

Simple maths for keywords

Adam Kilgarriff
Lexical Computing Ltd
adam@lexmasterclass.com

Abstract

We present a simple method for identifying keywords of one corpus *vs.* another. There is no one-size-fits-all list, but different lists according to the frequency range the user is interested in. The method includes a variable which allows the user to focus on higher or lower frequency words.

“This word is twice as common here as there.” Such observations are entirely central to corpus linguistics. We very often want to know which words are distinctive of one corpus, or text type, versus another.

The simplest way to make the comparison is expressed in my opening sentence. “Twice as common” means the word’s frequency (per thousand words, or million words) in the one corpus is twice its frequency in the other. We count occurrences in each corpus, divide each number by the number of words in that corpus, optionally multiply by 1,000 or 1,000,000 to give frequencies per thousand or million, and divide the one number by the other to give a ratio. (Since the thousands or millions cancel out when we do the division, it makes no difference whether we use thousands or millions. In the below I will assume millions and will use wpm for “words per million”, as in other sciences which often use “parts per million”.)

It is often instructive to find the ratio for all words, and to sort words by the ratio to find the words that are most associated with each corpus as against the other. This will give a first pass at two “keywords” lists, one (taken from the top of the sorted list) of corpus1 *vs* corpus2, and the other, taken from the bottom of the list (with scores below 1 and getting close to 0), for corpus2 *vs* corpus1. (In the below I will refer to the two corpora as the focus corpus *fc*, for which we want to find keywords, and the reference corpus *rc*: we divide relative frequency in the focus corpus by relative frequency in the reference corpus and are interested in the high-scoring words.)

There are four problems with keyword lists prepared in this way.

1. All corpora are different, usually in a multitude of ways. We probably want to examine a keyword list because of one particular dimension of difference between *fc* and *rc* – perhaps a difference of genre, or of region, or of domain. The list may well be dominated by other differences, which we are not at all interested in. Keyword lists tend to work best where the corpora are very well matched in all regards except the one in question. This is a question of how *fc* and *rc* have been prepared. It is often the greatest source of bewilderment when users see keyword lists (and makes keyword lists good tools for identifying the characteristics of corpora). However it is an issue of corpus construction, which is not the topic of this paper, so it is not discussed further.
2. “Burstiness”. If a word is the topic for one text in the corpus, it may well be used many times in that text with its frequency in that corpus mainly coming

from just one text. Such “bursty” words do a poor job of representing the overall contrast between *fc* and *rc*.

This, again, is not the topic of this paper. A range of solutions have been proposed, as reviewed by Gries (2007). The method we use in our experiments is “average reduced frequency” (ARF, Savický and Hlaváčová 2002) which discounts frequency for words with bursty distributions: for a word with an even distribution across a corpus, ARF will be equal to raw frequency, but for a word with a very bursty distribution, only occurring in a single short text, ARF will be a little over 1.

3. You can’t divide by zero. It is not clear what to do about words which are present in *fc* but absent in *rc*.
4. Even setting aside the zero cases, the list will be dominated by words with very few words in the reference corpus: there is nothing very surprising about a contrast between 10 in *fc* and 1 in *rc*, giving a ratio of 10, and we expect to find many such cases, but we would be very surprised to find words with frequency per million of 10,000 in *fc* and only 1,000 in *rc*, even though that also gives a ratio of 10. Simple ratios will give a list of rarer words.

The last problem has been the launching point for an extensive literature. The literature is shared with the literature on collocation statistics, since formally, the problems are similar: in both cases we compare the frequency of the keyword in condition 1 (which is either “in *fc*” or “with collocate *x*”) with frequency in condition 2 (“in *rc*” or “not with collocate *x*”). The literature starts with Church and Hanks (1989) and other much-cited references include Dunning (1993), Pederson. Proposed statistics include Mutual Information (MI), Log Likelihood and Fisher’s Exact Test, see Chapter X of Manning and Schütze (1999). I have argued elsewhere (Kilgarriff 2005) that the mathematical sophistication of MI, Log Likelihood and Fisher’s Exact Test is of no value to us, since all it serves to do is to disprove a null hypothesis - that language is random - which is patently untrue.

Sophisticated maths needs a null hypothesis to build on and we have no null hypothesis: perhaps we can meet our needs with simple maths.

A common solution to the zeros problem is “add one”. If we add one to all the frequencies, including those for words which were present in *fc* but absent in *rc*, then we have no zeros and can compute a ratio for all words. A word with 10 wpm in *fc* and none in *rc* gets a ratio of 11:1 (as we add 1 to 10 and 1 to 0) or 11. “Add one” is widely used as a solution to a range of problems associated with low and zero frequency counts, in language technology and elsewhere (Manning and Schütze 1999). “Add one” (to all counts) is the simplest variant: there are sometimes reasons for adding some other constant, or variable amount, to all frequencies.

