To appear in EUROSPEECH'95

BUILDING MULTIPLE PRONUNCIATION MODELS FOR NOVEL WORDS USING EXPLORATORY COMPUTATIONAL PHONOLOGY

*Gary Tajchman***, *Eric Fosler, and Daniel Jurafsky*

International Computer Science Institute
1947 Center Street, Suite 600
Berkeley, CA 94704, USA
& University of California at Berkeley
{tajchman,fosler,jurafsky}@icsi.berkeley.edu

ABSTRACT

In this paper we describe a completely automatic algorithm that builds multiple pronunciation word models by expanding baseform pronunciations with a set of candidate phonological rules. We show how to train the probabilities of these phonological rules, and how to use these probabilities to assign pronunciation probabilities to words not seen in the training corpus. The algorithm we propose is an instance of the class of techniques we call Exploratory Computational Phonology.

1. INTRODUCTION

One well-known difficulty in understanding speaker-independent continuous speech is variability in the pronunciation of words. This variability occurs across speakers and also across different contexts for a single speaker. In order to model this variation, recognition systems often use a richer lexicon in which each word has multiple pronunciations.

Using a multiple-pronunciation lexicon requires setting a probability for each pronunciation. The minimal algorithm, for example, would assign each of the n pronunciations of a word the zero-knowledge prior probability $1/n$. If a sufficient corpus and sufficient training time are available, these probabilities can be set simply by running a forced-Viterbi pass on some training data and deriving counts for each pronunciation of each word. But it is often the case that we lack sufficient training data or time to train each word. Our algorithm includes a new method for using phonological rules to estimate these pronunciation probabilities, which does not require retraining on each new corpus. The algorithm sets the probability of a word's pronunciation by combining the probabilities of the phonological rules used in each pronunciation.

We show two ways in which using these probabilities can help build an improved lexicon for speech lexicon. First, a lexicon with pronunciation probabilities has a lower word error rate on a Wall-Street Journal speech recognition task than a lexicon with equiprobable pronunciations, although not significantly better. Second, assigning probabilities to pronunciations gives us a metric for pruning away some pronunciations. A pruned lexicon allows for faster decoding and also, we show, for a statistically significantly reduced word error rate.

Deriving pronunciation probabilities by probabilistic rules also allows the development of a "parameterized" recognition lexicon, if the test set has sources of variability not present in the training set. For example, if the test set involves particularly fast speakers, the probabilities of phone deletion/reduction rules can be increased, increasing the probability of reduced pronunciations. Similarly, rules can be written to model the phonological effects of foreign accents. Then if a foreign speaker is detected, those rule probabilities can be changed dynamically.

We have been referring to our probabilistic rule training algorithm, along with other methods like phonological decision tree induction, as Exploratory Computational Phonology, a paradigm which uses pattern recognition tools to explore the space of phonological variation directly from acoustic data.

2. PHONOLOGICAL RULES FOR CREATING PRONUNCIATIONS

Given a base dictionary of pronunciations, we first apply optional phonological rules to produce an expanded lexicon, following the insights of Cohen (1989). Since the rules are optional, the surface lexicon will contain each underlying pronunciation unmodified, as well as the pronunciation resulting from the application of each relevant phonological rule. Each of our rules comes from our own error analysis of our recognizer or from the literature (Cohen (1989), Zwicky (1970, 1972b, 1972a), Kaisse (1985)). Table 1 gives the rules used in these experiments.

Our baseform dictionary was made by combining five different on-line pronunciation dictionaries: CMU (CMU 1993), LIMSI (Lamel 1993), PRONLEX (COMLEX 1994), BRITPRON (Robinson 1994), and a text-to-speech system. The pronunciations from all these sources were mapped into our 54-phone

**Currently at Voice Processing Corp, 1 Main St, Cambridge, MA 02142: tajchman@vpro.com

Name	Code	Rule
Reductions		
Mid vowels	RV1	-stress [aa ae ah ao eh er ey ow uh] $\rightarrow$ ax
High vowels	RV2	-stress [iy ih uw] $\rightarrow$ ix
R-vowel	RV3	-stress er $\rightarrow$ axr
Syllabic n	SL1	[ax ix] n $\rightarrow$ en
Syllabic m	SL2	[ax ix] m $\rightarrow$ em
Syllabic l	SL3	[ax ix] l $\rightarrow$ el
Syllabic r	SL4	[ax ix] r $\rightarrow$ axr
Flapping	FL1	[tcl dcl] [t d] $\rightarrow$ dx /V ___ [ax ix axr]
Flapping-r	FL2	[tcl dcl] [t d] $\rightarrow$ dx /V r ___ [ax ix axr]
H-voicing	VH1	hh $\rightarrow$ hv / [+voice] ___ [+voice]

