Using successor counts to do segmentation

(1) Agenda:
   • Finish up discussion of successor counts, seeing how they might be used for segmentation of word- and morpheme-like units even without a (potentially) infinite corpus to work from

(2) Example: trying to use successor counts to segment the utterance:
   but he explained that if the company had not focused on selling websites software for the last four years

(3) Approach:
   • Start at the beginning of the sentence, taking prefixes of increasing length:
     - b, bu, but, buth, buthe, etc...
     - For each prefix, compare to the utterances found in a corpus of English text
       * I used my (totally unscientific) cobbled-together corpus of newspaper articles and literature; 25 megs of text, divided into utterances at major punctuation [.,;!:?]
       * Total of 1,339,348 utterances in the corpus
   • For each prefix, counted how many possible successors among utterances in the corpus

(4) Successor counts for this sentence:
   (Drops to 0 at the first n)

   • Where do the peaks fall? Why?
   • Why does the successor count fall to 0 after just a short while?
   • How might this problem be solved?

(5) Backtracking: When the successor count drops to zero, go back to the previous peak and treat it as the start of a chunk
   • In this case, the end of the; new chunk beginning is explained that if the...

   (Drops to 0 immediately after a peak!)

(6) Next round:
Continuing through the sentence:

- How well does this procedure work?
- Where does it go wrong? Why? What throws it off?
- Would this be solved by a bigger corpus for calculating successors? Would it be solved by an infinitive corpus?

Other possible modifications: (some proposed by Harris, some since)

- Both forwards and backwards counts (successors/predecessors), to get better segmentation of both prefixes and suffixes
- Smoothing by considering not just immediate successors (character \( n + 1 \)) but also possible successors at time \( n + 2 \)
- Taking into account not only the number of possible successors, but also their relative frequencies. (??? Cairns, Shillcock, Chater, and Levy 1997 claim this helps, but I suspect that at least in some cases, it would actually hurt)

Summary of successor counts:

- Works a lot better than simple transition probabilities (why?)
- In theory, it requires an infinite corpus of (possible) utterances to draw on; could possibly relax this requirement with some sort of backtracking mechanism
- Backtracking procedure that I’ve suggested here would be thrown off if the syntax of the language prohibits certain classes of words from occurring at the beginning of an utterance.
- Can be thrown off by short words embedded inside longer words (e.g., \textit{we} inside \textit{web})

What’s next: brief overview of another important approach to finding words and morphemes: Minimum Description Length

- Rather than looking for boundaries by finding sequences that \textit{don’t} belong together, try to build a lexicon of actual words and morphemes by finding sequences that frequently \textit{do} occur together.