Connector Sets (Revised)

Linas Vepstas

August 6, 2017

Abstract

Extract from the language-learning diary, reporting on an initial dataset containing connector sets. This is a revised (6 August 2017) version of the original 11 May 2017 report. It re-analyzes and expands the original analysis on a newer, larger dataset. This was motivated in part due to several errors found and fixed in the processing pipeline in late June/early July. In retrospect, it appears these errors mostly did not affect the earlier analysis, as the most significant error was introduced after the inital analysis was made. None-the-less, it seemed prudent to redo the report. Sadly, several months were lost in the confusion, requiring large datasets to be discarded.

Introduction

This is a report on a dataset of disjuncts and connector sets, extracted from MST parses of a batch of sentences. First, a recap of what these are, then a characterization of the database contents, and finally, a report on the grammatical similarity of words in the dataset.

The errors

The original report, dated 7 May 2017, was prepared on a painfully small dataset, which also (may have?) incorporated a fatal bug in the disjunct code: disjuncts were being assembled incorrectly, due to a reversed sign in the MI calculations. This bug was eventually uncovered, and so it seemed best to entirely discard the initial analysis, and instead repeat it with a newer and larger dataset that was correctly assembled. The revised analysis was done mostly in July. Sadly, the discovery of this bug required that multiple large datasets be discarded and reconstructed. This caused a month of effort to be lost.

Simultaneously, there was a lot of confusion about the efficacy of the cosine similarity measure. Initial work on cosine similarity used a filtered dataset, with the goal of filtering to reduce "noise" in the dataset, as well as to manage dataset size. It turns out that this filtering also had the undesired side-effect of destroying much of the "signal" as well – it rendered many grammatically unrelated words to be judged to be very similar. Between the accidental sign reversal, and the excessively strong data cuts, it

was all very confusing, and has taken another month to recover from this – I'm back to where I was in May, just older and wiser, now.

Summary of results

The primary results reported below are these:

* Most scores and metrics that can be assigned to connector sets give a (scale-free) Zipfian ranking distribution, and are thus fairly boring. Although there are some oddities here and there.

* The greater the average number of observations per disjunct, the more grammatically acceptable (accurate) the disjunct seems to be. This is good news: it means that the general technique is not generating ungrammatical garbage.

* Connector sets can be given a mutual information score. The distribution for the MI scores appears to be Gaussian (i.e. bell curve). This comes as a bit of a surprise. I am not aware of what kind of network theory gives a natural rise to Gaussians.

* The MI score seems to be quite good at identifying words that participate in idioms, set phrases and institutional phrases.

* The average number of connectors per disjunct, which should have indicated the part-of-speech that the word belongs to, fails to do this. This seems to be due to the fact that the dataset is polluted with lists and tables (including tables-of-contents, and indexes), all of which are mis-interpreted as sentences by the processing software. This causes some very unusual disjuncts to be constructed.

* In the earlier sample, derived from Wikipedia, it became clear that there were very few verbs that aren't relationship verbs. Wikipedia articles describe concepts and events. The relationship between these require the copula and other relationship verbs: "is", "has", "was". Wikipedia is almost completely devoid of narrative verbs: "ran" "jumped" "hit", "ate" "thought" "took". Thus, we discern two very different styles of human communication: the exchange of facts, and the exchange of stories. Narratives contain a far richer selection of verbs, and thus, for language learning, a text corpus of narratives is required. Ideally, this would be from young-adult literature, which is a bit more direct in its kinesthetic content than adult literature might be.

* Cosine similarity applied to connector sets seems to be an effective way of determining the grammatical similarity of words. Yet, it is not so unambiguously great, that other kinds of measures shouldn't be contemplated.

Recap

The story so far: Starting from a large text corpus, the mutual information (MI) of word-pairs are counted. This MI is used to perform a maximum spanning-tree (MST) parse (of a different subset of) the corpus. From each parse, a pseudo-disjunct is extracted for each word. The pseudo-disjunct is like a real LG disjunct, except that each connector in the disjunct is the word at the far end of the link.

So, for example, in in idealized world, the MST parse of the sentence "Ben ate pizza" would produce the parse Ben <–> ate <–> pizza and from this, we can extract the pseudo-disjunct (Ben- pizza+) on the word "ate". Similarly, the sentence "Ben puked pizza" should produce the disjunct (Ben- pizza+) on the word "puked". Since

these two disjuncts are the same, we can conclude that the two words "ate" and "puked" are very similar to each other. Considering all of the other disjuncts that arise in this example, we can conclude that these are the only two words that are similar.

Any given word will have many pseudo-disjuncts attached to it. Each disjunct has a count of the number of times it has been observed. Thus, this set of disjuncts can be imagined to be a vector in a high-dimensional vector space, which each disjunct being a single basis element. The similarity of two words can be taken to be the cosine-similarity between the disjunct-vectors.

Equivalently, the set of disjuncts can be thought of as a weighted set: each disjunct has a weight, corresponding to the number of times it has been observed. A weighted set is more or less the same thing as a vector, and these two are treated as the same, in what follows. Note that the disjunct vectors are sparse: for any given word, almost all coefficients will have a count of zero. For example, the dataset that will be examined next has over a quarter of a million different pseudo-disjuncts in it; most words have fewer than a hundred disjuncts on them.

Some terminology and notation are introduced next, followed by a characterization of the dataset. This is followed by a statistical analysis of the word-disjunct pairs, and is followed by an analysis of the resulting word-similarity.

Terminology

It is useful to introduce some notation for counting words, disjuncts, and connectors. Let $N(w)$ be the number of times that the word w has been observed, in the dataset. Let $N(w,d)$ be the number of times that the disjunct d has been observed on word w. The pair (w,d) is referred to as a "connector set" or "cset" in the text below. Thus, for a word w, there is a set $(w,*) = \{(w,d)|N(w,d) > 0\}$ of associated csets, called the "support" of the word. The size of this set can be written using the standard notation for set-sizes as $|(w,*)|$. Similarly, a disjunct d, is supported by the set $(*,d) = \{(w,d)|N(w,d) > 0\}$ of associated csets.

The primary contents of the database are the counts $N(w,d)$ and everything else of interest in this section can be obtained from this. Note that $N(w,d)$ can be understood as a matrix, where the disjuncts identify columns, and the words identify rows. In general, this is a very sparse matrix: the number of non-zero entries $|(*,*)|$ is far less than the number of rows times the number of columns.

Every time a word is observed in an MST parse, a disjunct is extracted for it; thus, word observations and disjunct observations are on one-to-one correspondence. In notation:

$$\sum_d N(w,d) = N(w,*) = N(w)$$

Similarly, the total number of times that a disjunct was observed is just

$$N(*,d) = \sum_w N(w,d)$$

Frequencies can be obtained by dividing by the total number of observations, so that $p(w,d) = N(w,d)/N(*)$ and $p(w) = N(w)/N(*)$ with $N(*) = \sum_w N(w)$ the total number of observations of words.

A single disjunct is always composed of a fixed number of connectors, independently of any observations; let $C(d,c)$ be the number of times that connector c appears in disjunct d. Note that $C(d,c)$ is almost always either zero or one; however, a connector can appear more than once in a disjunct, so this count can rise to 2 or 3 or very rarely higher. The wild-card sum $C(d,*) = \sum_c C(d,c)$ is the total number of connectors in the disjunct; it is the vertex degree of all edges connecting to that disjunct. It is also useful to define $C(d,+)$ and $C(d,-)$ as the total number of right-linking and left-linking connectors.

Dataset characterization

This section was originally written in May 2017 and used to report data for a different dataset. However, a serious flaw was found in the code: all MI values had a minus sign in them, and thus all computed disjuncts were maximally-bad. The statistical analysis of this maximally-bad data wasn't horrible: it did behave reasonably. However, in the end, its still bad data, and so all charts and graphs are being revised with a new dataset. The old dataset was also terribly tiny. The new dataset is much larger. You can get the old version by digging in git, and pulling up commit 27a66643a52c0985adc5b38caf94fc25f5e2e684 (or maybe a bit earlier, circa late June 2017 as that is when the bug was spotted.).

The following charts and analyses are derived from a single dataset, called 'en_pairs_rfive_mtwo'. It contains data for word-pair statistics derived from parsing text from tranche-1,2,3,4,5 (See the download.sh scripts), followed by MST parsing of tranche-1 and 2. The word-pair statistics were obtained by applying random-tree parsing to entire sentences. The dataset is summarized in section and repeated here. The column labels are explained there.

Size	Pairs	Obs'ns	Obs/pr	Entropy	MI	Dataset
839K x 851K	30.1M	1.35G	44.9	18.54	1.84	en_pairs_rfive

The support and count for the pairs are given below.

Size		Support		Count		Length		Dataset
L	R	L	R	L	R	L	R	Name
839K	851K	80.6K	80.6K	249	230	28.2	24.5	en_pairs_rfive

The disjunct stats are these:

Size	Csets	Obs'ns	Ob/cs	Entropy	H_{left}	H_{right}	MI	Notes
137K x 6.24M	8.63M	18.5M	2.14	20.96	19.14	9.71	7.90	en_pairs_rfive_mtwo

The dataset contains 851964 words. Of these, 137078 words that have disjuncts attached to them. These words have been observed a total of 18489594 times, for an average of 18489594/137078 = 21.70 observations per word. This dataset contains 6239997 different, unique disjuncts, for an average of 18489594 / 6239997 = 2.963 observations per disjunct.

The period appears 849354.0 times, suggesting that this many sentences were observed. Each sentence thus has an average of 18489594 / 849354 = 21.77 words per sentence.

The dataset contains 6239997 unique connector-sets, for an average of 18489594 / 6239997 = 2.96 observations per cset. This last number that makes this dataset feel thin and sparse. Its not clear how accurate that perception is: an earlier dataset was about one-tenth to one-twentieth the size, in the number of words, disjuncts and observations, yet it had a ratio of 1.5 observations per cset. That is, making more than ten times the number of observations only doubled the observations per cset.

The dataset is sparse in a completely different sense: viewing $N(w,d)$ as a matrix whose size is 851964×6239997, but only a very small number of these is non-zero: this is $8629163/(851964 \times 6239997) = 1.623 \times 10^{-6}$. The sparsity of this matrix can be defined as $-\log_2$ of this number, which is 16.60. The sparsity appears to increase with the number of observations: the previous, ten-times-smaller dataset had a sparsity of 15.

The total word-entropy for the dataset is defined as

$$H_{word} = -\sum_w p(w) \log_2 p(w)$$

and was measured to be $H_{word} = 9.71$ bits.[1] The total connector-set entropy is much larger. It is defined as

$$H_{cset} = -\sum_{w,d} p(w,d) \log_2 p(w,d)$$

and is measured to be $H_{cset} = 20.96$ bits. The disjunct entropy is dual to the word entropy:

$$H_{disjunct} = -\sum_d p(*,d) \log_2 p(*,d)$$

and is measured to be $H_{disjunct} = 19.14$ bits. The total mutual information between the words and disjuncts is then

$$MI_{cset} = \sum_{w,d} p(w,d) \log_2 \frac{p(w,d)}{p(*,d)p(w,*)} = H_{word} + H_{disjunct} - H_{cset}$$

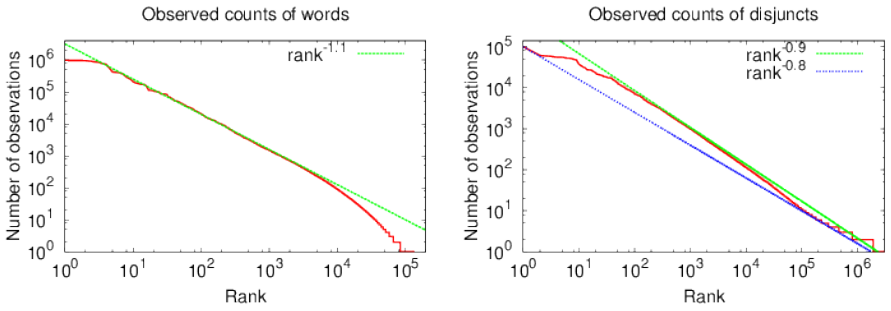
and is measured to be $MI_{cset} = 7.897$ bits.

Connector-set distribution

Some connector-sets will be observed far more often than others. Likewise for the two sides of the connector-set: some words will have far more observations, and some disjuncts will be seen more often.

[1] This and the following entropies were measured with the word-entropy-bits, disjunct-entropy-bits, etc. functions in disjunct-stats.scm Alternately, the print-matrix-summary-report now reports this.

Two graphs, dual to one-another. The one on the left shows $N(w,*)$, ranked by count. The one on the right shows $N(*,d)$, also ranked.[2]. The first follows the canonical Zipf distribution. The green line is an eyeballed, approximate fit, of exponent -1.1. The second has an exponent of about -0.85.



The first ten words in the word ranking are: "LEFT-WALL" "," "." "the" "and" "to" "of" "a" """ "in". This is the ranking of how often these words appear, overall, in the MST-parsed corpus. The number of connections to LEFT-WALL should be equal to the number of sentences in the corpus, as the parser is set up to make one LEFT-WALL connection to a sentence. Most sentences will end in a period; some with question marks of other punctuation. Commas and the word "the" can appear more than once in a sentence. The frequent occurrence of the straight double-quote mark is due to the fact that the corpus is heavily weighted with dialog: i.e. with fictional novels, where the characters are speaking a lot.

This list is repeated in the table below. The support is $|(w,*)|$, that is, the number of different kinds of disjuncts observed for that word. The count is $N(w,*)$, that is, the total number of times those disjuncts have been observed for that word. The frequency is just the count divided by 18489594. The length is $\text{len}(w,*) = \sqrt{\sum_d N^2(w,d)}$, that is, the root of the sum of the squares of the observations.[3]

word	support	count	frequency	$-\log_2$frequency	length
LEFT-WALL	64215	972963	0.05262	4.248	122353.9
,	243987	957593	0.05179	4.271	25475.4
.	106195	849354	0.04594	4.444	55168.0
the	215324	727027	0.03932	4.669	9264.2
and	126861	420942	0.02277	5.457	28694.7
to	117110	401967	0.02174	5.523	11480.0
of	108951	371211	0.02008	5.638	11047.5
a	102720	289631	0.01566	5.996	6855.1
"	51289	256785	0.01389	6.170	21388.8
in	64011	208745	0.01129	6.469	14758.1

[2]Obtained by running (print-ts-rank sorted-word-obs outport) from the disjunct-stats.scm file, on the en_pairs_rfive_mtwo database. The second one prints sorted-dj-obs. The graphs generated with ranked.gplot

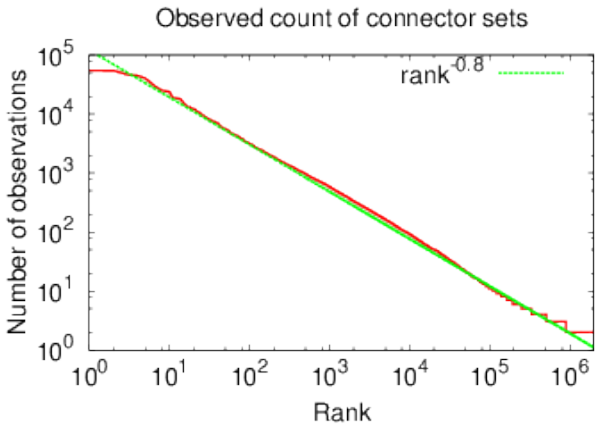
[3]A printing utility for these three is 'show-counts' in the 'disjunct-stats.scm' file.

