# LEARNING NON-PARAMETRIC MODELS OF PRONUNCIATION

*Brian Hutchinson*[*]

Electrical Engineering Department
University of Washington
brianhutchinson@ee.washington.edu

*Jasha Droppo*

Speech Technology Group
Microsoft Research
jdroppo@microsoft.com

## ABSTRACT

As more data becomes available for a given speech recognition task, the natural way to improve recognition accuracy is to train larger models. But, while this strategy yields modest improvements to small systems, the relative gains diminish as the data and models grow. In this paper, we demonstrate that abundant data allows us to model patterns and structure that are unaccounted for in standard systems. In particular, we model the systematic mismatch between the canonical pronunciations of words and the actual pronunciations found in casual or accented speech. Using a combination of two simple data-driven pronunciation models, we can correct 5.2% of the errors in our mobile voice search application.

***Index Terms*—** Pronunciation model, non-parametric model, casual speech

## 1. INTRODUCTION

As more data becomes available for a given speech recognition task, the natural way to improve recognition accuracy is to train larger acoustic models. Although this strategy provides modest improvements to small systems, it results in diminishing relative gains as the data and models continue to grow [1].

Alternatively, abundant data can be used to model structures and variabilities in the data that are unaccounted for in the standard system. One good example of this style is the cluster modeling approach of Beaufays [2], which automatically trains several specialized acoustic models instead of one model to cover all scenarios. This is the style of the approach presented here, where we use a very large data set to address the problem of pronunciation variability. When causal or accented speech cause the observed pronunciations diverge from their canonical forms, the acoustic model must bridge the gap. Although triphone models can account for some of this variation, variance in pronunciation has been found to be dependent on speaking rate and word frequency [3], which suggests that triphone context may be too limited.

Figure 1 illustrates several ways that pronunciation variability, the relationship between words and acoustics, can be modeled. One path represents the standard approach: a word sequence generates a phonetic sequence though a fixed pronunciation lexicon, which then generates acoustics through an acoustic model. The primary drawback of this approach is that the acoustic model must handle any differences between expected phones from the lexicon and realized phones in the acoustics. A substantial amount of research has been invested into addressing this issue (see [4] for an early survey paper). In particular, many researchers have worked on automatic dictionary learning [5, 6, 7, 8, 9]. However, simply increasing the number of available pronunciations has been found to increase search complexity, and in doing so offset the gains of better pronunciation modeling (e.g., [10]).

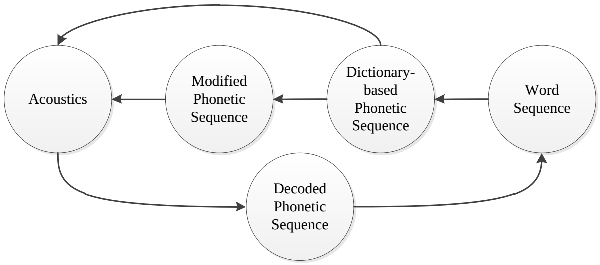
Acoustics
Modified Phonetic Sequence
Dictionary-based Phonetic Sequence
Word Sequence
Decoded Phonetic Sequence

**Fig. 1**. There are several ways that phonetic information bridges acoustics and word sequences.

Instead, one can incorporate a distortion model that bridges the gap between the dictionary-based phonetic sequence and the acoustics. Here any discrepancy between the dictionary phonemes and acoustics is decoupled into a distortion model, which governs phonetic distances, and the acoustic model, which is relieved of some of its burden. Researchers have considered a number of forms for the distortion model, including parametric context-independent and -dependent probabilistic phone edit models [11, 12, 13], tree-based models [14], and phonological rule-based models [15].

A third approach directly models word labels as being generated from a decoded phonetic sequence. Arbitrary features of an unconstrained phonetic decoding may be used, e.g. existence, expectation, or Levenshtein features employed in a segmental conditional random field framework [16, 17].

The method we propose in this work models distributions over possible decoded phone strings given hypothesized word strings. We interpolate a large non-parametric empirical model, that explicitly models a phone string distribution for each word, with a smaller parametric model. Using millions of utterances, we find that the empirical model approximately recreates the lexicon, while augmenting it with missing pronunciations, dialectal variants, and common reductions. A lattice rescoring experiment demonstrates a 5.2% relative reduction in word error rate.

