to improve the overall ATWV measure when compared to the baseline performance(L2P). The *X* mark in the DET curves refers to the False Alarm and Misses tradeoff point at the which the ATWV measure is the highest (Equal Error Rate is another commonly used metric in place of ATWV). When *Webpron 03* is used to derive pronunciations, the false alarms increase, alluding to the fact that there are too many confusion-causing pronunciations for the OOVs in this set, resulting in too many incorrect hits. On the other hand, *Webpron 05*, which are pronunciations that have been cleaned and normalized, beats the baseline with fewer false alarms and misses. It is interesting to note that *Webpron 06*, despite being manually converted from adhoc pronunciations, does not have fewer false alarms than the baseline although the ATWV score is higher. In all our experiments β was fixed at 1000 [15] to produce results consistent with the literature. However, varying β, and in effect varying the manner in which false alarms are weighted relative to misses, might indicate a different impact of web-derived pronunciations. Also, in observing the actual values of ATWV, one can observe that this value is much less than what is reported in the literature for the BN task, where the query terms comprise both in-vocaulary and OOV terms. This is because our query terms are all OOVs and there are no in-vocabulary query terms used in our evaluation.

Table 3 shows examples where the use of web-based pronunciations worked well for certain OOV queries and Table 4 illustrates examples where web-based pronunciations had a negative impact. In cases where the use of web-based pronunciations had a positive impact, the gains were obtained from far fewer misses (*ALBRIGHT* and *SHIMON*). In cases where the use of these pronunciations have a negative impact, it is usually due to the presence of several false alarms, sometimes resulting from the query term being a part of a bigger word or having too many variants. This can partly be attributed to the fact that on the web, especially for the case of ad-hoc pronunciations, people tend to put 'pronunciations by example,' using a common word as a base for the pronunciation of a rare word. For example in Table 4 the pronunciation of the word *FREUND* is based on the word *FRIEND* leading to a large number of false alarms since the word *FRIEND* occurs frequently in the speech index. Similarly, for the case of the word *THIERRY*, which has a large number of false alarms, the pronunciation extracted from the web was based on the pronunciation of *-TARY*, which is a common English suffix (*MILITARY, VOLUNTARY*). Figure 7 shows that the best performance was obtained when a union of all the web-derived and letter-to-sound system based pronunciations was used in retrieval.

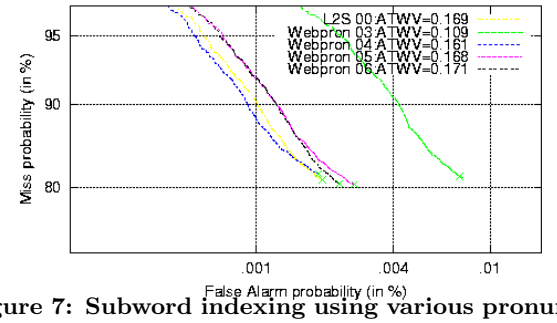


Figure 7: Subword indexing using various pronunciations

We also explored the effect of using the LVCSR system as a means to select the the best pronunciation for an OOV term, i.e. the LVCSR system has all variants in its lexicon and picks one based on either a likelihood metric (Forced-align) or during unsupervised decoding (unsup-reco) of speech containing the OOV words in question. The results are presented in Figure 8. It can be seen that the LVCSR-based pronunciations for OOVs performs just as well as $plex - best$ (which is the pronunciation that is the closest (in edit distance) to a reference pronunciation , i.e. an upper bound).

6. CONCLUSION

Large quantities of human-supplied pronunciations are available on the Web, which we exploit to build pronunciation lexica. Our methods yield more than 7M occurrences of raw English pronunciations (IPA plus ad-hoc). Our approach can be used to bootstrap pronunciation lexica for any lan-

Example	Pronunciations		Ref/Corr/FA/Miss	
	L2P	Web Based	L2P	Web Based
FREUND	f r oy n d	f r eh n d	9/3/0/6	9/3/470/6
SANTO	s ae n t ow	s ax/ey/ax/eh n t	9/2/4/7	9/2/194/7
THIERRY	th iy ax r iy	t eh r iy	7/0/16/7	7/1/1271/6

Table 4: Examples of L2P v/s Web-based pronunciations with negative impact

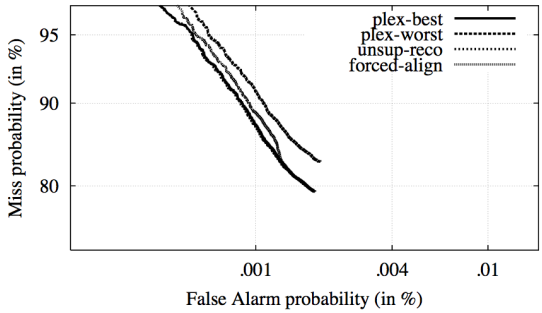


Figure 8: Effect of LVCSR selected pronunciations

guage where IPA or similar resources are available; e.g. extraction statistics (Section 2) for French and German are very promising. There is a need for accurate pronunciations

Method	Number of OOVs
Ad-hoc	**343**
Cleaned and Normalized IPA	**269**
Hand Cleaned Ad-hoc	**109**

Table 5: Number of OOVs that had pronunciations on the web

for named entities or foreign language terms in speech retreival applications like spoken term detection, and the web can serve as an important source. In this paper, we have presented an evaluation of the relative merits and weaknesses of the web as a pronunciation corpus for spoken term detection. Our experiments indicate that in general, web-extracted pronunciations perform better than those generated by L2P systems. However, we observed that in many cases people tend to put 'pronunciations by example,' using the pronunciation of a common word for a rare word. To address false alarms generated by the use of these pronunciations, we are investigating a filtering process based on expected counts of phone sequences in the speech index. Also, ad-hoc transcriptions of common words often highlight unusual pronunciations (e.g. 'cheenah' for 'China', which is a Spanish first name); the question of how many of these rare pronunciations to select and its dependence to the task at hand is a research topic for the future.

7. REFERENCES

[1] C. Allauzen et al. OpenFST: A general and efficient weighted finite-state library. In *CIAA*, 2007.

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

[3] M. Bisani and H. Ney. Investigations on joint-multigram models for grapheme-to-phoneme conversion. In *ICSLP*, 2002.

[4] A. Black, K. Lenzo, and V. Pagel. Issues in building general letter to sound rules. In *ESCA WSS-3*, 1998.

[5] 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.

[6] 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.

[7] S. Deligne and F. Bimbot. Inference of variable-length linguistic and acoustic units by multigrams. *Speech Commun.*, 1997.

[8] T. G. Dietterich. Machine learning for sequential data: A review. *LNCS*, 2396, 2002.

[9] H. Elovitz et al. Letter-to-sound rules for automatic translation of English text to phones. *IEEE Trans. ASSP*, 1976.

[10] 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.

[11] R. Kneser and H. Ney. Improved backing-off for m-gram language modeling. In *ICASSP*, 1995.

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

[13] Y. Marchand and R. I. Damper. A multi-strategy approach to improving pronunciation by analogy. *Comp. Ling.*, 2000.

[14] 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.

[15] NIST. The spoken term detection (std) 2006 evaluation plan, 2006. http://www.nist.gov/speech/tests/std/docs/std06-evalplan-v10.

[16] S. Parlak and M. Saraclar. Spoken term detection for Turkish broadcast news. In *Proc. ICASSP*, 2008.

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

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

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

[20] 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.

[21] P. Taylor. Hidden Markov models for grapheme to phoneme conversion. In *Interspeech*, 2005.

[22] 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.