Automatic extraction of collocations: a new Web-based method

Jean-Pierre Colson

Institut libre Marie Haps – BE-1050 Brussels – Belgium

Université catholique de Louvain – BE-1348 Louvain-la-Neuve – Belgium

Abstract

The automatic extraction of collocations from large corpora or the Internet poses a daunting challenge to computational linguistics. Indeed, previous statistical methods based on bigram extraction have shown their limitations, and there is besides no theoretical consensus on the extension of parametric methods to trigrams or higher n-grams. This is a key issue, because the automatic extraction of significant n-grams has important implications for computer-aided translation, translation quality assessment, automated text correction, terminology and computational lexicography. This paper reports promising results that were obtained by using a totally different approach to the automatic extraction of significant n-grams of any size. Instead of having recourse to statistical scores, the method is based on the testing of proximity algorithms that corroborate the native speaker's competence about existing collocations. It is argued that compound terminology and phraseology in the broad sense can be captured by algorithms based on linguistic co-occurrence phenomena. This is made possible by a subtle manipulation of the Application Programming Interface (API) of a Web search engine, in this case Yahoo. The algorithm presented here, the Web Proximity Measure (*WPR*), has been tested on about 4,000 collocations mentioned in traditional dictionaries and on 340,000 n-grams extracted from the Web 1T or 'Google n-grams'. The results show precision and recall scores superior to 0.9.

Keywords: automatic extraction, collocations, Web 1T, Google n-grams, phraseology

1. Introduction

The search for an algorithm that could extract all collocations from large corpora or the Internet has something of the alchemist's quest for the philosophical stone that would turn any metal to gold. Automatically recognizing structures that are semantically associated is indeed a particularly challenging mission, and it is no wonder that this research topic has attracted many researchers from the fields of computational linguistics, but also information theory, terminology, statistics, phraseology, corpus linguistics and even applied linguistics.

It comes as no surprise that the terminology used by the different researchers varies a lot according to their country, their native language or the linguistic school to which they belong. The notion of collocation goes back to traditional and corpus linguistics (Palmer, 1938; Firth, 1957) and has received very different definitions. An extensive overview of them falls beyond the scope of the present contribution, but it should be reminded that this term originates from British traditional linguistics. Collocations were broadly defined by Firth as the company that words keep (Palmer, 1968). Within the framework of systemic functional linguistics, M.A.K. Halliday sees them as a linear co-occurrence relationship among lexical items which co-occur together (Halliday, 1966). In corpus linguistics, John Sinclair's simple definition of collocations is also well known: the occurrence of two or more words within a short space of each other in a text (Sinclair, 1991). Some linguists integrate the notion of restricted collocation (Aisenstadt, 1953) as opposed to idioms and free word combinations, or distinguish between restricted, semantic and syntactic collocations (Moon, 1998).

In this short report of an innovative method for automatic extraction, we will use the notion of collocation in the broadest sense, corresponding to Hoey's definition: «Collocation has long been the name given to the relationship of a lexical item with items that appear with greater than random probability in its (textual) context» (Hoey, 1991: 6-7).

From a semantic point of view, those multi-word units will be recognized by native speakers as belonging together. The decisive criterion in Hoey's definition, however, is a statistical one, and therefore collocations are likely to correspond to a broad palette of more or less fixed expressions such as compound proper nouns (e.g. *Mount Rushmore*), compound nouns (*city breaks*), compound terms (*market capitalization*), noun-adjective combinations (*sharp criticism*), idioms (*spill the beans*), routine formulae (*long time no see*), proverbs and sayings (*it takes two to tango*), quotations (*Away, and mock the time with fairest show*) and even well-known song or film titles (*Gone with the wind*).

Church and Hanks (1990) have carried out pioneer work on word pair statistics and were followed by many other researchers. Evert (2004) has provided an extensive overview of those studies and a detailed analysis of all possible statistical formulae and parameters (more than 30 different ones). According to Deane (2005), however, the performance of those statistical scores is generally low, and the precision rate falls rapidly after the most frequent phrases in the list. Other critical discussions of the statistical approach (Heid, 2007) point to the very diverse results that were obtained by applying different association measures to the same corpus. Besides, most studies are based on bigrams, for which the 2x2 contingency table can be applied, whereas «an extension to higher dimensions (three, four, ... words) is not yet fully understood even theoretically» (Heid, 2007: 1042).

A closely related issue is that of the lexical bundles that were identified by Biber (1999) and defined as «the most frequent recurring lexical sequences in a register» (Biber, 2004: 376). It should be stressed that Biber's approach is purely frequency-driven. Biber (2004), for instance, extracted four-word lexical bundles (fourgrams) by selecting sequences that were used in at least five different texts, with a frequency cut-off of 40 occurrences per million words.