The remainder of the paper is organized as follows. Section 2 introduces parametric, non-parametric and interpolated models of pronunciation; in Section 3, we discuss our experiments and results; and we offer conclusions and areas of future research in Section 4.

[*] This work was conducted during an internship at Microsoft Research.

| **Missing Pronunciations** | | **Dialectal Variants** | |
|---|---|---|---|
| license | l ay s ax n z | french | f r ae n ch |
| corps | k ao r | cinemas | s eh n ax m ax s |
| **Common Reductions** | | **Common Phone Rec. Errors** | |
| donald | d aa n ax l | civic | s ih t ih k |
| redmond | r eh d m ax n | spice | s t b ay s |

**Fig. 2**. Example pronunciations not contained in the lexicon.

## 2. MODELS OF PRONUNCIATION

The goal of our pronunciation models is to estimate the distribution $P(p_o|w)$ of observed phone sequences $p_o$ for each word $w$ in the lexicon. These observed phone sequences may be obtained via human annotation, or more likely, an unconstrained phonetic decoding. We consider several forms for $P$, including a non-parametric empirical model, two kinds of parametric model, and an interpolated model that combines their relative strengths.

### 2.1. Non-Parametric Empirical Model $P_E$

When sufficient data is available, we can estimate $P(p_o|w)$ directly for all words appearing in our training data, using:

$$P_E(p_o|w) = \frac{C(w,p_o)}{\sum_{p'_o} C(w,p'_o)}.$$

Here $C(w,p_o)$ is the number of times word $w$ was has observed pronunciation $p_o$ in the training data. To obtain these counts, we separately perform an unconstrained phonetic decoding and a word alignment of all training utterances, and then assign phone sequences to words based on timing and sequence information.

Although the empirical model learns to assign probability mass near the existing pronunciation dictionary entries, it also learns several types of patterns that did not originally exist. Figure 2 contains pronunciations that were the most probable for the given word, but did not exist in the ASR pronunciation dictionary. They fall into four classes: legitimate pronunciations that were missing from the lexicon, dialectal variants, common phonetic reductions of words, and systematic phone recognition errors. The last category highlights the importance of matching the process used to produce observed pronunciations in training and test.

The empirical model has a number of advantages: it serves as a data driven way to heal mistakes the lexicon, it yields relatively sharp models of pronunciation, and it models lexically-dependent pronunciation variation. However, it has limitations: the empirical model may poorly estimate probabilities for infrequent words, and does not generalize to unseen words. To handle these cases, we need a parametric model to which we can back off.

### 2.2. Parametric Models $P_M$

Our parametric models are smooth in the sense that they assign a non-zero probability to all possible phone sequences. They compute $P(p_o|w)$ by marginalizing over dictionary pronunciations, $p_w$:

$$P_M(p_o|w) = \sum_{p_w} P(p_o, p_w|w) = \sum_{p_w} P_D(p_o|p_w)P_P(p_w|w)$$

where $P_D$ is a distortion (equivalently error or edit) model similar to the joint multigram model of [18], and $P_P$ is a standard (possibly uniform) pronunciation model.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| ae | p | ax | l | | ae | p | ax | l |
| ae | b | $\epsilon$ | l | | ae | $\epsilon$ | b | l |

**Fig. 3**. Two minimum Levenshtein alignments of reference sequence [ae p ax l] to observed sequence [ae b l].

$P_D$ gives a distance between two phone sequences. In the models we will consider, it decomposes over the distances between individual phones in $p_o$ and $p_w$. To do this decomposition, we first need to associate individual phones in $p_o$ with those in $p_w$; that is, we need to align $p_o$ to $p_w$.

#### 2.2.1. *Alignments*

The process of aligning two phone sequences pairs individual phones in one sequence with individual phones in another. If a phone is not paired with any phone in the second sequence (e.g. if the sequences are of different length), an $\epsilon$ is inserted into second sequence and the phone is aligned to $\epsilon$. We denote the alignment of $p_w$ to $p_o$ as $A = ((r_1, o_1), \ldots, (r_N, o_N))$, where $r_i$ is a reference phone (or $\epsilon$) and $o_i$ is the observed phone (or $\epsilon$) to which it is aligned.

One common approach to aligning two phone sequences is to compute the minimum Levenshtein distance alignment $A_{Lev}$. However, as Fig. 3 illustrates, there may be multiple minimizers of the Levenshtein distance. Although both alignments in this figure obtain the minimum distance, the first offers a more probable generative explanation of the observed phone sequence.

