Word Pair Distributions

Linas Vepstas

12 July 2019 - Final Version

Abstract

The distribution of the the mutual information (MI) for word-pairs is examined for English and Chinese. Understanding this distribution is important, as the MI is an important ingredient for the unsupervised learning of grammatical structure. The distribution is found to be bi-modal, decomposing into a zero-mode that appears to describe completely random pairings of words, and a true mode, that captures actual collocations (idioms, institutional phrases, *etc.*) This bimodal structure is found both in English and in Chinese, and is not only the same qualitatively, but is also quantitatively quite similar.

This report is an adjunct to, and an extract from the Language-Learning Diary, focused on describing recent (2015 and later) word-pair datasets. It is meant to provide context for a broader scope of work.

Introduction

The current conception of the language-learning pipeline[1, 2] attempts to extract grammatical structure from text corpora by overlaying a rough, preliminary connectivity network on the text. That preliminary network can then be chopped up into small pieces, and those pieces can then be compared, using statistical techniques, to refine it's accuracy, and to build a lexical description of the network. The lexical description then captures the structure of the preliminary network, which is presumably some combination of co-locational, syntactic, semantic and predicate-argument structure, the details of which depending on what structure is highlighted in the preliminary network.

In a general mathematical setting, the statistical properties of a network are captured by the so-called "partition function", an abstract mathematical structure that is given by the functional integral over all possible network configurations, each configuration weighted by a Boltzmann distribution, assigning a specific probability to that configuration. Given that partition function, many network properties can be obtained, including approximate lexical descriptions of the network (as perturbation theory expansions).[1] Unfortunately, the partition function is unwieldy, verging on impossible to

[1]In machine learning and biological settings, the mechanism of a partition function with a Boltzmann distribution is commonly given the name of a Maximum Entropy Model. This simply refers to the idea that the states of the system are equiprobable (after weighting) and that equiprobable distributions maximize the entropy. This holds as a fundamental theorem, and can be used to define the entropy, or, equivalently, the notion of equiprobability.

compute with numerically, and so a wide variety of approximations to it are commonly used, ranging from Ising models to neural networks.[3]

One particularly common and important task is finding a decompositional, lexical description of the network. In physics, such a decomposition is given in terms of a perturbation theory derived from a Hamiltonian, and the associated Green's functions (propagators) obtained therefrom. In linguistics, such a decomposition is commonly given as a dependency grammar of a natural language. It is useful to recognize that these two descriptions are effectively equivalent; both seek to describe the network structure in terms of local interactions with not-too-distant neighbors, using a tensor algebra to capture connectivity information. Toggling between these two descriptions, one of which assigns weights and probabilities, the other of which invokes the algebraic structure of grammar,[4, 5] can provide considerable insight into the problem of the machine understanding of natural language.

The leading order structure of a network can often be described in terms of the pair-wise interaction of network elements. This is a foundational assumption of perturbation theory. Conversely, simple pair-wise interactions can lead to rich and complex network behaviors; this is the primary lesson taught by the Ising models. Despite their structural simplicity, Ising models still pose a tremendous computational barrier in all but the simplest cases. One common simplifying assumption is to approximate the interactions between pairs of elements in the system by the mutual information (MI) in that network. The MI approximation is sometimes a reasonable approximation to the Ising model, and sometimes less so.[6]

Natural language clearly does not have simplicity of the Ising model; however, one can still pursue an MI approximation thereof.[7] In this approximation, one samples a large corpus of text, and computes word-pair MI. These can be used as the weights of links of an overlay network. In particular, if one discards all links with the lowest MI, one is left with a spanning tree connecting all of the nodes in the network. This is referred to as the Minimum Spanning Tree (MST) or the Maximum Spanning Tree, depending on whether one is maximizing the MI of all of the links, or minimizing minus the MI. Such trees have been applied in a wide variety of settings, from immunoglobulin interactions in zebrafish[6, 8] to natural language.[7, 9] No claim is made here that MST models are the *sine qua non* of unsupervised natural language learning. Rather, they provide a particularly simple and easy preliminary guide to the underlying network structure. This guide is meant to be refined to obtain the "true" syntactic, referential (endophora and diectic), and general semantic structure of text.