This suggests a solution to problem 4. Consider what happens when we add 1, 100, or 1000 to all counts-per-million from both corpora. The results, for the three

words *obscurish*, *middling* and *common*, in two hypothetical corpora, are presented below:

Add 1:

word	wpm in <i>fc</i>	wpm in <i>rc</i>	adjusted, for <i>fc</i>	adjusted, for <i>rc</i>	Ratio	Rank
<i>obscurish</i>	10	0	10+1=11	0+1=1	11.0	1
<i>middling</i>	200	100	200+1=201	100+1=101	1.99	2
<i>common</i>	12000	10000	12000+1=12001	10000+1=10001	1.20	3

Add 100:

word	wpm in <i>fc</i>	wpm in <i>rc</i>	adjusted, for <i>fc</i>	adjusted, for <i>rc</i>	Ratio	Rank
<i>obscurish</i>	10	0	10+100=110	0+100=100	1.10	3
<i>middling</i>	200	100	200+100=300	100+100=200	1.50	1
<i>common</i>	12000	10000	12000+100=12100	10000+100=10100	1.20	2

Add 1000:

word	wpm in <i>fc</i>	wpm in <i>rc</i>	adjusted, for <i>fc</i>	adjusted, for <i>rc</i>	Ratio	Rank
<i>obscurish</i>	10	0	10+1000=1010	0+1000=1001	1.01	3
<i>middling</i>	200	100	200+1000=1200	100+1000=1100	1.09	2
<i>common</i>	12000	10000	12000+1000=13000	10000+1000=11000	1.18	1

Tables 1-3: Frequencies, adjusted frequencies, ratios and keyword ranks for “add 1”, “add 100” and “add 1000” for rare, medium and common words.

All three words are notably more common in *fc* than *rc*, so all are candidates for the keyword list, but they are in different frequency ranges.

- When we add 1, *obscurish* comes highest on the keyword list, then *middling*, then common.
- When we add 100, the order is *middling*, *common*, *obscurish*.
- When we add 1000, it is *common*, *middling*, *obscurish*.

Different values for the “add-N” parameter focus on different frequency ranges.

For some purposes a keyword list focusing on commoner words is wanted, for others, we want to focus on rarer words. Our model lets the user specify the keyword list they want by using N as a parameter. Note that other statistics proposed for this problem offer no such parameter so are not well suited to the fact that different research concerns lead to researchers being interested in words of different frequency ranges.

This approach to keyword list generation has been implemented in the Sketch Engine, a leading corpus tool in use at a number of universities and dictionary publishers

(<http://www.sketchengine.co.uk>) After some experimentation, we have set a default value of $N=100$, but it is a parameter that the user can change: for rarer phenomena like collocations, a lower setting seems appropriate.¹

To demonstrate the method we use BAWE, the British Academic Written English corpus (Nesi 2008). It is a carefully structured corpus so, for the most part, two different subcorpora will vary only on the dimension of interest and not on other dimensions, *cf* problem 1 above. The corpus comprises student essays, of which a quarter are *Arts and Humanities* (ArtsHum) and a quarter are *Social Sciences* (SocSci). Fig 1 shows keywords of ArtsHum vs. SocSci with parameter $N=10$.

