CSME 206A Natural Language & Speech Processing **Spring Semester**

Lecture 6: Computational Phonology

Lecturer: K.R. Chowdhary *: Professor of CS*

6.1 Introduction

Formal, computational models are important in linguistics for two main reasons. First, the project of making our vague ideas about language elegant and fully formal. It improves our understanding of the real claims of the grammar, and enables careful comparisons of competing ideas. Second, the best models we have of human language acquisition and use are computational. That is, they regard people using language as going through some changes which can be modelled as formal derivations. The idea that the relevant changes in language learning and language use are derivations of some kind is an empirical hypothesis which may well be false, but it is the best one we have. The main project of linguistic theory is to provide this computational account.

Since theoretical linguistics provides formal generative models of language, it implicitly treats human language learners and language users as computers. Computers are useful in the development of linguistics in just the way that they are useful in physics or biology: they facilitate calculations. These calculations are not the reason that our pursuit is called computational. The reason the subject at hand is called computational phonology is that we adopt the programmatic hypothesis that the abilities we are modelling are computational.

Symbols can be used to represent sounds that make words, and the pronounciation of a word can be represented by these symbols, called *phonetic alphabets*. How the words are spelled in a language culture specific. The sound based writing, i.e, spoken words are composed of smaller units of speech, underlines the modern theory of *Phonology*.

The *Computational Phonology* comprises: algorithms for *speech recognition*, *speech synthesis*, and the *linguistics*. The aim of speech recognition is to take sound wave as input and produce as output a string of words. Where as text-to-speech (TTS) synthesis is to take input the words and produce output the acoustic sound waveform. The important applications of TTS are: speaking dictionaries, speech based interface with computers, and voice based input for disabled.

We will study how the words are pronounced in terms of individual units of speech called "phones". So, it appears that for the pronounciation of a word if we concatenate the phones of that word, it should serve the purpose. The challenge is that different phones are pronounced differently in different contexts. The computational phonology is study of computational means for modeling phonological rules. We will be making use of FSTs (Finite State Transducers), studied earlier, for computational phonology [Jurafsky].

6.1.1 Speech Sound and its Phonetic Transcription

The study of pronunciation is called *phonetics*. We will represent pronunciation of words as strings of symbols, which represent phones or segments. These symbols are phonetic symbols. For example, phone p

corresponds to letter p and phone l corresponds to letter l. The IPA (International Phonetic Alphabets) are used as standard for phonetics. Some examples of IPA are given in the Table 6.1.

Table 6.1: Examples of IPA

IPA-Symbol (Consonants)	Word x	IPA Transcript of x
1. [p]	Parsely	[parsli]
2. [k]	Catnip	[kaetnie]
3. [d]	dill	[dil]
4. [θ]	thistle	[θisl]
5. [v]	clove	[kloυv]
6. [z]	hazlnet	[heizlnut]
IPA-Symbol (Vowels)	**Word x**	**IPA Transcript of x**
1. [i]	lily	[lili]
2. [ei]	daisy	[deizi]
3. [υ]	woodruff	[w$\upsilon druf$]
4. [u]	tulip	[tulip]
5. [aυ]	sunflower	[sunflaυr]
6. [oυ]	lotus	[loυtes]

When we speak, these phones are generated by sound organs of human mouth (see chapter 1 for details).

6.1.2 Relation between Phonological Rules and Phonemes

A phoneme is the smallest unit of sound in a word that makes a difference in its pronunciation, as well as its meaning, from other word. A phoneme does not have any inherent meaning by itself, but when you put phonemes together, they can make words. The 'm' sound, often written as /m/, is an example of phoneme.

When comparing phonemes and morphemes, morphemes are the smallest meaning bearing units of a language. Phonemes are the smallest units of sound in a language. In chinese, each phoneme corresponds to a morpheme and each morpheme corresponds to a phoneme. A syllable is cluster of sounds with at least one vowel.

Example. Phonemes in the word "Ball."
There are three phonemes in "ball". These are: /b/, /aw/, and /l/. Each of its sound affect the meaning of the word. □

Allphone vs. Phoneme The difference between these two is that changing the phoneme changes the meaning of the word, where as changing the allphone changes the sound of the realization of the word, but does not change the meaning of the word. A further explanation is – a phonemes is a set of allphones or individual non-contrastive speech symbols. Allphones are sounds, whilst, a phoneme is a set of sounds. Allphones are relatively similar sounds, which are in mutually exclusive or complementary distributive.