Given that such MST models depend on the the mutual information between word-pairs, it is important to gain some reasonable understanding of the structure and distribution of word-pairs arising in natural language. This is performed in the current report, for several datasets obtained from English and Chinese corpora. This report is structured as a diary entry from an ongoing research project, and thus, subsequent sections will document to a large variety of different datasets, how they were constructed, and their status. This documentation is mostly irrelevant to the primary findings of this report; rather, it is a formal record of the progress of ongoing research.

This report does describe multiple important findings and observations. These include the discovery that the distribution of the MI of word-pairs is bi-modal, with one mode, the zero-mode, appearing to correspond with essentially random pairings of words, and another mode capturing the true colocational quality of natural language. These two different modes can be seen to manifest in several different ways, in multiple different statistical quantities, and thus can be fairly easily identified. From this easy identification follows the idea that the two modes can be separated or cut-apart, with the zero-mode discarded, as it appears to be just "noise" in the data. Several different cuts are explored, to see what they do to the data.

A hypothesis is then advanced: after discarding the zero model, the overlay network provided by the minimum spanning tree will be more accurate, or at least, more representative of the syntactic structure of natural language. This hypothesis is not tested here; it is the subject of ongoing research.

To facilitate that research, the idea of a Maximum Planar Graph (MPG) is introduced. The MPG is constructed by starting the MST tree, and then adding additional edges, for progressively lower MI, until a complete, maximal planar graph is obtained. The utility of the MPG parse is that the grammar obtained from it can be filtered after the fact: whenever a root word attaches to a connector with a low MI, that connector can be discarded. This can result in a huge savings of compute time and dataset management effort: rather than recomputing MST parses over and over again, with different data cuts applied, a single MPG parse can be computed once, with data cuts applied at later stages.

Finally, it is pointed out that relatively little is known (by this author, at least) about the distribution of MI in random Zipfian (scale-free) networks. An earlier 2009 paper[10] had begun an exploration of the distribution of MI in random networks, comparing it to the distribution of MI found in natural language texts. It provides some initial guidance for how to tell apart "true" natural language from a random network; unfortunately, it leaves too many questions open. Carefully characterizing random networks, and how they differ from natural language networks remains an open, important task.

The general organizational structure of this report is chronological. Again: it is primarily a diary of research results, with a fair amount of interpretation and description added, so that it can be read by a wider audience.

Definition of Mutual Information

The mutual information of word pairs has a non-obvious form. This is because a pair is already structured: one word necessarily precedes the other. Thus, one must use a definition that acknowledges that structure. For a word pair (w_l, w_r) consisting of a word w_l on the left, and a word w_r on the right, the MI is defined as

$$MI(w_l, w_r) = \log_2 \frac{N(w_l, w_r) N(*,*)}{N(w_l, *) N(*, w_r)}$$

where $N(w_l, w_r)$ is the observation count: the number of times that particular word pair was observed in text. The other quantities are wild-card sums: $N(w_l, *) = \sum_{w_r} N(w_l, w_r)$ and so on. The correctness of this definition, and it's precise interpretation and meaning is not given here; the reader is encouraged to consult other texts for this.

Word pairs can be observed in several different ways.In what follows, two of these ways are referred to as the "clique counting" method, and the "random planar tree" method. The "clique counting" method considers a window of some fixed size, and then counts all possible word-pairings within that window. The random planar tree method constructs a random planar tree connecting all of the words in a given sentence, and then counts only those pairs appearing in that tree. It is expected that these two methods give similar results, however this has not been characterized, and the true differences between these methods remains unknown. Almost all of the following results are obtained with the random-planar-tree method.

Distribution of Mutual Information – 31 May 2015

A preliminary look at the distribution of mutual information in large English language datasets was performed by Rohit Shinde, Google Summer-of-Code student, in 2015. These results are recorded in this section.[2] These are based on the EN_PAIRS_ROHIT dataset, which contained about 400K words (exactly 396255 words), between which there were about 8.9M word-pairs observed (exactly 8880914 of them), for which there were about 418M observations (exactly 418235277 of them). These were obtained by generating random planar graphs for English-language Wikipedia articles.