2. Methodology

The starting point of our methodology is Web-based. It is beyond the scope of this paper to go into all the details of using the Web as a linguistic corpus. Kilgariff and Grefenstette (2003) have argued that there are far more advantages than drawbacks in such an approach, in the first place of course the immediate access to huge corpora in many languages. Baroni (2008) has shown that the distributions in text obtained from traditional corpora on the one hand and Web corpora on the other are roughly the same. Besides, the huge lexical frequencies on the Web make it possible to extract all types of collocations, including idioms and compound terms (Colson, 2007; 2008). There have been attempts to adapt the statistical extraction of n-grams (mainly by means of Mutual Information) to the Web (Turney, 2001; Baroni and Bisi, 2004; Baroni and Vegnaduzzo, 2004). For all their interest, those studies are again limited to bigrams and cannot be easily extended to trigrams or higher n-grams.

How could we possibly separate the wheat from the chaff and extract all significant collocations from the Web? After months of experimentation with simple and more complex algorithms, the most promising results of this research were obtained by directly exploiting the associative power of search engines and their underlying algorithms. The basic principle is easy to understand: if you type collocations such as *separate the wheat from the chaff* or *sharp criticism* (without the quotation marks) on Google (search for all the words), the first results appearing on your screen are indeed instances of these English phrases, as if you had searched for the *exact* phrase. The reason is obvious: Web search algorithms, in much the same way as complex text search algorithms (*e.g*. regular expressions), are designed to put the best matches on top of the result list. It would therefore be tempting to simply compare the frequencies in order to extract significant collocations.

Our provisional score, the Web Proximity Measure at gram level n (*WPR* with n=2 to 7) would then correspond to the division of the frequency with the exact search (F_e) by the frequency with all words (F_a):

$$WPRn = \frac{Fe(GR2, GR3...GRn)}{Fa(GR2, GR3...GRn)} \qquad (1)$$

Unfortunately, equation (1) does not work very well, because frequency figures yielded by search engines are not quite reliable, as already pointed out by Lüdeling et al. (2007). They are based on approximations and may vary a lot according to the search engine used, and from one month to another. In many cases, the frequency for the exact phrase will even be higher than the frequency for all the words. Thus, typing the collocation *gift of the gab* in October 2009 on Bing (www.bing.com) yielded 961,000 matches for 'all the words' and 28,500,000 for 'the exact phrase'.

Instead of working with the global frequencies as described in equation (1), the method proposed here takes advantage of the natural associative power of the search engine, by considering the proportion of best matches. The results returned by the search engine API (in this case Yahoo [1]) for each request constitute the sample. Our solution has to be workable on average computers, so we will restrict the search to one request per n-gram. One API request sends back 50 results on the Bing API and 100 on Yahoo. The terminology used by the search engines varies slightly, but the results returned include here the title of the Web page and the summary. In order to take these elements into account, we come to the following formula.

S_x will stand for the sample of size x that will be used; the frequency of exact matches for the n-grams is then checked on the sample and the Web Proximity Measure *WPR* is the score obtained after the division by 2x (because we check both the title and the summary). *WPR* will receive a value between 0 and 1.

In short, we propose the following formula:

$$WPRn(Sx) = \frac{Fe(GR2, GR3...GRn)}{2x} \qquad (2)$$

[1] See: http://developer.yahoo.com/about/. The API (Application Programming Interface) makes it possible to send requests programmatically to the search engine. The Yahoo! Search Web Services allow for 5,000 requests per IP per day. Microsoft's new Bing API (http://msdn.microsoft.com/en-us/library/dd900818.aspx) imposes no limitation, but yields less accurate results at the moment of this writing. Google has put an end to its Soap API, and its new AJAX Search API (http://code.google.com/apis/ajaxsearch/) prohibits automated search.

Equation (2) translates for the computer the observation that, checking a collocation on the search engine without the quotation marks (*all the words* but not *the exact phrase*), you are struck by the many *exact results* that will appear on the screen. The *Web Proximity Measure* is the proportion of those exact matches in the first results.

The method may at first sight look simple, as compared to sophisticated statistical parameters. It should be pointed out, however, that the underlying algorithms used by search engines in order to produce these results are far from trivial. Besides, the sampling method proposed here offers the advantage of relying on real corpus evidence, as if an instant picture had been made of language in use. To some extent, the tendency that certain grams will display to co-occur on the Web is related to the natural way in which a native speaker will combine elements that are frequently associated. Dell Hymes (1972) already pointed out that «the capabilities of language users do include some (perhaps unconscious) knowledge of probabilities». The main advantage of the methodology proposed here is that it can (theoretically) be used for n-grams of any size. It is besides easy to reproduce by anyone who has a basic mastery of programming with the APIs.

Before we turn to an evaluation of the results, an important note should be made about *spamdexing*. This notion refers to the sometimes inaccurate results yielded by search engines because of manipulation techniques designed to increase the ranking of a Web page. For lack of space, the technical details will not be specified here, but the methodology includes a number of filters in order to cope with spamdexing. The results sent back by the search engine sometimes display a lot of identical lines, even repetitions of the same paragraph, and filters have to be designed if we wish to constitute a valid sample.