#### 2.2.2. *Context Independent Model*

Given an alignment $A$, the context independent edit model assumes a very simple form:

$$P_{D_{CI}}(p_o|p_w, A) = \prod_{i=1}^{N} p(o_i|r_i) \tag{1}$$

The only parameters of this model are the $O(|\mathcal{P}|^2)$ entries its conditional probability table, where $\mathcal{P}$ is the alphabet of phones. Given an alignment, these parameters are estimated using maximum likelihood: $P(o|r) = C(r,o)/\sum_{o'} C(r,o')$, where $C(r,o)$ is the number of times reference symbol $r$ is aligned to observed symbol $o$ in the training data.

Given a trained CI model, one can efficiently compute the alignment $A_{CI}$ that maximizes Eqn. 1. This alignment can be found in $O(N^2)$ time using a variant of the Levenshtein dynamic programming algorithm, with pairwise alignment cost $-\log(p(o|r))$. We can therefore train the CI model iteratively, starting with a Levenshtein alignment:

1. Given an alignment of the training data, update parameters
2. Given an updated model, re-align the training data

In practice, we find that training takes 5-15 iterations to converge.

#### 2.2.3. *Context Dependent Model*

It is intuitive that the probability of a given edit is influenced by its context. A straightforward generalization of the context independent model is to additionally condition on the the previous $M$ edits; that is, to be build models of the form:

$$P_{D_{CD}}(p_o|p_w, A) = \prod_{i=1}^{N} P(o_i|r_i, o_{i-1}, r_{i-1}, \ldots, o_{i-M}, r_{i-M})$$

This model has $O(|\mathcal{P}|^{2M+1})$ parameters, the elements of the conditional probability table. For small $M$ (e.g. $M = 2$ or 3) and moderate amounts of data (e.g. 500K words), the parameters can be well estimated with maximum likelihood.

### 2.3. Interpolated Model $P_I$

Ideally, we would like to rely on the empirical model when it is well estimated, and back off to a (smoother) parametric model when it is not. Our interpolated model $P_I$ does exactly this:

$$P_I(p_o|w) = \alpha_w P_E(p_o|w) + (1 - \alpha_w) P_M(p_o|w).$$

The word dependent interpolation weight is given by $\alpha_w = C(w)/(C(w) + K)$. $C(w)$ is the number of times word $w$ is observed in the training data. $K$ is a tunable parameter, and can be interpreted as the amount of count mass allocated to $P_D$. For words not seen during training, $\alpha_w = 0$, and the parametric model is used exclusively; on the other hand, $\alpha_w \approx 1$ for words that have been observed a large number of times. In practice, we find that $K \in [1, 100]$ works well.

## 3. EXPERIMENTS

### 3.1. Lattice Rescoring

We conducted lattice rescoring experiments to compare the effectiveness of our various models of pronunciation:

1. We generate a word lattice using an HMM based system, whose acoustic model contains 135K diagonal Gaussian components shared by 22K states, which in turn are shared by 9.7K HMMs; its language model is a trigram model over a 65K lexicon, with 4.7M n-grams.
2. We perform a phonetic decoding using the same acoustic model with a trigram language model containing 16K n-grams over a vocabulary of 45 phones. This provides our one-best observed phone sequence.
3. For each word $w$ in the lattice (with recognized pronunciation variant $p_w$), we use $w$'s time boundaries to identify the observed phone sequence $p_o$ in the span of $w$, and augment the lattice with the pronunciation model score $P(p_o|w, p_w)$ (using any of the models in Section 2).
4. The new lattice score for each word is a weighted sum of the existing acoustic model score, the existing language model score, and the new pronunciation model score. These weights are tuned on development data.

Because the empirical model has no means to assign probabilities to unseen words, it cannot be used alone in step 3 above. To bypass this problem, we assign an $\epsilon$ probability to all pronunciations for words that were not seen in training. This permits us to use the empirical pronunciation models in the above rescoring framework.