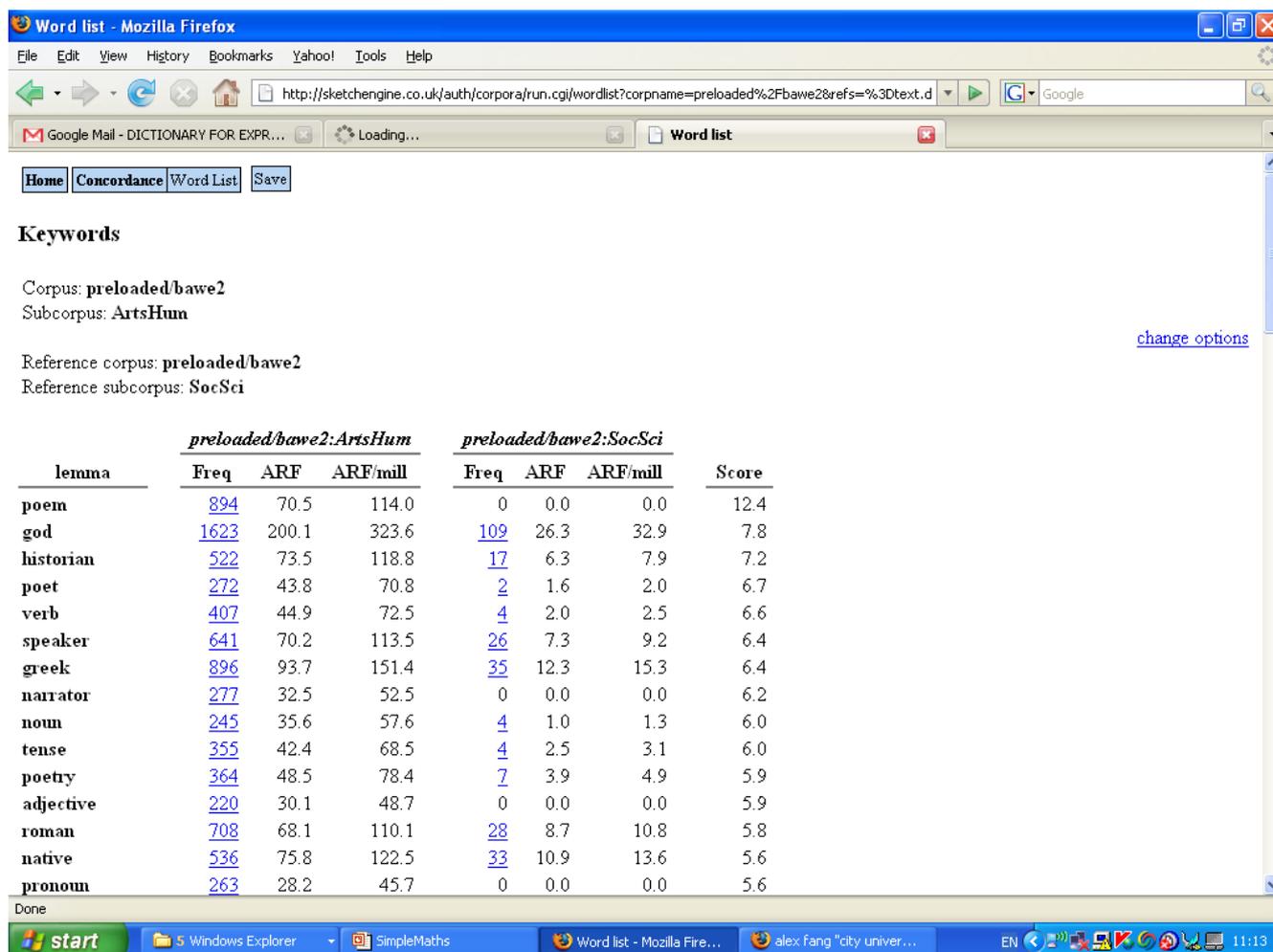


Fig 1: Keywords for ArtsHum compared with SocSci essays in BAWE, with parameter $N=10$, using the Sketch Engine.

We see here a range of nouns and adjectives relating to the subject matter of the arts and humanities, and which are, plausibly, less often discussed in the social sciences.