3. Results and discussion

The results of automatic extraction of collocations are measured according to their degree of precision and recall [2] (Evert, 2004; Deane, 2005; Heid, 2007). The precision score measures whether all n-grams that were identified as collocations are indeed collocations, whereas recall will check if all collocations have been extracted by the algorithm.

In section 1, we have defined collocation as statistically significant co-occurrence, but also as the natural tendency that words will display to co-occur with a given set of other words, a crucial element for the identification of collocations by a native speaker. However, there remains some margin of interpretation in this process, because different native speakers will not always have the same judgment about the collocational character of a given n-gram. For this reason, the first evaluation of the Web Proximity Measure does not rely on the meaning of a few native speakers, but rather on existing lists of collocations.

3.1. Measuring recall on a list of collocations

In the first verification step of the Web Proximity Measure, we wish to check if phrases that are recognized as collocations by traditional dictionaries will indeed be extracted by the algorithm.

[2] We use the standard definition of precision and recall, based on the relationship between the elements that should have been detected and were indeed found, called the true positives (tp), the elements that were detected but wrongly, the false positives (fp); and the elements that were not detected and should have been, the false negatives (fn). This yields the following formulas:
$\text{Precision} = \dfrac{tp}{tp + fp}$ and $\text{Recall} = \dfrac{tp}{tp + fn}$

For this purpose, 4,000 collocations were randomly selected from two English dictionaries that may be considered as reliable tools for learners of English: the Oxford Dictionary of Collocations and the Longman Dictionary of Contemporary English. As we use collocation in its statistical meaning, including all types of set phrases, the sample does not only contain weakly idiomatic combinations such as noun-adjective combinations, but also idioms.

Some of the randomly selected collocations could not be used in the final list, because they included generic pronouns such as *someone*, *something*, *your*, as in *to pull someone's leg*. These collocations are seldom used in this form, so they were removed from the list. This left us with 3,807 collocations (1,700 weakly idiomatic and 2,107 strongly idiomatic cases).

The weakly idiomatic cases were selected from a list of collocations in combination with high frequency nouns: *argument*, *business*, *car*, *criticism*, *day*, *door*, *example*, *family*, *hand*, *home*, *house*, *idea*, *job*, *mark (noun)*, *message*, *money*, *news*, *problem*, *question*, *reaction*, *remark*, *table*, *thing*, *time*, *way*, *word.* Most of those collocations are bigrams: *trenchant criticism*, *wacky idea*, *complimentary remark*. There are however several cases of trigrams or higher grams: *not the faintest idea*, *have better things to do.*

The strongly idiomatic collocations were randomly selected and mainly consist of trigrams or higher grams: *ace in the hole*, *in dribs and drabs*, *rest on your laurels.*

The Web Proximity Measure (*WPR*) was computed for the 3,807 collocations. This took about 2 seconds per collocation on an average PC. For lack of space, the whole results cannot be mentioned in this report, but Tab. 1 presents a sample of collocations in combination with *argument*, and the *WPR*-score for each collocation:

angry argument:	0.2	powerful argument:	0.47	logical argument:	0.435
bitter argument:	0.44	sound argument:	0.335	rational argument:	0.525
heated argument:	0.61	strong argument:	0.245	reasoned argument:	0.155
violent argument:	0.295	valid argument:	0.36	economic argument:	0.635
big argument:	0.395	compelling argument:	0.425	moral argument:	0.465
little argument:	0.275	conclusive argument:	0.02	political argument:	0.505
silly argument:	0.49	convincing argument:	0.485	theoretical argument:	0.425
stupid argument:	0.46	persuasive argument:	0.4	advance the argument:	0.03
basic argument:	0.44	plausible argument:	0.25	deploy the argument:	0.015
general argument:	0.27	spurious argument:	0.21	put forward an argument:	0.14
main argument:	0.08	tenuous argument:	0.285	develop an argument:	0.075
good argument:	0.585	weak argument:	0.415	accept the argument:	0.07
major argument:	0.4	balanced argument:	0.24	dismiss the argument:	0.045

Table 1: Sample results from the collocation list

It should be stressed that the Web Proximity Measure is just a mirror of actual language use. Thus, a figure of 0.2 means that there were 40 instances of the exact phrase in the 100 results sent back by the search engine (as both the title and the summary were checked, see equation 2 above). The statistical scores and algorithms underlying the search engine are complex and are of course kept secret. It is therefore no easy matter to establish the significance threshold for the Web Proximity Measure. The solution proposed here relies rather on experiments corroborating the native speaker's intuition on the collocational character of the combinations. On the basis of a systematic analysis of the scores obtained in hundreds of texts in which all n-grams were given a *WPR*, we come to the hypothesis of a significant threshold level at *WPR = 0.065* for

English, if the Yahoo search engine is used. With such a score, we consider a collocation as significant if at least 13 results were exact matches in the 100 first results sent by the search engine.

