Learning Phonological Rule Probabilities from Speech Corpora with Exploratory Computational Phonology

Gary Tajchman, Daniel Jurafsky, and Eric Fosler
International Computer Science Institute and
University of California at Berkeley
{tajchman,jurafsky,fosler}@icsi.berkeley.edu

1 Introduction

Phonological rules have formed the basis of phonological theory for decades, although their form and their coverage of the data has changed over the years. Until recently, however, it was difficult to determine the relationship between hand-written phonological rules and actual speech data. The current availability of large speech corpora and pronunciation dictionaries has allowed us to connect rules and speech in much tighter ways. For example, a number of algorithms have recently been proposed which automatically induce phonological rules from dictionaries or corpora (Gasser 1993; Ellison 1992; Daelemans *et al.* 1994).

While such algorithms have successfully induced syllabicity or harmony constraints, or simple obligatory phonological rules, there was been much less work on non-obligatory (optional) rules. In part this is because optional rules like flapping, vowel reduction, and various coarticulation effects are postlexical and often products of fast speech, and hence have been accorded little significance in phonological theory. In part, however, this is because optional rules are inherently probabilistic. Where obligatory rules apply to every underlying form which meets the environmental conditions, producing a single surface form, optional rules may not apply, and hence the underlying form may appear as the surface form, unmodified by the rule. This makes the induction problem non-deterministic, and not solvable by the above algorithms.

While optional rules have received less attention in linguistics because of their probabilistic nature, in speech recognition, by contrast, optional rules are commonly used to model pronunciation variation. In this paper, we employ techniques from speech recognition research to address the problem of assigning probabilities to these optional phonological rules. We introduce a completely automatic algorithm that explores the coverage of a set of phonological rules on a corpus of lexically transcribed speech using the computational resources of a speech recognition system. This algorithm belongs to the class of techniques we call Exploratory Computational Phonology (ECP), which use statistical pattern recognition tools to explore phonological spaces.

We describe the details of our probability estimation algorithm and also present the probabilities the system has learned for ten common phonological rules which model coarticulation effects. Our probabilities are derived from a corpus of 7203 sentences of read speech from the Wall Street Journal (NIST 1993). We also benchmark the probabilities generated by our system against probabilities from phonetically hand-transcribed data, and show a relatively good fit. Finally, we analyze the probability differences between rule use in male versus female speech, and suggest that the differences are caused by differing average rates of speech.

2 The Algorithm

In this section we describe our algorithm which assigns probabilities to hand-written, optional phonological rules like flapping, The algorithm takes a lexicon of underlying forms and applies phonological rules to produce a new lexicon of surface forms. Then we use a speech recognition system on a large corpus of recorded speech to check how many times each of these surface forms occurred in the corpus. Finally, by knowing which rules were used to generate each surface form, we can compute a count for each rule. By combining this with a count of the times a rule did not apply, the algorithm can compute a probability for each rule.

The rest of this section will discuss each of the aspects of the algorithm in detail.

2.1 The Base Lexicon

Our base lexicon is quite large; it is used to generate the lexicons for all of our speech recognition work at ICSI. It contains 160,000 entries (words) with 300,000 pronunciations. The lexicon contains underlying forms which are very shallow; thus they are post-lexical in the sense that there is no represented relationship between e.g. 'critic' and 'criticism' (where critic is pronounced *kritik* and criticism *kritisizm*). However, the entries do not represent flaps, vowel reductions, and other coarticulatory effects.

In order to collect our 300,000 pronunciations, we combined seven different on-line pronunciation dictionaries, including the five shown in Table 1[1].

Source	Num Words	Num Base Prons	Num Expanded Prons
CMU	95,781	99,279	399,265
LIMSI	32,873	37,936	49,597
PRONLEX	30,353	30,354	81,936
BRITPRON	77,685	85,450	108,834
TTS	77,383	83,297	111,028

Table 1: Pronunciation sources used to build fully expanded lexicon.