Statistics of some Languages The languages !XOO has 112 phonemes, and spoken in the African country Botswana, it has 31 vowels. The language with fewest sounds (phonemes) is *Rotokas*, which has got 11 phonemes, and has 12 letters, and spoken by approx. 4300 people in Papua New Guinea. A language with largest number of alphabets is *Khmer*, with 74 letters, and it is official language of Cambodia, spoken by 1.2 million people. The language with fewest words is: Taki-Taki (also called, *Sranan*), which has 340 words, and spoken in south American country of *Suriname*.

All the identical phones, for example, all the occurrences of [t] are not created equal. Accordingly, the phones are pronounced differently in different contexts. For example, [t] in t*unafish* and [t] in st*arefish*, are pronounced differently. In the first, it is *aspirated* and in second it is not. The aspirated is period of voicelessness after the [t] closure.

There are other contextual variants of [t]. For example, when [t] occurs between two vowels, particularly when the first is stressed, it is pronounced as *tap*. In tap, the tongue is curled and pressed against the alveolar ridge. Another variant of [t] occurs before the *dental consonant* [θ]. Here [t] becomes [t̪], where the tongue touches the back of the teeth. So the challenge is to represent the relation between [t] and its different variants of pronunciations, in different contexts. We do it by the abstract class, called, *phoneme*, which is realized as different *allophones* in different contexts. We write the phonemes inside the slashes. So /t/ is phoneme whose allophones includes [t^h], [ɾ], and [t̪].

The relationship between phonemes and its allophones is specified as phonological rules, e.g., the *Chomsky and Hall* rule indicated as:

$$/t/ \rightarrow [\underline{t}]/ - - - \theta \tag{6.1}$$

In above, the symbols /t/ is sound produced by letter *t* and [t̪] is the *allophone* of *phoneme* [t]. The surface allophone appears to the right of arrow, i.e., [t̪] and phonetic environment is indicated by symbols surrounded by "– – –".

A flap is a constant sound produced by a single quick flip of the tongue against the upper part of the mouth, often heard as a short *r* in Spanish language.

The following is version of *flapping rule*[1]. The rule indicates that, when a word ends with *dental consonant*, i.e., having the sound like of letter *t* in *thistle*, then replace sound /t/ by its allphone [t̪].

$$/\left\{ \begin{array}{c} t \\ d \end{array} \right\}/ \rightarrow [ɾ]/V^{'} - - - V \tag{6.2}$$

The symbols $V^{'}$ and V stand for stressed and unstressed vowels, respectively. The rule indicates that the sounds /t/ and /d/ will become the sound of allphone [ɾ], when former is appearing between stressed and unstressed vowels.

6.2 Phonological Rules and Transducers

The phonological rules can be implemented by transducers as we did in the case of morphological analysis. We will use a system like in two level morphology. As a first example, we use the *flapping* rule:

[1]Flap is Verb, sound produced when it moves its wings up and down during the time of flying or when it runs before flying. Similar word is flutter.

$$/t/ \rightarrow [\text{ɾ}]/V^{'} - -V \tag{6.3}$$

The corresponding finite state transducer is shown in Fig. 6.1.

In phonetics, a *flap* or *tap* is a type of consonantal sound, which is produced with a single contraction of the muscles so that one articulator (such as the tongue) is thrown against another.

The main difference between a flap and a stop is that in a flap, there is no buildup of air pressure behind the place of articulation, and consequently no release burst. Otherwise a flap is similar to a brief stop.

Flaps also contrast with trills, where the air-stream causes the articulator to vibrate. Trills may be realized as a single contact, like a flap, but are variable, whereas a flap is limited to a single contact.

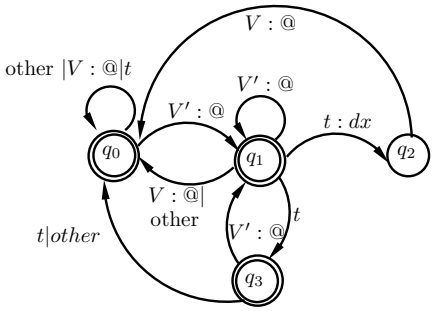

Figure 6.1: Flapping Transducer.

