The problem of overgeneration

In our discussion of *n*-gram models, we were largely concerned with specifying grammars for specific phenomena. Among other things, you wrote grammars for word-final devoicing, intervocalic voicing, and penultimate stress. This shows that many phenomena can be accounted for by very simple models. But there is a problem: the opposite also holds. This whole unit is dedicated to explaining what this means, why it is a problem, and how it can be addressed in at least some domains. The first two points are handled in this notebook, whereas the latter is spread out over the remainder.

Overgeneration and undergeneration

There is a major problem with *n*-gram models, in fact every computational model. All these models take for granted that there is a fixed alphabet, and the elements of these alphabet are treated as unanalyzable atoms without any additional properties. As far as an *n*-gram grammar is concerned, *s* and *f* do not differ in any relevant sense from *z* and *v*. So just like one can write a grammar for intervocalic voicing, one can also write one for intervocalic devoicing. Remember, intervocalic voicing means that voiceless sounds like *s* and *f* may not appear between vowels. Intervocalic devoicing would be the opposite: voiced sounds like *z* and *v* may not appear between vowels. Intervocalic voicing is a very natural and common process, whereas intervocalic devoicing has not been found in even a single language. Apparently, language simply does not work like that.

Exercise 1 Write a negative grammar for intervocalic devoicing, assuming that the alphabet consists only of *a*, *i*, *u*, *s*, *z*, *f*, and *v*.

Exercise 2 Assuming the same alphabet as before, write an *n*-gram grammar (it may be positive or negative) that requires every word to consist of exactly 5 symbols. This will be a large grammar. But keep in mind that *n*-gram grammars are just sets, and there's various way to compactly define large sets.

Exercise 3 Assuming the same alphabet as before, write an *n*-gram grammar (it may be positive or negative) that requires every word to start with *a* and end with *f*.

Exercise 4 Assuming the same alphabet as before, write an *n*-gram grammar (it may be positive or negative) for "penultimate *f*": if a word has at least two symbols, then the last but one symbol must be *f* and no other position may be *f*.

Exercise 5 Suppose that English only contains the words *the*, *old*, *man*, *woman*, *sleep*, *sleeps*, *snore*, and *snores*. Assume furthermore that the English subject verb agreement system works as follows: if the subject does not contain an adjective (like *old*), then use the inflected verb form; otherwise, use the base form. So we would get *The old man snore* but *The man snores*. Write an *n*-gram grammar that captures this unnatural condition.

These exercises show that our models **overgenerate** from a typological perspective. Not only can they express constraints that occur in natural languages, but also phenomena that exist in no known language at all. Overgeneration is a better position to be in than **undergeneration**, when a formalism cannot capture all relevant phenomena. It's better to do too much than too little. Many formalisms actually suffer from both shortcomings: not all phenomena can be handled, and the formalism can also express unnatural patterns (yes, n-gram models are in this unfortunate situation as we will learn in a later unit).

Ideally, a formalism would provide a perfect fit for natural language, which means that it neither overgenerates nor undergenerates. No such formalism exists at the point of writing. There have been many attempts to design more expressive models that do not undergenerate. Overgeneration, on the other hand, has seen a lot less work. This is not ideal because the larger the range of phenomena that can be described, the harder it is to design a learning algorithm for the model. The intuition here is that you'll have a much easier time picking the right language among, say, 100 candidates rather than 1000. A learning algorithm that assumes that a natural language could display an unnatural process like intervocalic devoicing needs more data to rule out this hypothesis. Data isn't cheap, so overgeneration should be avoided if possible.

Universals

Linguists use the term **universals** to refer to certain invariable properties of language. Natural languages simply do not vary in all logically conceivable ways.

Example 1 No known language enforces any of the following conditions:

- The further to the right a syllable occurs in a word, the more consonants it must have.
- Any sequence of sounds is a possible word as long as it contains at least as many vowels as consonants.
- When we sort the words of a language by their length, we get the **Fibonacci series**: 1, 1, 2, 3, 5, 8, 13, ...
- The first word in a sentence must rhyme with the last word in a sentence.
- Every sentence must have an odd number of words.
- To negate a sentence, utter it backwards.
- Adjectives that start with a vowel go before the noun, adjectives that start with a consonant go after the noun.
- Relative clauses follow the noun they modify if it is the subject, but otherwise precede the noun.

Apparently languages can only enforce constraints of a specific kind, and the examples above do not fit the bill.

Exercise 6 For each constraint above, give a concrete example from English that violates it.

Exercise 7 Can you think of a constraint that you are fairly certain does not arise in any natural language?
Hint: Mathematical concepts like prime numbers are very fruitful for this.

The existence of universals means that there might be phenomena or constraints that can be described by an *n*-gram grammar yet never show up in the real world because they violate certain universals.

Exercise 8 Can any of the constraints listed above be enforced by an *n*-gram grammar? If so, explain how.

The idea of universals is very powerful: if we can identify a reliable list of universals, then we know what constraints in natural language may look like. That would give us a very good idea of what our models have to be capable of and what is superfluous, and thus we could design more restrictive and efficient models that can be learned from less data. Unfortunately we do not have conclusive list of universals yet — linguists keep discovering new phenomena, and new data might invalidate our current assumptions about what is universal. This is one of the reasons why computer models still do much worse than humans in several respects. Presumably, the human mind somehow comes with the full list of universals, and that makes language a lot easier. Thanks to universals, children learn their native language effortlessly with relatively little input. Current machine learning models, on the other hand, have no knowledge of universals and thus need huge amounts of data to weed out lots of crud that is logically conceivable but nonetheless never occurs in any natural language. Quite simply, children will never try to do anything like intervocalic devoicing, whereas a computer needs to learn from the data that the language does not have such a process.

Linguists distinguish two types of universals:

1. **Formal universals**
   These identify abstract properties of the "grammar machine". For example, if all linguistic phenomena could be described by *n*-gram grammars (they can't, unfortunately), that would be a formal universal.

2. **Substantive universals**
   These identify properties of the "building material" used by the grammar machine. For instance, that consonants and vowels aren't just arbitrary symbols but very different kinds of sounds with very different roles in language. A substantive universal might help explain, for example, why intervocalic voicing is common and intervocalic devoicing unattested even though both look the same from a formal perspective and both could be produced by the grammar machine.

The next unit discusses some mathematical properties that might be substantive universals.