Allomorphy

David R. Mortensen

July 19, 2023

Morphemes Can Have More than One Realization

In English, there is a morpheme HOP, meaning, roughly, 'jump like a rabbit'. There are two ways of spelling hop:

(1) a. Peter was afraid to **hop** past Mr. McGregor's gate.
    b. Peter **hop**s past Mr. MacGregor's gate.
    c. Peter **hopp**ed past Mr. MacGregor's gate.
    d. Peter is **hopp**ing past Mr. MacGregor's gate.

When HOP occurs before the past tense suffix *-ed* or the present participle suffix *-ing* it is realized as *hopp*, but it is realized as *hop* elsewhere.

This is an example of what is called allomorphy—the state of affairs when a single morpheme has more than one realization. The realization can be in terms of spelling. It can also be in terms of pronunciation or (in a sign language) in the motor or visual representation of the morpheme.

The basic definition of allomorphy:

(2) a. The signified remains constant
    b. The signifier varies
    c. The distribution of the various signifiers is predictable

The various signifiers are called ALLOMORPHS. The *hopp* allomorph occurs before a suffix that starts with a vowel . The *hop* allomorph occurs in all other situations (elsewhere) .

This is true of spelling (orthography) but not pronunciation. *hopped* is pronounced /hɑpt/, with no vowel before the suffix.

Phonologically-Conditioned Allomorphy

The best studied type of allomorphy is that in which the ENVIRONMENTS[1] that determine which allomorph will surface are based on sound. Since we have not dealt with phonology yet, we will use spelling as a proxy for pronunciation (even though these two things—phonology and orthography—are rather different beasts).

Examples from English

A classic example of allomorphy is presented by the English plural suffix:

Sometimes the plural is written as *-s*. More rarely, it is written as *-es*. Whether one writes *-s* or *-es* is perfectly predictable: When the plural suffix follows *ch*, SH, *s*, or *z*, one writes *-es*. Otherwise, one writes *-s*, as seen in Table 1.

One can express this pattern of allomorphy as a rule:

This is an application of a principle, first known from the Sanskrit grammarian Pāṇini, which is sometimes called the ELSEWHERE PRINCIPLE. It holds that grammatical rules act like `case` statements in programming languages, with the most specific cases given priority, more general cases following, and a fallback case (the elsewhere case) applying when none of the other cases do.

[1] Linguists call general contexts ENVIRONMENTS.

kit	kits	kiss	kisses
kid	kids	buzz	buzzes
pick	picks	pitch	pitches
bud	buds	bus	buses
puff	puffs		
fin	fins		
jam	jams		
path	paths		
pill	pills		
fear	fears		

Table 1: Orthographic allomorphs of the English plural suffix.

1. Start by adding the suffix *s*. We'll call *s* the UNDERLYING FORM of the morpheme.
2. Apply a rule that adds an *e* between any sequence of *s*, *z*, *sh*, or *ch* and an *s* (at the end of a word).

For those who are familiar with Python regular expression syntax, the rewrite rule could be expressed with the following function:

```python
import re

def e_insertion(form: str) -> str:
    return re.sub("(ch|sh|s|z|)(s)$", "\1e\2", form)
```

Figure 1: A Python implementation of the English e-insertion rule.

Applying this to the underlying form of the word (the concatenation of the underlying forms of both morphemes) sometimes results in a change and sometimes does not. When there is a change, our plural morpheme looks like *-es* (or, at least, we have an *e* before the final *s*).

The same function can be applied to all words. It will only insert *e* in cases where the *-es* allomrph is expected. Or almost.

What about words like *pass*? If we pass *pass* to our `e_insertion` function, we get *pases*, which is not what we want. We need some way of saying that our rule only applies at morpheme boundaries (not in the middle of morphemes like *pass*). Let us say that rather than concatenating the underlying forms of morphemes to get the underlying form of the word that we join them with `^`, indicating a morpheme boundary, and that we delete all of the `^` symbols when we are done with them. We can then revise our function to be:

```python
import re

def e_insertion(form: str) -> str:
    return re.sub("(ch|sh|s|z|)[^](s)$", "\1^e\2", form)
```

Figure 2: A revised Python implementation of the English e-insertion rule.

inek	‘cow’	ineği	‘his cow’
kuyruk	‘tail’	kuyruğiu	‘its tail’
köpük	‘foam’	köpüğü	‘its foam’
yatak	‘bed;	yatağı	‘its bed’

Table 2: Turkish k/ğ alternation

Examples from Turkish

Other Examples

Suppletive Allomorphy

Morphologically Conditioned Allomorphy

Implications for Tokenization

All widely used tokenization schemes treat different allomorphs of the same morpheme as different vocabulary items. This is suboptimal, especially for less common morphemes, since embeddings of each of the separate types are likely to be less informative than the embedding of a type that subsumes all of them. Take the case of one of the two negative prefixes in English:

(3)
a. **im**balanced
b. **im**patient
c. **im**penetrable
d. **im**ponderable
e. **im**mortal
f. **im**moral

(4)
a. **in**ordinate
b. **in**applicable
c. **in**decipherable
d. **in**dissoluble
e. **in**tangible
f. **in**terminable
g. **in**separable
h. **in**secure
i. **in**finite

(5)
a. **il**legal
b. **il**liberal
c. **il**logical
d. **il**limitable
e. **il**legible

(6)
a. **ir**reversible
b. **ir**revocable
c. **ir**resistible
d. **ir**reproachable
e. **ir**reconcilable
f. **ir**religious
g. **ir**rational
h. **ir**regular

	'hand'	'köy'	'oda'	'korku'
unmarked	el	köy	oda	korku
accusative	eli	köyü	odayı	korkuyu
genitive	elin	köyün	odanın	korkunun
dative	ele	köye	odaya	korkuya
locative	elde	köyde	odada	korkuda
ablative	elden	köyden	odadan	korkudan

Table 3: Turkish vowel harmony

meng	+	urus	mengurus	'take care'
meng	+	tulis	menulis	'write'
meng	+	kirim	mengirim	'send'
meng	+	pakai	memakai	'use'
meng	+	sewa	menyewa	'rent'

Table 4: Nasal substitution in Indonesian

We could learn representations for *im-*, *in-*, *il-*, and *ir-* separately. However, the number of actual examples in our training data will be, in the final analysis, not that large. As a result, the embeddings may, given the vagaries of small numbers, end up being quite different from one another. A tokenization scheme in which allomorphy was factored out would have the advantage of reducing sparsity and increasing generality.

If, on formal grounds, it can be known that two units, *A* and *B* realize one morpheme *M* (in different context), *A* and *B* should be given the same representation.

References