In the Fig. 6.1, representing FST for *flapping sound*. The symbol *dx* (in ARPAbet) indicates a flap, "other" means any feasible pair not used any where, @ indicates any symbol not used on any edge of the graph. In IPA *dx* is represented as [ɾ]. The FST accepts any string in which there is *flaps* (sound) in the correct places (i.e. after a stressed vowel, and before an unstressed vowel), and rejects all other strings. The examples of flapping sound words are: *butter*, *mutter*, and *ruttr*, where *u* (for V') is stressed and *e* is unstressed vowel (for V). The *tt* is flapping sound. However, "letter" is not a flapping sound, though *tt* is surrounded by vowels because there is no pair of stressed and unstressed pair of vowels, as both together are vowels (letter 'e').

A *flap* is a consonant. The flap sound occurs, e.g., when number of birds flock take-off for fly, and the sound is produced by feathers together for all these is flapping sound.

6.3 Mapping text-to-phones for TTS

An important application of text-to-phones is *pronunciation dictionaries*. These dictionaries are used for TTS (text-to-speech) systems. The simplest pronunciation dictionaries are to have a list of words and their pronunciations, like shown in Table 6.2.

However, the challenge is due to large number of words, over one hundred thousand at the minimum. The other challenge is that two distinct words are many times pronounced the same, called *homographs*. For example, the verb *wind* (You should wind up soon!) is pronounced [waind], while the noun *wind*, in "Tough winter wind", is pronounced as [wind]. Hence, the correct context should be supplied to TTS, to identify as which form of that word is right at that moment.

The other challenge is proper nouns (the name of person, or place, or company, etc). These account for millions in number.

Table 6.2: Mapping Text to phones in a pronunciation Dictionary.

Word	Pronunciation
cat	[kaet]
cats	[caets]
fox	[fax]
foxes	[fak.siz]
pigs	[pigz]
geese	[gis]

6.4 TTS Beyond the Lookup Dictionaries

The method discussed above is fine if all the words are built up in the dictionary, in advance. However, it is not possible to represent every word in dictionary, e.g., names of places, multilingual words, names of persons or things in different dialects, and numbers. Both the speech synthesis (TTS) and automatic speech recognition (ASR) systems need to be able to guess the pronunciation of words that are not in their dictionary.

6.4.1 FST-based Pronunciation Lexicons

Early work in pronunciation modeling to TTS relied heavily on letter to sound rules, where each rule specified how a letter or combination of letters was mapped to phones; as given in the Table 6.3.

Table 6.3: Text fragment v/s Pronunciation mapping.

Text Fragment	Pronunciation (Phone)
-p-	[p]
-ph-	[f]
-phe-	[fi]
-phes-	[fiz]
-place-	[pleis]
-placi-	[pleisi]
-plement-	[pliment]

These systems comprised a long list of such rules and a very small dictionary of exceptions (often function words such as a, are, as , both, does, etc.).

More recent systems rely on very large dictionaries, with letter to sound rules only used for small number of words that are neither in the dictionary nor are morphological variants of words in dictionary.

The new approach, which is based on FST has following components:

1. An FST to represent pronunciation of individual words and morphemes in a lexicon (Table 6.4). Note that it requires two tapes.
2. FST to represent possible sequence of morphemes,

3. Individual FST for each pronunciation rule (e.g., for -s in plural),
4. Letter to sound rules for names and acronyms, as they are not part of the lexicon. Note that it requires two tapes.
5. Default letter to sound rules for other unknown words.

The first of above is simple extension of what was discussed earlier. The extended version is given in the Table 6.4, which corresponds to lexical tapes (top most) in Fig. 6.3. The arcs for regular and irregular nouns in Fig. 6.2 corresponds to the mapping shown in Table 6.4, where, each symbol in lexicon is a pair of symbols separated by "|", the first representing "orthographic" lexical entry and second "phonological" lexical entry[2].

