Stop word removal ~ phonological tiers

We started out this unit with *n*-gram grammars as a basic model for studying the linguistic domains of phonotactics and morphotactics, i.e. the rules that govern the arrangement of sounds and parts of word like *un-*, *re-*, *-ly*, *-ize*, *-ation*, *-s*, and so on. After that, we switched to more applied issues and the bag-of-words model. In order to be truly effective, the bag-of-words model must be combined with stop word removal. You might think that stop word removal, being a technique for real-world applications, offers nothing of value to linguistics. But stop word removal is actually closely related to what linguists call *phonological tiers*.

The limits of *n*-gram grammars

We already know that (positive/negative) *n*-gram grammars can be used to describe all kinds of conditions on natural language phonotactics and morphotactics. But these grammars can only enforce locally bounded conditions like intervocalic voicing or penultimate stress. These phenomena have the property that it suffices to consider a finitely bounded number of adjacent segments. Not all natural language phenomena obey this condition.

Example 1

One of these phenomena is attested in **Samala**, which belongs to the group of Chumash languages spoken in southern California. Samala displays a constraint known as **long-distance sibilant harmony**. All sibilants in a word must agree in anteriority, no matter how far apart they are. This means that a word can certainly contain multiple instances of *s* or multiple instances of ʃ (ʃ is the IPA symbol corresponding to English *sh*). But it may never contain a mixture of the two. So *haʃxintilawaʃ* is well-formed, whereas *hasxintilawaʃ* and *haʃxinitilawas* are both ill-formed. The form *hasxintilawas*, while not an actual word of Samala, would also obey the sibilant harmony constraint.
Even a 10-gram grammar cannot capture these contrasts. A negative 10-gram grammar that generates both *hasxinitilawas* and *haʃxintilawaʃ* must not forbid any of the following:

- *sxintilawa*
- *xintilawas*
- *ʃxintilawa*
- *xintilawaʃ*

But then this grammar would also allow for the illicit *hasxintilawaʃ* and *haʃxintilawas*. Only an 11-gram grammar could capture the contrast.

Exercise 1

Write an 11-gram grammar that generates *hasxintilawas* and *haʃxintilawaʃ*, but not *hasxintilawaʃ* and *haʃxinitilawas*. The grammar may be positive or negative, whichever you prefer.

Exercise 2

Extend the grammar so that it also captures the fact that *ʃtajanowonowaʃ* is licit whereas *stajanowonowaʃ* is illicit. You might have to move beyond 11-grams.

Samala's long-distance sibilant harmony can be handled by an n-gram grammar only if there is some upper bound $k < n$ such that sibilants in a Samala word are never separated by more than k symbols. The examples above show that this k is at least 12, so one would need at least a 13-gram grammar to handle the process. Such large n-grams simply aren't feasible in practice.

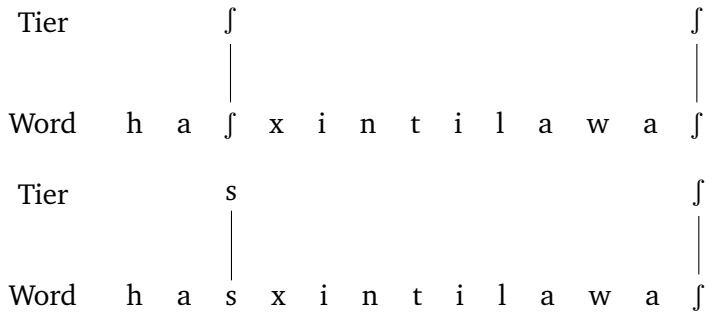
Example 2 Suppose for the sake of argument that Samala has only three sounds: a vowel, *s*, and ʃ. Then there are $3^{13} = 1,594,323$ distinct n-grams. Only the n-grams that do not mix *s* and ʃ are well-formed, of which there are $2 \times 2^{13} = 2^{14} = 16,384$. So a positive 13-gram grammar for Samala's sibilant harmony would contain 16,384 distinct 13-grams (plus a few with $\rtimes$ or $\ltimes$), and a negative one $1,594,323 - 16,384 = 1,586,131$. That's a lot.