For further information about these sources please refer to CMU (CMU 1993), LIMSI (Lamel 1993), PRONLEX (COMLEX 1994), BRITPRON (Robinson 1994), and a text-to-speech system (TTS).

We represent pronunciations with the set of 54 ARPAbet-like phones detailed in Table 2. All the lexicon sources except LIMSI use ARPABET-like phone sets[2]. CMU, BRITPRON, and PRONLEX phone sets include three levels of vowel stress. The pronunciations from all these sources were mapped into our phone set using a set of obligatory rules for stop closures [b$^\circ$, d$^\circ$, g$^\circ$, p$^\circ$, t$^\circ$, k$^\circ$], and optional rules to introduce the syllabic consonants [l̩, m̩, n̩], reduced vowels [ə, ɨ, ɚ], voiced h [ɦ], and alveolar flap [ɾ].

2.2 Applying Phonological Rules to Build a Surface Lexicon

We next apply phonological rules to our base lexicon to produce the surface lexicon. Since the rules are optional, the surface lexicon must contain each underlying pronunciation unmodified, as well as the pronunciation resulting from the application of each relevant phonological rule. Table 3 gives the 10 phonological rules used in these experiments:

One goal of our rule-application procedure was to build a tagged lexicon to avoid having to implement a phonological-rule parser to parse the surface pronunciations. In a tagged lexicon, each surface pronunciation is annotated with the names of the phonological rules that applied to produce it. Thus when the speech recognizer 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 algorithm applies rules to pronunciations recursively; when a context matches the left hand side of a phonological rule "RULE," two pronunciations are produced: one unchanged by the rule (marked -RULE), and one with the rule applied (marked +RULE). The procedure places the +RULE pronunciation on the queue for later recursive rule application, and continues trying to apply phonological rules to the -RULE pronunciation. See Figure 1 for details of the algorithm. While our procedure is not guaranteed to terminate, in practice the phonological rules we apply have a finite recursive depth.

[1] Although it was not relevant to the experiments described here, our lexicon also included two sources which directly supply surface forms. These were 13,362 hand-transcribed pronunciations of 5871 words from TIMIT (TIMIT 1990), and 230 pronunciations of 36 words derived in-house from the OGI Numbers database (Cole *et al.* 1994).

[2] The LIMSI pronunciations already included the syllabic consonants and reduced vowels. For this reason, the words found only in the LIMSI source lexicon did not participate in the probability estimates for the syllabic and reduced vowel rules.

IPA	ARPABET	ICSI	IPA	ARPABET	ICSI
b	B	B	b$^{\circ}$	-	BCL
d	D	D	d$^{\circ}$	-	DCL
g	G	G	g$^{\circ}$	-	GCL
p	P	P	p$^{\circ}$	-	PCL
t	T	T	t$^{\circ}$	-	TCL
k	K	K	k$^{\circ}$	-	KCL
ɑ	AA	AA	s	S	S
æ	AE	AE	z	Z	Z
ʌ	AH	AH	ʃ	SH	SH
ɔ	AO	AO	ʒ	ZH	ZH
ɛ	EH	EH	f	F	F
ɝ	ER	ER	v	V	V
ɩ	IH	IH	θ	TH	TH
i	IY	IY	ð	DH	DH
o	OW	OW	tʃ	CH	CH
ʊ	UH	UH	dz	JH	JH
u	UW	UW	h	HH	HH
ɑ^{w}	AW	AW	ɦ	-	HV
ɑ^{y}	AY	AY	y	Y	Y
e	EY	EY	r	R	R
ɔ^{y}	OY	OY	w	W	W
l̩	-	EL	l	L	L
m̩	-	EM	m	M	M
n̩	-	EN	n	N	N
ə	-	AX	ŋ	NG	NG
ɨ	-	IX	ɾ	-	DX
ɚ	-	AXR	silence	#H	#H

Table 2: Baseform phone set used was the ARPABET. This was expanded to include syllabics, stop closures, and reduced vowels, alveolar flap, and voiced h.