The main point of this table is to demonstrate the log-likelihood column. At this point, these numbers won't seem to have much meaning; however, they provide an overall scale that will be seen, repeatedly, in the analysis below. The range of magnitudes – 4 to 7 – is no accident, and similar ranges will be seen later.

The first ten pseudo-disjuncts in the disjunct-ranking are ""+" ",-" "the+" "He+" "The+" "the-" "LEFT-WALL-" "I+" "“+" "to-". The meaning of the plus and minus signs was explained above; but to recap: the disjunct "xxx+" means that there are many words that expect to be followed by the word "xxx" (on the right). The disjunct "the-" means that there are many words that want to link to the word "the" on the left. This is grammatically correct: "the" is a determiner, and it is always the dependent of some noun. The disjunct "The+" is at first appears to be grammatical garbage/nonsense: it states that there are many words that want to link to the word "The" on the right. Naively, this is never correct for English; determiners always precede the noun that they modify. The capitalization gives away what is really happening: the word "The" is a sentence opener, and it is being linked by the LEFT-WALL, indicating the start of the sentence; *ergo*, it is lining backwards. Similar remarks apply to ""+" "the+" "He+" "I+" "“+": Clearly, the capitalized "He" is a sentence opener, and "I" is plausibly so. The two different styles of quotation marks (symmetrically vertical and right-leaning) open up dialog in fictional novels, which make up a large portion of the corpus.

The above avoids the question of whether its is syntactically correct to link the LEFT-WALL to "The". This is determined not by raw frequency counts, but by mutual information. This is explored later. At this point we can only say that such a linking is frequent, and cannot judge whether it is correct.

The ranking of connector sets is shown below. It's a graph of the ranked counts $N(w,d)$. Recall that we define a "connector set" as the pairing of a word, and one particular disjunct that is associated with that word.[4]



The top-ten connector-sets are "LEFT-WALL: He+;" "LEFT-WALL: The+;" "LEFT-WALL: “+;" "LEFT-WALL: "+;" ".: "+;" "LEFT-WALL: I+;" "and: ,-;" "LEFT-WALL: It+;" "LEFT-WALL: She+;" ".: ”+;"

[4]Graph of sorted-cset-obs, *op cit*.

These are hard to read, so, decoded: the four are connectors from the left side of the sentence to the words “He” and “The”, and two different styles of double-quote marks: a right-leaning double-quote, and a vertical double-quote. This was commented on before: the corpus has many novels, and so many sentences will begin with quotes. Next comes a period which links to a double-quote on it’s right. Last comes period followed by a leaning double-quote. Clearly, this is expected in the corpus. Also visible are the sentence openers “I” “It” “She”. In that list is the word “and”, which connects to the *left* to a comma. Not a surprise.

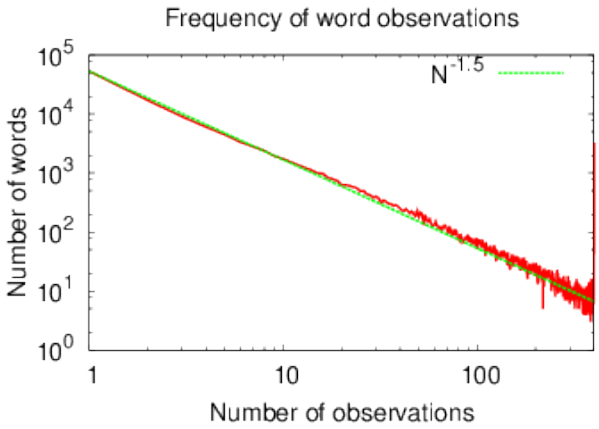
Not visible in the top-ten, but can be seen in the top-fifty are mirror-images, for example: ",: and+;" in 14th place, which states that the comma expects to be followed by the “and”. The counts differ: the sixth-place "and: ,-;" had 32755 counts, while 14th place had 17745. Presumably this is due the comma having a different or more complex linkage about half the time. Other items in the top-50 that are not sentence openers include "”: ?-;" ",: but+;" ".: him-; "in: the+;" ".: ’+;" "”: .-;" "of: the+;". The first disjunct with more than one connector in is is "It: LEFT-WALL- was+;" which states that “It” wants to be a sentence opener, but wants to be followed by the word “was” on the right. Not surprising. Proceeding down the list, the disjuncts continue in this manner. This does not necessarily mean that they are “high quality”, only that they were frequently observed in MST parses. It might be the case that the quality is given by the MI between the word and its disjunct; this is explored later.

Word distribution

It is also interesting to turn the word distribution graph “on it’s side”. This is meant to be a simple exercise, so as to place some later graphs into context. Despite the simplicity, the analysis turns out to be somewhat surprising, and somewhat subtle. In particular, the last graph elicits some features in the dataset that are not otherwise easily visible.

In this dataset, there were 53076 words observed exactly once (out of a total of 18.5M observations of 137K words). This is quite something: of all the words observed, almost half were seen only once. More than half were seen twice, or less. These are presumably rare typos, foreign words, IPA pronunciation guides: any word that appears only once must be unusual; and yet, there are a lot of them! There are 17120 words that appear twice, 9081 that appear exactly 3 times, *etc.* These counts are graphed below.[5]

[5] Graph of binned-word-counts.dat, generated in disjunct-stats.scm



This graph indicates that most (almost all) words were observed less than 100 times. In this dataset, there were only 9425 words that were observed 100 or more times, 5683 words that were observed 200 or more times, and 3263 words that were observed 400 or more times. Percentage-wise: about 6.8% of the words were observed 100 times or more. This should be enough to give confidence in the syntactic usage of the commonly-used English words; but most of the rest of this dataset includes oddities of various sorts, including place names and given names. The challenge will be to see if these can be grouped into grammatical categories.

Writing N for the number of times that some word was observed, it appears that there are approximately $53076 \times N^{-3/2}$ words observed that many times. In formulas, the size of the set of words $\{w|N(w) = N\}$is given by

$$|\{w|N(w) = N\}| \sim N^{-3/2} \tag{1}$$

where $\{w|\text{cond}\}$ is a set of words (subject to the condition cond) and $|\{w|\text{cond}\}|$ denotes the size of that set of words.

There is something interesting about this chart: it is more stable under varying dataset sizes than the Zipfian distribution. The slope of the Zipfian distribution changes, as datasets grow larger, typically trying to approach a slope of 1.0 very slowly. By contrast, the above $N^{-3/2}$ behavior seems to provide a much better description, even as the size of the dataset varies. I cannot demonstrate that assertion here, but have noticed it to be true when looking at other datasets.

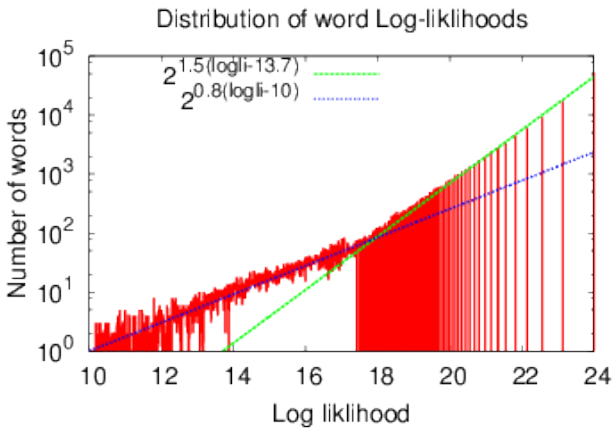
The next graph belabors the point, and yet it's important.[6] It shows nothing new, but it does show it in a format that will be recur frequently, later. Thus, its worth understanding now. This graph shows exactly the same data as the previous graph: it *is* the same graph, except that the x-axis is now labeled differently, and some of the counts have been binned together. So first: note that $-\log_2(1/18489594) = 24.140$ and so this is the location of the first spike on the far-right. Next, $-\log_2(2/18489594) = 23.140$ and $-\log_2(3/18489594) = 22.555$ are the locations of the second and third spikes: these correspond to words that have been observed 1,2 and 3 times. Words that have

[6] Generated from binned-word-logli.dat

been observed exactly N times will have a log-likelihood of $-\log_2(N/18489594) = 24.140 - \log_2 N$. The formula 1, which was based on the graph above, can be rewritten as

$$|\{w|N(w) = N\}| \sim 2^{-3/2 \times \log_2 N}$$

which effectively predicts the height and location of the spikes. This is clearly demonstrated by the straight green line.



But then something else happens. As long as the bins are narrow, so that they are either full, or empty, then the nice power law holds. Once the bins become too wide to just hold single, discrete counts, but instead lump together different logli's, the apparent distribution changes. It is worthwhile to understand this phenomenon.

This graph was generated by bin-counting. The x-axis was divided into 1200 equal-sized bins, and whenever the log-likelihood of a word landed within a particular bin, the count was accumulated into that bin. On the right side of the graph, many (most) bins are empty, because they do not correspond to logarithms of integers; this results in the spikes. The width of each bin is (24-9)/1200, and so when

$$\log_2 N - \log_2(N-1) \approx \log_2\left(1 + \frac{1}{N}\right) \approx \frac{1}{N \log 2} < \frac{24-9}{1200} = \frac{15}{1200}$$

then multiple counts will be shoved into one bin. For this chart, this happens when $N \approx 115$ so there are about 115 distinct spikes, and then they merge when the logli is $24.140 - \log_2 115 \approx 17.3$ which is the spot where the above graph bends from the green line to the blue line. From this point on, when multiple counts are being jammed into one bin, we expect the distribution measure to be given by the Jacobian determinant[7] of the point measure. We can compute this explicitly. Changing notation slightly, write

$$C(N) = KN^{-3/2}$$

[7] https://en.wikipedia.org/wiki/Jacobian_matrix_and_determinant

for the number of words that were observed N times. This is the same formula as before. For this dataset, $K = 53076$. The bincount at logli$= x$ is

$$\text{bincnt}(x) = \sum_{x \le \log_2(T/N) \le x+\varepsilon} C(N)\Delta N$$

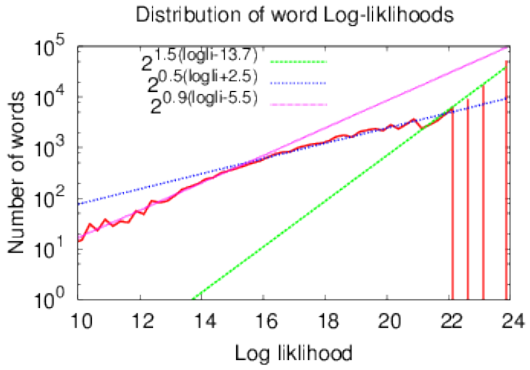
For this dataset, $T = 18489594$. The binsize used in the above graph was $\varepsilon = 15/1200 = 1/80$. In the above formula, $\Delta N = 1$ is a notational trick to allow us to convert the sum into an integral when N gets large. We replace $\sum$ by $\int$ and replace ΔN by dN and write

$$\text{bincnt}(x) \approx \int_x^{x+\varepsilon} C(N)\frac{dN}{dy}dy$$

Here, there is a change-of-variable to $y = \log_2(T/N)$ or equivalently $N = T2^{-y}$. The Jacobian determinant is then $|dN/dy| = N\log 2$ and so

$$\text{bincnt}(x) \approx \frac{K\log 2}{\varepsilon\sqrt{T}}2^{x/2}$$

Comparing this to the graph above, with logli$= x$, we expect the bincounted region to have a slope of 0.5, and yet, the eyeballed fit above clearly shows 0.8. What's going on? WTF? Blame the data. Try again. The graph below shows exactly the same data, but this time there are only 60 bins grand-total, and so only the first three spikes show. After that, the spikes merge together into bins. The green line is drawn exactly with the same slope and offset as before: that's because the first three spikes are in exactly the same locations as before, and have the same height. The blue line shows an eyeballed fit to the merged counts, and initially, it really does have a slope of 0.5, which is exactly what the Jacobian determinant was telling is it should be. Yayy! Declare victory and go home!



The purple line is an eyeballed fit to what the data is doing, when the number of observations really does become large. The knee in the graph is at about logli$= 15.5 = 24.140 - \log_2 N$ or, equivalently at $N = 400$. Thus, we have to revise the apparent distribution. It is, for this dataset:

$$|\{w|N(w) = N\}| \sim \begin{cases} N^{-3/2} & \text{for } N < 400 \\ N^{-1.8} & \text{for } N > 400 \end{cases}$$

It was noted before that there are 3263 words in the dataset that were observed 400 or more times.

What does this mean? What is this saying? Its not entirely clear. It seems to suggest that there are about 3.3K words that are used preferentially more often than the rest. That is, they are used more often not only in absolute terms, but also in relative terms: the form a core of the vocabulary, enjoying a popularity exceeding the trend line for less-frequently used words.

Its not clear if this is a generic feature of the English language, or if this is peculiar to the particular corpus. Let's review the corpus again. The corpus comprises assorted late-19th and early 20th century texts from Project Gutenberg, a dozen sci-fi/fantasy novels, and a sampling of fan-fiction. These texts will contain stray markup, including tables of contents, chapter headings, indexes, figure captions and itemized lists. Quite often, ASCII artwork is used to delimit chapters or sections. Chapter headings are often written in all-upper-case. There are stray quotations in Latin, snippets of Latin prose and poetry. Travelogues will include miscellaneous foreign sayings and unusual place-names. All of this stuff adds up: it will be observed only once, twice, maybe a a few dozen times. It may as well be random text, from the point of view of the word-pair MI statistics, and from the point of view of the MST parser. There is no easy way to remove this "garbage" in any *a priori* fashion. It is there, and it is unavoidable. It is indistinguishable from random sentences. However, it seems that the fact that some significant portion is "ungrammatical" should not affect word-count statistics, unless it just so happens that there is a core of 3.3K vocabulary words, followed by 130K words that are given names, arcane terms, and other "junk". This does not seem plausible.