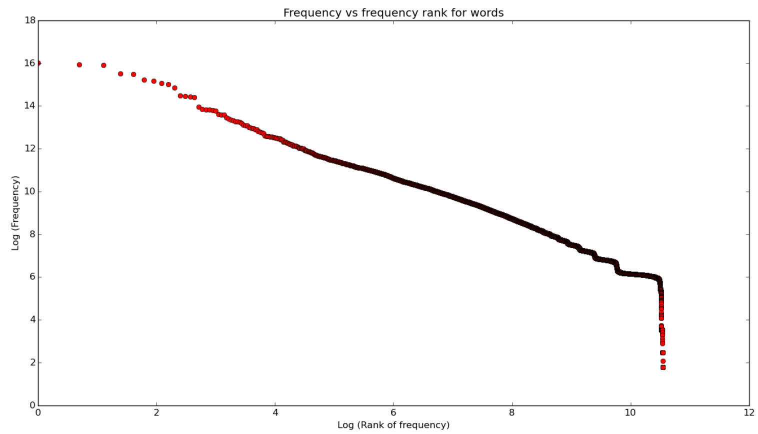
As a "balanced English-language corpus", Wikipedia articles have several deficiencies:

- A near-complete absence of action verbs (run, jump, hit, ...) and a vast surplus of ontological descriptions (was, is, has, ...)
- An unusually high proportion of proper nouns (geographical place names, commercial product names, names of famous persons, ...)

For these reasons, this dataset is perhaps less than ideal for studying the English language, although in most ways it should still be generally representative.

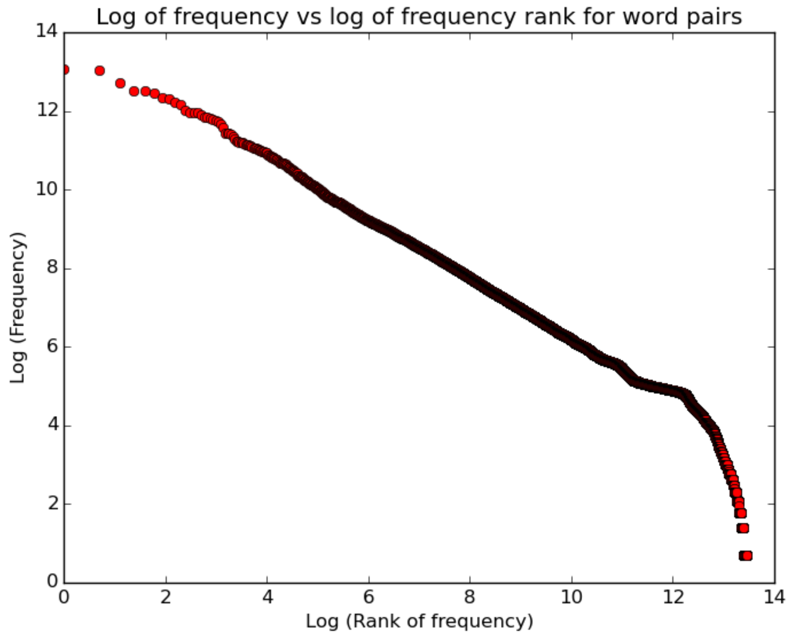
First graph: word observation frequency vs. frequency rank for single words.

[2] Rohit's scripts for generating the plots were lost. The datasets were EN_PAIRS_2016 and EN_PAIRS_ROHIT; see the `notes` file for a detailed description of the contents of this dataset. In short: these hold 8.9M word-pairs obtained from random planar parses of Wikipedia articles. Archived in the `data/sql-dumps` directory.



Frequency is the number of times the word was observed. Eyeballing this, it appears to be a classic Zipfian distribution. The log is the natural log. There's something not entirely right with this graph/dataset: given that number of words is 396255 and that $\log(396255) = 12.89$, the x-axis of the graph should have extended a bit farther. The reason for the premature drop-off is unknown. Perhaps the graph was generated early in the data-collection process, before the complete set of data had been assembled. It does seem to be exactly Zipfian, with an exponent of what seems to be exactly 1.0, so this seems to be in good shape.