The nondeterministic mapping produces a tagged equiprobable multiple pronunciation lexicon of 510,000 pronunciations for 160,000 words. For example, Table 4 gives our base forms for the word "butter":

The resulting tagged surface lexicon would have the following entries:

b$^{\circ}$bʌɾə:+BPU +FL1; +CMU +FL1 +RV1; +PLX +FL1 +RV1
b$^{\circ}$bʌɾɚ:+TTS +FL1; +BPU +FL1; +CMU +FL1 -RV1 +RV3; +LIM +FL1; +PLX +FL1 -RV1 +RV3
b$^{\circ}$bʌt$^{\circ}$tə:+BPU -FL1; +CMU -FL1 +RV1; +PLX -FL1 +RV1
b$^{\circ}$bʌt$^{\circ}$tɚ:+TTS -FL1; +BPU -FL1; +CMU -FL1 -RV1 +RV3; +LIM -FL1; +PLX -FL1 -RV1 +RV3
b$^{\circ}$bʌt$^{\circ}$tɝ:+CMU -RV1 -RV3; +PLX -RV1 -RV3

2.3 Filtering with forced-Viterbi

Given a lexicon with tagged surface pronunciations, the next required step is to count how many times each of these pronunciations occurs in a speech corpus. The algorithm we use has two steps; PHONETIC LIKELIHOOD ESTIMATION and FORCED-VITERBI ALIGNMENT.

In the first step, PHONETIC LIKELIHOOD ESTIMATION, we examine each 20ms frame of speech data, and probabilistically label each frame with the phones that were likely to produce the data. That is, for each of the 54 phones in our phone-set, we compute the probability that the slice of acoustic data was produced by

Name	Code	Rule
Low/Mid Vowel Reduction	RV1	Unstressed [ɑæʌɔɛɝeoʊ]/ ___ → ə
High Vowel Reduction	RV2	Unstressed [iɩu]/ ___ → ɨ
R-vowel reduction	RV3	Unstressed ɝ/ ___ → ɚ
Syllabic n-reduction	SL1	[əɨ] n/ ___ → n̩
Syllabic m-reduction	SL2	[əɨ] m/ ___ → m̩
Syllabic l-reduction	SL3	[əɨ] l/ ___ → l̩
Syllabic r-reduction	SL4	[əɨ] r/ ___ → ɚ
Flapping	FL1	[t°d°] [td]/ (VOWEL) ___ [əɨɚ] → ɾ
Flapping with r	FL2	[t°d°] [td]/ (VOWEL) ___ [əɨɚ] r→ ɾ
H-voicing	VH1	h/ (VOICED) ___ (VOICED) → ɦ

Table 3: Phonological Rules

```
For each lexical item, L, do:
    Place all base pronunciations of L onto the queue Q
    While Q is not empty do:
        Dequeue pronunciation P from Q
        For each phonological rule R, do:
            If the context of R could apply to P
                Apply R to P, giving P'
                Tag P' with +R, and place onto queue Q
                Tag P with -R
        Output P with tags
```

Figure 1: Applying Phonological Rules to the Base Lexicon

that phone. The result of this labeling is a vector of phone-likelihoods for each acoustic frame.