Table 6.4: Phonetic - Transcription

Orthographic-lexicon (For *Regular noun*)	Phonological Lexicon
cat	$c\|k\ a\|ae\ t\|t$
fox	$f\|f\ o\|a\ x\|ks$
dog	$d\|d\ o\|a\ g\|g$
For Irregular noun	
goose	$g\|g\ oo\|a\ s\|s\ e\|\varepsilon$

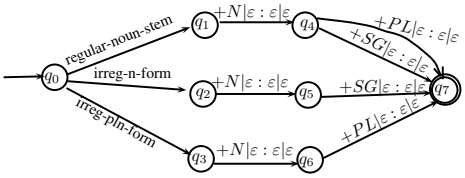

Figure 6.2: FST for text-to-speech.

The FST shown in Fig. 6.2 is for nomial singular and plural inflection. The automata adds the morphological features [+N], [+PL], and [+SG] at the lexical level and also adds the plural suffix $s|z$ at the intermediate level.

The final FST is produced by combining the composition shown in Table 6.4 along with the FST shown in Fig. 6.2. The Fig. 6.3 shows the mapping between the lexicon and surface form for orthography and phonology simultaneously, hence there are two tapes at every level, one for sound and other for spelling. The system can be used to map from a lexical entry to its surface pronunciation or from surface orthography to surface pronunciation via the lexical entry.

The FST in Fig. 6.2 shows two level, an underlying or "lexical" level and Intermediate level. The only thing which remains to be added is transducers which apply the spelling rules; and pronunciation rules to map intermediate level into surface level. These include the spelling and pronunciation rules discussed in ε-insertion and e.g., flapping sound rules.

Recall that lexicon FST maps between the "lexical" level, with its stems and morphological features, and an "intermediate" level which represents a simple concatenation of morphemes. Then a host of FSTs, each

[2]In entry $a|b$, where a is orthographic lexical entry, and b is phonological lexical entry.

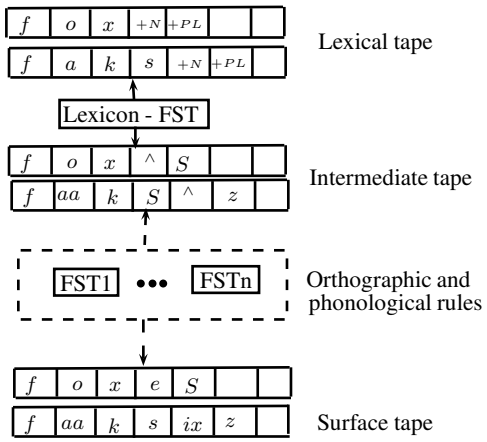

Figure 6.3: Mapping between lexicon and surface.

representing either representing a single spelling rule constraint, or a single morphological constraint, all run in parallel so as to map between this intermediate level and the surface level.

Note that, each level has both orthographic and phonological representations. For text to speech applications, in which the input is lexical form (e.g., for text generation, where the system knows the identity of the word, its part-of-speech, its inflection, etc), the cascade of FTSs can map from lexical form to surface pronunciations. For text-to-speech applications in which the input is a surface spelling (e.g., for "reading text out loud" applications), the cascade of FTSs can map from surface orthographic form to surface pronunciation va underlying lexical form.

Review Questions

1. What you understand by phonemes and allphones? Explain the difference.
2. What is flapping rule?
3. Which language has largest number of phonemes? Which has smallest number of phonemes? Which has smallest number of alphabets? Which has largest number of alphabets? In which of the world these languages are spoken?
4. What are the words called which are pronounced the same?
5. How such words are correctly pronounced?

Exercises

1. Explain, how a FA can be useful in Computational Phonology. Give examples.
2. What are different sounds of phoneme [t], which occur due to its contexts?
3. Explain the flapping sound with examples of words that produce flapping sound when pronounced.
4. Give examples of five words having stressed and unstressed vowels.

5. Give examples of FST to produce the pronunciation of suitable words using Chomsky-and-Hall rule.

References

[1] Jurafsky D and Martin J, *Speech and Language Processing, 3rd Ed.*, Pearson India, isbn: 3257227892, Nov. 2005.

[2] Buchsbaum, Adam L. and Giancarlo, Raffaele, Algorithmic Aspects in Speech Recognition: An Introduction, *J. Exp. Algorithmics*, Jan. 1997, Vol. 2, `http://doi.acm.org/10.1145/264216.264219`, doi = 10.1145/264216.264219, ACM, USA.