If we use this threshold for the *WPR*, we can now compute the recall for our sample list of 3,807 English collocations. We will start from the reasonable hypothesis that the information found in the above mentioned dictionaries is reliable, which means that all 3,807 collocations should be considered by the algorithm as true positives. If we get a *WPR*-score that is lower than 0.065 for any of them, we will consider that the algorithm has produced false negatives. We can now apply the formula presented in note (2) above, which yields the following recall scores for both parts of the sample:

Sample collocations (N=3807)	*Recall*
Weakly idiomatic (N=1700)	0.91
Strongly idiomatic (N=2107)	0.92
Average	0.915

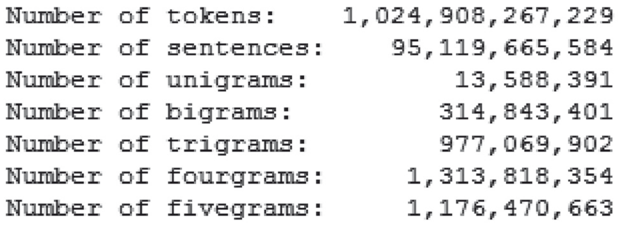
Table 2: Measure of recall in the collocation sample

As a first evaluation of the recall yielded by the *WPR*, these results are quite satisfactory, because more than 90 percent of all collocations are indeed extracted by the algorithm.

3.1. Measuring precision on the Web1T

A note of caution is necessary on the performance of any extraction algorithm. As pointed out by Deane (2005) and Heid (2007), there is a need for algorithms that are not limited to bigrams (this first requirement is met by the *WPR*), and also for reproducible experiments with huge corpora. In the second evaluation step of the *WPR*, we will therefore try to measure precision on the largest English corpus available: the Web1T, also known as *the Google n-grams*.

In 2006, Google Inc. decided to share an n-gram-database with researchers by creating a gigantic list of English n-grams from the Web [3]. As shown in Tab. 3, the Web1T presents the linguist with an impressive collection:



```
Number of tokens:    1,024,908,267,229
Number of sentences:    95,119,665,584
Number of unigrams:         13,588,391
Number of bigrams:         314,843,401
Number of trigrams:        977,069,902
Number of fourgrams:     1,313,818,354
Number of fivegrams:     1,176,470,663
```

Table 3: Distribution of n-grams on the Web1T
(source: http://googleresearch.blogspot.com/2006/08/all-our-n-gram-are-belong-to-you.html)

[3] The Web1T is distributed by the Linguistic Data Consortium, http://www.ldc.upenn.edu/. It consists of 6 DVDs. The n-grams were generated from about 1 trillion word tokens from the Web and the corpus is therefore called the Web1T. The Web1T can be easily manipulated by several free progams such as ssgnc (http://code.google.com/p/ssgnc/wiki/Introduction). The Computation Linguistics group of the University of Osnabrück (http://www.cogsci.uni-osnabrueck.de/~korpora/ws/cgi-bin/Web1T5/Web1T5_freq.perl) offers a very useful tool on its Web site.

It is noteworthy that the Web1T is not a corpus in the strict sense, because it was not assembled by linguists according to the accurate methodology of corpus linguistics. However, Google produced this list by applying a strict language selection to the Web and by choosing only relevant text pages, so that it is more reliable than a direct search on unsorted Web material. Besides, the figure of one trillion words corresponds to 10,000 times the British National Corpus in its commercial version (100 million words), and the basic principles of statistics predict that such a huge collection will probably cover most aspects of the actual English usage.

If a standard reference corpus must be established for measuring the efficiency of automatic collocation extraction (in English), the Web1T is certainly one of the best candidates. In just a few seconds, the researcher receives a list of thousands of n-grams starting with, ending by, or containing any given gram. The challenge is again to separate the wheat from the chaff in this linguistic abundance.

The verification of all n-grams from the Web1T is a huge task, and one should besides take the cumulative aspect of n-grams into account: they may be collocations, not as a whole, but simply because they are part of a collocation at a higher level of n-grams. For example, the bigram *spill the* will probably be extracted as a significant collocation, because it is part of the collocational trigram *spill the beans*. Like Russian puppets, collocations at the level of sixgrams of sevengrams may contain several lower level collocations. This problem is particularly acute when huge collections are the object of analysis.

In the case of the Web1T, it is therefore reasonable to start, not from bigrams, but from the highest level available: fivegrams. This does not mean that further verification of the lower levels is superfluous, but this paper focuses on this important step in the evaluation of the Web Proximity Measure.