Table 1: Phonological Rules with Estimated Probabilities

ARPAbet-like phone set using a set of obligatory rules for stop closures [bcl, dcl, gcl, pcl, tcl, kcl], and optional rules to introduce the syllabic consonants [el, em, en], reduced vowels [ax, ix, axr], voiced h [hv], and alveolar flap [dx]. One goal of our rule-application procedure was to build a tagged lexicon to avoid the complexity of parsing the surface pronunciations with a phonological-rule parser. In a tagged lexicon, each surface pronunciation is annotated with the names of the phonological rules that applied to produce it. Thus when the decoder finds a particular pronunciation in the speech input, the list of rules which applied to produce it can simply be looked up in the tagged lexicon.

The resulting tagged surface lexicon would have the entries in Table 2.

3. RULE PROBABILITY ESTIMATION

Given the rule-tagged lexicon, we next run an embedded forced-Viterbi procedure on a segment of the Wall Street Journal 1993 corpus to find the optimal alignment of word pronunciations to the corpus. From this alignment we can then produce counts for each pronunciation of each word. The counts and the lexicon can then be combined to form a tagged lexicon that also has counts for each pronunciation of each word.

Notice in Table 2 that each pronunciation of a word may contain multiple derivations, each consisting of the list of rules which applied to give the pronunciation from the base form. These tags are either positive, indicating that a rule applied, or negative, indicating that it did not.

To produce the initial rule probabilities, we need to count the number of times each rule applies, and compare it to the number of times it had the potential to apply. If each pronunciation only had a single derivation, this would be computed simply as follows:

$$P(R) = \sum_{p \in \text{PRON}} \frac{\text{Ct (Rule R applied in } p)}{\text{Ct (Rule R could have applied in } p)}$$

This could be computed from the tags as :

$$P(R) = \sum_{p \in \text{PRON}} \frac{\text{Ct(+R tags in } p)}{\text{Ct(+R tags in } p) + \text{Ct(-R tags in } p)}$$

However, since each pronunciation can have multiple derivations, the counts for each rule from each derivation need to be weighted by the probability of the derivation. The derivation probability is computed by multiplying together the probability of each of the applications or non-applications of the rule. Let

- *DERIVS(p)* be the set of all derivations of a pronunciation p.
- *POSRULES(p,r,d)* be 1.0 if derivation d of pronunciation p uses rule r, else 0.
- *ALLRULES(p,r)* be the count of all derivations of p in which rule r could have applied (i.e. in which d has either a +R or -R tag).
- $P(d|p)$ be the probability of the derivation d of pronunciation p.
- PRON be the set of pronunciations derived from the forced-Viterbi output.

Now a single iteration of the rule-probability algorithm must perform the following computation:

$$P(r) = \sum_{p \in \text{PRON}} \sum_{d \in \text{DERIVS}(p)} P(d|p) \frac{\text{POSRULES(p,r,d)}}{\text{ALLRULES(p,r)}}$$

Since we have no prior knowledge, we make the zero-knowledge initial assumption that $P(d|p) = \frac{1}{|DERIVS(p)|}$. The algorithm can then be run as a successive estimation-maximization to provide successive approximations to $P(d|p)$. For efficiency reasons, we actually compute the probabilities of all rules in parallel, as shown in Figure 1.

4. ESTIMATING PROBABILITIES OF UNSEEN WORDS

Given the probabilities of phonological rules and the tagged lexicon, we can now estimate the probabilities of pronunciations for each word. Let:

- $P(+R)$ be the probability of rule R applying.
- $P(-R) = 1 - P(+R)$ be the probability of rule R not applying.

bcl b ah dx ax:+BPU +FL1; +CMU +FL1 +RV1; +PLX +FL1 +RV1
bcl b ah dx axr: +TTS +FL1; +BPU +FL1; +CMU +FL1 -RV1 +RV3; +LIM +FL1 ; +PLX +FL1 -RV1 +RV3
bcl b ah tcl t ax:+BPU -FL1; +CMU -FL1 +RV1; +PLX -FL1 +RV1
bcl b ah tcl t axr:+TTS -FL1; +BPU -FL1; +CMU -FL1 -RV1 +RV3; +LIM -F L1; +PLX -FL1 -RV1 +RV3
bcl b ah tcl t er:+CMU -RV1 -RV3; +PLX -RV1 -RV3