### 3.2. Data

Our data come from the Windows Live Search for Mobile voice search task [19]. We divide the data into a 2.9M utterance (6.3M word) training set, 8.7K utterance (18.9K word) development set, and 12.7K utterance (27.2K word) test set. Some of training data labels are not human annotated, but instead are "lightly supervised" transcriptions in which we have high confidence.

| Method | Dev | Test |
|---|---|---|
| Baseline | 34.3 | 34.8 |
| CI | 32.8 | 33.6 |
| CD ($M = 1$) | **32.4** | **33.5** |
| CD ($M = 2$) | **32.4** | 33.6 |
| Pron Dict | 33.0 | 33.6 |
| Empirical | **32.4** | **33.1** |
| Interpolated | **32.3** | **33.0** |

**Table 1**. WER by model of pronunciation.

| | Lev. Aligned | CI Aligned |
|---|---|---|
| CI Model | 0.9% | 3.4% |
| CD Model ($M = 1$) | 3.4% | 3.7% |

**Table 2**. WER reduction over baseline by model and alignment.

### 3.3. Results

We first compare models in Table 1. This table contains four sections: the baseline, parametric models, non-parametric models, and the interpolated model.

Of the parametric models, the context dependent model conditioned on the previous $M = 1$ edits performs the best, though the context independent model performs comparably. All three parametric results use the alignment $A_{CI}$ instead of $A_{Lev}$; see Section 3.4 for some discussion of the effect of alignment on performance.

The "pron dict" method rescores using a pronunciation dictionary with relative frequencies estimated on the training set to rescore the lattice. This method rewards phonetic "exact matches" (observed pronunciation matches dictionary pronunciation), which does improve performance over the baseline. This is consistent with previously reported results [17]. Performance is further improved using our empirical model.

Finally, moving to our more general interpolated model preserves the performance gains. We set parameter $K = 1$, having tuned on development data. This interpolated model corrects 5.2% of the errors made by the baseline.

### 3.4. Analysis

There are a number of useful observations that can be drawn from these experiments. First, Table 2 illustrates the interaction between context and alignment method. The left column, which uses the minimum Levenshtein alignment $A_{Lev}$, is consistent with previous findings [13]: introducing context improves the performance. The right column, however, suggests that using CI alignment $A_{CI}$ gives a similar boost in performance without the need for context. This suggests that one of benefits of context in the Levenshtein case is to learn to correct its sub-optimal alignments.

In Table 3, statistics of the empirical model by count cutoff are given. A model with count cutoff $C$ means that all word-pronunciation pairs occurring fewer than $C$ times are pruned from the empirical model. The first observation is that the empirical model tends to contain fewer words than the pronunciation dictionary, and more pronunciations per word. However, as listed in the Match column, these empirical pronunciations "cover" a larger portion of the test data; that is, they include the observed pronunciations in test utterances. The second observation is illustrated by the pronunciations per match (P/M) column: although there are a small

| Count Cutoff | Words | Prons. | P/W | Match | P/M |
|---|---|---|---|---|---|
| 1 | 44814 | 1739981 | 38.8 | 7784 | 223.5 |
| 5 | 10522 | 103417 | 9.8 | 5808 | 17.8 |
| 20 | 4423 | 23166 | 5.2 | 4579 | 5.1 |
| 50 | 2461 | 9037 | 3.7 | 3804 | 2.4 |
| 100 | 1541 | 4440 | 2.9 | 3238 | 1.4 |
| 200 | 995 | 2245 | 2.3 | 2830 | 0.8 |
| 1000 | 327 | 511 | 1.6 | 1988 | 0.3 |
| 5000 | 84 | 120 | 1.4 | 1095 | 0.1 |
| Pron Dict | 64601 | 91513 | 1.4 | 2484 | 36.8 |

**Table 3**. Effect of count cutoff pruning on the empirical model. "Match" indicates how many test set utterances (out of 12,758) are covered by the model. P/W, P/M are the prons. per word, per match.

number of very common word-pronunciation pairs, an exponential number of new pronunciations must be added to continue to cover utterances in the test data. One can also interpret the P/M column as a measurement of efficiency. It takes only minor pruning of counts to create a more efficient model than the pronunciation model; this better fit is one advantage of learning pronunciations in a data-driven fashion.

## 4. CONCLUSIONS

We introduce a non-parametric empirical model that exploits abundant training data to directly learn pronunciation variation. Interpolating the empirical model with a parametric model yields the best performance, with a relative improvement of 5.2% in WER over the baseline. For the voice search task, the empirical model alone contributes most of the gain, and provides good coverage of the test data. We also find evidence that much of the benefits of introducing context-dependency to a parametric distortion model that uses Levenshtein distance alignments can be obtained by a context-independent model that uses an optimal alignment. Further, this optimal alignment can be computed with no additional run-time cost over the standard Levenshtein dynamic programming algorithm.