Our algorithm is based on a multi-layer perceptron (MLP) which is trained to compute the conditional probability of a phone given an acoustic feature vector for one frame, together with 80 ms of surrounding context. Bourlard & Morgan (1991) and Renals *et al.* (1991) show that with a few assumptions, an MLP may be viewed as estimating the probability $P(q|x)$ where q is a phone and x is the input acoustic speech data. The estimator consists of a simple three-layer feed forward MLP trained with the back-propagation algorithm (see Figure 2). The input layer consists of 9 frames of input speech data. Each frame, representing 10 msec of speech, is typically encoded by 9 PLP (Hermansky 1990) coefficients, 9 delta-PLP coefficients, 9

Source	Pronunciation
TTS	bʌtɚ
BPU	bʌtə
BPU	bʌtɚ
CMU	bʌtɝ
LIM	bʌtɚ
PLX	bʌtɝ

Table 4: Base forms for “butter”

delta-delta PLP coefficients, delta-energy and delta-delta-energy terms. Typically, we use 500-4000 hidden units. The output layer has one unit for each phone. The MLP is trained on phonetically hand-labeled speech (TIMIT), and then further trained by an iterative Viterbi procedure (forced-Viterbi providing the labels) with Wall Street Journal corpora.

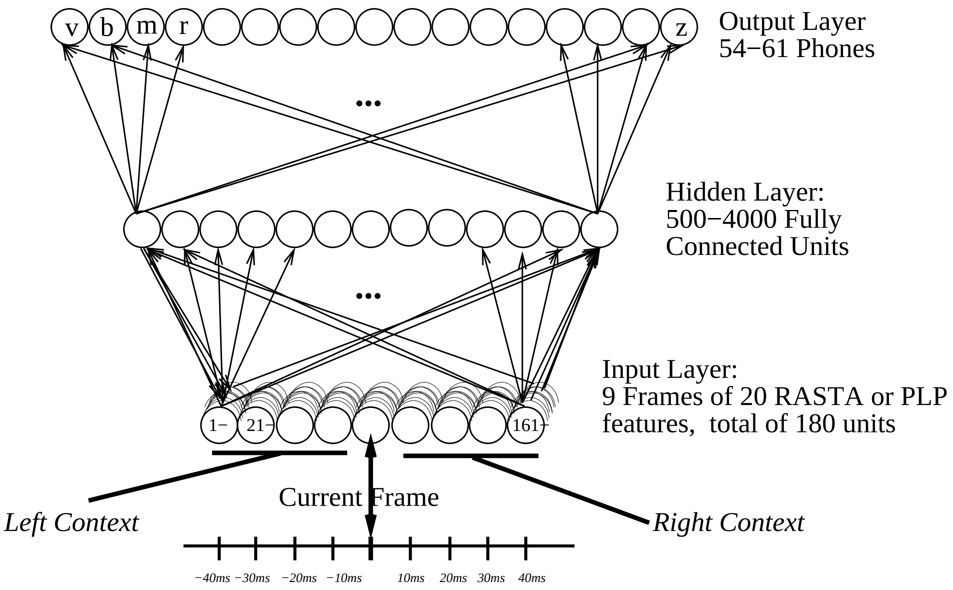

Figure 2: Phonetic Likelihood Estimator

The probability $P(q|x)$ produced by the MLP for each frame is first converted to the likelihood $P(x|q)$ by dividing by the prior $P(q)$, according to Bayes' rule; we ignore $P(x)$ since it is constant here:

$$P(x \mid q) = \frac{P(q \mid x)P(x)}{P(q)} \tag{1}$$

The second step of the algorithm, forced-Viterbi alignment, takes this vector of likelihoods for each frame and produces the most likely phonetic string for the sentence. If each word had only a single pronunciation and if each phone had some fixed duration, the phonetic string would be completely determined by the word string. However, phones vary in length as a function of idiolect and rate of speech, and of course the very fact of optional phonological rules implies multiple possible pronunciations for each word.

The Viterbi algorithm is a dynamic programming search, which works by computing for each phone at each frame the most likely string of phones ending in that phone. Consider a sentence whose first two words are "of the", and assume the simplified lexicon in Figure 3.

