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 at this time.

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 number associated with the rule (possibly modified based on properties of a candidate word) having a higher value the more the word seems likely to naturally take the proposed rule, 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.

Other than some notes at the end of this document about inventing names, the rest of the paper has lots of details of calculation and example; if you don’t care about numbers, you can still read it and see if you think that the right concepts and properties are being considered.

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, attestation of the inferred form, 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.

The base weight for a particular generation rule will be the square root[[4]](#footnote-4) of the fraction of eligible words in some reference corpus for which the rule applies. For example, to compute the generation rule for OUT+ on verbs, start with the datum that NI3 contains about 40,000 verbs. There are about 136 which start with OUT,[[5]](#footnote-5) so the weight would be .

The base weight need not, however, be the end of the story. It appears that measurable properties of a candidate word can provide a more refined value based on how the definitions of the word relate to words for which the rule is explicitly supported. For every word in the reference corpus, create a set of up to three root senses (or six for verbs, which have separate transitive and intransitive lists of definitions) by taking the first substantive word in each of the first three definitions.[[6]](#footnote-6)

Now look at every word which appears as a root sense for at least 5% of either the full corpus or the words for which the reference corpus supports the generation rule; in this case, sampling produces the following table:

|  |  |  |
| --- | --- | --- |
| **Verb** | **Out** | ***all*** |
| be | 11% | 4% |
| become | 12% | 6% |
| bring | 6% |  |
| cause | 23% | 14% |
| come | 7% |  |
| form | 4% | 6% |
| give | 8% | 6% |
| go | 5% | 4% |
| have | 6% | 2% |
| make | 22% | 20% |
| move | 10% | 6% |
| perform | 9% | 4% |
| produce | 5% |  |
| put | 8% | 6% |
| set | 7% |  |
| subject | 1% | 8% |
| take | 6% | 6% |
| utter | 7% |  |

For a given candidate word, look at every word in its set of root senses which is in the table. For example, SPLASH has {become, dash, move, strike}, two of which are in the table. Take the one with the highest ratio (greater than one) between roots found in supported words and roots in the whole corpus—become, with 2:1—and multiply the base value by the square root of that, increasing it from .058 to .082. Take the next one (if any) and multiply by the cube root—in this case, move with 1.67:1, increasing it to the final (for now) value of .097.

If the word (also) has words for which the ratio is less than one, repeat the process for those. PASSAGE has {cause, engage, go, move, subject}. “Cause” and “move” bring it up to .088, but “subject” reduces it by to .031.

For any word which is otherwise unmodified by this process, modify the weight by .9 to make up for all the other words which are increased.

The other part is to look at the total number of definitions of the word[[7]](#footnote-7) and see how that relates to the averages for all words and for supported candidates. These last two are 4.96 (a) and 1.48 (b), respectively; the ratio is 3.35. As a final step, determine *d,* the number of definitions for the candidate word, and multiply the value by . So getting back to SPLASH, it has twelve definitions; . Multiplying this by the value of .097 we had before gives a truly final weight of .279.

Names are often used in wordplay. For example, [Peter Gretton](https://en.wikipedia.org/wiki/Peter_Gretton) allows for a long palindrome deletion and [Saltalamacchia](https://en.wikipedia.org/wiki/Jarrod_Saltalamacchia) has a letter bank of *chitals.* Some people do not feel constrained by reality and are willing to invent names, such as Ross Eckler’s isogram Emily Jung Schwartzkopf. Or if someone feels a mysterious compulsion to create a letter bank on [Chwałowice Coal Mine](https://en.wikipedia.org/wiki/Chwałowice_Coal_Mine), they could benefit by finding a Wilma Cohen. A database to be used for wordplay, therefore, should have both an exhaustive list of famous (or “notable,” in Wikipedia-speak) people and a long list of possible names.

Let’s suppose that want to have about a million inferred names. There are many online lists of [common first names](https://names.mongabay.com/male_names.htm) and [last names](https://names.mongabay.com/most_common_surnames.htm). A simple approach would be to take the thousand most common first names and the thousand most common last names and create the cross-product. I would suggest several possible refinements to get more useful and likely names.

1. Start with two thousand of each and take the cross product, then filter to the million most likely.[[8]](#footnote-8)
2. For some particularly common names, include common middle initials, such as James C. Smith,[[9]](#footnote-9) who may turn up some day as a transposal of [HMS Majestic](https://en.wikipedia.org/wiki/HMS_Majestic_(1785)).
3. Add a bonus factor, maybe 50%, to isograms, which I think are more likely to be of logological interest.
4. Using Wikipedia data, identify the distribution of national origins and ethnicities for each first and last name, then tweak the scores to reflect the degree of harmony in each inferred name. This would, for example, add to Julio Chavez and reduce Jacek Dubois.

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)
4. Taken in a spirit of expansiveness. [↑](#footnote-ref-4)
5. Plus a few, such as OUTGENERAL, which are based on nouns. [↑](#footnote-ref-5)
6. If there are subsenses, such as 1a, 1b, …, take only the first. Ignore definitions, such as the ones in small caps in NI3, which indicate that the word is synonymous with another word. Only look at definitions, of course, for the right part of speech, but if the word has multiple, entirely separate entries, repeat the entire process for each. [↑](#footnote-ref-6)
7. Counting subsenses as a half and subsubsenses as a quarter. [↑](#footnote-ref-7)
8. I’d likely do a small sample to identify the cutoff, rather than trying to sort all four million. [↑](#footnote-ref-8)
9. Okay, it turns out that [he is in Wikipedia](https://en.wikipedia.org/wiki/James_C._Smith) already, but you get the point. [↑](#footnote-ref-9)