Lexemes, or what Dictionaries Know about Morphology

David R. Mortensen

January 27, 2024

Take a corpus of text, tokenized by punctuation and whitespace, and identify all of the word types (unique words). Then take all of the headwords in a dictionary and compare these two lists. Are all the common words on the first list in the second list? What is missing?

(1) a. houses
    b. drinks
    c. passed
    d. moving

Note that the dictionary does have entries for

(2) a. house
    b. drink
    c. pass
    d. move

Figure 1: A dictionary entry with the headword *virus*.

Why are the examples in (1) present but those in (2) absent? Are the wordforms in (1) not covered?

The dictionary has one entry for *drink*, *drinks*, *drank*, *drunk*, and *drinking*. Together, these form what is called a LEXEME. The other words shown above are treated similarly. A lexeme is a set of wordforms that are related by inflection. One of the forms is often used to identify the whole set, like *drink*. This is called the LEMMA. The headwords in dictionaries are lemmas.

Lexemes are not simply sets, though. They have structure. The structure a lexeme has is called its paradigm. Take an English verb like *drink*. Leaving aside the past participle *drunk* and the present participle *drinking*, their are three parameters in which English verbs can vary: person, number, and tense (which we say in the lecture on inflection). We can then ask what the form of the verb *drink* is for each of the combinations of properties. We often lay this out as a table to make it easier to visualize, as in Table 1: Note that this English paradigm is rather monotonous—most of the cells are just *drink*. Consider the equally boring paradigm for *move* in Table 2: Note that everywhere where the first paradigm has *drink*, the second paradigm has *move*, wherever the first paradigm has *drank*, the second paradigm is *moved*, and wherever the first paradigm has *drinks*, the second paradigm has *moves*. We can look at paradigms in two ways:

- From the perspective of a lemma or another "base" form
  1SG is lemma and 3SG is lemma + *s*

	PAST	PRES	FUT
1SG	drank	drink	drink
2SG	drank	drink	drink
3SG	drank	drinks	drink
1PL	drank	drink	drink
2PL	drank	drink	drink
3PL	drank	drink	drink

Table 1: A paradigm for *drink* (person, number, and tense)

	PAST	PRES	FUT
1SG	moved	move	move
2SG	moved	move	move
3SG	moved	moves	move
1PL	moved	move	move
2PL	moved	move	move
3PL	moved	move	move

Table 2: A paradigm for *move* (person, number, and tense)

- From the perspective of paradigm cells
  The other past cells are always the same as, e.g., the 1SG.PAST and 3SG.PRES is always 1SG.PRES + *s*.

  The verb paradigms in Totonac are much more complicated than the English noun paradigms. A partial paradigm is shown in Table 3.

'SWIM, BATHE'	IMPERFECT	PERFECTIVE	PERFECT
1SG	kpáʃ	kpáɬ	kpaʃní̱:t
2SG	páʃa̱	paʃt	paʃní̱:ta̱
3SG	páʃ	paɬ	paʃní̱:t
1PL.EXCL	kpaʃá̱w	kpaʃw	kpaʃni̱:tá̱w
1PL.INCL	paʃá̱w	paʃw	paʃni̱:tá̱w
2PL	paʃá̱:ti̱t	páʃti̱t	paʃni̱:tátti̱t
3PL	paʃqóh	paʃqó:ɬ	paʃqo:ní̱:t

Table 3: Partial paradigm for Totonac *paʃ* 'swim, bathe'

As a lemma, would choose 3SG—all of the other wordforms can be infer easily from this one. The relationships are much more complicated, but the principles are the same. One can either infer each of the cells given the lemma and the properties (like first person, plural, inclusive, or perfective) or infer some cells in the paradigm given wordforms in the other cells.

Computational Tasks with Paradigms

One can infer all of the cells in (this part of) a Totonac verb paradigm (roughly) by substituting the lemma for the Xs in Table 4.

'SWIM, BATHE'	IMPERFECT	PERFECTIVE	PERFECT
1SG	kX	kXɬ	kXnɺ́:t
2SG	Xa̱	Xt	Xnɺ́:ta̱
3SG	X	Xɬ	Xnɺ́:t
1PL.EXCL	kXá̱w	kXw	kXnɺ:tá̱w
1PL.INCL	Xá̱w	Xw	Xnɺ:tá̱w
2PL	Xá̱:tɺ̱t	Xtɺ̱t	Xnɺ:tátɺ̱t
3PL	Xqóh	Xqó:ɬ	Xqo:nɺ́:t

Table 4: Partial paradigm for Totonac

Alternatively, given a subset of wordforms in a paradigm, we can fill in the missing cells by analogy with other paradigms (Table 5).