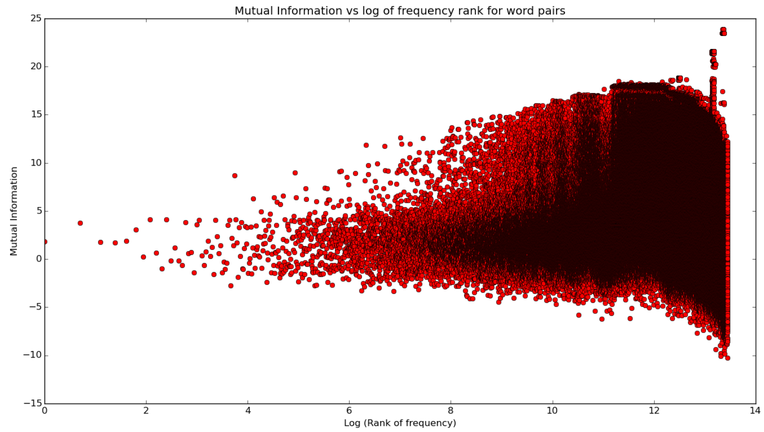
Next, the distribution for word-pairs.



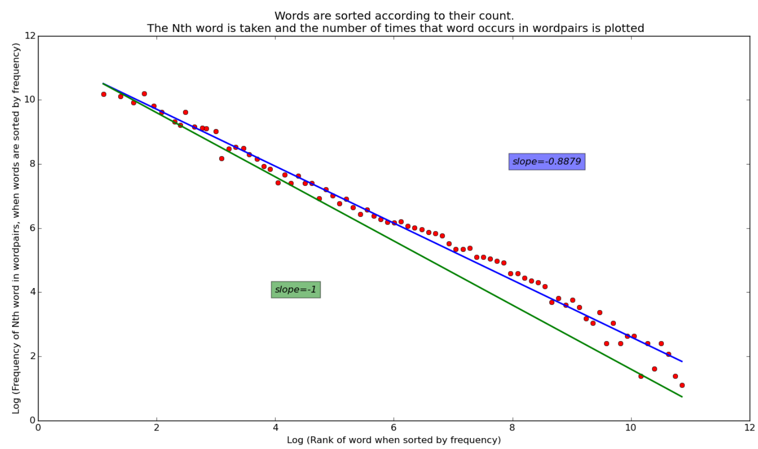
The final dataset had 8880914 word-pairs, and $\log(8880914) = 16.00$ and so again, this appears to be graphing only a preliminary, earlier version of the final dataset. Again, it has a Zipfian distribution, although now the exponent seems to be 0.7 and not 1.0 (just eye-balling it).

Below is a scatter-plot of the mutual information of word-pairs vs. the rank. The distribution seems to be bi-modal: there seem to be two contributions. One is an isosceles triangle (or Gaussian?), centered on an MI=0; then another, second distribution, containing only lower-ranked word pairs, having an MI centered on 10 or so. This suggests an interesting hypothesis: the larger mode, centered at MI=0, captures an essentially random distribution of word-pairs in text. The second, smaller high-MI mode captures collocations: idioms, institutional phrases, set phrases, etc. The bi-modality suggests several ways of arranging data-cuts to separate these two modes. One cut would be to simply discard all word-pairs with an MI of about 5 (again, just eye-balling this particular graph). A completely different cut would be to discard the $e^6 = 403$ top-ranked word-pairs. This cut doesn't separate the two modes, but does decapitate the zero-mode.

The core question is, of course, how do these two modes contribute to grammatical structure. That is, does the zero-mode contribute to grammar, or not? Is the zero mode "just random noise", or does it contain signal? Perhaps its the other way around: the collocations should be treated as "single words", whereas all "true grammar" is to be found in the zero mode.



The final graph shows the number of word pairs, as function of single-word rank. It appears Zipfian.



Summary: the first two, and the last graph show a classic Zipfian distribution, which is exactly what is expected. The third graph appears to be bimodal, and suggests the presence of a mode with “true collocations” (high MI) and a dominant mode of random word-pair associations (centered on an MI of zero.) It is not clear if that bi-modality is actually present in the data, or is merely a trick of the eye. If it is present, it is not clear how these two modes contribute to grammar. The presence (or absence) of bi-modality, and it’s effect on grammar seems to be an important question to pursue.

Word-pair Asymmetry

This section reports on the asymmetry of the distribution of word-pairs. Some words appear more often at the beginning of sentences (and thus, appear more often on the left-hand-side of a word-pair) while others appear more often at the end (the period ending a sentence, treated as a word, is dominant on the right-hand side of pairs).