Large grammars are undesirable for multiple reasons. They are almost impossible to decipher for humans. Who is supposed to look at 16,384 distinct 13-grams and deduce what condition they encode? And the larger the grammar, the more computational resources it consumes. Small grammars are fast and easy to figure out, large grammars are slow and indecipherable. Small grammars trump large grammars.

Phonological tiers allow for smaller grammars

If you have some background in phonology, you might already be thinking that there is a much simpler solution: project a **tier** that contains only sibilants (*s* and ʃ), and regulate the shape of this sibilant tier with a negative bigram grammar containing only *s*ʃ and ʃ*s*. That's one darn small grammar.

Example 3 The sibilant tier of *hasxintilawas* is *ss*, which is allowed by the negative grammar. The illicit *hasxintilawa*ʃ, on the other hand, has the sibilant tier *s*ʃ, which is not allowed.



Exercise 3 Carry out the same calculations for

- *ha*ʃ*xintilawa*ʃ,
- *ha*ʃ*xinitilawas*, and
- ʃ*tajanowonowa*ʃ.

Exercise 4 As an abstract example, suppose that our alphabet consists of *a*, *b*, and *c*, and that all symbols except *c* should be projected on the tier. What is the tier of *aabaccacb*?

Tiers are a nice linguistic metaphor, but what is going on here at a formal level? Exactly the same thing as with stop word removal. We have a function that removes all irrelevant elements, and then we apply a specific procedure to the output of this function. For stop word removal, that follow-up procedure was the construction of a bag of words. With tier projection, we instead test the output for well-formedness with respect to an n-gram grammar.

The mathematics of tiers

Remember that del_S is a function that takes a string as its input and deletes all symbols that belong S. Tier projection works pretty much the same, except that one usually specifies which symbols to keep rather than which to delete. So given a set T of tier symbols, del_T takes a string as its input and keeps only those symbols that belong to T. This difference is very similar to the split between positive and negative grammars: T specifies what may be kept, S what must not be kept. We can unify stop word removal and tier projection to a single function del_X, where X is some set with a specified polarity. With del_{+X}, only the symbols in X are kept, whereas del_{-X} removes all symbols that belong to X (and only those).

Exercise 5 Compute the values for all of the following:

- $del_{-\{a,b\}}(aaccbad)$
- $del_{+\{a,b\}}(aaccbad)$
- $del_{+\{a,b\}}(aababad)$
- $del_{-\{a,b\}}(aababad)$
- $del_{-\{a,b\}}(\varepsilon)$
- $del_{+\{a,b\}}(\varepsilon)$

You might think that this isn't a true unification, we have just moved the difference between stop word removal and tier projection into the polarity distinction. But just as with n-gram grammars, polarity doesn't actually matter. For every set S of a given polarity, there is some set T of opposite polarity such that $del_X(s) = del_Y(s)$ for every string s over some fixed alphabet Σ.

Exercise 6 Explain why this holds. Illustrate your argument with a few examples.

This is a pretty nifty result. Intuitively, stop word removal and tier projection seem completely unrelated. One is about cleaning up data for practical applications, the other about simplifying linguistic analysis. But mathematically, they are exactly the same thing. In later units, we will see many more examples of this unifying power of math.

Exercise 7 The term **culminativity** refers to the property that every word has exactly one primary stress. Suppose that our alphabet is $\{\sigma, \acute{\sigma}\}$, where σ denotes an unstressed syllable and $\acute{\sigma}$ one with primary stress. Specify a set ${}^{+}T$ of tier symbols and a bigram grammar G to capture culminativity (*hint*: $\rtimes$ and $\ltimes$ can be used with tiers, too).

Recap

- No n-gram grammar provides an elegant account of long-distance phenomena.

- The larger the *n*-grams, the larger the grammar; large grammars are unwieldy and computationally inefficient.
- Phonological tiers allow for more compact grammars by filtering out irrelevant material.
- The stop word removal function *del* is also the function for constructing phonological tiers.
- Math allows us to unify things that look very different at the surface. In particular, linguistic theory and language technology seem like very different beasts, but they actually share a lot of math.