The Diaphonemes of English Vowels: A Prolog Implementation

Julian Wilkinson

UC Davis

LIN 177 Winter 2018

The Problem

In Sociolinguistics, it is well known that dialects vary systematically. I aim to model this phenomenon in English vowels across different dialects by exploring the concept of diaphonemes, or abstract phonological units that identifies a correspondence between related sounds of two or more varieties of a language or language cluster (Crystal, 2011). My model will be able to translate the vowels within a broad IPA transcription of a word in a certain dialect of English into another dialect of English. This will be done using information from phontetican John Wells's books *The Accents of English* (1982). Furthermore, I will implement diaphonemes within the translator, which will allow the program to give the variations of the word across the implemented dialects.

The Approach

I first collected information about the vowels used across different English dialects from *The Accents of English* (Wells, 1982). Some of this data can be seen in Figure 1. There is an entry corresponding to Wells's standard lexical sets, or the example words in the final column. These represent all the possible vowel distinctions in known English dialects.

	Wells Index	Canadian English	Irish English	Received Pronunciation	Standard American English	Example
1	w1	ɪ	ɪ	ɪ	ɪ	Kit
2	w2	ɛ	ɛ	e	ɛ	Dress
3	w3	æ	æ	æ	æ	Trap
4	w4	ɑ	ɒ	ɑ	ɑ	Lot
5	w5	ʌ	ʌ	ʌ	ʌ	Strut

6	w6	ʊ	ʊ	ʊ	ʊ	Foot
7	w7	æ	æ/ aː	ɑː	æ	Bath
8	w8	ɑ	ɒ/ ɔː	ɒ	ɔ	Cloth
9	w9	ɜr	ʌr/ ɛr	ɜː	ɜr	Nurse
10	w10	i	iː	iː	i	Fleece
11	w11	eɪ	eː	eɪ	eɪ	Face
12	w12	ɑ	aː	ɑː	ɑ	Palm
13	w13	ɑ	ɔː	ɔː	ɔ	Thought
14	w14	o	oː	əʊ	o	Goat
15	w15	u	uː	uː	u	Goose
16	w16	aɪ	aɪ	aɪ	aɪ	Price
17	w17	ɔɪ	ɔɪ	ɔɪ	ɔɪ	Choice
18	w18	ɑʊ	aʊ	aʊ	aʊ	Mouth
19	w19	ɪr	iːr	ɪə	ɪr	Near
20	w20	ɛr	eːr	ɛə	ɛr	Square
21	w21	ɑr	aːr	ɑː	ɑr	Start
22	w22	or	ɔːr	ɔː	ɔr	North
23	w23	or	oːr	ɔː	or	Force
24	w24	ʊr	uːr	ʊə	ʊr	Cure
25	w25	i	iː	i	i	Happ***y***
26	w26	ər	ər	ə	ər	Lett***er***
27	w27	ə	ə	ə	ə	Comm***a***

Vowels across Partial Set of English Dialects

Figure 1. Table of various dialects and their vowels according to *Accents of English* (Wells, 1982). Wells Index

added in along with standard lexical sets provided by Wells. Full set used in program available in Appendix A.

Before transforming this data into Prolog facts, I first considered how I wanted the data to be linked together. There were two approaches I could take:

Approach One: link each dialect's vowels together like a web, or, *Approach Two*: link each dialect together through one central dialect. *One* was overly complicated and the number of predicates I would have to have added with each new dialect would have grown at an immense rate.

This left *Approach Two*, however, just from looking at Canadian English's w12 and w13 entries, we can see that vowels are not always uniquely mapped to Wells's lexical sets. And, even if one could be narrowed down, this raised the question of principledness, as the linking dialect would effectively become diaphonemes, as they linked variations of a sound, which also brought the implications of what is the "standard" dialect of English and should it be determined solely on a basis of the unicity of its vowels relative to other dialects. Last, there were instances where there were multiple instances of a vowel within a dialect relative to lexical sets. An example of this is in Irish English where the vowel in the word *nurse* can be realized as either /ʌr/ or /ɛr/.

Instead, I based the link off of the order in which Wells presented his the vowels of English for a specific dialect, and their corresponding example word. For example, for every dialect gathered from *Accents of English*, there was an entry with the example word followed by the vowel's IPA transcription; each example word is in the same order across dialects and a number was then assigned to each word and its corresponding vowel. Yielding 27 possible unique instances of vowels occuring in English, this can be seen further in Figure 1, where each

example word has the possibility of being unique in one dialect and is assigned a value in the *Wells Index* column. It should be noted that the term *Wells Index* was is coined for this paper only and is not used elsewhere. These Wells Indices, however, effectively have become diaphonemes, as they are abstract phonological units that identify a correspondence between related sounds in different dialects.

