Succinctness and choosing between grammars

PREREQUISITES

- functions(basics, function growth)

Thanks to all the math we have put to good use, we now have three expressively equivalent models of phonotactics:

1. fixed negative n-gram grammars, and
2. mixed negative n-gram grammars, and
3. fixed positive n-gram grammars.

By "expressively equivalent" we mean that every string language that can be generated by a grammar of one of those tree types can also be generated by grammars of the other two types. Or in other words, we can freely translate between these three grammar types as we see fit. But this also means that we cannot distinguish between these three types of grammars based purely on typological data. There is no empirical phenomenon that allows us to advocate, say, for fixed negative n-gram grammars and against the other two types. However, we should not conflate expressive equivalence with total equivalence. These three grammar types can still differ in other respects, and one of them is succinctness: how many n-grams does the grammar need to capture a given phenomenon?

1 Differences in grammar size

The three grammar types above vary hugely in how compactly they can model specific phenomena. You already saw a glimpse of this in earlier exercises, but the true extent only becomes evident once we consider a few artificial examples.

EXAMPLE 1.
Suppose our alphabet contains the symbols a, b, c, d (and nothing else). Now consider the language L that contains ab, aab, $aaab$, and so on (more succinctly, we can write L as a_+b). This is very easy to express as a positive grammar:

1. ⋊a
2. aa
3. ab
4. b⋉

The smallest mixed negative grammar for this language isn't much bigger:

1. c
2. d

3. ⋊⋉
4. ⋊b
5. ba
6. bb
7. a⋉

And the fixed negative grammar is huge by comparison:

1. ⋊⋉
2. ⋊b
3. ⋊c
4. ⋊d
5. ac
6. ad
7. ba
8. bb
9. bc
10. bd
11. a⋉
12. b⋉
13. c⋉

Example 2.
Suppose that our alphabet still contains only *a*, *b*, *c*, *d*, but *L* now follows a more general pattern: 1 or more instances of *a*, followed by exactly one instance of *b* or *c* or *d*. Hence *L* contains *ab*, *ac*, *ad*, *aab*, *aac*, *aad*, *aaab*, *aaac*, *aaad*, and so on.

The positive grammar is still fairly small.

1. ⋊a
2. aa
3. ab
4. ac
5. ad
6. b⋉
7. c⋉
8. d⋉

The mixed negative grammar, by comparison, grows a lot in size:

1. ⋊⋉
2. ⋊b
3. ⋊c
4. ⋊d

5. ba
6. bb
7. bc
8. bd
9. ca
10. cb
11. cc
12. cd
13. da
14. db
15. dc
16. dd
17. a⋉

In fact, this also happens to be the fixed negative grammar. For L, allowing n-grams of variable length does not help at all.

Example 3.
Now suppose our alphabet is $\{a\}$ and consider the language L that contains all strings over a whose length is at least 2 (i.e. aa, aaa, and so on). The mixed negative grammar is incredibly small:

1. ⋊⋉
2. ⋊a⋉

The fixed negative grammar looks slightly different, but has the same size.

1. ⋊⋊⋉
2. ⋊a⋉

This time the positive grammar is the largest:

1. ⋊⋊a
2. ⋊aa
3. aaa
4. aa⋉
5. a⋉⋉

Example 4.
Finally, assume that the alphabet Σ is $\{a, b, c, d, e, f\}$ and that L contains all strings over this alphabet except that no string may have 5 or more instances of a in a row. For instance, *baaaab* and *caaadaaaf* are well-formed, but not *baaaaab* or

ffaaaaaaacabec. The negative grammar for this is maximally simple:

1. aaaaa

The positive grammar, on the other hand, is enormous. It contains all n-grams in Σ_E^5 except *aaaaa*. That's 32,767 n-grams: since there are 6 symbols in Σ and 2 edge markers, Σ_E^5 contains $8^5 = 32,768$ 5-grams.

Overall, there doesn't seem to be much regularity. Sometimes a positive grammar is smaller, sometimes a negative grammar one, and sometimes it matters whether the negative grammar is mixed or fixed while in other cases the two look exactly the same. Sometimes the differences is only one or two n-grams, sometimes it's tens of thousands. So is this a case of anything goes where one can never be quite sure how things will pan out? No, quite to the contrary.

2 Upper bounds and rates of growth

Even though it is difficult to tell how things may pan out for a specific phenomenon or string language, that does not mean that there are no regularities. It's just that these regularities are a bit more abstract in nature as they take the form of **upper bounds**. Since every fixed-length grammar, whether positive or negative, is built from members of Σ_E^n for some alphabet Σ and some choice of n, its size cannot exceed that of Σ_E^n. And that size is easy to calculate. Each n-gram furnishes n positions, each one of which must be a symbol from Σ or one of the two edge markers. So if Σ contains m symbols, there are $m + 2$ choices for each position, and since there are n positions, this means there are $(m + 2)^n$ different combinations. Hence the size of Σ_E^n is $(m + 2)^n$, and that's a fixed upper bound on the size of any n-gram grammar over Σ.

Example 5.
Suppose $\Sigma := \{a, b\}$. Then Σ_E^2 has $(2 + 2)^2 = 4^2 = 16$ members. We can list them all:

1. ⋊⋊
2. ⋊⋉
3. ⋊a
4. ⋊b
5. a⋊
6. a⋉
7. aa
8. ab
9. b⋊
10. b⋉

11. ba
12. bb
13. ⋉⋊
14. ⋉⋉
15. ⋉a
16. ⋉b

Exercise 1.
For $n \geq 2$, no grammar ever needs to contain every member of Σ_E^n. Explain why.

This insight provides us with a fixed upper bound for any given choice of Σ and n such that no grammar can be bigger than that. But that by itself isn't really that interesting, we want to know how that upper bound changes as we vary Σ and n. We can make this more visual by drawing a table, where rows indicate the size of the alphabet (plus both edge markers) and columns indicate the length of the n-grams.

	1	2	3	4	5
3	3	9	27	81	243
4	4	16	64	256	1024
5	5	25	125	625	3125
6	6	36	216	1296	7776
7	7	49	343	2401	16807
8	8	64	512	4096	32768
9	9	81	729	6561	59049
10	10	100	1,000	10,0000	100,000
100	100	10,000	1,000,000	100,000,000	10,000,000,000

As you can see, the numbers grow quite a bit from the top to the bottom, but much faster from left to right. In other words, n plays a much bigger role in determining the size Σ_E^n. The number of bigrams over an alphabet with 100 symbols (including edge markers) is still smaller than the number of 5-grams over an alphabet with 7 symbols. Our upper bound grows **exponentially** with n, but only **polynomially** with Σ.

What does this tell us? While we can freely choose between fixed negative grammars, mixed negative grammars, and positive grammars because they are interchangeable, grammar size can vary a lot depending on the phenomenon. This does not matter too much as long as Σ and n are both small, but as we increase the size of the alphabet and the length of the n-grams, it becomes more noticeable. In fact, we do not even need to worry too much about Σ as n has a much bigger impact on this. Even with very small alphabets, Σ_E^n is giant for $n > 5$.

Example 6.
For an alphabet with just two symbols, Σ_E^6 is 4,096, and Σ_E^10 is 1,048,576. That's more than the number of trigrams over an alphabet with close to 100 symbols.

3 Recap

- Depending on the phenomenon at hand, a positive or a negative grammar may be more succinct.
- The difference in grammar size may not always be very pronounced, but it can be.
- The difference cannot exceed the size of Σ_E^n, which grows polynomially with Σ and exponentially with n.