WEB-DERIVED PRONUNCIATIONS

Arnab Ghoshal[*]   *Martin Jansche*[†]   *Sanjeev Khudanpur*[*]   *Michael Riley*[†]   *Morgan Ulinski*[‡]

| [*]Johns Hopkins University | [†]Google, Inc. | [‡]Cornell University |
| Baltimore, MD 21218 | New York, NY 10011 | Ithaca, NY 14853 |

ABSTRACT

Pronunciation information is available in large quantities 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 from ad-hoc transcriptions. We show improvements on a letter-to-phoneme task when using web-derived vs. Pronlex pronunciations.

Index Terms— Speech processing.

1. INTRODUCTION

Knowing how to pronounce a word is important for automatic speech recognition and synthesis. Previous approaches have either employed trained persons 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 [1, 2]. 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.

We study two kinds of pronunciations commonly found 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 pronunciations 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 clarified. 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 we must predict it 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.

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 [3], pronunciation-by-analogy [4], and hidden Markov models [5]. We chose to use *n*-gram models over pairs [6].

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 [7]. Once built, the *n*-gram model is represented as a weighted finite-state transducer (FST), mapping letters to phonemes, using the OpenFst Library [8], which allows easy implementation of the operations that follow. Note that this results in a *joint* model $\Pr_u[\lambda, \pi]$.

---

[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.

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.

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 twenty 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.

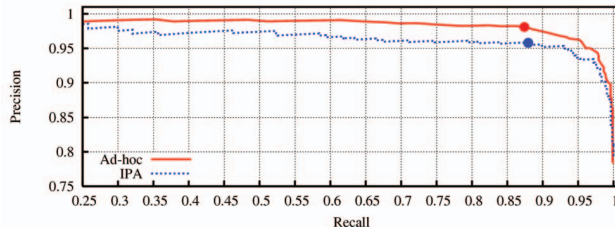### 2.2. Ad-hoc Pronunciation Extraction

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

We then extract that contiguous sequence of terms λ_2, among the eight 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

Fig. 1. Precision vs. recall in pronunciation extraction validation.

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).

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.

Our training and test data are based on a subset of words in the web-derived data whose orthographies also occur in Pronlex. By using only the set of words that appear in both the lexica, we eliminate any overall sampling bias in either of the lexica, and focus solely on the pronunciations. 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 pronunciations per word in Pronlex, and 1.87 pronunciations per word in Web-IPA. In the next subsection, we also consider smaller subsets of this dataset that were derived by a similar methodology. 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').

Some of our test sets, further described below, are drawn at random from the 30K Pronlex ∩ Web-IPA lexicon. For others, we set aside rare words as test data, chosen by low counts in the HUB4 Broadcast News corpora. We are interested in rare words because they are less likely to occur in existing lexica. Handling these otherwise out-of-vocabulary (OOV) words is important in many applications.

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
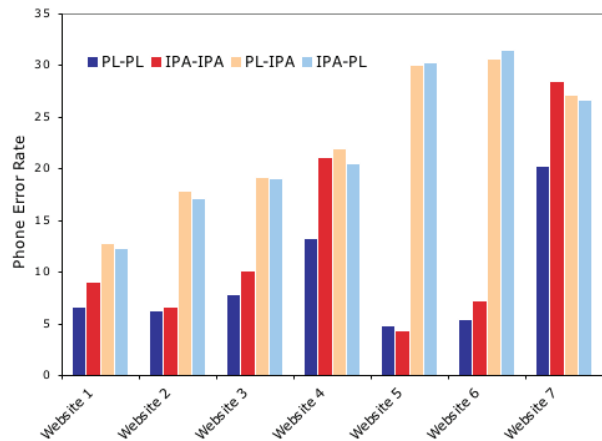
Fig. 2. Cross-validation results by website.

Fig. 3. Effect of pronunciation normalization – L2P models trained using normalized data are better at predicting ref. pronunciations.

the number of insertions, deletions, and substitutions divided by the number of phonemes in the reference (times 100%). In cases of multiple predicted[4] 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 with Pronlex, by performing 5-fold cross-validation experiments on their orthographic intersection described above. For each cross-validation run, two L2P models are trained on the same 24K subset of the intersection – one using the Pronlex pronunciations, and the other using the Web-IPA pronunciations. Each of the models is then used to generate candidate pronunciations for the same 6K subset left out of the training. The two sets of generated candidate pronunciations are then scored against the test pronunciations from both lexica, giving us four PhER numbers. The overall PhER for these four cases are shown in Table 2.

Train Test	Pronlex	Web-IPA
Pronlex	6.35	17.10
Web-IPA	14.33	12.98

Table 2. Phoneme error rates (in %) for 5-fold cross validation on the intersection between Pronlex and Web-IPA pronunciations.

At a first glance the PhER numbers presented in Table 2 may suggest that the Web-IPA data is inherently of lower quality. But a very different picture emerges when one breaks down these PhER numbers by individual websites. We repeated the above cross-validation experiments, but instead using the entire Web-IPA data, the orthographic intersection was done using data collected from individual websites. Fig. 2 shows the same 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 pronunciations are almost as good at predicting the website 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, and vice versa.