Each pronunciation of the words 'of' and 'the' is represented by a path through the probabilistic automaton for the word. For expository simplicity, we have made the (incorrect) assumption that consonants have a duration of 1 frame, and vowel a duration of 2 or 3 frames. The algorithm analyzes the input frame by frame, keeping track of the best path of phones. Each path is ranked by its probability, which is computed by multiplying each of the transition probabilities and the phone probabilities for each frame. Figure 4 shows a schematic of the path computation. The size of each dot indicates the magnitude of the local phone probability. The maximum path at each point is extended; non-maximal paths are pruned.

The result of the forced-Viterbi alignment on a single sentence is a phonetic labeling for the sentence (see Figure 5 for an example), from which we can produce a phonetic pronunciation for each word. By running this algorithm on a large corpus of sentences, we produce a list of "bottom-up" pronunciations for each word in the corpus.

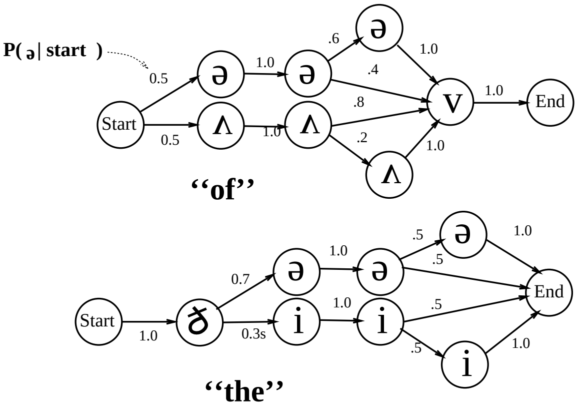

Figure 3: Pronunciation models for “of” and “the”

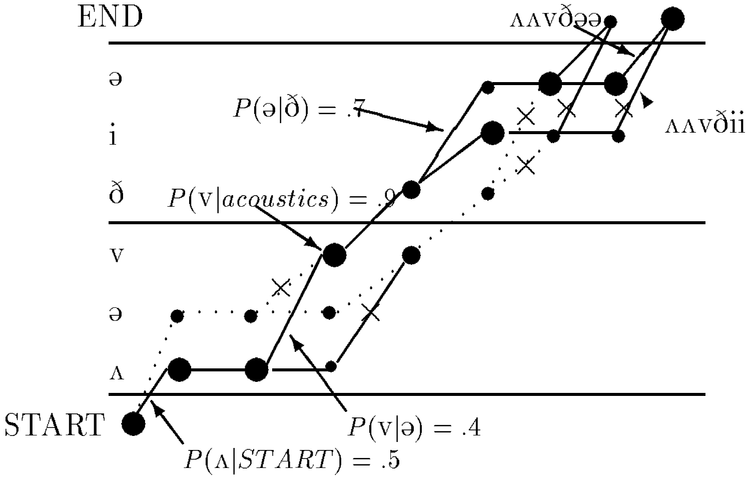

Figure 4: Computing most-likely phone paths in a Forced-Viterbi alignment of ‘of the’

2.4 Rule probability estimation

The rule-tagged surface lexicon described in §2.1 and the counts derived from the forced-Viterbi described in §2.3 can be combined to form a tagged lexicon that also has counts for each pronunciation of each word. Following is a sample entry from this lexicon for the word *Adams* which shows the five derivations for its single pronunciation:

Adams: *ae dx ax m z*: count=2

derivation 1: `+ATS +FL1 -SL2`
derivation 2: `+BPU +FL1 -SL2`
derivation 3: `+CMU +FL1 +RV1 -SL2`
derivation 4: `+LIM +FL1 -SL2`
derivation 5: `+PLX +FL1 -SL2`