Thus, it is unclear on what the meaning of this knee in the graph really is, and how it should be explained. Note that this knee is NOT visible in the Zipfian distribution – nothing happens at $N = 400$ - it is smooth as silk. It maybe could have been visible in the on page 9 graph, except that the far edge of that graph ends at exactly $N = 400$, and does not continue past there! This seems to be a fairly subtle effect.

Ranked average observations per disjunct

A more interesting distribution arises by looking at the average number of observations per disjunct (per word). That is, a single word may have hundreds of disjuncts, observed thousands of times; what is the average number of times that a disjunct is observed? By "average", it is explicitly meant $N(w,*)/|(w,*)|$, the number of observations divided by the support for those observations.

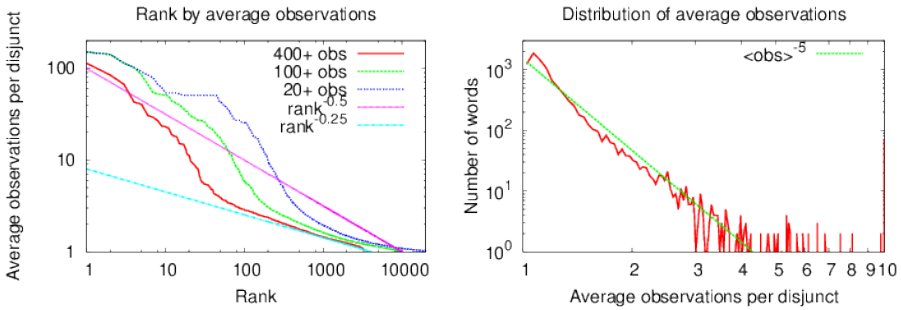
This number gives a hint of how "narrow" the grammatical usage of a word is. If the average is high, it suggests that the word just does not have very many disjuncts on it; the few that it does have are observed a lot. Recall that these disjuncts (pseudo-disjuncts) connect to individual words, and not to word-classes. Thus, if a disjunct is seen a lot, it probably connects to another word, forming a high-MI pair. This can be explicitly seen in the example further below.

A graph of the ranked average number of observations, per disjunct, per word, is shown on the left, below.[8] The ranking is distinctly not Zipfian; this is confirmed by

[8] Computed with the sorted-avg list in disjunct-stats.scm

slicing the data three ways: excluding words with less than 400 observations (leaving 3263 words), excluding words with less than 100 observations (leaving 9425 words), and excluding words with less than 20 observations (leaving 25505 words out of 137K words).

The graph on the right expresses an alternate view of the same idea: it shows a bin-count of all of the words. Reaffirming the graph on the left, it indicates that almost all words have an average disjunct observation count of less than four. It also conveys the sense that when the average disjunct count is greater than about four, that this is unusual, and perhaps meaningful in some way.



The first ten on the ranked list are "*" "Literary" "Archive" "Gutenberg" "Notes" "...." "|" "Foundation" "Project" "Summary". This suggests that these all come from exactly the same parse of a small group of sentences having a very regular, formulaic structure, occurring repeatedly in multiple texts. One of those sentences is easily found; it begins as: "The Project Gutenberg Literary Archive Foundation has been created..."

Closer examination indicates that more or less all words having an average of more than three observations per disjunct are associated with the Project Gutenberg legal boilerplate. The first 80 entries in the 400+ list have an average observation count of above three, and they are all boilerplate words: "fee" "copies" "trademark" "agreement" "electronic" "copyright" "donations", and so on. This suggests that pretty much all of the "bump" on the above-left graph is entirely due to license boilerplate!

Its entertaining to look at some of these close-up.[9] The word "Prince" shows up 99th on the list, with an average of 2.885 observations per disjunct. It has a total of 773 different disjuncts on it. The top six disjuncts are just single links, shown in the table below. Clearly all are princes. This leaves 767 other disjuncts with far fewer counts.

disjunct	number of observations
Andrew+	626
Vasili+	149
Andrew's+	46
Edouard+	46
Hans+	33
Bagration+	32

[9]View disjuncts by saying (filter (lambda (cset) (< 10 (get-count cset))) (cog-incoming-by-type (Word "foo") 'Section)) where 10 is the minimum number of counts.

The word “think” appears as number 140 on the list, with an average of 2.6104 observations per disjunct. It has 5462 disjuncts in total; only six are observed more than 200 times. The last in the list below is the first clear appearance of a transitive verb.

disjunct	number of observations
I-	1321
you-	308
don’t-	265
to-	244
do- you-	211
I- it+	202

The word “long” appears as 195 on the list, with an average of 2.3825 observations per disjunct. It has 5412 different disjuncts on it; only five are seen more than 200 times. These are:

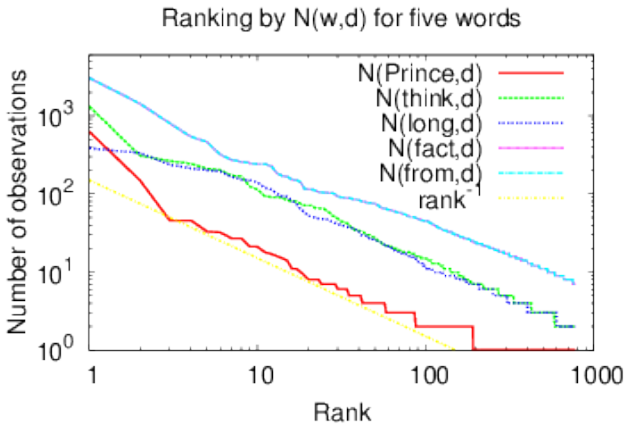
disjunct	number of observations
a-	385
as+	335
how-	236
a- time+	212
enough+	201

The skewness appears to be very sharp. This suggests that we should not waste time looking at mean-square variations in the average, although we’ll do this anyway. But first, its worth graphing the skewness directly. Again, this is done on a log-log graph, in a Zipfian way.

Disjunct count distribution

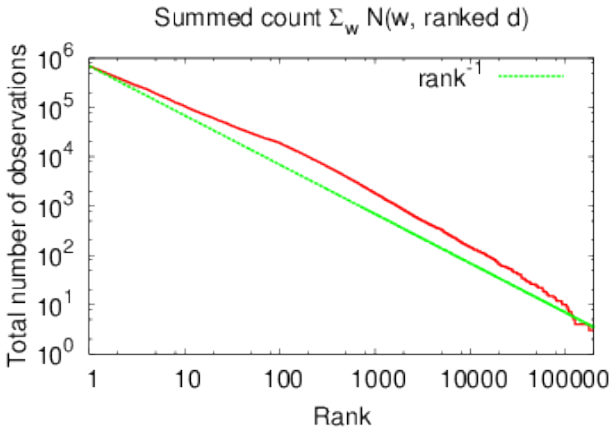
The graph below shows the distribution of the disjunct observations on the five words “Prince”, “think”, “long”, “fact” and “from”. Indeed, it looks Zipfian; since we know that all but the first three or four disjuncts are noise, this graph illustrates “pink noise” or “1/f noise”.[10]

[10]Computed with the dj-prince and dj-think etc. arrays in disjunct-stats.scm



Can one get a smoother distribution by summing together these two graphs? Sure... and one can sum together not just these two words, but all words (that have been observed at least 100 times).

That graph is shown below.[11]



Its kind of a strange graph. Yes, the x-axis of this graph does imply that there are thousands of words with more than a thousand disjuncts on them, and hundreds that have more than ten-thousand (unique, different) disjuncts on them! Exactly what does this mean? This is covered in the next section.

Disjunct Support Distribution

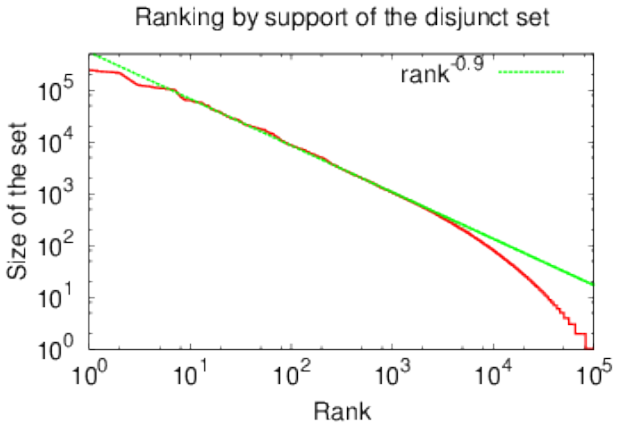
Is it possible that some words have a large number of disjuncts on them?

Yes, it is. For example, the comma was observed to have 243987 unique, different disjuncts associated with it. The word "the" has 215324 unique, different disjuncts,

[11] Computed with the accum-dj-all function in disjunct-stats.scm

the word “and” has 126861. Rounding out this list are "to" "of" "." "a" "was" "LEFT-WALL". Its not clear what fraction of these disjuncts are grammatically valid, and what fraction are junk.

The graph below shows the distribution of the size of the support: the ranking of $|(w,*)|$. Again, the graph appears to be approximately Zipfian.[12] The eye-balled fit has a slope of 0.9, but, from the eyeball-perspective, this is not all that different from a slope of 1.0.



Terminology: the “support” of a vector is the number of basis elements that have a non-zero coefficient. This is the set $(w,*)$ defined earlier. Equivalently, this is the size of the set of disjuncts associated with a word, when counted *without* multiplicity.

Ranked Euclidean length (RMS Size)

A different distribution arises by looking at the ranked RMS sizes of the disjunct sets[13]. Here, the RMS size[14] is computed by taking the root-mean-square of the counts on each disjunct in the set, that is, by computing $\sqrt{\sum_d N(w,d)^2}$ for each word w and then ranking. Interpreting d as a basis element of a vector space, this can be recognized as the Euclidean length of the count-vector.

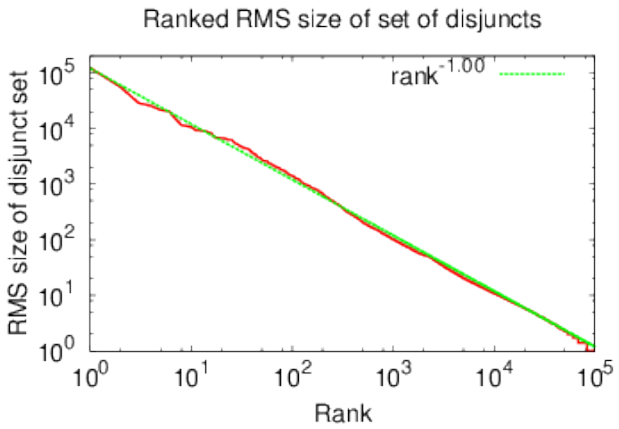
The RMS size of the set is thus larger not only when more disjuncts have been observed, but also when most of the observations are made of only a small handful of disjuncts. That is, the RMS size should be relatively larger, if the word is less grammatically flexible. So for example, prepositions tend to be very flexible; adjectives, not so much. Thus, we expect adjectives to appear higher-up on this ranking list, than

[12] Generated by sorted-support in disjunct-stats.scm

[13] Obtained by running (print-ts-rank sorted-lengths outport) from the disjunct-stats.scm file

[14] The word “length” can be used to describe the root-mean-square size of the set of disjuncts associated with a word. That is, each element of the set is a disjunct, and that disjunct has a count, the number of times it has been observed. The root-mean-square of these counts can be taken as the set-size. But this set can also be interpreted as a vector, and so the RMS size is the same thing as the Euclidean length of the vector. Thus, the word “length” is sometimes used for the RMS size; they’re the same thing.

the observation-based list. And this might be true, relatively, but certainly not true absolutely.



The first dozen words of the RMS-size-list are: "LEFT-WALL" "." "and" "," """ "”" "in" "to" "of" "“" "as" "the". Not that interesting: these are all words that were observed a lot in the text. The RMS size is dominated by the total number of observations of a word in text. In and of itself, its insufficient to indicate how “concentrated” the disjuncts are, how grammatically narrow a word is. For this, some other quantity is needed.

The slope appears to be exactly -1.0, continuing the scale-free trend.

Mean-square to size ratio

More interesting is the ratio of mean-square size to the total size. In formulas, by ranking according to

$$\frac{\sqrt{\sum_d N^2(w,d)}}{N(w,*)} = \frac{\sqrt{\sum_d p^2(w,d)}}{p(w,*)}$$

This seems like the interesting ratio, because the Zipf exponent of -0.65 would be doubled, when working with mean-square sizes, thus making the two rankings comparable.

Words high in this score will be words that have relatively few disjuncts on them, or at least, few that matter much, that rise above the level of noise. The first ten words on this list, excluding punctuation, and excluding all words with fewer than 100 observations: "It" "and" "but" "in" "Two" "as" "Notes" "not" "There" "Summary". A review of the input corpus shows that the word "Summary" appears in only one input text: Charles Darwin’s “On the Origin of Species”, and then only to indicate an actual summary! The word “Notes” appears in less than a dozen input texts, with almost every usage being formulaic and rigid. The grammatical usage of these two words in the input corpus is fairly constrained, and thus it is no surprise that these have relatively narrower disjunct distributions on them.

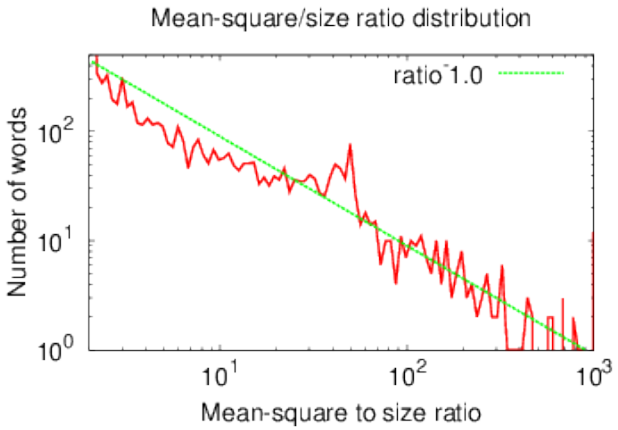
Excluding the capitalized words, and punctuation, in this list what remains is quite surprising. It is shown in the table below.[15]

rank	score	word
7	1956	and
10	1147	but
12	1043	in
14	814	as
16	776	not
21	590	been
22	568	be
26	453	one
27	440	at
30	386	own

rank	score	word
34	329	of
35	328	to
36	324	with
37	319	other
38	315	it
42	284	when
43	275	for
45	271	he
46	265	have
49	248	that

It is surprising because, grammatically, we expect most of these words to have a large number of varied disjuncts attached to them. We expect them to be diffuse, not sharp: we expect that these would have a large number of observations smeared over a large variety of disjuncts, instead of having their weight concentrated in only a handful of disjuncts, the way that "Prince" was, above. So what is going on, here?