There are a number of ways in which this work could be extended. First, closer integration with acoustic model training is likely to yield sharper distributions and a tighter fit to the data. Second, estimating word-pronunciation co-occurrence counts in semi-supervised fashion (e.g. through word recognition instead of forced alignment) would broaden its applicability to a wide range of speech genres and tasks. Finally, it would be of interest to modify our models to factor out the distinct phenomena that affect pronunciation (e.g. accent, dialect, recognition errors). Aside from better modeling the data, such an approach could be used for speaker adaptation.

## 5. ACKNOWLEDGMENTS

The authors would like to thank Patrick Nguyen and Alex Acero for their insightful suggestions and feedback.

## 6. REFERENCES

[1] M. J. F. Gales, D. Y. Kim, P. C. Woodland, H. Y. Chan, D. Mrva, R. Sinha, and S. E. Tranter, "Progress in the CU-HTK broadcast news transcription system," *IEEE Trans. ASLP*, vol. 14, no. 5, pp. 1513–1525, September 2006.

[2] F. Beaufays, V. Banhoucke, and B. Strope, "Unsupervised discovery and training of maximally dissimilar cluster models," in *Proc. Interspeech*, 2010.

[3] E. Fosler-Lussier and N. Morgan, "Effects of speaking rate and word frequency on pronunciations in convertional speech," *Speech Communication*, vol. 29, no. 2-4, pp. 137–158, November 1999.

[4] H. Strik and C. Cucchiarini, "Modelling pronunciation variation for ASR: Overview and comparison of methods," in *Proc. ETRW Workshop on Modelling Pronunciation Variation for ASR*, May 1998.

[5] M. Riley and A. Ljolje, "Automatic generation of detailed pronunciation lexicons," in *Automatic Speech and Speaker Recognition: Advanced Topics*, C.-H. Lee, F. K. Soong, and K. K. Paliwal, Eds., chapter 12. Kluwer, Boston, 1995.

[6] T. Sloboda and A. Waibel, "Dictionary learning for spontaneous speech recognition," in *Proc. ICSLP*, October 1996.

[7] C. M. Westendorf and J. Jelitto, "Learning pronunciation dictionary from speech data," in *Proc. ICSLP*, October 1996, pp. 1045–1048.

[8] T. Holter and T. Svendsen, "Maximum likelihood modelling of pronunciation variation," *Speech Communication*, vol. 29, no. 2-4, pp. 177–191, 1999.

[9] O. Vinyals, L. Deng, D. Yu, and A. Acero, "Discriminative pronunciation learning using phonetic decoder and minimum classification error criterion," in *Proc. ICASSP*, April 2009.

[10] M. Finke and A. Waibel, "Speaking mode dependent pronunciation modeling in large vocabulary conversational speech recognition," in *Proc. Eurospeech*, 1997, pp. 2379–2382.

[11] E. S. Ristad, P. N. Yianilos, and S. Member, "Learning string edit distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 522–532, 1998.

[12] G. Zweig and J. Nedel, "Empirical properties of multilingual phone-to-word transduction," in *Proc. ICASSP*, 2008.

[13] J. Droppo and A. Acero, "Context dependent phonetic string edit distance for automatic speech recognition," in *Proc. ICASSP*, 2010.

[14] J. E. Fosler-Lussier, *Dynamic Pronunciation Models for Automatic Speech Recognition*, Ph.D. thesis, University of California, Berkeley, 1999.

[15] I. Amdal, F. Korkmazskiy, and A. C. Surendran, "Joint pronunciation modelling of non-native speakers using data-driven methods," in *Proc. ICSLP*, October 2000.

[16] P. Nguyen and G. Zweig, "Speech recognition with flat direct models," *IEEE Journal of Selected Topics in Signal Processing*, 2010.

[17] G. Zweig, P. Nguyen, J. Droppo, and A. Acero, "Continuous speech recognition with a tf-idf acoustic model," in *Proc. ICASSP*, 2010.

[18] S. Deligne, F. Yvon, and F. Bimbot, "Variable-length sequence matching for phonetic transcription using joint multigrams," in *Proc. Eurospeech*, Madrid, Spain, September 1995, pp. 2243–2246.

[19] A. Acero, N. Bernstein, R. Chambers, Y. C. Ju, X. Li, J. Odell, P. Nguyen, O. Scholz, and G. Zweig, "Live search for mobile: Web services by voice on the cellphone," in *Proc. ICASSP*, 2008.