The following figures are the result of the application of the *WPR*-score to the same list of 26 nouns as in section 3.1. (*argument*, *business*, *car…*), which was completed by a few high-frequency verbs (*do*, *make*, *meet*, *take*). Position 5 in the fivegram was chosen for the nouns (e.g. *two sides to every argument*) in order to observe adjective combinations and other nominal constructions, whereas position 1 was selected for the verbs, in order to capture a complement structure (e.g. *makes the world go round*). For verbs, the infinitive form and the third person singular were checked (*make/makes*). All in all, 34 lists of 10,000 fivegrams were submitted to the *WPR*-score, totalizing 340,000 fivegrams.

One of the major findings of this experiment is that *frequent co-occurrences are not necessarily collocations.* In all examples that were studied, there was a high discrepancy between the frequency rank and the *WPR*-rank.

To illustrate this, Tab. 4 presents the first part of the listing for one of the series: all fivegrams from the Web1T beginning with MAKES. The listing displays the 20 first results in order of decreasing *WPR*-score. Tab. 4 also mentions the frequency rank on the Web1T (FrRank), and the total frequency of the fivegram on the Web1T (Fr).

It takes no statistiscal formula to notice that the Web Proximity ranks and the frequency ranks do not correlate. *Makes every yesterday a dream*, for instance, gets a *WPR*-rank of 14 and a frequency rank of 1618. Is it a collocation? Obviously, because it is part of a popular quotation: "Today well lived makes every yesterday a dream of happiness and every tomorrow a vision of hope. Look well therefore to this day" [4]. In the same way, the second *WPR*-rank, *makes the heart*

[4] The origin of this quotation appears to be a Sanskrit poem.

grow fonder, is included in the well-known phrase *Absence makes the heart grow fonder*, and gets only 101 as a frequency rank. We might go on like this with hundreds of examples. Further in the list (not in Tab. 4), the fivegram *makes a cameo appearance* is obviously a collocation, and it is recognized by the algorithm (the *WPR*-score is 0.225 and the *WPR*-rank is 57), but its frequency rank is 1526 (Fr: 2741), and it is less frequent that many other non-collocational fivegrams such as *makes it so you can* (Fr: 2790), *makes broadband friends PUN pushes* (sic) (Fr: 2817), *makes and models looking to* (Fr: 3104), *makes the raincoat flashers look* (Fr: 3492), *makes citroen fiat ford isuzu* (Fr: 4035) or *makes acura alfa romeo amc* (Fr: 5771).

```
(WPrRank 1) makes up for lost time (FrRank 89) (Fr 24390): 0.785
(WPrRank 2) makes the heart grow fonder (FrRank 101) (Fr 21301): 0.515
(WPrRank 3) makes the world go around (FrRank 221) (Fr 12709): 0.415
(WPrRank 4) makes no representation that materials (FrRank 136) (Fr 17643): 0.385
(WPrRank 5) makes searching the web easy (FrRank 320) (Fr 9671): 0.385
(WPrRank 6) makes her clothes fall off (FrRank 624) (Fr 5614): 0.38
(WPrRank 7) makes no sense at all (FrRank 49) (Fr 35993): 0.38
(WPrRank 8) makes no warranty or representation (FrRank 121) (Fr 19005): 0.37
(WPrRank 9) makes me want to puke (FrRank 340) (Fr 9284): 0.37
(WPrRank 10) makes no representations or warranties (FrRank 1) (Fr 354898): 0.365
(WPrRank 11) makes integer from pointer without (FrRank 208) (Fr 13175): 0.365
(WPrRank 12) makes no guarantee or warranty (FrRank 112) (Fr 19897): 0.365
(WPrRank 13) makes me want to vomit (FrRank 453) (Fr 7168): 0.365
(WPrRank 14) makes every yesterday a dream (FrRank 1618) (Fr 2601): 0.36
(WPrRank 15) makes up the bulk of (FrRank 391) (Fr 8161): 0.36
(WPrRank 16) makes use of the zend (FrRank 321) (Fr 9655): 0.355
(WPrRank 17) makes pointer from integer without (FrRank 14) (Fr 79858): 0.345
(WPrRank 18) makes no warranty or guarantee (FrRank 557) (Fr 6110): 0.34
(WPrRank 19) makes extensive use of adobe (FrRank 1773) (Fr 2397): 0.335
(WPrRank 20) makes the world go round (FrRank 30) (Fr 47917): 0.335
```

Table 4: WPR-score for fivegrams with 'makes' on the Web1T

Let us now pay attention to another example, with the noun *argument* at the end of the fivegram. Thus, all fivegrams from the Web1T ending with *argument* were submitted to the *WPR*-score. In Tab. 5 are the 30 most frequent fivegrams of this type on the Web1T:

14898331	PUN PUN PUN supplied argument	14239	<S> warning PUN missing argument
889940	NUM warning PUN supplied argument	14223	PUN PUN warning PUN argument
127110	NUM warning PUN invalid argument	13110	has been deprecated - argument
98867	pointer targets in passing argument	13009	PUN warning PUN passing argument
94064	for the sake of argument	11657	<S> it is the argument
66454	<S> warning PUN invalid argument	11543	<S> there is an argument
54076	wrong datatype for second argument	11487	this is not an argument
32681	builtin and then its argument	10926	determined unanimously that oral argument
30981	PUN PUN PUN the argument	10895	the i - th argument
27022	PUN PUN PUN first argument	10871	- merge UNK PUN argument
25963	both sides of the argument	10863	PUN NUM PUN query argument
25231	warning - UNK param argument	10787	there is a strong argument
18018	PUN PUN the first argument	10649	int PUN PUN but argument
16455	politics PUN essays PUN argument	10444	NUM warning PUN missing argument
14866	ordered submitted without oral argument	10277	who has the better argument

Table 5: Raw frequencies on the Web1T for fivegrams ending with 'argument'

Raw frequencies of this type do not make much sense. Tab. 6 presents the first 30 results obtained by the Web Proximity Measure:

(WPrRank 1) main thrust of his argument (FrRank 1731) (Fr 511): 0.43

(WPrRank 2) to make a convincing argument (FrRank 478) (Fr 1317): 0.41

(WPrRank 3) the crux of his argument (FrRank 430) (Fr 1427): 0.405

(WPrRank 4) to make a compelling argument (FrRank 796) (Fr 874): 0.405

(WPrRank 5) to make a coherent argument (FrRank 1828) (Fr 484): 0.395

(WPrRank 6) became involved in an argument (FrRank 1129) (Fr 688): 0.385

(WPrRank 7) engaged in a heated argument (FrRank 1549) (Fr 546): 0.375

(WPrRank 8) has automatically lost whatever argument (FrRank 1099) (Fr 699): 0.375

(WPrRank 9) got into a heated argument (FrRank 270) (Fr 1929): 0.365

(WPrRank 10) options prepended to the argument (FrRank 1645) (Fr 530): 0.365

(WPrRank 11) got into a huge argument (FrRank 1512) (Fr 552): 0.36

(WPrRank 12) has been deprecated - argument (FrRank 18) (Fr 13110): 0.355

(WPrRank 13) the crux of my argument (FrRank 425) (Fr 1437): 0.34

(WPrRank 14) pointer targets in passing argument (FrRank 4) (Fr 98867): 0.335

(WPrRank 15) to make a closing argument (FrRank 1453) (Fr 569): 0.325

(WPrRank 16) flip side of the argument (FrRank 1992) (Fr 453): 0.32

(WPrRank 17) get into a heated argument (FrRank 732) (Fr 944): 0.315

(WPrRank 18) to make a strong argument (FrRank 1308) (Fr 611): 0.315

(WPrRank 19) can make a strong argument (FrRank 825) (Fr 848): 0.31

(WPrRank 20) had gotten into an argument (FrRank 753) (Fr 913): 0.3

(WPrRank 21) perhaps the most compelling argument (FrRank 1348) (Fr 601): 0.3

(WPrRank 22) either side of the argument (FrRank 219) (Fr 2250): 0.295

(WPrRank 23) event handler receives an argument (FrRank 355) (Fr 1616): 0.295

(WPrRank 24) he got into an argument (FrRank 421) (Fr 1444): 0.295

(WPrRank 25) how to win an argument (FrRank 190) (Fr 2484): 0.295

(WPrRank 26) opposite side of the argument (FrRank 1777) (Fr 498): 0.295

(WPrRank 27) the force of his argument (FrRank 991) (Fr 753): 0.295

(WPrRank 28) they got into an argument (FrRank 431) (Fr 1423): 0.29

(WPrRank 29) two sides to every argument (FrRank 1135) (Fr 684): 0.285

(WPrRank 30) given by the pathname argument (FrRank 692) (Fr 991): 0.27

Table 6: WPR-score for fivegrams ending by 'argument'

While several collocations presented in Tab. 6 belong to the domain of computer science (ranks 12, 14, 23, 30), it is obvious that this list, which does not correlate with raw frequency either, contains clear cases of collocation. *Perhaps the most compelling argument* (*WPR*-rank 21, Frequency rank 1348) for using the Web Proximity Measure, is its pedagogical use. For non-native speakers of English (even at advanced level), it is particularly enlightening to be confronted in a matter of a few seconds with useful collocational fivegrams such as *the main thrust of his argument*, *the crux of his argument*, *the flip side of the argument*, etc.