Table 2: Resulting tagged entries

For each word/pron pair $p \in PRON$ from forced-Viterbi alignment
  Let $DERIVS(p)$ be the set of rule derivations of p
  For every $d \in DERIVS(p)$
    For every rule $R \in d$
    if $(R = +RULE)$
    then
      $ruleapp\{RULE\}$ += $\frac{1}{|DERIVS(p)|}$
    else
      $rulenoapp\{RULE\}$ += $\frac{1}{|DERIVS(p)|}$
For every rule $RULE$
  $P(RULE) = \frac{ruleapp(RULE)}{ruleapp(RULE)+rulenoapp(RULE)}$

Figure 1: Parallel computation of rule probabilities

The probability of pronunciation pron_i of word W with tags $\pm R_1, \ldots, \pm R_n$ is estimated by two steps. First, we take the geometric mean of pronunciation probabilities, to avoid penalizing words in which a large number of phonological rules apply.

$$Q'(\text{pron}_i|W) = \sqrt[n]{\prod_{k=1}^{n} P(\pm R_k)}$$

Next, these geometric means are normalized.

$$P(\text{pron}_i|W) = \frac{Q'(\text{pron}_i|W)}{\sum_j Q'(\text{pron}_j|W)}$$

Probabilities of identical pronunciations with different derivations are summed together, to give a complete probability for the pronunciation.

5. RESULTS

We ran the estimation algorithm on 7203 sentences (129,864 words) read from the Wall Street Journal. The corpus (1993 WSJ Hub 2 (WSJ0) training data) consisted of 12 hours of speech, and had 8916 unique words. Table 1 shows the resulting phonological rule probabilities.

We first attempted to judge the reliability of our automatic rule-probability estimation algorithm by comparing it with hand-transcribed pronunciations. We took the hand-transcribed pronunciations of each word in TIMIT, and computed rule probabilities by the same rule-tag counting procedure used for our forced-Viterbi output. Figure 2 shows the fit between the automatic and hand-transcribed probabilities. Since the TIMIT pronunciations were from a completely different data collection effort with a very different corpus and speakers, the similarity of the two sets of probabilities is quite encouraging.

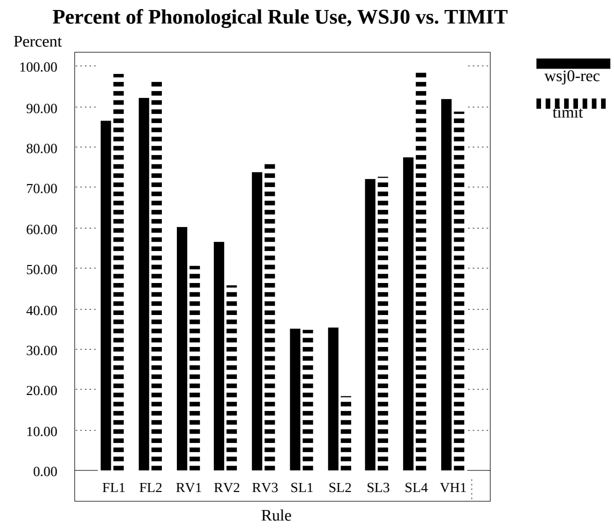

Figure 2: Automatic vs Hand-Transcribed Probabilities for Phonological Rules

We next attempted to determine whether our algorithm for assigning pronunciation probabilities performed better than the zero-knowledge equiprobable lexicon. We compared a lexicon with pronunciation probabilities set by our algorithm to the same lexicon in which each pronunciation had the same probability. The experiment was run on a 5 kword closed-vocabulary test database taken from the female subset of spoke 5 of the 1993 WSJ development set. For these experiments, the training set consisted of all of the female speakers in the WSJ0 database. The speech database contained 3538 sentences, with 8195 unique words.

Table 3 shows that adding pronunciation probabilities reduced word error from 32.6% for the equiprobable lexicon to 30.4% for the probabilistic lexicon.

The reduction in word error due to assigning probabilities is not statistically significant. We do get a statistically significant error reduction, however, by using the pronunciation probabilities to prune the lexicon. We define a pruning parameter, μ, such that $0 \leq \mu < 1$, where if

$$P^*((\text{pron}_j|W) = max_j P(\text{pron}_j|W)$$

We then prune all pronunciations pron_i such that

$$P(\text{pron}_i|W) \leq \mu P^*(pron_i|W)$$

Lexicon	Word Recog. Error
Equiprobable	32.6%
Probabilities, No Pruning	30.4%
Probabilities, Pruned μ=0.2	25.5%
Probabilities, Pruned μ=0.4	23.1%
Probabilities, Pruned μ=0.6	23.8%
Probabilities, Pruned μ=0.8	24.4%
Probabilities, Most-Likely	24.6%
Forced-Viterbi (lower bound)	20.8%

Table 3: Results of Probabilistic Rule Algorithm

The results can be seen in Table 3. Note that the absolute word error values are quite high, since we are using a simple bigram system, and only giving it half the normal training time to facilitate experimental turnaround.