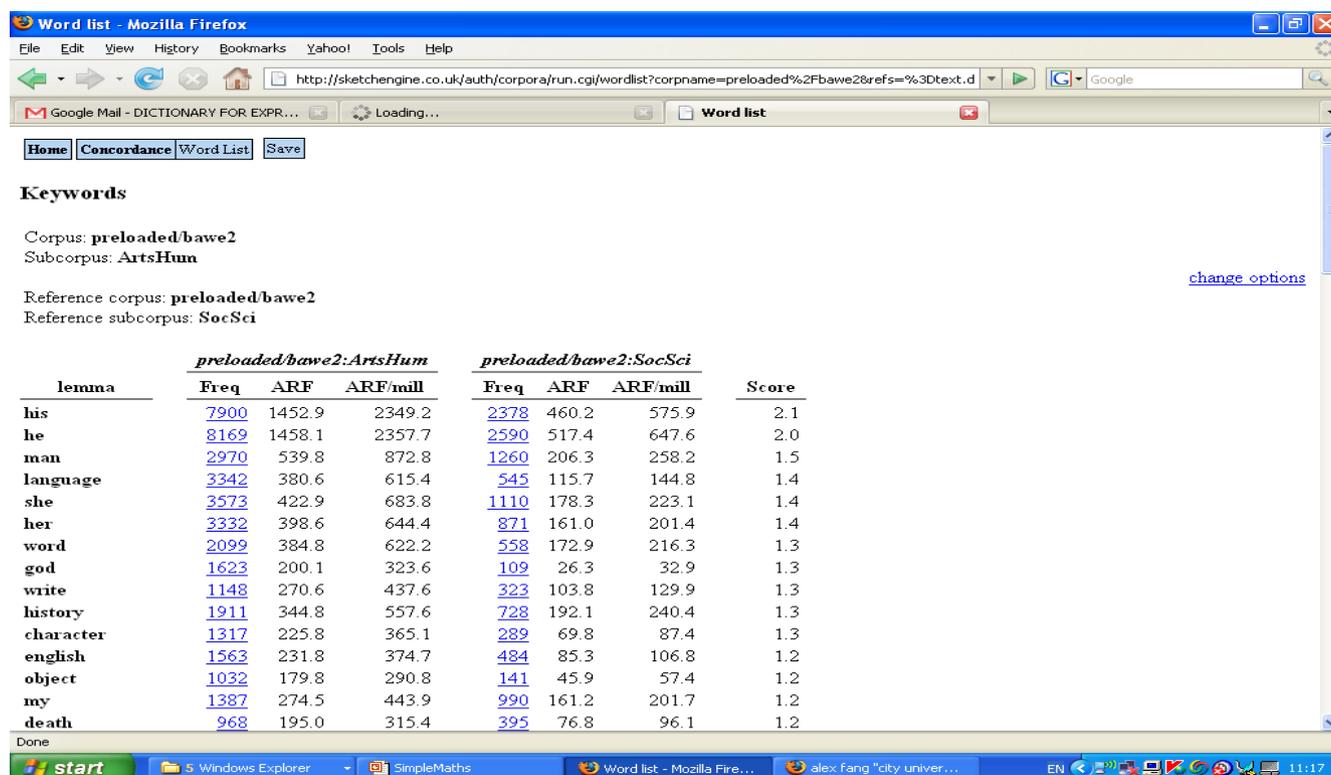


Fig 2: Keywords for ArtsHum compared with SocSci essays in BAWE, with parameter N=1000, using the Sketch Engine.

Fig. 2 shows the same comparison but with N=1000. Just one word, *god* (lowercased in the lemmatisation process) is in the top-15 samples of both lists, and Fig 1 has *historian* against Fig 2's *history*. Many of the words in both lists are from the linguistics domain, with them providing corroborating evidence for this being distinctive of ArtsHum in BAWE, but none are shared. Fig 1 has the less-frequent *verb*, *noun*, *adjective*, *pronoun*, *tense*, *native* and *speaker* (the last two possibly in the compound *native speaker*) whereas Fig 2 has the closely-allied but more-frequent *language*, *word*, *write*, *english* and *object*. (*Object* may be in this list by virtue of its linguistic meaning, 'the object of the verb', or its general or verbal meanings, 'the object of the exercise', 'he objected'; this would require further examination to resolve.)

The most striking feature of the Fig 2 is the four third-person and one first-person pronouns. Whereas the first list told us about recurring topics in ArtsHum vs. SocSci, the second also tells us about a grammatical contrast between the two varieties.

In summary

I have summarised the challenges in finding keywords of one corpus vs. another. I have proposed a method which is both simpler than other proposals and which has the advantage of a parameter which allows the user to specify whether they want to focus on higher-frequency or lower-frequency keywords. I have pointed out that there are

no theoretical reasons for using more sophisticated maths and I have demonstrated the proposed method with corpus data. The new method is implemented in the Sketch Engine corpus query tool and I hope it will be implemented in other tools shortly.

References

- Church, K. and P. Hanks 1990. Word association norms, mutual information and lexicography. *Computational Linguistics* 16(1), 22_29.
- Dunning, T. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics* 19(1), 61–74.
- Gries, S. 2008. Dispersion and Adjusted Frequencies in Corpora. *Corpus Linguistics* 13 (4), pp 403–437.
- Kilgarriff, A. 2005. [Language is never ever ever random](#). *Corpus Linguistics and Linguistic Theory* 1 (2): 263–276.
- Manning, C. and H. Schütze 1999. *Foundations of Statistical Natural Language Processing*, MIT Press. Cambridge, MA.
- Nesi, H. (2008) BAWE: [An introduction to a new resource](#). In Proc. 8th Teaching and Language Corpora Conference. Lisbon, Portugal: 239–246.
- Pedersen, T. 1996. Fishing for exactness. *Proceedings of the Conference of the South-Central SAS Users Group*, 188–200.
- P. Savický and J. Hlaváčová. 2002. Measures of word commonness. *Journal of Quantitative Linguistics*, 9: 215–231.

ⁱ Note that the value of the parameter is related to the value of one million that we use for normalising frequencies. If we used ‘words per thousand’ rather than ‘words per million’, the parameter would need scaling accordingly.