Left-right asymmetry - Definition

Define the relation $E(w_1, w_2)$ as being the relation that both words w_1 and w_2 occur at the ends of an edge in the same sentence, but in arbitrary order. It is symmetric: $E(w_1, w_2) = E(w_2, w_1)$.

Given n words, define the observation count of a relation $R(w_1, w_2, \cdots, w_n)$ between them as $N(R, w_1, w_2, \cdots, w_n)$. Use the symbol A to represent the ordered pair (w_1, w_2); that is, the relation is $A(w_1, w_2)$.

By these definitions, one has that

$$N(E, w_1, w_2) = N(A, w_1, w_2) + N(A, w_2, w_1)$$

This is the symmetrized count. It is useful to mod out one of the two words, and to consider the sum

$$N(E, w) \equiv N(E, w, *) = \sum_{w_2} N(E, w, w_2)$$

This counts how often the word w occurs at one end or the other of a word-pair. It is a distinct count from $N(w)$, which, by definition, counts only once per word in a sentence.

Left-right asymmetry - Results

This section explores how often a given word occurs on the left side of a word pair, *vs.* how often it occurs on the right. This, of course, depends on the word. If sentences and words were randomly generated, one would expect that a given word would occur on the left, or on the right, exactly half the time. That is, in a random world, one expects

$$N(w, *) \approx N(*, w)$$

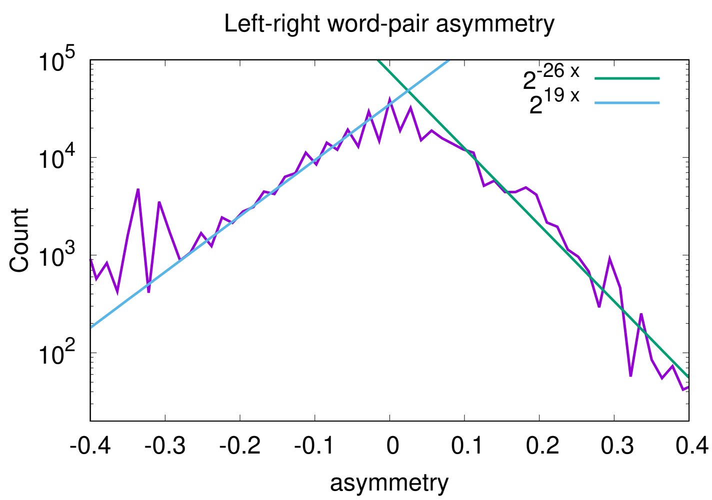
This cannot hold for human languages: the exclamation point, question mark and period occur exclusively on the right hand-side of any randomly-generated pair. This is not limited to punctuation: for Japanese, sentences usually end in a verb, and thus, for Japanese verbs v, one expects that $N(v, *) \ll N(*, v)$.

Some typical values for English words are given in the table below. Here, by definition, $2N(w) \equiv N(*, w) + N(*, w)$ is (twice) the number of times the word w is observed in a pair relation. The word-pairs were generated by creating random planar parse trees of sentences in the corpus, and then counting a pair, if two words are connected by a parse link.[3]

[3] This is the dataset collected by Rohit, summer of 2015, described above.

w	$N(*,w) - N(w,*)/2N(w)$
how	-0.0058
when	-0.0051
,	-0.0021
will	-0.00114
usually	-0.00112
a	-0.00085
the	0.0035
finally	0.0043
word	0.0094
hope	0.0128
?	0.3197
.	0.8578

A histogram summarizing the above table is shown below. It bin-counts the asymmetry $N(*,w) - N(w,*)/2N(w)$. About 100 bins are used, from an observation of 8.88 million distinct English-language word pairs, where were observed for a total of 420 million times. These pairs connected about 400K distinct words. The number of distinct words is large, because these include surnames, and geographical names, as well as an assortment of foreign-language words as might be encountered in a sampling of English Wikipedia pages.



Observe that the y-axis is drawn with a logarithmic scale. The two sides to this peak are conjectured to be linear. Two lines guessing at the slopes are indicated in the graph; the (natural logarithm) slopes are +13 and -18 (or +19 and -26 base two).