Now that there was principled way to link the vowels of the dialects together, the facts were formed. The diaphoneme facts have an arity of four, and take the parameters of the dialect's vowel in IPA, the dialect's name, the corresponding Well's Index, and the name "wells_index" listed as the corresponding "dialect". An example fact can be seen in Figure 2.

```
diaphoneme(['ɪ'],['canadian_english'],[['w'],['1']],['wells_index']).
```

An Example Prolog Fact From dialects.swipl

Figure 2. This is an example fact for Canadian English. The predicate name is diaphoneme. Here the letter /ɪ/ from Canadian English is mapped to the Wells Index 'w1'. A fact was created for each entry in the table, corresponding to a unique Wells Index.

Next, the actual translator itself was created. The translator was based off a typical list copier and recursed through the sublists in a list. The predicate itself was named diaphonemeSwap (arity four) and takes the word to be translated, the words original dialect, the translated word, and the dialect to be translated into. There are multiple instances of the predicate declared for the various cases of recursion. Figure 3 has code from the four cases:

```
% Case 0 (Base Case)
diaphonemeSwap([], _, [], _).
```

```
% Case 1
diaphonemeSwap([Original_Phoneme|Rest], FromDialect, [Original_Phoneme|ToAdd], ToDialect) :-
        cns(Original_Phoneme), %Original_Phoneme is consonant
        !,
        diaphonemeSwap(Rest, FromDialect, ToAdd, ToDialect).
% Case 2
diaphonemeSwap([Original_Phoneme|Rest], FromDialect, [Changed_Phoneme|ToAdd], ToDialect) :-
        not(cns(Original_Phoneme)),
        diaphoneme(Original_Phoneme, FromDialect, Wells_Index, ['wells_index']),
        diaphoneme(Changed_Phoneme, ToDialect, Wells_Index, ['wells_index']),
        diaphonemeSwap(Rest, FromDialect, ToAdd, ToDialect).
% Case 3
diaphonemeSwap([Original_Phoneme|Rest], FromDialect, [Original_Phoneme|ToAdd], ToDialect) :-
        not(cns(Original_Phoneme)),
        not(diaphoneme(Original_Phoneme, FromDialect, Wells_Index, ['wells_index']);
                diaphoneme(_, ToDialect, Wells_Index, ['wells_index'])),
        !,
        diaphonemeSwap(Rest, FromDialect, ToAdd, ToDialect).
```

Four Cases in Translating Words

Figure 3. The four cases in translating a word.

Case 0 was created was created as the base case for when the list was finished being recursed on.

Case 1 was created to detect if a sublist to be copied was a consonant. Consonants were defined in prolog in PhonesAndClasses.swipl much as they were in FullProperties.swipl from lecture.

Case 2 did two things. The first was if the sublist was check if the sublist was a consonant. If it was not, then it proceeded and mapped the sublist to a Wells Index from the

original dialect. From there it mapped the Wells Index to the corresponding vowel in the other dialect. This was also the only case that was not cut, which allowed for cases of multiple vowels being mapped to a Wells Index via backtracking.

Case 3 was reached if the sublist was not a consonant or yielded a wells index. In this case, the sublist was just copied to the new list. For every case but 0, recursion proceeded at the end of the predicate.

Finally, for testing purposes, two more predicates were created. The first, *saeToOthers*, takes the IPA transcription of Wells's example words in Standard American English and translates them to the IPA in all other available dialects and the second, *wellsToOthers*, takes the IPA transcription of a word with the vowels as Wells Indices, or diaphonemes, and then realizes them in every available dialect.

diaphonemeSwap can also be used for manual entry to translate. It should be noted, however that some vowels must be listed specially, where diphthongs, extended vowels, and rhotic-final vowels must be nested inside one list. Diacritics are also listed in separate entries. This will be discussed in the next section.

Issues and Shortcomings

The first issue is that due to my own capacity to write Prolog programs, I could not create the main function, *diaphonemeSwap*, without making it necessary for word inputted to have their vowels grouped in their own list. This violates principledness in that this is not necessarily how vowels are grouped in the brain. Next, this program is limited to both the number of available dialects in *Accents of English* and the capacity of its author. There are many more dialects of English, and the data used is based on the research of a single person, which leaves room for

error. In terms of IPA, this program struggles in that due to the nature of character encoding, diacritics such as ¨ are placed in their own sublist, aside from the vowel they are altering. Finally, there are some words were it may appear as though a word is duplicated in translation when backtracking. In this case, however, if one looks closer they can see that this is a duplicate, but it is a duplicate mapping to the diaphonemes listed. I.e., the dialect the word is originating has vowel entries that are not unique, and the dialect the word it is being translated may also have copies in the corresponding vowels being mapped to. Though not necessarily an error, the repeated output could be considered to border overgeneration.