Each pronunciation of each word in this lexicon is annotated with rule tags. Since each pronunciation may be derived from different source dictionaries or via different rules, 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

new	*york*	*city's*	*fresh*	*kills*	*landfill*	*on*	*staten*
n y uw	y ao r kcl k	s ih tcl t iy z	f r eh sh	kcl k ih l z	l ae n dcl f ih l	aa n	s tcl t ae tcl t en

island	*for*	*one*	*dumps*	*four*	*million*	*gallons*	*of*
ay l ax n dcl	f ao r	w ah n	dcl d ah m pcl p s	f ao r	m ih l y ix n	gcl g ae l ax n z	ax f

toxic	*liquid*	*into*	*nearby*	*freshwater*
tcl t aa kcl k s ix kcl	l ih kcl k w ih dcl	en tcl t uw	n ih r bcl b ay	f r eh sh w ao dx axr

streams	*every*	*day*
s tcl t r iy m z	eh v r iy	dcl d ey

Figure 5: A forced-Viterbi phonetic labelling for a Wall Street Journal sentence

did not.

To produce the initial rule probabilities, we need to count the number of times each rule applies, out of 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:

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

This could be computed from the tags as :

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

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 simply by multiplying together the probability of each of the applications or non-applications of the rule. Let

- *DERIVATIONS(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).
- $P(d|p)$ is the probability of the derivation d of pronunciation p.
- PRON is the set of pronunciations derived from the forced-Viterbi output.

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

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

Since we have no prior knowledge, we make the zero-knowledge initial assumption that $P(d|p) = \frac{1}{|DERIVATIONS(p)|}$. The algorithm can the 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 follows:

- for every word/pron pair $P \in PRON$ from forced-Viterbi alignment

    let $DERIVATIONS(P)$ be the set of rule derivations that could produce P

```
    for every $d \in DERIVATIONS(P)$
      for every rule $R \in d$
      if $(R = +RULE)$ then increment $ruleapp\{RULE\}$ by $\frac{1}{|DERIVATIONS(P)|}$
      else increment $rulenoapp\{RULE\}$ by $\frac{1}{|DERIVATIONS(P)|}$
• for every rule RULE
    $P(RULE) = \frac{ruleapp(RULE)}{ruleapp(RULE)+ruleapp(RULE)}$
```

3 Results

We ran the estimation algorithm on 7203 sentences (129,864 words) read from the Wall Street Journal. The corpus (1993 WSJ Hub 2 (WSJ 0) training data) consisted of 12 hours of speech, and had 8916 unique words. Table 3 shows the probabilities for the ten phonological rules described in §2.2.

Name	Code	Rule	Probability
Low/Mid Vowel Reduction	RV1	Unstressed [ɑæʌɔɛɝeoɷ]/___ → ə	.60
High Vowel Reduction	RV2	Unstressed [iɪu]/___ → ɨ	.57
R-vowel reduction	RV3	Unstressed ɝ/___ → ɚ	.74
Syllabic n-reduction	SL1	[əɨ] n/ ___ → n̩	.35
Syllabic m-reduction	SL2	[əɨ] m/ ___ → m̩	.35
Syllabic l-reduction	SL3	[əɨ] l/ ___ → l̩	.72
Syllabic r-reduction	SL4	[əɨ] r/ ___ → ɚ	.77
Flapping	FL1	[t°d°] [td]/(VOWEL) ___ [əɨɚ] → ɾ	.87
Flapping with r	FL2	[t°d°] [td]/(VOWEL) ___ [əɨɚ] r→ ɾ	.92
H-voicing	VH1	h/ (VOICED) ___ (VOICED) → ɦ	.92

Table 5: Results of the Rule-Probability-Estimation Algorithm

Note that all of the rules are indeed quite optional; even the most commonly-employed rules, like flapping and h-voicing, only apply on average about 90% of the time. Many of the other rules, such as the reduced-vowel or reduced-liquid rules, only apply about 50% of the time.

We next 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 6 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 closeness of the probabilities is quite encouraging.

Figure 7 breaks down our automatically generated rule probabilities for the Wall Street Journal corpus into male and female speakers. Notice that many of the rules seem to be employed more often by men than by women. For example, men are about 5% more likely to flap, more likely to reduce vowels *ih* and *er*, and slightly more likely to reduce liquids and nasals.