The distribution, shown below, is a power-law.



Are there other interesting measures? One could contemplate the ratio of the mean-square size to the support $\sum_d N(w,d)^2/|(w,*)|$. Another possibility would be this, minus the average-squared, which would give the second moment, aka, the mean-square deviation from the average, specifically

$$\frac{\sum_d N(w,d)^2}{|(w,*)|} - \left[\frac{N(w,*)}{|(w,*)|}\right]^2 = \frac{1}{|(w,*)|}\sum_d \left[N(w,d) - \frac{N(w,*)}{|(w,*)|}\right]^2$$

Neither of these variations seem promising; they seem to offer up more of the same, at least on this dataset. A larger and more refined dataset might reveal otherwise.

[15] Extracted from ranked-sqlen-norm.dat

Mutual information

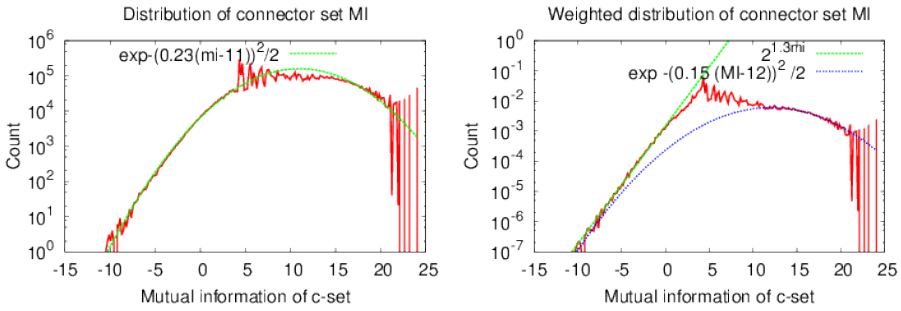
The concept of the "fractional mutual information" for a pair is interesting to explore. Define this as

$$MI_{pair}(w,d) = -\log_2 \frac{p(w,d)}{p(w,*)p(*,d)} \tag{2}$$

This is "fractional" in the sense that the total MI for the set of all pairs can be written as

$$MI_{cset} = \sum_{w,d} p(w,d) MI_{pair}(w,d)$$

Fractional MI is interesting because it usually has a reasonably nice distribution. For this particular dataset, it ranges from about -11 to +24. The distribution is shown in the graphs below. [16]



These graphs are generated by computing the value for $MI_{pair}(w,d)$ for each of the 8629163 (w,d) pairs (aka 'connector sets'), and approximating it's distribution by bin counting. In each graph above, there are 200 bins, each of width of about 35/200, and each pair is assigned to one of the bins, according to it's MI value. The graph on the left then shows how many pairs there are in each bin. The graph on the left is similar, but not the same: it sums the frequencies for all the pairs in each bin. In formulas: the graph on the left shows the value of

$$\text{sizeof } \left\{(w,d) | MI_{pair}(w,d) \text{ is in bin}\right\}$$

while the graph on the right shows

$$\sum_{MI_{pair}(w,d) \text{ is in bin}} p(w,d)$$

where 'x is in bin' simply means $lo \le x < hi$ with the bin being the interval $[lo, hi)$.

Both of these graphs show "combs" in the right side. These combs are exactly the same combs as noted in the last figure in section Word distribution on page 10. The combs are due to the large number of words that have been observed only a small handful of times. In essence, the combs attest that the bulk of the high-MI pairs have been observed more than just a few times; *i.e.* the high MI values are meaningful.

[16] These are graphed by binned-cset-mi and weighted-cset-mi in the disjunct-stats.scm file.

Both graphs show an eyeballed fit in green. The left graph shows that the distribution can be approximated by a Gaussian (visually a parabola, due to the logarithmic scale), given by $\exp -(0.23(MI_{pair} - 11))^2/2$. The shape of the graph on the right is harder to pin down. It has hints of parabolic behavior, yet the left edge appears more straight than curved, that is, to be exponential, given by $2^{1.3MI_{pair}} = \exp 0.9MI_{pair}$. The eyeballed blue parabola is given by $\exp -(0.15(MI_{pair} - 12))^2/2$, it's clearly missing a large excess in the middle of the graph.

Marginal Mutual Information of Words

The marginal mutual information of a single word can be defined by summing the (fractional) mutual information between a word, and all of it's disjuncts:

$$MI_{word}(w) = \frac{1}{p(w)} \sum_d p(w,d) MI_{pair}(w,d)$$

This is also written in the "fractional" style, so that, again, the total MI of the entire dataset can be written as

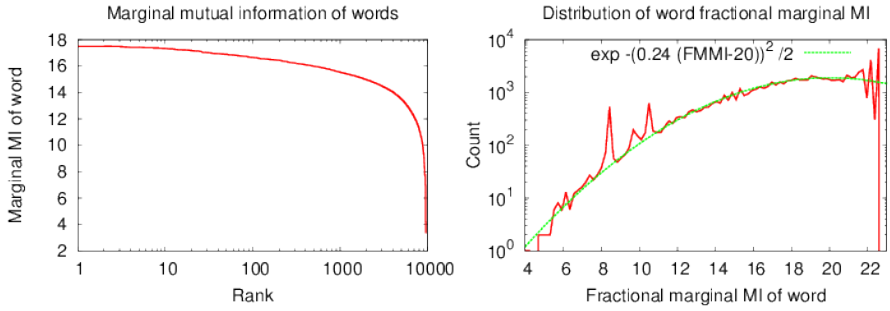
$$MI_{cset} = \sum_w p(w) MI_{word}(w)$$

That is, $MI_{word}(w)$ is the fractional contribution of the word to the total MI. The fractional marginal MI is very convenient for comparing different words, since it factors out the frequency of how often a word is observed: the MI of two words with two very different frequencies can be directly compared.

As can be seen from the graph below, the fractional marginal MI ranges between +3 and +18 for this dataset. The total MI for the dataset is measured to be 7.8969 bits. The distribution can be visualized in two different ways. The graph on the left, below[17] shows the ranked MI of the 9425 words that have been observed more than 100 times in this dataset. Note that it is a semi-log plot; it is NOT Zipfian. Note that the MI seems to decay logarithmically, for a good long ways, and then drops off a cliff.

The graph on the right shows the distribution, for all 137078 words, bin-counted into 100 bins. The combs on the far right are again the same combs as noted in the last figure in section Word distribution on page 10. The distribution appears to be Gaussian, and of approximately the same width as before, although located at a different center: the green line in the graph is the Gaussian given by $\exp -(0.24(MI_{word} - 20))^2/2$ as compared to 0.23 for the non-marginal MI distribution above. The difference between 0.24 and 0.23 seems to be significant: changing one to the other seems to give a noticeably poorer fit.

[17] This is graphed by sorted-word-mi-hi-p in disjunct-stats.scm



What are the disjuncts like at either end of this distribution? The first few words in the ranking are: "LICENSE" "FULL" "formats" "BREACH" "AGREE" "WARRANTIES" "WARRANTY" and are clearly parts of the set phrases that make up the Project Gutenberg license agreement. The word “Prince”, examined previously, is 7378th in this rank.

Its entertaining to look at some of these.[18] The word "LICENSE" is surely undersampled, as, in this all-capitals form, it only appears in the Gutenberg boilerplate, and nowhere else. This we cannot expect accurate MST parses, and cannot expect accurate disjuncts for this word. Yet other set phrases quickly top the list. The word “San”, number 33 in the list, is seen only with the disjuncts “Francisco+” and “Antonio+”. The word “Tomb” is 40th in the list and has three disjuncts observed more than twice:

count	disjunct for “Tomb”
19	Great- of+ Nazarick+
15	Underground- of+ Nazarick+
7	The- Great- of+ Nazarick+

It is already clear from this one example that the high marginal-MI words will be those that take part in idioms, “set phrases” or “institutional phrases”, and that the disjunct identifies the words taking part in the setting. The word “prominently” appears 50th in the list, and suggests that it is only used only in a rather rigid and formulaic way:

count	disjunct for “prominently”
51	appear- whenever+
51	without- displaying+

The word “Corps” appears 95th in the list. Note that both the English and French word-ordering appears: “Diplomatic Corps” and “Corps Diplomatique”, as witness on the different sign on the disjuncts:

[18] View disjuncts by saying (filter (lambda (cset) (< 10 (get-count cset))) (cog-incoming-by-type (Word "foo") 'Section)) where 10 is the minimum number of counts.

count	disjunct for “Corps”
8	Supply-
6	Marine-
4	Medical-
3	Diplomatic-
3	Diplomatique+

At the other extreme, the ten words with the lowest MI are: "it" "in" "of" "that" "to" "and" "the" "LEFT-WALL" "," "." ranging from 5.07 for "it" down to 3.33 for the period. These are already familiar from previous rankings: they occur with very high frequency; the disjunct lists on them will be lengthly variable, diffuse.

These samplings of disjuncts are sharply reminiscent of the technique of collocation used in corpus linguistics, but with a big, important difference. There, the linguist examines a window that is some 6-8 words wide, and examines the frequency of all phrases appearing within that window, containing the word-of-interest in the center. Here, we again have a word-of-interest, but this time, instead of seeing phrases, we see the grammatical structure revealed directly, by means of the disjuncts extracted from MST parses. The philosophical basis used to justify corpus linguistics, i.e. that of frequentism, is accepted and applied here as well. In this case, it is used to obtain grammatical structure.

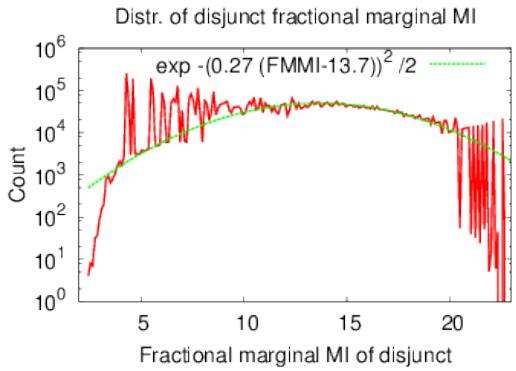
Mutual information of disjuncts

Symmetrically, one also has the mutual information of a disjunct, in comparison to all of the words it connects to:

$$MI_{disjunct}(d) = \frac{1}{p(*,d)} \sum_w p(w,d) MI_{pair}(w,d)$$

Again, this is presented in the “fractional” style, so that the total MI of the entire dataset can be written as before:$MI = \sum_d p(*,d) MI_{disjunct}(d)$The dataset contains 6239997 (6.24 million) unique disjuncts, observed for a total 18489594 (18.5 million) times. The distribution is shown below, with the disjuncts sorted into 200 equal-sized bins.[19] The combs on the far right are again the same combs as noted in the last figure in section Word distribution on page 10. The green line shows an eyeballed fit to a Gaussian. As before, the width appears to be almost 1/4th, but not quite. It is given by $\exp -(0.27(MI_{disjunct} - 13.7))^2/2$

[19] Use binned-dj-mi to get this.



The origin of the noise that seems to build up for MI<10 is unclear.

Fractional Entropy

There is a simpler variant than the mutual information, that is also worth understanding: the fractional contribution to the total entropy. This is given by the sum

$$H_{word}(w) = -\frac{1}{p(w)} \sum_d p(w,d) \log_2 p(w,d)$$

This is written in the "fractional" style, so that the total entropy of the entire dataset can be written as

$$H_{cset} = \sum_w p(w) H_{word}(w)$$

Analogously, one also has the fractional contribution of the disjuncts:

$$H_{disjunct}(d) = -\frac{1}{p(*,d)} \sum_w p(w,d) \log_2 p(w,d)$$

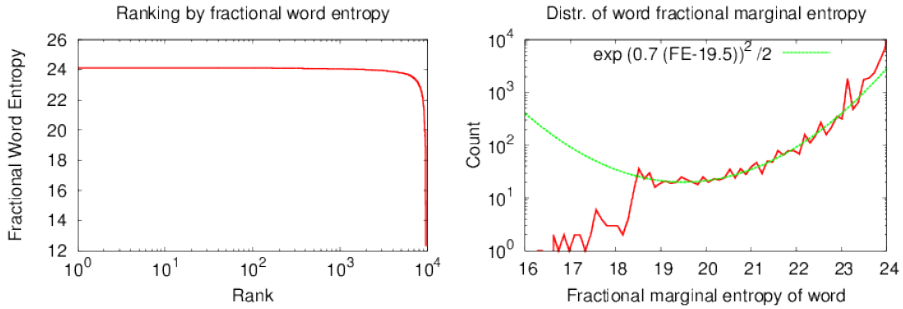
where, again, one has that

$$H_{cset} = \sum_d p(d) H_{disjunct}(d)$$

The ranked fractional entropy is shown in the left graph below.[20] It only shows those words that have been observed 100 times, or more.

It resembles the graph for the ranked fractional marginal MI, above. The graph on the right shows the distribution of the entropy, for all of the words. This affirms (or explains?) the sharp knee in the graph on the left: the knee occurs because almost all words have a large disjunct-entropy. Remarkably, the distribution is anti-Gaussian, in that it appears to diverge, the larger the entropy. In that respect, it cannot even be a proper distribution, as it cannot be normalized to a probability of 1.0 - the distribution increases without bound! Yet, as the graph illustrates, that is what it seems to be. The green curve is the anti-Gaussian, given by $\exp(0.27(FME - 13.7))^2/2$. As before, it has a width of approximately 1/4th.

[20] Generated from sorted-word-ent in disjunct-stats.scm.



The top-ranked words (which have been observed 100 times or more) are "clawing" "manages" "noting" "anyways" "circling" "choke" "neared" "urging" "pursuing" "exited". This is not a list that has been exposed with other statistics. Almost all of these are verbs, a grammatical class that never appeared in any of the previous lists. Why these?

All of these words have an entropy of exactly $24.14021 = \log_2 18489594$. Since there are a total of 18489594 observations of disjuncts in this dataset, the only way in which this entropy is possible is if every disjunct on these words was observed once and only once.

Looking deeper into the disjunct set, the sensation that the words with the highest entropies are almost all verbs continues quite strikingly. A sampling is given in the table below. All of these words have a large support (i.e. have at least 100 observations of disjuncts on each), but each disjunct is observed only once, rarely twice; almost never more than that. For example, “gripped” has only one disjunct that was observed three times, seven that were observed twice, and 320 that were observed only once. A quick examination shows that many, maybe most, are grammatically reasonable, for example, for “gripped”: (He- staff+) is the disjunct observed three times. Four of the seven observed twice are (had- him+) (she- his+ hand+) (He- staff+ tightly+) (his+ hand+) and look to be a part of rather formulaic sentences (possibly from the fan-fiction part of the corpus). The other three disjuncts observed twice are strange nonsense: (hands- Ross- at+ edge+ vanity+) (hand- jaw- other- top+) (LEFT-WALL- jockeyings- by+ them+). These seem to be the result of failed MST parses, depending, presumably, on word-pairs that were witnessed only a few times.