---

[4]Only one best-guess pronunciation is produced for each extracted token. Multiple predicted pronunciations for an orthographic form result only if it was *extracted* with multiple distinct IPA (or ad-hoc) pronunciations.
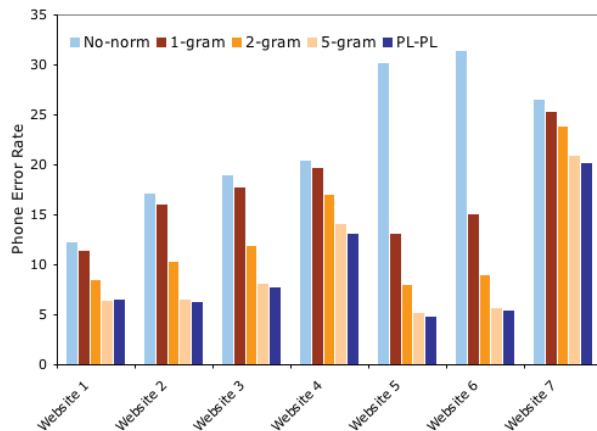
These experiments demonstrate that websites vary in the quality of pronunciations available from them. Moreover, the websites list different pronunciations than what one would obtain from Pronlex. The differences can be caused both by improper use of IPA symbols, as well as other site-specific conventions. For instance, 'graduate' is pronounced as either /ɡɹædʒuɪt/ or /ɡɹædʒueɪt/ in Pronlex, but appears as /ɡɹadjʊeɪt/, /ɡɹædjʊɪt/, /ɡɹædjuət/, /ɡɹædjuɪt/, and /ɡɹædʒʊət/ 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. As before, we find the orthographic intersection of the lexicon obtained from a website and Pronlex. If multiple pronunciations are present, then the two with the smallest phoneme edit distance are selected. Using these pronunciation pairs we train a phoneme-to-phoneme (P2P) transduction model, which takes a pronunciation obtained from the website and converts it to a Pronlex-like form.

The model for the P2P normalizing transducer is identical to the L2P models described earlier, the only difference being that the P2P models are trained on aligned (phoneme, phoneme) pairs, instead of (letter, phoneme) pairs. For the cross-validation experiments, the normalizing transducer is trained on the pronunciation pairs collected from the two training lexica, and is then used to normalize the pronunciations in the training lexicon obtained from the website. An L2P model trained on the normalized pronunciations is then used to generate candidate pronunciations for the test set words, which are scored against their Pronlex references.

The P2P transducers are trained using varying *n*-gram orders, and the results are presented in Fig. 3. As one can clearly see, normalization helps to improve the quality of the pronunciations obtained from the web. Notice in particular that the 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). Moreover, while creating a hand-built lexicon, it
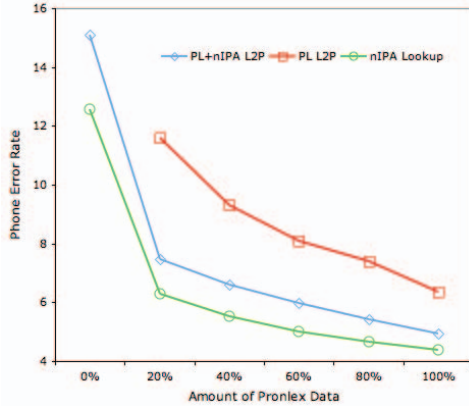
Fig. 4. Performance on rare words – normalized web pronunciations help a lot on rare words.

is natural to annotate the most frequent words. To replicate this we subdivide the Pronlex words, with BN frequency of at least 2, into 5 subsets based on decreasing frequency – the first one contains 20% of the most frequent words, the second one 40%, third with 60%, fourth 80%, and the fifth 100%.

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 is trained on the subset of Pronlex, and then used to generate pronunciations for the rare words. (2) Pronunciations from the 10 most frequent websites are normalized using only the subset of Pronlex, and are then pooled together. Rare words are then looked up in this lexicon. (3) The normalized Web-IPA and the Pronlex subset are combined together and used to train another L2P model, which is then used to generate the pronunciations.

Fig. 4 shows the PhER on the rare words using each of the three methods described above, for varying amounts of Pronlex data used. The normalized Web-IPA data clearly produces better pronunciations for rare words. Of particular interest is the fact that the Web-IPA, when normalized using only 20% of the hand-crafted Pronlex dictionary (roughly 10K most frequent words), already produces pronunciations that are as good as those generated by an L2P model trained on the whole of Pronlex.

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 test 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 $\Pr[\lambda_1, \lambda_2, \pi] = \Pr[\lambda_1 \mid \pi] \Pr[\lambda_2 \mid \pi] \Pr[\pi]$ 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. CONCLUSION

Large quantities of human-supplied pronunciations are available on the Web, which we exploit to build lexica comparable in quality to Pronlex and larger in size. To our knowledge, this is the first study of its kind; we are aware of [9], which addresses a much more restricted problem for Japanese that exploits its multiple writing systems. Our approach can be used to bootstrap pronunciation lexica for any language where IPA or similar resources are available; e.g. extraction statistics (cf Sec. 2) for French and German are very promising.

One issue that we did not address is the usefulness of a pronunciation. For example, ad-hoc transcriptions of common words often highlight unusual pronunciations (e.g. 'cheenah' for 'China', which is a Spanish first name). This would be scored correct in Sec. 4.2, but there is a question of how many of these rare pronunciations we would want to put in our lexicon.

6. REFERENCES

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

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

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

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

[5] P. Taylor, "Hidden Markov models for grapheme to phoneme conversion," in *Interspeech*, 2005.

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

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

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

[9] E. Sumita and F. Sugaya, "Word pronunciation disambiguation using the Web," in *HLT/NAACL*, 2006.