Since these are are coarticulation or fast-speech effects, our initial hypothesis was that the difference between male and female speakers was due to a faster speech-rate by males. By computing the weighted average seconds per phone for male and female speakers, we found that females had an average of 71 ms/phone, while males had an average of 68 ms/phone, a difference of about 4%, quite correlated with the similar differences in reduction and flapping.

4 Related Work

Our algorithm for phonological rule probability estimation synthesizes and extends earlier work by (Cohen 1989) and (Wooters 1993). The idea of using optional phonological rules to construct a speech-recognition

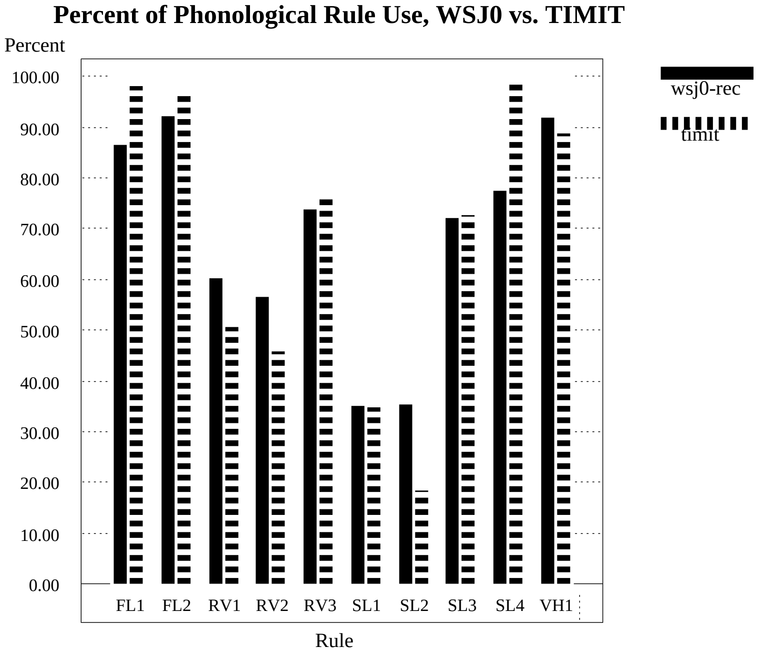

Figure 6: Automatic vs Hand-transcribed Probabilities for Phonological Rules

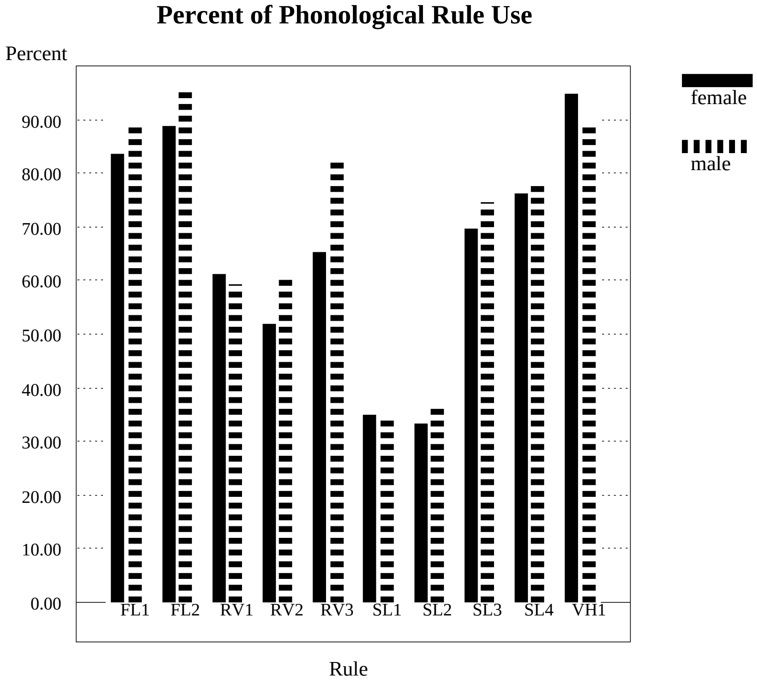

Figure 7: Male vs Female Probabilities for Phonological Rules

lexicon derives from Cohen (1989), who applied optional phonological rules to a baseform dictionary to produce a surface lexicon and then used TIMIT to assign probabilities for each pronunciation. The use of a forced-Viterbi speech decoder to discover pronunciations from a corpus was proposed by Wooters (1993). 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. A decision tree is learned for each underlying phoneme specifying its surface realization in different contexts. These completely automatic techniques, requiring no hand-written rules, can allow a more fine-grained analysis than our rule-based algorithm. However, as a consequence, it is more difficult to extract generalizations across classes of phonemes to which rules can apply. We think that a hybrid between a rule-based and a decision-tree approach could prove quite powerful.

5 Conclusion and Future Work

Although the paradigm of Exploratory Computational Phonology is only in its infancy, we believe our rule-probability estimation algorithm to be a new and useful instance of the use of probabilistic techniques and spoken-language corpora in computational linguistics. We plan in future work to address a number of shortcomings of these experiments, for example including some spontaneous speech corpora, and looking at a wider variety of rules.

In addition, we have extended our algorithm to induce new pronunciations which generalize over pronunciations seen in the corpus (Wooters & Stolcke 1994). We now plan to augment our probability estimation to use the pronunciations from this new HMM-induction-based generalization step, This will require extending our tag-based probability estimation step to parse the phone strings from the forced-Viterbi.