This observation, and the table below, reinforces the need to the urgency of clustering words: by clustering words together into grammatical categories, this should increase the number of observations of category-pairs, improving MST parses, as well as increasing the number of observations of disjuncts, hopefully drowning out the weak and bizarre disjuncts. Given that the high-entropy words seem to be predominantly verbs, this suggests that clustering will be absolutely required to pick out the grammatical form of verbs.

Entropy	rank	Word	$\|(w,*)\|_3$	$\|(w,*)\|_2$	$\|(w,*)\|_1$
24.1402	1	clawing	0	0	102
24.1293	41	licking	0	1	183
24.1241	79	toys	0	1	124
24.1202	164	grabbing	0	3	300
24.0846	567	gripped	1	7	328

The table employs a bit of notation worth reviewing. Recall the definition of the set that supports a word:

$$(w,*) = \{(w,d)|N(w,d) > 0\}$$

The notation $|(w,*)|$ was used to indicate the size of this set. Extend this notation as

$$|(w,*)|_k = \text{sizeof } \{(w,d)|N(w,d) \geq k\}$$

That is, $|(w,*)|_k$ is the size of the support for when the disjuncts on word w have been seen at least k times.

From this table, it is now clear why "large entropy" can be intuitively understood to mean "many possibilities". Each of these words was seen in a very broad setting of possibilities: in a sense, the broadest possible. Each of these words was observed with a vary large set of different disjuncts, and this set was as spread-out as possible: the vast majority of disjuncts were observed exactly once.

More terms

Some curious terms show up in relating the fractional mutual information to the fractional entropy. Expanding out the above summations, one obtains

$$MI_{word}(w) = H_{word}(w) + \log_2 p(w,*) + \frac{1}{p(w)} \sum_d p(w,d) \log_2 p(*,d)$$

The last term is bizarre...

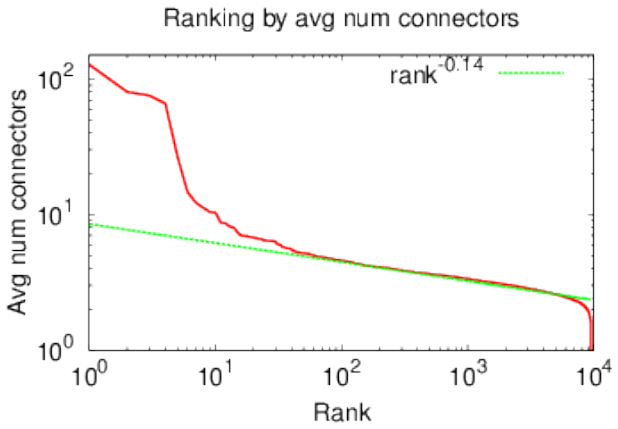
Vertex degrees and hubiness

Vertex degrees can be defined as the average number of connectors per disjunct. In principle, the vertex degree is an excellent indicator of the part of speech. For example, determiners, adjectives and adverbs typically have a degree of one: they have one connector, which is modifying the noun (or verb) that they act on. By contrast, nouns typically have a degree of two: one connector to attach to a verb, another to a modifier, and that's it. Verbs have a degree of three: one connector to a subject, one to a direct object, a third to an indirect object or a modifier. Of course, nouns might have two or more modifiers, or maybe zero modifiers; verbs are also quite variable, but the general concept of vertex degree is appealing. Closely related to this is the idea of "hubiness", which can be defined as the second moment of the degree.

Thus, its worth looking at this. Define the average degree as

$$K(w) = \frac{\sum_{d,c} N(w,d)C(d,c)}{N(w,*)}$$

This is graphed, below, for all words that have at least 100 observations.[21]



This graph is unexpected. Having an average number of connectors that exceed 5 or 6 is intuitively surprising. In proper grammar, it would be hard to reach even this degree without having a transitive or ditranstive verb with several modifiers, and a particle or preposition. The first ten items are out of control. What's happening here? The first ten items in the ranking are: "|" "#" "+" "||" "_" "ASCII" "electronically" "Northup" "disclaimer" "u". The first five are presumably formating markup or possibly decorative markup in the texts. Three of the words appear to be license boilerplate. The name "Northup" appears in only one text in the corpus: an autobiography, "Twelve Years a Slave", by Solomon Northup. Besides the title-page, the word Northup appears repeatedly in the table of contents, which is bound to make for awkward parsing. This presumably "explains" why the average connector count would be 11.2 for this word.

Moving further down this list, many of the words and symbols suggest that they appear in tables or lists embedded in the corpus. The "grammar" of tables and lists is necessarily awkward, and seems unlikely to get much of a meaningful parse from the MST parser. This is further strengthened by an earlier analysis of a Wikipedia-based dataset, re-reported below.

Vertex degree in Wikipedia

A similar analysis, performed on a much smaller dataset consisting entirely of Wikipedia articles found a similar behavior. The first ten items in the ranking were: "-" "de" - "y" ":" "(" ")" "General" "Department" "x" "Act".

[21] Computed with sorted-avg-connectors in disjunct-stats.scm

Consider “de”. There are 12 observations of the disjunct “Janeiro+”. There are 9 observations of the disjunct “la+”. There are 51 observations of a disjunct that has 117 connectors on it!! This starts out as “Diego- Francisco- Francisco- Alonso- Carlos- Fernández- Carlos-” and ends with “Figueroa+ (+ (+ y+” suggesting that there were possibly 51 really bad parses of a very long table of Spanish kings, which was mistaken for being a single sentence. Clearly, its junk; its frequently-occurring junk, which suggests that the table was repeatedly transcluded in maybe 51 different Wikipedia pages.

Similarly, “Department” has 18 observations of a disjunct with 41 connectors on it. It starts with “Education- Education- Health- Services- Services- Immigration-” and ends with “Veterans+ of+ Treasury+ Treasury+”, again suggesting a bad parse of a table mistaken for a sentence, and included in 18 different Wikipedia pages.

The list continues in a similar way, for quite a while. The green line suggests that if some 30 or so pathological cases are ignored, the system settles down to a more respectable behavior. Entries 30 through 50 in the rankings are "Bay" "Street" "Island" "of" "century" "right" "Game" "Georgian" "or" "a" ";" "near" "Party" "team" "law" "Australia" "her" "research" "Church" "east" "Government". Notable is a preponderance of capitalized words, suggesting more tables of various sorts, and a complete lack of verbs. A spot-check of words like “team” and “law” shows that the pathological behavior continues. Several conclusions are possible.

One conclusion is that there is a severe shortage of verbs in Wikipedia articles, and this makes sense: its primarily descriptive, rather than active: running, jumping, hitting, putting, mixing, giving, setting are not the kinds of verbs that are required to describe a typical encyclopedia topic.

Another conclusion is that perhaps the number of observations of pairs are insufficient to get deep, reliable MST parsing. Junk links get used because there were not enough appropriate word-pairs seen to give a good-quality MST parse. A related conclusion is that the connector-set dataset is also too thin: The grammatically reasonable connectors are observed not even a few dozen times, barely pushing them out of the noise-floor of onesie-twosie observations of junk.

So: bigger datasets, and an urgent need for non-Wikipedia content. Fiction, and presumably teen fiction should be filled with the kinds of active verbs describing human motions and actions, and should be absent of tables and lists masquerading as sentences.

Hubiness

Similar to the above, hubiness can be defined as the second moment of the connector count:

$$hub(w) = \frac{\sum_{d,c} N(w,d) C^2(d,c)}{N(w,*)} - K^2(w)$$

Given the earlier Zipfian results on the average degree $K(w)$, it should be no surprise that a ranked listing of words by hubiness is very nearly identical to the listing for average degree. This is, after all, what the scale-free nature of the Zipfian distribution

really means. Not only is the ranking nearly the same, but one also has the approximate equality $hub(w) \approx 2K(w)$ to some ten or twenty percent.

Disjunct Cosine Similarity

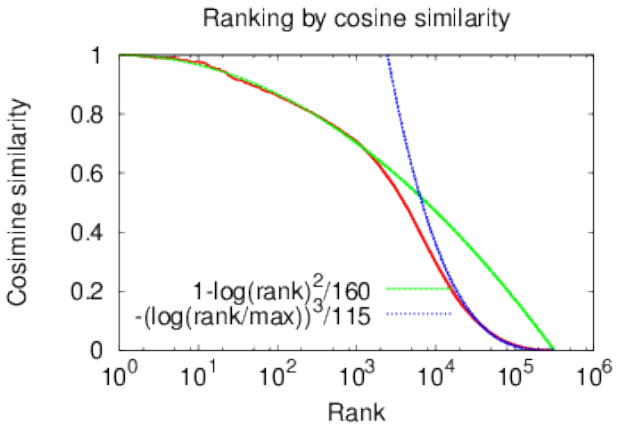
The cosine similarity between two vectors is simply their inner product. In this case, given two words w_1 and w_2, it is given by

$$\text{sim}(w_1, w_2) = \frac{\sum_d N(w_1, d) N(w_2, d)}{\text{len}(w_1)\text{len}(w_2)}$$

where $\text{len}(w)$ is the root-mean-square length (Euclidean length) of the connector-set vector:

$$\text{len}(w) = \sqrt{\sum_d N^2(w, d)}$$

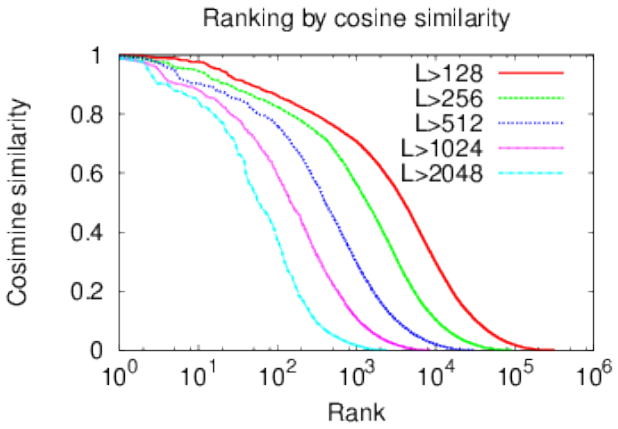
The current dataset being analyzed (the EN_PAIRS_RFIVE_MTWO dataset, same as above) contains 797 words whose length is greater than or equal to 128. The ranking-by-length was already shown up above, in the graph Ranked Euclidean length (RMS Size) on page 17. The similarity between all pairs of these was computed; this resulted in $797 \times 796/2 = 317206$ pairs. These can be sorted and ranked.[22] They are shown below.



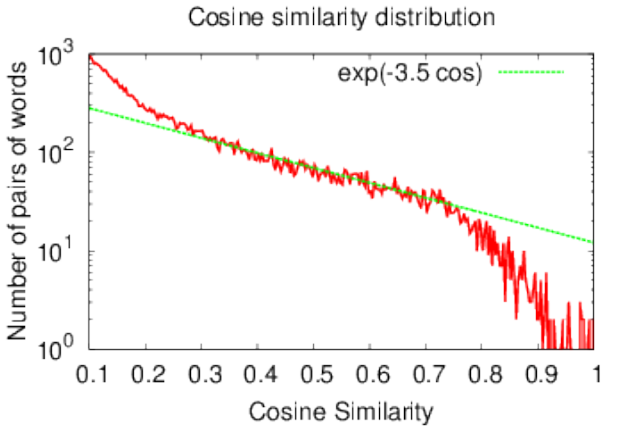
Well, that's new! The similarity ranking is very well fit by a parabola at the high end, and a cubic at the low end! In the above, the green line is given by $1 - \log^2(rank)/160$ and while the blue line is given by $-\log^3(rank/317206)$.

There are 427 words observed 256 or more times, these form $427 \times 426/2 = 90951$ pairs. There are 245 words observed 512 or more times; these form $245 \times 244/2 = 29890$ pairs. There are 130 words observed 1024 or more times, these form $130 \times 129/2 = 8385$ pairs. The ranking for these are shown below. There are 69 words observed 2048 or more times; these form $69 \times 68/2 = 2346$ pairs. The ranking of these are shown below.

[22] See the good-sims and ranked-sims in disjunct-stats.scm file.



The graph below shows the distribution of cosine similarity.[23] There are 797 words for which $128 < \text{len}(w)$. This shows the distribution of the 317206 word-pairs formed from these words, for which $0.1 \leq \text{sim}(w_1, w_2)$. The eyeballed fit is for $\exp(-3.5 \times \text{sim})$.



The top-ten similarity pairs (for which $128 < \text{len}(w)$) are: 'Stats .. Category' 'Notes .. Summary' 'Stats .. Rating' 'Category .. Rating' 'She .. He' 'Category .. Fandom' 'Stats .. Fandom' 'Category .. Character' 'Stats .. Character' 'she .. he'. The first one has a cosine of exactly 1.0, and the rest are above 0.975.

The two words "Stats" and "Category" have exactly one disjunct: it is 'LEFT-WALL- :+' That is, these appear as the first word in a sentence, and are immediately followed by a colon. The word 'Notes' has 27 distinct disjuncts, but only three are observed more than 6 times: 'Chapter- or LEFT-WALL- or End-' That is, 'Notes' appears either all by itself, or as the phrase 'Chapter Notes' or as 'End Notes'. The word

[23] Produced by binned-good-sims.

'Summary' has 47 distinct disjuncts, but only two appear more than 6 times: 'Chapter- or LEFT-WALL-' i.e. either as a solitary word, or as the phrase 'Chapter Summary'. This is why 'Notes' and 'Summary' are considered to be so similar.

Continuing in this fashion: 'Rating' has 7 disjuncts, four of which are observed more than 6 times: '(LEFT-WALL- :+) or (LEFT-WALL- :+ Audiences+) or (LEFT-WALL- :+ Explicit+) or (LEFT-WALL- :+ Mature+)' This suggests it appears in a table of some sort. The first disjunct '(LEFT-WALL- :+)' accounts for its high similarity to both 'Stats' and 'Category'. So, yes, all these words have been discovered to behave in a grammatically similar fashion; however, this behavior is somewhat boring: they arise from the regularity of some table listing.

