Source processing on a base dictionary consists of applying generation rules to eligible words in the dictionary and determining scores for the resulting letter strings. This proposal will provide some sample generation rules and discuss how the scores are calculated. Choices of base dictionaries, generation rules, and minimum scores for subsequent actions are outside the scope of this document.

Generation rules are of several types:

1. Addition of affixes, such as *anti-, out-, re-,* and *-ville*.
2. Creation of plurals and verb forms by analogy with base words, such as *womanchild****ren***, *acre****feet*** (but not *bears****feet***), and *testdrive****n***. There are not very many of these and it’s not clear to me that it is worth attempting to automate this category.
3. Creating variant spellings, such as *mythici****s****e*. Not sure of this one either.
4. Treating words listed in the dictionary with one part of speech as if they were a different part. For example, *beanball* (n) 🡪 *beanball* (v) 🡪 *beanball****ed***.
5. Combining rules; for example, ***re****beanball****ing****.*

The score will be the product of three factors based on the quality of the original word, a value associated with the rule (generally modified based on properties of the word), and the degree of attestation of the inferred form. There are several arbitrary constants proposed. There is no objectively correct value for these; I am using my intuition. It would be possible to run the software with different values and see what output you like, but I am fairly confident that the ability to set a threshold on searches will be adequate customizability.

The score for the quality of the original word is determined via the GooGoo rule—it is the sum of the Google and Gooch subscores. The Google subscore ranges from .05 for a word which is not found in the Google corpus (or possibly some similar one,[[1]](#footnote-1) or a combination) or a Google search, such as *myelinizing;* a slightly higher value () for a word not in the corpus but found N times in a search; and .1 for a word which occurs once per billion (10-9), such as *defenestrating;* increasing logarithmically to .5 for a word which occurs once per thousand (10-3) or more, such as *new*. The Gooch subscore is , where *N* is the number of sources listed for the word in the Gooch corpus. This same approach is used for the last factor, but recognizing that many words will be attested very little or not at all, the Google numbers will be replaced by .25 for missing words, () for words found only in a search, .35 for a word with frequency 10-9, and .5 for a word with frequency 10-6 or more and the Gooch formula is

Let’s work through an example: *gatherability.* The original word is *gatherable.* It has a frequency of 2×10-9 in the Google ngram corpus,[[2]](#footnote-2) which gives a weight of .12 by logarithmic interpolation. It is in two of the Gooch sources (W and O), for a weight of .255, so the total score for the original word is .375. Assume that the rule VERB+*able* 🡪 VERB+*ability* has a value of .2. The modified word does not appear in either corpus, but has about 2200 Ghits,[[3]](#footnote-3) giving it a GooGoo value of .317-.136=.181. Multiplying these three values gives .0136, or about ⅐4.

1. For example, the Corpus of Contemporary American English (COCA) with 560 million words, available for a modest price via [https://www.wordfrequency.info](https://www.wordfrequency.info/). [↑](#footnote-ref-1)
2. This is a visual average over the last couple of hundred years. I will assume that we can find an API to get a value. [↑](#footnote-ref-2)
3. There are actually 4900, but most of them are for *gatherable.* We will review the first hundred hits (I did twenty in this case, manually) to see how many of them actually include the target word and use that to factor down the reported value. [↑](#footnote-ref-3)