In other current work we have also been using this algorithm to model the phonological component of the accent of non-native speakers. Finally, we hope in future work to be able to combine our rule-based approach with more bottom-up methods like the decision-tree or phonological parsing algorithms to induce rules as well as merely training their probabilities.

Acknowledgments

Thanks to Mike Hochberg, Nelson Morgan, Steve Renals, Tony Robinson, Florian Schiel, Andreas Stolcke, and Chuck Wooters

References

Bourlard, H., & N. Morgan. 1991. Merging multilayer perceptrons & Hidden Markov Models: Some experiments in continuous speech recognition. In *Artificial Neural Networks: Advances and Applications*, ed. by E. Gelenbe. North Holland Press.

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.

Cole, R. A., K. Roginski, & M. Fanty., 1994. The OGI Numbers Database. Oregon Graduate Institute.

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

Daelemans, Walter, Steven Gillis, & Gert Durieux. 1994. The acquisition of stress: A data-oriented approach. *Computational Linguistics* 208.421–451.

Ellison, T. Mark, 1992. *The Machine Learning of Phonological Structure*. University of Western Australia dissertation.

Gasser, Michael, 1993. Learning words in time: Towards a modular connectionist account of the acquisition of receptive morphology. Draft.

Hermansky, H. 1990. Perceptual linear predictive (plp) analysis of speech. *J. Acoustical Society of America* 87.

Lamel, Lori, 1993. The Limsi Dictionary.

NIST, 1993. Continuous Speech Recognition Corpus (WSJ 0). National Institute of Standards and Technology Speech Disc 11-1.1 to 11-3.1.

Renals, S., N. Morgan, H. Bourlard, M. Cohen, H. Franco, C. Wooters, & P. Kohn. 1991. Connectionist speech recognition: Status and prospects. Technical Report TR-91-070, ICSI, Berkeley, CA.

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.

TIMIT, 1990. TIMIT Acoustic-Phonetic Continuous Speech Corpus. National Institute of Standards and Technology Speech Disc 1-1.1. NTIS Order No. PB91-505065.

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, CA: University of California dissertation. available as ICSI TR-92-062.

Wooters, Chuck, & Andreas Stolcke. 1994. Multiple-pronunciation lexical modeling in a speaker-independent speech understanding system. In *ICSLP-94*. To appear.