The first non-capitalized word-pair appears at position ten. Of the top thirty, the table below shows the non-capitalized word-pairs.

rank	cosine	word-pair
10	0.977	'she .. he'
16	0.955	'guess .. suppose'
17	0.954	'city .. house'
19	0.950	'would .. might'
20	0.945	're .. non'
21	0.944	'village .. city'
24	0.931	'father .. mother'
25	0.928	'village .. house'
26	0.928	'don .. sama'
27	0.923	'world .. city'
28	0.921	'son .. daughter'
29	0.918	'suppose .. hope'

The table below shows several more pairs worth closer examination.

rank	cosine	word-pair
32	0.915	'though .. but'
33	0.914	'should .. might'
54	0.893	'should .. must'
61	0.885	'when .. until'
67	0.881	'leave .. take'
68	0.8794	'believe .. think'
69	0.879	'in .. by'

It is worth looking at a few of these, to see how they work out. The word 'she' has 27578 distinct disjuncts on it; the word 'he' has 57330 disjuncts. The table below shows the top-ranked disjuncts for each. It makes quite clear why the two have a high similarity.[24]

[24] Print the disjuncts with the 'print-disjuncts' function.

she		he	
count	disjunct	count	disjunct
2501	said+	4673	said+
1060	was+	2643	was+
983	had+	1959	had+
656	asked+	1293	could+
593	”-	1281	asked+
474	could+	909	that- was+
372	that- was+	813	”-

The list above contains the pairs ’guess .. suppose’ and also ’suppose .. hope’. The word ’guess’ has 859 disjuncts on it. The word ’suppose’ has 805 disjuncts on it. The word ’hope’ has 1735 disjuncts on it. The table below indicates why their cosine distance is close.

guess		suppose		hope	
859 total obs		805 total obs		1735 total obs	
count	disjunct	count	disjunct	count	disjunct
272	I-	377	I-	393	I-
57	I- .+	66	I- you+	160	I- you+
21	I- it’s+	45	I- .+	77	that+
20	I- you’re+	39	I- that+	59	of+
16	I- so+	33	I- it+	55	I- that+
15	could-	27	I- ?+	51	we- that+ you+
15	I- you+	25	I- ,+	33	I- so+

The pair ’re .. non’ seems strange, but closer examination shows that both are usually followed by a hyphen, and that this accounts for all of the observed similarity between these two. The pair ’don .. sama’ is also strange. The similarity is due entirely to their linking to dashes and other punctuation. It seems to be due to a Finnish text that has snuck into the corpus.

Some more verbs are worth looking at. Here is ’leave .. take’, below. Almost all of the similarity is due the infinitive form; the only other shared disjunct appearing in the table is ’to- it+’, although there are more common disjuncts at lower counts.

leave		take	
3017 total obs		5693 total obs	
count	disjunct	count	disjunct
298	to-	619	to-
40	to- the+	105	would-
33	to- .+	95	to- care+
32	took-	88	will-
31	you-	87	to- a+
28	to- him+	74	to- it+
26	to- room+	70	to- her+
25	to- it+	67	I- it+

The pair 'believe .. think' is as follows; several kinds of constructions are clearly shared.

believe		think	
2190 total obs		5462 total obs	
count	disjunct	count	disjunct
339	I-	1321	I-
116	I- that+	308	you-
94	to-	265	don't-
79	to- that+	244	to-
54	that+	211	do- you-
51	I- you+	202	I- it+
51	I- it+	168	I- you+

The first prepositions in the list are 'in .. by' and so are worth a look. Most of the similarity is accounted for by having them precede a determiner. Why there is a strong link to a determiner is unclear. Whether there is also a strong link between the determiner and a subsequent noun is also unclear.

in		by	
64011 total obs		20907 total obs	
count	disjunct	count	disjunct
12484	the+	2064	the+
4776	a+	497	a+
4720	his+	271	followed-
1529	front+	266	his+
1429	her+	252	surrounded-
1113	this+	214	which+
1092	my+	207	means+

The pair 'would .. might' is accompanied by 'should .. might' and 'should .. must'. The disjuncts are shown below. The reason for the similarity is readily apparent, and corresponds with what might be expected.

would		should		might		must	
20039 total obs		6388 total obs		6861 total obs		5860 total obs	
count	disjunct	count	disjunct	count	disjunct	count	disjunct
1648	have+	1064	have+	654	have+	666	have+
1510	be+	840	be+	468	be+	558	be+
621	it- be+	415	I-	154	he- have+	215	it- be+
445	he- have+	246	I- be+	150	it- be+	195	you-
315	he- be+	155	we-	91	he- be+	183	I-
290	not+	134	he- be+	89	I-	137	we-

The words 'city', 'house', 'village' and 'world' are all similar. These are shown below.

city		house		village		world	
1332 total obs		3330 total obs		1019 total obs		3024 total obs	
count	disjunct	count	disjunct	count	disjunct	count	disjunct
258	the-	598	the-	157	the-	646	the-
96	the- ,+	275	the- ,+	57	the- .+	265	the- .+
94	the- .+	226	the- .+	57	the- ,+	202	the- ,+
27	the- and+	80	a-	23	street+	158	in- the-
24	this-	79	into- the-	21	the- Mbonga+	117	in- the- .+
17	the- of+	58	the- and+	14	a-	112	in- the- ,+
15	The- the- .+	54	his-	12	their-	82	this-
14	the- the- .+	48	my-	11	the- and+	43	.+
13	a-	45	the- was+	9	the- of+	39	the- is+

The above table clearly indicates why the cosine similarity between these four words is high. Yet, it is also disappointing: the primary reason is that they all take the determiner 'the', and occur at the end of phrases (the- ,+) or at the end of sentences (the- .+). This seems superficial, at best. After that, there's not so much similarity, and what there is still falls back onto the presence of determiners.

The similarity above is predicated on the frequent pairing of a word with a determiner. The pairing itself is correct, but perhaps not entirely significant: the word-pairs that needed to appear in the MST parse to extract these disjuncts had almost abysmally low MI values. They were clearly high enough to allow the MST parse to move forward, but are not otherwise terribly promising. They are shown below.

word-pair	fractional mi
the city	2.7827
the house	2.4228
the village	2.6635
the world	3.0578
city .	1.0167
house .	1.0095
village .	0.8765
world .	1.2255

Recall that, for word-pairs, that MI scores below 4 are considered to be quite poor. That the scores above are poor is not surprising: determiners link with almost any noun, but not to verbs: this is enough to get them into the 2-3 range for MI. By contrast, the period can appear after almost any word, except maybe for prepositions, adjectives and adverbs. This is apparently enough to drive the MI positive, but that's all.

The above suggests that using the raw observation count to compute the cosine similarity is perhaps not the best way to compute similarity.

Score Cosine

The above results suggest that perhaps the correct way to assign a score is to total up the word-pair MI scores for each disjunct, and use that to form a cosine. That is, define

$$sc(w,d) = \sum_{c \in d_+} mi(w,c) + \sum_{c \in d_-} mi(c,w)$$

where $d_+ \subseteq d$ is the subset of connectors that connect to the right, and $d_- \subseteq d$ is the subset of connectors that connect to the left. Here, $c \in d$ is simply one of the words that the connector is connecting to. Then, $mi(w,c)$ and $mi(c,w)$ are the word-pair MI scores. The cosine similarity might then be defined as

$$\text{sos}(w_1,w_2) = \frac{\sum_d sc(w_1,d)sc(w_2,d)}{\sqrt{\sum_d sc^2(w_1,d)}\sqrt{\sum_d sc^2(w_2,d)}}$$

The idea here is that connectors to punctuation, and connectors to determiners, which might be observed quite often, but have a naturally low MI score, will contribute relatively little to the overall cosine similarity.

Why is this a reasonable thing to do? Well, the previous two tables already suggested that the four words 'city' 'house' 'village' and 'world' should be placed into the same grammatical class, simply because of the high frequency with which they connect to a determiner. More subtle differences carrying a semantic signal might be getting washed out in this process. Perhaps the sos cosine score would be more sensitive to those processes.

But, if we are inventing scores, much more is possible.

Overlap Similarity

Cosine similarity is perhaps too strict: it judges that two words are similar when they share not only the same collection of disjuncts, but have these occur with the same frequencies. Perhaps its enough to say that two words are grammatically similar, if they simply share the same set of disjuncts. This suggests ignoring the relative counts, or at least, sharply filtering them. Thus, the overlap similarity can be defined as

$$\mathrm{ovl}(w_1,w_2) = \frac{\sum_d \sigma(N(w_1,d))\sigma(N(w_2,d))}{\sum_d \sigma(N(w_1,d)+N(w_2,d))}$$

where $\sigma(x)$ is some sigmoid function. In the most basic case, its a step function: $\sigma(x) = 1$ if $x > C$ for some constant C and zero otherwise. Each term in the numerator sum is one only when both disjuncts are present. Each term in the denominator is one when either disjunct is present.

This similarity measure essentially boils down to a normalized cylinder-set measure. That is, instead of interpreting the disjuncts as a basis for a vector space, they can be taken as independent observations in a Cartesian product space. This makes more sense than pretending that these form a vector space. Why is that? What makes the cosine angle, and vector dot products so appealing, is that they are preserved by rotations; yet there really is no reason to expect or desire rotational symmetry. In short, the idea that we are dealing with vector spaces give the wrong idea of what's going on: its more appropriate to view the set of disjuncts as elements of a product space. Product spaces have a natural measure: the cylinder-set measure. The overlap similarity is essentially $\mu(A \cap B)/\mu(A \cup B)$ with μ the Borel measure, and the intersection/union being taken over the associated disjunct sets.

Continuing in this vein, what should be taken as the measure $\mu(A)$ of A? The counting measure is a natural choice; that is, $\mu(A) = |A|$ being the number of elements in set A. Because we also have a count associated with the members of the set, we can consider using the l_p norms for the measure. That is, $|A|$ is just the l_0 norm; perhaps $\mu(A) = \|A\|_p$ could also work. For the purposes here, this may be enough; note, however, that only l_0 and l_1 satisfy one of the axioms of measure theory: namely, that when $A \cap B = \varnothing$ then $\mu(A \cup B) = \mu(A) + \mu(B)$. The other l_p norms do not satisfy this.

The (opencog matrix) module currently implements an overlap similarity which computes the l_0-norm based similarity:

$$\mathrm{ovl}_0(w_1,w_2) = \frac{|\{d \text{ s.t. } 0 < N(w_1,d)\} \cap \{d \text{ s.t. } 0 < N(w_2,d)\}|}{|\{d \text{ s.t. } 0 < N(w_1,d)\} \cup \{d \text{ s.t. } 0 < N(w_2,d)\}|}$$

where $\{d \text{ s.t. } 0 < N(w,d)\}$ is the set of disjuncts d such that the cset (w,d) was observed at least once. As always, $|\{x\}|$ is the number of elements in the set $\{x\}$.

Some experimentation was done with this similarity measure, but the results were not particularly impressive. What does become quickly apparent is that the most frequently observed words will necessarily have a low similarity to the low-frequency words. This is because the high-frequency words will have a large number of disjuncts observed with them, thus causing the denominator to grow large. The numerator, by

contrast, stays small, essentially limited by the number of disjuncts observed on the low-frequency word. This behavior is not what we want. Intuition suggests that we really do want to be able to compare words, independent of how frequently they occur.

This suggests a modified form of overlap similarity, by normalizing to a common count. That is, one should work with $N(w,d)/N(w,*)$, which can be understood as the normalized probability of observing a disjunct d on some word w. So, arrange the words such that $N(w_1,*) > N(w_2,*)$ and define $K(w_1,d) = N(w_1,d)N(w_2,*)/N(w_1,*)$ – this makes the counts on w_1 directly comparable to those on w_2, giving

$$\text{rovl}(w_1,w_2) = \frac{|\{d \text{ s.t. } 1 \le K(w_1,d)\} \cap \{d \text{ s.t. } 1 \le N(w_2,d)\}|}{|\{d \text{ s.t. } 1 \le K(w_1,d)\} \cup \{d \text{ s.t. } 1 \le N(w_2,d)\}|}$$

That is, the condition $1 \le N(w_2,d)$ simply tests for the presence or absence of disjunct d on w_2, while $K(w_1,d)$ gives the probability of observing disjunct d on w_1, if w_1 had been observed as often as w_2.

The above considerations suggest a more appropriate definition for overlap similarity that allows for frequency-independent observations: namely

$$\text{ovl}(w_1,w_2) = \frac{\sum_d \sigma(f(w_1,d))\sigma(f(w_2,d))}{\sum_d \sigma(f(w_1,d)+f(w_2,d))}$$

with $f(w,d) = N(w,d)/N(w,*)$. In this form, the numerator now starts to resemble the numerator of the cosine similarity measure: both the cosine numerator, and this numerator are computing a kind-of overlap or intersection of sets of disjuncts. The denominators differ. Based on gut intuition, the above measure may be be more suitable for judging similarity than the cosine measure, precisely because it emphasizes the set-like qualities of connector sets, as opposed to vector-like qualities.

Disjunct Subsets

The above line of thinking suggests that another interesting comparison can be made by looking for a subset relationship between the disjuncts on different words. For example, transitive verbs should have all the connectors that intransitive verbs do, plus some more.

The overlay similarity can be adapted for this purpose: two sets obey a subset relation $A \subset B$ if $A \cap (1-B) = \varnothing$.

This re-affirms the previous observation: we expect connector-sets to behave in a set-like fashion, not a vector-like fashion.

Connector Similarity

There is another, dual kind of similarity that is very different from the above proposals. A single disjunct is an ordered list of (pseudo-)connectors:

$$d = (c_1, c_2, \cdots, c_k)$$

where each pseudo-connector is a word, and a direction indicator. We can judge two connectors to be similar, and thus, two words to be similar, if they appear in a large number of disjuncts that would be identical, but for that one connector.

There are some practical difficulties of writing code to discern this.

Word-Pair Cosine Similarity

To correctly gauge the advantage of disjunct-based techniques, one could compare them to the same kinds of measures, but applied to word-pairs, instead of word-disjunct pairs. For example, the disjunct-based cosine similarity, can be contrasted against the simpler word-pair-based cosine similarity. This is given by

$$\text{sim}_{pair}(w_1, w_2) = \frac{\sum_w N_{pair}(w_1, w) N_{pair}(w, w_2)}{\text{len}(w_1)\text{len}(w_2)}$$

where $\text{len}(w)$ is the root-mean-square length (Euclidean length) of the pair vector:

$$\text{len}(w) = \sqrt{\sum_v N_{pair}(w, v) N_{pair}(v, w)}$$

and $N_{pair}(w, v)$ is the count of having observed the ordered word-pair (w, v). Equivalently, writing the normalized frequency of observing a word pair as $p(w, v) = N(w, v)/N(*, *)$, this similarity can be written in the form

$$\text{sim}_{pair}(w_1, w_2) = \frac{\sum_w p(w_1, w) p(w, w_2)}{\sqrt{\sum_v p(w_1, v) p(v, w_1)} \sqrt{\sum_v p(w_2, v) p(v, w_2)}}$$

Note that this similarity measure is NOT symmetric: $\text{sim}(w_1, w_2) \neq \text{sim}(w_2, w_1)$. This is because it's built out of a manifestly non-symmetric count: $p(w, v) \neq p(v, w)$ and should really be written as $p(w, v) = p(R; w, v)$ with the relation R encompassing all of the constraints of pair-wise word relationships (including, for example, that the pair might have been extracted from a random planar tree parse). Of course, one could construct a symmetrized similarity measure.