References

Crystal, D. (2011). A dictionary of linguistics and phonetics. Malden, Mass.: Blackwell.

Wells, J. (1982). *Accents of English* (Vols 1-3). Cambridge [Cambridgeshire] ; New York: Cambridge University Press.

Appendix A

	Wells Index	Australian English	Canadian English	Irish English	Received Pronunciation	Scottish English	Standard American English	New England English	New York English	New Zealand English	Newfoundland English	Example
1	w1	ɪ	ɪ	ɪ	ɪ	ɪ	ɪ	ɪ	ɪ	ə	ɪ	Kit
2	w2	e	ɛ	ɛ	e	ɛ	ɛ	ɛ	ɛ	e	ɛ	Dress
3	w3	æ	æ	æ	æ	a	æ	æ	æ/ æə/ ɛə	æ	æ	Trap
4	w4	ɒ	ɑ	ɒ	ɑ	ɔ	ɑ	ɒ	ɑ/ ɑə	ɒ	ɑ	Lot
5	w5	ʌ	ʌ	ʌ	ʌ	ʌ	ʌ	ʌ	ʌ	ʌ	ɔ¨	Strut
6	w6	ʊ	ʊ	ʊ	ʊ	u	ʊ	ʊ	ʊ	ʊ	ʊ	Foot
7	w7	aː	æ	æ/ aː	ɑː	a	æ	a/ æ	æə/ ɛə	aː	æː	Bath
8	w8	ɑ	ɑ	ɒ/ ɔː	ɒ	ɔ	ɔ	ɒ	ɔə	ɑ	ɑː	Cloth
9	w9	ɜː	ɜr	ʌr/ ɛr	ɜː	ɜr	ɜr	ɜ	ɜ/ ɜr	ɜː	ɜrː	Nurse
10	w10	iː	i	iː	iː	i	i	i	i	iː	iː	Fleece
11	w11	ʌɪ	eɪ	eː	eɪ	e	eɪ	eɪ	eɪ	ʌɪ	ɛː/ eɪ	Face
12	w12	aː	ɑ	aː	ɑː	a	ɑ	a	ɑə	aː	æ/ ɑː	Palm
13	w13	ɔː	ɑ	ɔː	ɔː	ɔ	ɔ	ɑ	ɔə	ɔː	ɑː	Thought
14	w14	ʌʊ	o	oː	əʊ	o	o	o	oʊ	ʌʊ	ʌʊ	Goat
15	w15	uː	u	uː	uː	u	u	u	u/ ɪu	uː	uː	Goose
16	w16	ɑɪ	aɪ	aɪ	aɪ	ae/	aɪ	aɪ	ɑɪ	ɑɪ	əi	Price

						ʌi						
17	w17	ɔɪ	ɔɪ	ɔɪ	ɔɪ	ɒɪ	ɔɪ	ɔɪ	ɔɪ	ɔɪ	əi	Choice
18	w18	æʊ	ɑʊ	aʊ	aʊ	ʌu	aʊ	aʊ	aʊ	æʊ	əu	Mouth
19	w19	ɪə/ iːə/ iː	ɪr	iːr	ɪə	ir	ɪr	iə	ɪə/ ɪər	ɪə/ iːə/ iː/ eə	ɛr	Near
20	w20	eə	ɛr	eːr	ɛə	er	ɛr	æə	ɛə/ ɛər	eə	ɛr	Square
21	w21	aː	ɑr	aːr	ɑː	ar	ɑr	a	ɑə/ ɑər/ ɑr	aː	ær	Start
22	w22	ɔː	or	ɔːr	ɔː	ɔr	ɔr	ɒ	ɔə/ ɔər	ɔː	ɔ¨r	North
23	w23	ɔː	or	oːr	ɔː	or	or	oə	ɔə/ ɔər	ɔː	ɔ¨r	Force
24	w24	ʊə/ ɔː/ uːə/ uː	ʊr	uːr	ʊə	ur	ʊr	uə	ʊə/ ʊər	ʊə/ uːə/ uː/ ɔː	ʊr	Cure
25	w25	iː	i	iː	i	e/ ɪ/ i	i	ɪ/i	i	iː	i	Happy
26	w26	ə	ər	ər	ə	ər	ər	ə	ə/ ər	ə	ər	Letter
27	w27	ə	ə	ə	ə	ʌ	ə	ə	ə	ə	ə	Comma