The precision and recall are not easy to establish for this fivegram sample from the Web1T. Indeed, even experienced linguists would not be able to detect all significant parts of phrases, proverbs, quotations and clichés that are included in the fivegrams. The following fivegrams starting with *give me*, for instance, were all extracted by the *WPR*-score from the Web1T and identified as significant combinations, but they are not easily recognized as (parts of) collocations:

Give me that old (*give me that old time religion*)*;* give me the heebie (*give me the heebie-jeebies*); give me here John Baptist (*give me here John Baptist's head in a charger*, Matthew 14:8); give me I will surely (*and of all that thou shalt give me I will surely give the tenth* *unto thee,* Genesis 28:22); give me a home where (*give me a home where the buffaloes roam,* Cowboy song); give me the strength to save (*Give me the strength to save some life*, Firemans Prayer); give me would be greatly (*any help you give me would be greatly appreciated*); give me convenience or give (*Give me convenience or give me death*, song by the Dead Kennedys); give me a ballpark figure (*a rough estimate*); give me five loaves of (*give me five loaves of bread, or whatever can be found*, Samuel 21:3); give me no lip child (*Don't give me no lip child lyrics*, song by the Sex Pistols); give me excess of it (*If music be the food of love, play on, Give me excess of it.* Shakespeare, Twelfth Night, Act 1, Scene 1).

Web-driven collocational fivegrams indeed include, among others, a lot of pop song lyrics, but also many biblical and literary quotes.

It would be an impossible mission to manually check in this way the recall score for the 340,000 fivegrams of this experiment. The precision score, however, was computed for the 200 first results obtained for each of the 34 series of fivegrams, which yielded an impressive average precision of 0.965.

From a distributional point of view, all significant collocational fivegrams obtained by means of the *WPR*-score display a curve that comes fairly close to that of Zipf-Mandelbrot. To illustrate this, Tab. 7 shows a log-log table for 5 series of 2,000 fivegrams. This time, the *WPR*-score was computed for 2,000 fivegrams containing resp. *argument*, *choice*, *criticism*, *family*, *time* as the last gram (e.g.: *main thrust of his argument*). Tab. 7 presents as x-axis the log10 of the *WPR*-rank and starts therefore at 0 for rank 1, while the y-axis corresponds to the log10 of the *WPR*-score. As this score lies between 0 and 1, it was first multiplied by 100 in order to visualize the results more easily. Thus, the *WPR*-rank 1 for *time* corresponds to a *WPR*-score of 0.735. If we compute the log10 of 73.5, we obtain 1.86, as shown in Tab. 7. The dramatic fall of the *WPR*-score occurs for all fivegrams a little before x=3 (*WPR*-rank 1,000), more precisely around rank 800.

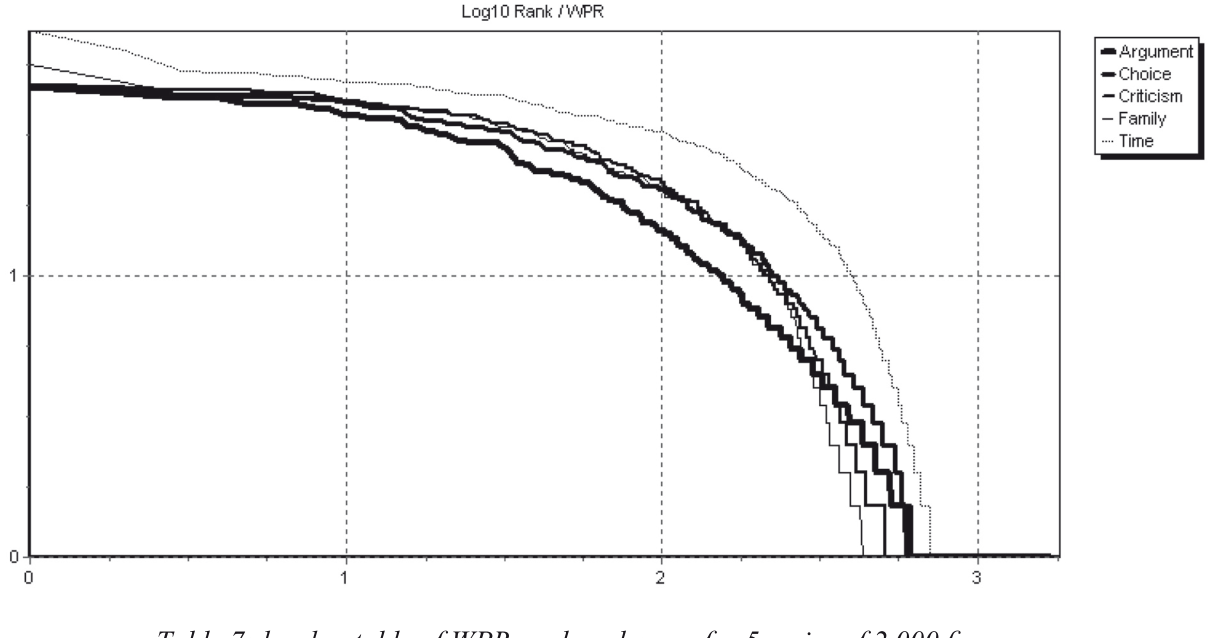


Table 7: log-log table of WPR-rank and score for 5 series of 2,000 fivegrams

Presenting those results in a log-log table makes it possible to check the Zipfian character of the distribution. It is well-known that Zipf's law predicts in this case a straight line, with an abrupt fall at the end (Mandelbrot's correction). The picture we get from this sample is fairly *zipfian*, which may have implications for the relationship between collocations and information theory. If verified by other experiments, the zipfian character of collocational n-grams may shed some light on the natural efficiency of information in language.