The point here is that this measure treats words as similar with they link, pair-wise, to the same kinds of words, with the same kinds of frequencies. This is not unlike the similarity that the disjunct-cosine is measuring, except that the disjunct carries additional grammatical information with it: it captures more complex relationships between the words in a sentence.

Cosine Information

The cosine similarity was defined as

$$\text{sim}(w_1, w_2) = \frac{\sum_d N(w_1, d) N(w_2, d)}{\sqrt{\sum_d N^2(w_1, d)} \sqrt{\sum_d N^2(w_2, d)}}$$

which, after dividing by $N(*, *)$ so that $p(w, d) = N(w, d)/N(*, *)$, gives the equivalent expression

$$\text{sim}(w_1, w_2) = \frac{\sum_d p(w_1, d) p(w_2, d)}{\sqrt{\sum_d p^2(w_1, d)} \sqrt{\sum_d p^2(w_2, d)}}$$

Comparing this to the expression for mutual information suggests that using the vector support, instead of the vector length, could be interesting. In particular, these might be interesting:

$$\mathrm{com}(w_1,w_2) = -\log_2 \frac{\sum_d p(w_1,d)p(w_2,d)}{p(w_1,*)p(w_2,*)}$$

Perhaps the following is closer to the "original intent" of mutual information: let $\mathrm{dot}(w_1,w_2) = \sum_d p(w_1,d)p(w_2,d)$

$$\mathrm{dmi}(w_1,w_2) = -\log_2 \frac{\mathrm{dot}(w_1,w_2)\mathrm{dot}(*,*)}{\mathrm{dot}(w_1,*)\mathrm{dot}(*,w_2)}$$

Or even

$$\mathrm{cmi}(w_1,w_2) = -\log_2 \frac{\mathrm{sim}(w_1,w_2)\mathrm{sim}(*,*)}{\mathrm{sim}(w_1,*)\mathrm{sim}(*,w_2)}$$

The $\mathrm{sim}(*,*)$ serves to normalize the entire calculation, so that one is effectively computing with norm-$\mathrm{sim}(w_1,w_2) = \mathrm{sim}(w_1,w_2)/\mathrm{sim}(*,*)$. Some experiments cmi were performed,, and it quickly became apparent that cmi does "the wrong thing": it singles out pairs that have a lot to do with each other, but little to do with anything else. Which discriminates *against* the clusters of similar words, which is not what we want. We don't want strange, unusual pairings; we want common, likely pairings. The level of discrimination can be severe: some really, really bad pairings show up, but only because they are unlike anything else. This score is strong when sim is weak, and is essentially picking up the tail-end of the sim distribution.

One can be inspired to write some crazy concoctions:

$$\mathrm{mim}(w,d) = -\log_2 \frac{p(w,d)}{\sum_{w,d} p^2(w,d)}$$

I don't know what to call these; the first seems to be some kind of "cosine information", the second, some sort of "mutual length" device.

Quality Cosine

The motivation for using cosine similarity is to find pairs of words that act in a grammatically similar fashion: they are used in the same way, with the same kinds of disjuncts. However, observational counts are subject to the vicissitudes of the input text: perhaps, instead of using vectors where the components are given by the frequency, once could instead use components based on, say, the "quality" of the disjunct itself. A disjunct could be judged as being "high quality" if $mi(w,d) = MI_{pair}(w,d)$ as defined in equation 2 on page 19 is high. This motivates a "quality cosine":

$$\mathrm{qim}(w_1,w_2) = \frac{\sum_d mi(w_1,d)mi(w_2,d)}{\sqrt{\sum_d mi^2(w_1,d)}\sqrt{\sum_d mi^2(w_2,d)}}$$

But what if the high quality is based on a pathetically small number of observations? Whenever one has a small number of observations, one has, almost automatically, a high MI value, simply because these two things are seen together, and nothing else is.

Perhaps there should be some observational weighting. One can contemplate defining $pi(w,d) = p(w,d)mi(w,d)$ and so the cosine

$$\text{pim}(w_1,w_2) = \frac{\sum_d pi(w_1,d)pi(w_2,d)}{\sqrt{\sum_d pi^2(w_1,d)}\sqrt{\sum_d pi^2(w_2,d)}}$$

The suitability of these different means of judging similarity is not clear.

Cosine Similarity scatterplots

The scatterplot in 1visualizes the cosine similarity between 797 word-pairs.[25] The rows and columns are ranked by frequency of word occurrence, so that the single-most-frequently occurring word is at the upper-left, with less and less frequently occurring words proceeding to the right, and downwards. The color scale is such that red represents 1.0, yellow represents 0.75, green is exactly 0.5, and blue is 0.25 or less, fading to black. This is for the same EN_PAIRS_RFIVE_MTWO dataset as above, and examined in detail in section on page 28. That is, it computes the similarity between the 797 words whose length is greater than 128. Note that cosine similarity is symmetric; this figure is necessarily symmetric about the diagonal. The diagonal can be seen as a red line, representing a similarity of 1.0.

The tartan pattern indicates that some words are very unlike others, and that most words are very unlike one-another.

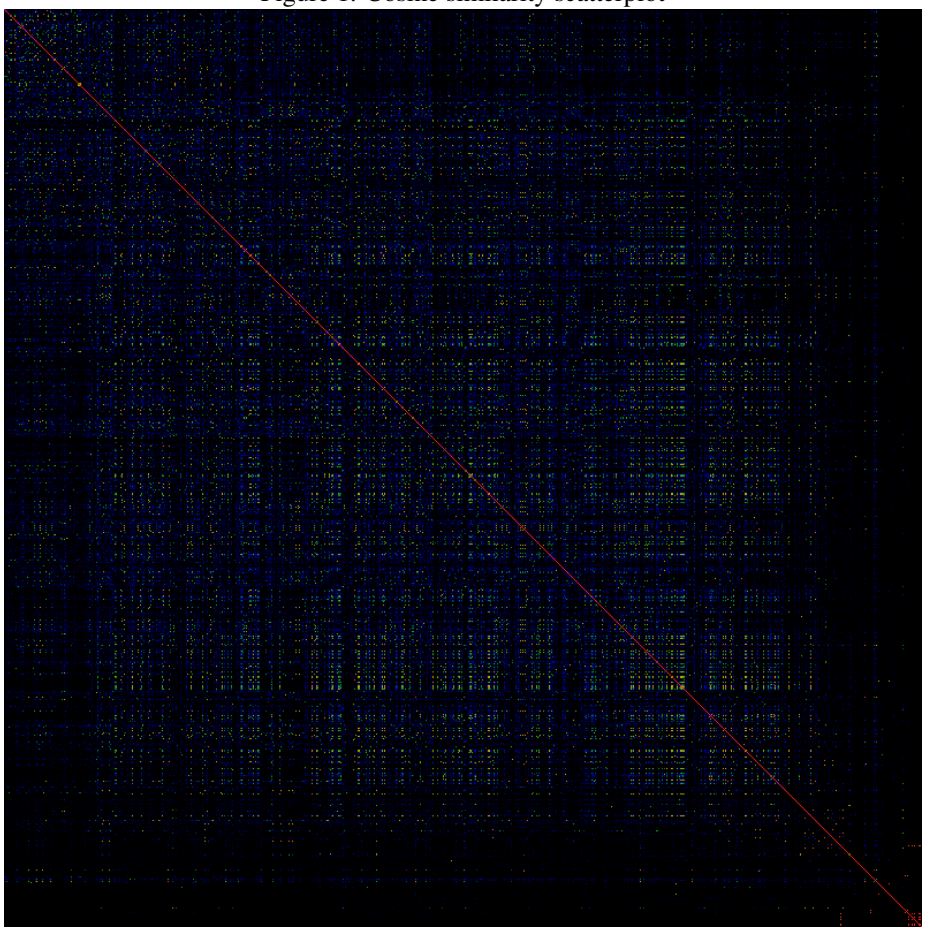
It would be nice to re-arrange (permute) the rows and columns to bring the matrix into quasi-diagonal form. This is easier said than done. The graph 2 shows the same data, with a different ranking. This time, the first word is the the LEFT-WALL, and the next word is the one word, out of 796, that has the highest cosine similarity to the period. Next comes the word, out of the remaining 795, that has the highest cosine similarity to the last.[26] This continues on down the list, so that word-pairs with the highest similarity are next to each-other, in the list. Effectively, we get the two-highest similarities for each word: the highest being to the word right before, and second-highest to the word right after. The list is organized so that the highest similarity pair occurs in the upper-left. The pairs least-similar to anything else end up on the lower-right. The color scheme is as before.

This ranking exposes block-diagonal structure in the dataset. The block in the upper-right consists entirely of punctuation and various capitalized words. The middle portions are quite interesting. The full list of 797 words is shown in table on page 42. Its sort of interesting to read. So, the word that is most similar to the LEFT-WALL is

[25] Generated with 'scatter.scm'

[26] Computed with the 'dranked' function, and the 'dranked-long' list graphed, from the 'disjunct-stats.scm' file.

Figure 1: Cosine similarity scatterplot



the double-dash – perhaps not to surprising, as the double dash is often used to start new sentence phrases. That various forms of punctuation follow is not surprising.

After this are various capitalized words. As noted previously, capitalized words are observed far less frequently than their u-capitalized counterparts – usually only one capitalized word per sentence! Thus, capitalized words have far fewer disjuncts on them, thus making it easier for them to appear superficially similar, when they really should not be. Despite this, it is reassuring to see "Where Who What Why How" occur in succession. Other confidence-instilling sequences include "2 1 4 3 A" and some names: "Mai Demelza Richard John George". Then another reassuring sequence: "nothing something anything everything". Why "George" should be similar to "nothing" is best left unasked. But we'll ask anyway.

Figure 2: Quasi-diagonal cosine similarity

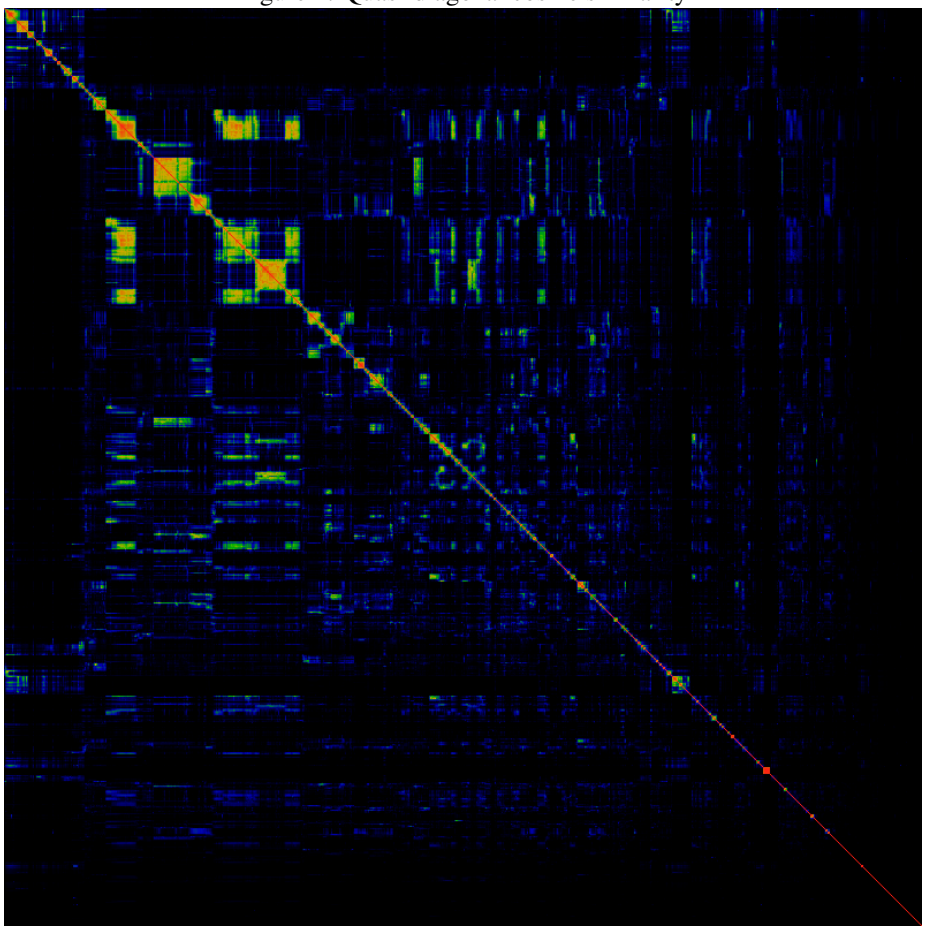


Table 1: Block-ranked words

LEFT-WALL – ... . *** Two Bit - The “ And But So Perhaps Though When As Then While If Have Can Did Do Will Thank Are Where Who What Why How Or Now Here This That It There He She I You We They Some One Most Instead However Besides Well Oh Ah No Yes Yeah Good Ham 2 1 4 3 A Mai Demelza Richard John George nothing something anything everything it there who which that what where when until as because since if before while little small large good great second woman man child boy girl king house city village world town country ship sun latter land others children people men things them us me him her his my your their our the this every another leave take get find see hear speak move stop fight be make give show keep bring meet follow call ask tell help go live stay talk turn run change use read say understand remember know think suppose guess hope promise believe mean am myself fear thought knew saw felt found called happy fine certain new single word moment minute year day night evening morning way book letter party distance light fire forest room air sea earth ground floor table subject story line case question place time thing person family soul heart body own wife brother sister father mother daughter son hands face feet mouth eyes hand mind friends friend thoughts arm arms chest staff words office past water door window truth same whole best present first last least once home back down up at over on in by from upon under through into for with such having like quite only also still born gone already been seen done taken known given brought made making taking after above among between of all near against to will can cannot may must might would should could did does was is seems seemed appeared used began tried wanted needed meant came went turned continued said says spoke had has always never ever usual possible well soon far much late bad strong big young white black dark blue closed opened open hold put stand sleep answer try return come enough ready trying going supposed beginning rest sound corner direction presence name side power nature one most part kind sort number lot bit couple matter group other two three four five ten several many some out length force death life voice hair husband lips head shoulder eye bed business point state view sense feeling thinking saying now indeed therefore however sir yes God dear lady poor old dead wrong right here doing glad afraid sure sorry coming close hard long short real different clear probably certainly simply just almost not hardly easily feel have shall wish need want seem next following Queen captain wall future general London work himself herself smiled sighed nodded asked replied cried got stood sat looked look smile chance reason means longer doubt idea end edge bottom top account instead perhaps and or but though although then thus yet even really heard left lost passed met were are do love told gave took held followed behind within half an raised free better less more rather often true alone again) too so very pretty a its these those anyone you we they he she North Jack Pitch Ross Jim Dallas Telzey Goth Sir san together along off away around round about how why – ; , : . . . ! . ? — All From In On By To With After For It’s It’s Not Just Even At See paragraph spirit cause middle sight front spite charge form attention throat breath money law human few hundred thousand years hours minutes days times later ago ’ " ” ’ _ etc Mr Mrs Dr sama don ul looking especially nor either each any no finally suddenly slowly quietly Mary King Lord Lady Miss Captain I’m I’m My Her His Your (‘ Illustration Stats Category Rating Fandom Character Warning Is Really Please Father Mother May course being without let set sitting drawing living high heavy cold public fact particular full relief order hour individual General outside both talking able happened exactly else hell non re self S t six natural than feel deep wide deal agree don’t don’t didn’t didn’t received paid care drop provide York agreement ape medium author laws terms works forward forth shook enjoyed originally access permission instance chapter comment St Princess Old New Summary Notes Our THE # * Of { including Prince notes states copies copy YOU THIS OF OR Section CHAPTER Chapter + | Project eBook eBooks Gutenberg } P tax States fee electronic associated Archive Foundation United & copyright Literary donations License trademark distribution refund Van distributing Ooal Additional Posted archive Tags