The lexica pruned at μ=0.4 and μ=0.6 are significantly better than the unpruned and equiprobable lexica. In addition, pruning the lexicon allows for a faster decoding time. In order to establish a lower bound for our experiment, we report in the final value in Table 3 results from using a lexicon in which the probabilities of each pronunciation are generated by doing a forced-Viterbi on a training set. Notice that the pruned, untrained lexicons produce a word-error only about 3% worse than the fully trained lexicon.

We also ran an experiment in which we used our rules to augment the LIMSI lexicon (Lamel 1993) with extra pronunciations. We found that our pronunciations did not improve the LIMSI lexicon at all; in fact the word error rose with the new pronunciations. However, since the LIMSI pronunciations were tuned on exactly this corpus, this is perhaps not the best comparison. In addition, since most of our rules were designed with fast, spontaneous speech in mind, a read-speech corpus like the Wall Street Journal is probably the wrong testbed for this algorithm.

6. RELATED WORK

An important component of our technique, the use of a forced-Viterbi speech decoder to discover pronunciations from a corpus is based on the work of Wooters (1993), while Wesenick & Schiel (1994) independently propose a very similar forced-Viterbi-decoder-based technique which they use for measuring the accuracy of hand-written phonology. Chen (1990) and Riley (1991) model the relationship between phonemes and their allophonic realizations by training decision trees on TIMIT data for each phone specifying its surface realization in different contexts. Decision tree methods potentially allow great flexibility in the analysis of of contextual influences. We believe that a hybrid between our acoustic-trained rule-based algorithm and a decision-tree approach could prove quite powerful.

7. CONCLUSIONS

We have only performed preliminary experiments with our algorithm for assigning probabilities to phonological rules and using them to build multiple-pronunciation lexicons. However, we already have significant results on the usefulness of the rules to enable lexicon pruning. If there is not sufficient time or corpora to prune via forced-Viterbi counts, or if the test set has sources of variability not present in the training set (e.g. accent or fast rate of speech), our algorithm provides a useful method for pronunciation probability estimation. Our algorithm runs only about 3% worse than actually training the pronunciation probabilities directly with a complete forced Viterbi alignment.

We hope in the future to test out these rules on databases of fast speakers (Mirghafori *et al.* 1995) and spontaneous speech, and to create new rules modeling foreign-accent effects.

Acknowledgments

Thanks to Mike Hochberg, Nelson Morgan, Steve Renals, Tony Robinson, Florian Schiel, Andreas Stolcke, and Chuck Wooters. This work was partially funded by ICSI and an SRI subcontract from ARPA contract MDA904-90-C-5253. Partial funding also came from ESPRIT project 6487 (The Wernicke project).

8. REFERENCES

CHEN, F. 1990. Identification of contextual factors for pronounciation networks. In *IEEE ICASSP-90*, 753–756.

CMU, 1993. The Carnegie Mellon Pronouncing Dictionary v0.1. Carnegie Mellon University.

COHEN, M. H., 1989. *Phonological Structures for Speech Recognition*. University of California, Berkeley dissertation.

COMLEX, 1994. The COMLEX English Pronouncing Dictionary. copyright Trustees of the University of Pennsylvania.

KAISSE, E. 1985. *Connected Speech: the Interaction of Syntax and Phonology*. Academic Press.

LAMEL, LORI, 1993. The Limsi Dictionary.

MIRGHAFORI, NIKKI, ERIC FOSLER, & NELSON MORGAN. 1995. Fast speakers in large vocabulary continuous speech recognition: Analysis & antidotes. In *Eurospeech-95*.

RILEY, MICHAEL D. 1991. A statistical model for generating pronunciation networks. In *IEEE ICASSP-91*, 737–740.

ROBINSON, ANTHONY, 1994. The British English Example Pronunciation Dictionary, v0.1. Cambridge University.

WESENICK, MARIA-BARBARA, & FLORIAN SCHIEL. 1994. Applying speech verification to a large data base of German to obtain a statistical survey about rules of pronunciation. In *ICSLP-94*, 279–282.

WOOTERS, CHARLES C., 1993. *Lexical Modeling in a Speaker Independent Speech Understanding System*. Berkeley: University of California dissertation. Available as ICSI TR-92-062.

ZWICKY, A. 1970. Auxiliary Reduction in English. *Linguistic Inquiry* 1.323–336.

—— 1972a. Note on a phonological hierarchy in English. In *Linguistic Change and Generative Theory*, ed. by R. Stockwell & R. Macaulay. Indiana University Press.

—— 1972b. On Casual Speech. In *Eighth Regional Meeting of the Chicago Linguistic Society*, 607–615.