4. Conclusion

Extracting significant collocations at the level of bigrams, but also of trigrams and higher grams remains one of the major challenges for corpus-driven linguistics. Lexicography, terminology, but also other fields such as automatic translation may in the future profit from new extraction algorithms that would be applicable to all n-gram levels. In this paper, the first results of an exploratory research indicate that a proximity-based method may be worth exploring. By exploiting the fascinating proximity algorithms used by search engines, the method proposed here for extraction of significant fivegrams already produces results that can be directly used by foreign language students and lexicographers. The underlying statistical mechanisms involved may offer fresh insights into the complex relationship between co-occurrence and collocation. The Web Proximity Measure also provides a new key to the *Google n-grams*. Finally, it seems to confirm that collocations can be defined by statistical criteria, even at the level of fivegrams.

References

Aisenstadt E. (1953). Restricted collocations in English lexicology and lexicography. *Review of Applied Linguistics*, vol. 1: 53-61.

Baroni M. (2008). Distributions in text. In Lüdeling, A. and Kytö, M., editors, *Corpus linguistics. An international handbook*. Berlin: Mouton de Gruyter.

Baroni M. and Bisi S. (2004). Using cooccurrence statistics and the web to discover synonyms in a technical language. In *Proceedings of LREC 2004*, pp. 1725-1728.

Baroni M. and Vegnaduzzo S. (2004). Identifying subjective adjectives through web-based mutual information. In *Proceedings of KONVENS 2004*, pp. 17-24.

Biber D. (1999). Lexical bundles in conversation and academic prose. In Hasselgard, H. and Oksefjell, S., editors, *Out of corpora: studies in honor of Stig Johansson*. Amsterdam: Rodopi.

Biber D. (2004). If you look at... Lexical bundles in university teaching and textbooks. *Applied Linguistics*, vol. 25: 371-405.

Church K.W. and Hanks P. (1990). Word association norms, mutual information and lexicography. *Computational Linguistics*, vol. 16: 22-29.

Colson J.-P. (2007). The World Wide Web as a corpus for set phrases. In Burger, H., Dobrovol'skij, D., Kühn, P. and Norrick, N., editors, *Phraseology. An international handbook.* Berlin: Mouton de Gruyter.

Colson J.-P. (2008). Cross-linguistic phraseological studies: an overview. In Granger, S. and Meunier, S., editors, *Phraseology. An interdisciplinary perspective*. Amsterdam: John Benjamins.

Deane P. (2005). A nonparametric method for extraction of candidate phrasal terms. In *43d Annual Meeting of the Association for Computational Linguistics*, University of Michigan.

Evert S. (2004). *The statistics of word cooccurrences -word pairs and collocations*. Ph.D. thesis, University of Stuttgart.

Firth J.R. (1957). *Modes of meaning. Papers in Linguistics 1934-1951*. Oxford: Oxford University Press.

Halliday M. (1966). Lexis as a linguistic level. In Bazell, C., Catford, J., Halliday, M. and Robins, R., editors, *In Memory of J.R. Firth*. London: Longman.

Heid U. (2007). Computational linguistic aspects of phraseology. In Burger, H., Dobrovol'skij, D., Kühn, P. and Norrick, N., editors, *Phraseology. An international handbook.* Berlin: Mouton de Gruyter.

Hoey M. (1991). *Patterns of Lexis in Text*. Oxford: Oxford University Press.

Hymes D. (1972). On communicative competence. In Pride, J. and Holmes, J., editors, *Sociolinguistics*. London: Penguin.

Kilgariff A. and Grefenstette G. (editors) (2003). Web as corpus. introduction to the special issue. *Computational Linguistics*, vol. 29: 1-15.

Lüdeling A., Evert S. and Baroni M. (2007). Using Web data for linguistic purposes. In Hundt, M., Nesselhauf, N. and Biewer, C., editors, *Corpus linguistics and the Web.* Amsterdam: Rodopi.

Moon R. (1998). *Fixed Expressions and Idioms in English*. Oxford: Clarendon Press.

Palmer F. (editor) (1968). *Selected Papers of J.R. Firth 1952-1959*. Indiana University Press.

Palmer H. (1938). *A Grammar of English Words*. London: Longman.

Sinclair J. (1991). *Corpus, Concordance, Collocation*. Oxford: Oxford University Press.

Turney, P. (2001). Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of ECML 2001*, pp. 491-502.