The block-ranked words are a kind of "stream-of-consciousness" with regard to similarity. This is shown in the table below.[27] So, the numbers are all quite similar to each other, but the letter 'A' is not very similar to '3'. It just so happens that 'A' is more similar to '3', than any other subsequent word. 'A' and 'Mai' are not similar, but then all of the given names are quite similar. Next, it turns out that 'George' and 'nothing' really don't have very much to do with one-another. And so the block-diagonal structure is exposed.

Pair	Similarity
2 .. 1	0.881
1 .. 4	0.791
4 .. 3	0.835
3 .. A	0.375
A .. Mai	0.351
Mai .. Demelza	0.615
Demelza .. Richard	0.515
Richard .. John	0.602
John .. George	0.648
George .. nothing	0.342
nothing .. something	0.483
something .. anything	0.549
anything .. everything	0.369
everything .. it	0.496
it .. there	0.743
there .. who	0.416
who .. which	0.618
which .. that	0.580
that .. what	0.730
what .. where	0.722
where .. when	0.830
when .. until	0.886

Reading through the list of 797 words in table on the preceding page reveals a lot of interesting runs. All this suggests that cosine similarity really is doing the right thing. Equally reassuring are the failures of similarity: the end of the list of 797 words is populated with Gutenberg license boilerplate words. This is good: they are at the end of the list because they don't fit into any other grammatical usage patterns. They don't fit because they have been observed hundreds of times, which propels them into the high-frequency category; yet, since they always appear in set phrases, they can't be similar to anything else.

[27] Created with the 'prt-sim' function.

Filtering

An earlier draft of this report showed a very different figure, which turned out to have been created based on what seemed to be a valid assumption, but turns out not to be so wise. It deserves a some discussion here, which is resumed in a later section.

The idea (recaptured below) is that it can be a good thing to filter out noisy data. For example, one may choose to ignore words that have not been seen many times. This is the cut done up above: there are only 797 words observed with a length of 128 or greater. This is a convenient number only because it allows a 797-pixel-wide color plot to be made. A different cut might exclude words that were observed less than N times.

Another possibility, and this seems to be the fatal one, is to exclude disjuncts that are seen only a handful of times. Superficially, this seems like a very plausible thing to do. In practice, this turns out to be an almost completely disastrous cut. Setting it too high renders almost all capitalized words identical to one-another, having a similarity of exactly one. Capitalized words are observed only infrequently; discarding low-frequency disjuncts leaves the capitalized words almost naked, and thus essentially identical. Just about all of them connect to the LEFT-WALL, since they are the first word in a sentence, and so all of them share a very high observation count of links to LEFT-WALL, and low observation counts of links to anything else. Ergo: they are all similar, since they all start sentences.

The damage doesn't stop there. It turns out that such trimming also raises similarity across the board: it ends up so that everything seems to be pretty darn similar to everything else, thus erasing too much of the "signal" that we are looking for. The original hope was to raise the signal-to-noise ratio by applying judicious cuts; but hope is not enough. Injudicious cuts can destroy the signal all too easily. Careful data analysis is needed; blind trust is not enough.

Dataset report 3 June 2017

Some summary reports from various different datasets.

Word-Pair datasets

First, datasets that hold word-pairs, parsed using the LG "ANY" link type: i.e. random parse trees.

Size	Pairs	Obs'ns	Obs/pr	Entropy	MI	Dataset
395K x 396K	8.88M	418M	47.0	19.28	3.02	en_pairs_sim
138K x 140K	4.89M	140M	28.6	17.73	2.03	en_pairs_tone_mst
183K x 187K	8.05M	268M	33.3	17.83	1.84	en_pairs_ttwo_mst
425K x 432K	15.2M	557M	36.6	18.32	1.93	en_pairs_tthree
134K x 135K	5.54M	174M	31.4	17.67	1.94	en_pairs_rone_mst
185K x 188K	8.95M	321M	35.9	17.77	1.79	en_pairs_rtwo_mst
428K x 434K	16.4M	639M	38.9	18.27	1.90	en_pairs_rthree
839K x 851K	30.1M	1.35G	44.9	18.54	1.84	en_pairs_rfive
158K x 159K	5.92M	729M	123	18.45	2.02	zh_pairs_sone
60K x 60K	1.68M	87.8M	52.3	17.47	2.88	zen_pairs

The legend is as follows:

Size The dimensions of the array. This is the number of unique, distinct words observed occurring on the left-side of a word pair, times the number of words occurring on the right. We expect the dimensions to be approximately equal, as most words will typically occur on both the left and right side of a pair.

Pairs The total number of distinct pairs observed.

Obsn's The total number of observations of these pairs. Most pairs will be observed more than once. Distributions are typically Zipfian, as previous sections point out.

Obs/pr The average number of times each pair was observed.

Entropy The total entropy of these pairs in this dataset, as defined previously: for word-pairs (w_L, w_R) it is $H = -\sum_{w_L,w_R} p(w_L,w_R) \log_2 p(w_L,w_R)$.

MI The total mutual information for the pairs in this dataset, as defined previously: $MI = \sum_{w_L,w_R} p(w_L,w_R) \log_2 [p(w_L,w_R)/p(w_L,*)p(*,w_R)]$

The datasets are as below.

en_pairs_sim This contains text parsed from Wikipedia, only. As noted previously, Wikipedia is painfully short of verbs and pronouns. Compared to the Gutenberg datasets below, it is also very rich in foreign words and proper names (product and brand names, geographical place names, biographical mentions and other named entities). Issue: missing connectors the LEFT-WALL.

en_pairs_tone_mst Text from Project Gutenberg "tranche one", mostly all "famous authors", popular, well-known 19th century books. Includes six modern sci-fi/fantasy novels from other sources, and some 20th century non-fiction, including a military appraisal of Vietnam.

en_pairs_ttwo_mst Tranche two - Everything from tranche one, plus fan-fiction from http://archiveofourown.org. Most of the selected texts were 10K words or longer. See the 'download.sh' file for the precise texts. Issues: tone_mst and ttwo_mst are missing connectors the LEFT-WALL. Certain types of punctuation is mishandled.

en_pairs_tthree Tranche three - Everything in tranche two, plus several hundred of the most recently created Project Gutenberg texts, whatever they may be. See the 'download.sh' file for the precise texts. The _mst version has the same issues that ttwo_mst has, although some connectors to LEFT-WALL do get added. The _mst version is probably not useful for similarity measurements.

en_pairs_rone_mst Same as en_pairs_tone_mst, but with minor issues fixed. However, links to LEFT-WALL still missing.

en_pairs_rtwo_mst As above, tranche 1 & 2.

en_pairs_rthree As above, tranche 1,2 & 3.

en_pairs_rfive As above, tranche 1,2,3,4 & 5.

zh_pairs_sone A parse of Mandarin Wikipedia, with each individual character (hanzi) treated as a single item (so that, during pair-counting, pairs are formed between items). Non-Chinese characters are grouped into words in the normal way, by splitting according to white-space (and punctuation). Thus, the total dimensions of the dataset are given by the number of observed Chinese characters (hanzi) plus the number of observed non-Chinese words (and punctuation).

zen_pairs A parse of a small set of Mandarin novels, with text segmented into words by external third-party tools (provided by Ruiting).

Now, for some commentary, as to the summary stats. For English, as the number of pair observations increase, so do the number of unique, distinct words. The relation even seems to be linear: double the number of pair observations, and the number of different words also increases. This suggests something Zipfian at work. The explosion of words is hypothesized to be given names, although these datasets all fail to split hyphenated words, and so some may be due to that. The point is that the average observations per pair increases with difficulty, and the entropy and MI does not budge at all.

Comparing the English _sim dataset to the _rone, _rtwo and _rthree datasets does provide some contrast: The _sim dataset, built from Wikipedia, is distinctly different from the Gutenberg datasets. Certainly, the prose style in the two datasets is quite different, with Wikipedia consisting of statements of facts ("is", "has" relational statements) concerning a broad range of named entities, whereas the Gutenberg texts are primarily narrative adventures ("did", "went" activity statements) involving fictional personages.

Comparing English to Chinese is very interesting. The Chinese dataset has three times, almost four times more observations per pair; equivalently about 3-4 fewer "words". This is partly due to the fixed number of ideograms in the language. Remarkably, the entropy and MI are untouched. This suggests that the entropy and MI

are capturing something about the human nature of language use, as opposed to something descriptive of the language itself. However, a lot more data would be needed to see if this is really true.

There's something else interesting going on, shown in the table below.

Size		Support		Count		Length		Dataset
L	R	L	R	L	R	L	R	Name
158K	159K	6819	6411	548	487	41.7	37.7	zh_pairs_sone
60K	60K	8170	8702	191	156	18.1	15.1	zen_pairs
839K	851K	80.6K	80.6K	249	230	28.2	24.5	en_pairs_rfive
428K	434K	45.6K	45.1K	208	187	22.9	19.4	en_pairs_rthree
185K	188K	24.7K	23.8K	199	173	21.5	17.8	en_pairs_rtwo_mst
134K	135K	17.4K	17.4K	143	129	16.6	14.0	en_pairs_rone_mst

The columns are as follows:

Size The left and right dimensions, as before. Viz, the number of unique, distinctly different words observed on the left and the right side of a pair. Viewed as a matrix, this is the number of columns and rows in the matrix.

Support The support is the average number of word-pairs that a word participates in (on the left, or on the right). Viewed as a matrix, this is the average number of non-zero entries in each row or column. Viewed as (row or column) vectors, this is the "support" of a (row or column) vector. Mathematically, this is the l_0 norm of each vector: $|(w_L,*)| = \sum_{w_R} (0 < N(w_L,w_R))$ and likewise $|(*,w_R)| = \sum_{w_L} (0 < N(w_L,w_R))$.

Count The count is the average number of observations that a word-pair was observed, for a given word. Viewed as a matrix, this is the average value of each non-zero entry (averaged over rows, or columns). Viewed as vectors, this is the l_1 norm divided by the l_0 norm. The l_1 norm is just the wild-card counts $N(w_L,*)$ and $N(*,w_R)$, where as always, the wild-card counts are defined as $N(w_L,*) = \sum_{w_R} N(w_L,w_R)$. The count shown in the table is then the average count: $N(w_L,*)/|(w_L,*)|$ for the rows, and likewise for the columns.

Length The length is the average length of the row and column vectors. This is the l_2 norm divided by the l_0 norm. The l_2 norm is just the standard concept of the length of a vector in Euclidean space. Here, $L(w_L,*) = \sqrt{\sum_{w_R} N^2(w_L,w_R)}$, and likewise $L(*,w_R) = \sqrt{\sum_{w_L} N^2(w_L,w_R)}$. The length is interesting, because it "penalizes" word-pairs with only a small number of counts. The act of squaring the count has the effect of giving much higher "confidence" to large observation counts: a word-pair observed twice as often is given four times the credit. The length shown in this table is the "average" length: it is $L(w_L,*)/|(w_L,*)|$ for the rows, and likewise for the columns.

So here's what is so interesting in this table: the support, for Chinese, is outrageously different than it is for English. For a given item (hanzi, for Chinese, word, for English), the Chinese hanzi participates in three to four fewer item-pairs! Since pairs are formed on a sentence-by-sentence basis, this means that the variety of different hanzi that can occur in a single sentence is much more constrained, much more strongly correlated. Now, perhaps this comparison is not quite valid: because we are not comparing words to words, but rather English words to Chinese "morphemes" (in the sense that Chinese words are typically composed of 1, 2 or 3 hanzi). Still, its interesting and surprising. This has knock-on effects: the observational counts are much higher, as are the average lengths. It would be interesting to repeat the previously given analysis of the various distributions, and see how they differ.

Disjunct datasets

Next, datasets that hold disjuncts. This section used to report more data, but it was all flawed: the MI had a minus sign in it, causing all computed disjuncts to be maximally bad. Despite this, the results were similar to the below: observations and entropy fit in line, as expected. The H_{left} entropy values were lower, hovered around 15, and the MI was int the 3-5 range, while H_{right} was unchanged and fit in line. You can find the original data in the git commit 9244905afdff191a39af8c5a6deab592d5a1558c.

Size	Csets	Obs'ns	Ob/cs	Entropy	H_{left}	H_{right}	MI	Notes
37K x 291K	446K	661K	1.48	18.30	16.00	10.28	7.98	en_pairs_sim
137K x 6.24M	8.63M	18.5M	2.14	20.96	19.14	9.71	7.90	en_pairs_rfive_mtwo
60K x 602K	801K	1.19M	1.48	18.86	17.99	10.13	9.26	zen_pairs_mst

An updated legend for the columns:

Size The dimensions of the array. The left dimension counts words, the right dimension counts the number of unique, distinct pseudo-disjuncts.

Left-Right The left and right entropies, as defined previously. Note that $MI = H - H_{left} - H_{right}$ holds, by definition. Not given for the word-pairs table, because these two are nearly equal, and are half the difference between the entropy and the MI.

The End