'SWIM, BATHE'	IMPERFECT	PERFECTIVE	PERFECT
1SG	kpáʃ	kpáʃɬ	kpaʃnɺ́:t
2SG	páʃa̱	paʃt	paʃnɺ́:ta̱
3SG	[mask]	paʃɬ	paʃnɺ́:t
1PL.EXCL	kpaʃá̱w	kpaʃw	[mask]
1PL.INCL	paʃá̱w	paʃw	paʃnɺ:tá̱w
2PL	paʃá̱:tɺ̱t	páʃtɺ̱t	paʃnɺ:tátɺ̱t
3PL	paʃqóh	[mask]	paʃqo:nɺ́:t

Table 5: Partial paradigm for Totonac *paʃ* 'swim, bathe'

These two tasks are called REINFLECTION and PARADIGM COMPLETION.

Morphemes, Paradigms, and Theories of Morphology

In the previous section, we concentrated on MORPHEMES as minimal signs and treated words as the concatenation of strings of morphemes. This approach to morphology is called ITEM-AND-ARRANGEMENT morphology. We further saw that non-concatenative processes could act like morphemes. A kind of morphological theory that generalizes morphemes to processes (treats concatenation of morphemes as a special case of string-to-string functions) is called ITEM-AND-PROCESS morphology.

WORD-AND-PARADIGM morphology is a much different morphological theory in which morphology is understood **not in terms of morphemes** but in terms of relationship between wordforms within a paradigm.

item-and-arrangement Words are built up by concatenating morphemes. Meaning is computed compositionally.

item-and-process Words are build up by applying zero or more functions of the type $f : \Sigma* \rightarrow \Sigma*$ to a root morpheme. Meaning is computed compositionally.

word-and-paradigm Wordforms are realizations of lexemes given inflectional properties (within a paradigm). Morphemes, as such, do not exist.

Lexemes, Paradigms, and Subword Representations

Lexemes are sets of signifiers than have only one signified. They also tend to share a common substring (so the signifiers are not completely distinct). What does this mean for tokenization?

- If words have a similar meaning, we expect them to have a similar representation.
- It would help words have a similar representation if there were an overlap between subword units
- If the lemma is a distinct unit, most or all wordforms belonging to a lexeme will share a token in common
- Is is less clear what this means for other subword units of a wordform

UniMorph and Paradigms in Computational Morphology

Perhaps the best-know multilingual computational resource for morphological analysis is UniMorph (`https://unimorph.github.io/`). It is a collaborative effort to collect and annotate paradigms of the world's languages. It consists of wordforms, the corresponding lemmas, and the corresponding bundle of inflectional properties. An excerpt from the Catalan dataset (a few wordforms for the verb *abaixar* 'lower') is given below:

```
abaixar abaixada V.PTCP;PST;SG;FEM
abaixar abaixades V.PTCP;PST;PL;FEM
abaixar abaixant V.PTCP;PRS
abaixar abaixaran V;IND;FUT;3;PL
abaixar abaixaràs V;IND;FUT;2;SG
abaixar abaixarà V;IND;FUT;3;SG
abaixar abaixarem V;IND;FUT;1;PL
abaixar abaixàrem V;IND;PST;1;PL;PFV
abaixar abaixaren V;IND;PST;3;PL;PFV
abaixar abaixares V;IND;PST;2;SG;PFV
abaixar abaixareu V;IND;FUT;2;PL
```

```
abaixar abaixàreu V;IND;PST;2;PL;PFV
abaixar abaixaré V;IND;FUT;1;SG
abaixar abaixaria V;COND;1;SG
abaixar abaixaria V;COND;3;SG
abaixar abaixaríem V;COND;1;PL
abaixar abaixarien V;COND;3;PL
abaixar abaixaries V;COND;2;SG
abaixar abaixaríeu V;COND;2;PL
abaixar abaixar V;NFIN
```

The first column is the lemma; the second column is the inflected wordform; the third column is a list of inflectional properties in standardized representation, separated by semicolons. Part of speech, represented by `V` for verb, is treated as a property. A complete catalog of all of these labels, and their definitions, is provided by the UniMorph Schema.

UniMorph has provided the framework for a number of shared tasks. The most prominent of these are reinflection and paradigm completion, as mentioned above. In reinflection, the model is provided with the lemmas, properties, and wordforms for a subset of the dataset as training data. At test time, the model is required to generate a wordform given the lemma and the properties. For example, given *abaixar* as a lemma and *V;IND;FUT;1;PL* as properties, the model is expected to generate *abaixarem*.