English Dataset Sample - 28 April 2017

This section reports on data collected for a small sample of English sentences, - taken from Wikipedia, late April 2017.[4] It was collected over the course of a few days, and is a medium-sized sample: larger datasets, collected over weeks or months, are possible, as well as smaller samples collected over a few hours.[5]

There were 106696 unique words observed in the dataset[6]. This number is fairly large, as it includes not only common nouns, but also surnames, geographical location names, and a variety of foreign-language words, as would be observed in typical Wikipedia articles. These words were observed for a total of 24417409 times[7].

A total of 80613 sentences were observed[8] with 15.88 parses per sentence[9]. On average, there were 19.07 words per sentence[10].

Word-pairs were obtained via 'clique counting': given a window size of 6, all-possible word-pairs were considered (a graph clique) and each such word-pair was counted once. The window is then shifted over by one word, and the process is repeated. This means that, on average, word-pairs in the middle of the sentence are counted six times. There were 9376710 (about 9M) unique 'clique pairs' observed[11] for a total of XXX observations.[12]

This dataset is not used in any further experiments; effectively all of the major datasets and experiments are built on counts obtained from random planar graph parses, rather than from clique-pair counts.

Word-pair dataset report 3 June 2017

Some summary report of various different datasets holding word-pairs. All of the word-pair-counts were obtained by using the LG "ANY" parser, which generates random planar graphs (random planar parse trees). Many of these datasets refer to "MST", and that is because those datasets *also* contain disjuncts (Sections) from MST parses. This additional data has no effect on the word-pairs; it just makes the datasets fatter.

The common procedure used during this era was to take a tranche of text, perform random planar-graph parsing, and count the resulting word-pairs. Next, marginal statistics and word-pair MI is computed. Then the same tranche of text was run through the

[4]The 'EN_SNAPSHOT' dataset

[5]Note that the AtomSpace infrastructure does have some serious performance limitations: the AtomSpace is designed to be a very general-purpose hyper-graph manipulation and analysis tool, and not a fast statistics-collecting tool for narrow tasks. One could probably write custom word-pair-counting software that would run an order of magnitude faster than the AtomSpace. But doing so would just lose the flexibility over later, more complex analysis.

[6]Obtained with `(fetch-all-words)(length (get-all-words))`

[7]`(get-total-atom-count (get-all-words))` which is the same as `(total-word-observations)`

[8]`(get-sentence-count)`

[9]`(/ (get-parse-count) (get-sentence-count))`

[10]`(avg-sentence-length)`

[11]`(fetch-clique-pairs) (length (get-all-clique-pairs))`

[12]`(get-total-atom-count (get-all-clique-pairs))`

MST parser, and so disjunct stats are added. Then the next, and the next tranche are processed (up to a total of five distinct tranches, described below.)

There were several problems encountered:

- Early datasets include as "words" character sequences that contain infix punctuation, typically because the input text used hyphens, dashes, double-dashes and commas to delimit words, but did not put spaces before or after the punctuation. The tokenizer was to "stupid" to recognize these, and treated them as single words. Some of the datasets filter these out, after-the-fact.
- Same as above, with with sentence-ending periods. These can be recognized as "words" that have an embedded period, followed by a capital letter. None of the datasets have these filtered out.
- The MST parser was (mildly) buggy; when a sentence contained multiple instances of the same word, the generated disjuncts were incorrect. See bug `opencog/atomspace#2252` for the fix. This does not affect word-pair data, but does produce low-quality disjuncts that contribute to the overall noise.
- The inclusion of the MST data makes the datasets large and slow to load.

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
619K x 581K	27.9M	1.25G	44.7	18.65	1.80	EN_PAIRS_CFIVE_MST
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
351K x 351K	14.6M	632M	43.4	19.35	3.37	ZEN_PAIRS_THREE

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 word-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 mis-handled.

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.

EN_PAIRS_RFIVE_MTWO The MST parses of tranches 1-2, performed on the word-pairs computed from EN_PAIRS_RFIVE. That is, the word-pair stats for tranches 1-5 were accumulated to completion first, before the MST parsing is started.

EN_PAIRS_RFIVE_MST As above, except that this is the MST parse of all of the tranches 1-5. That is, the MST corpus is the same corpus as the word-pair corpus. (Same as EN_PAIRS_RFIVE_MFIVE, but with dj marginals)

EN_PAIRS_CFIVE_MST Same as EN_PAIRS_RFIVE_MST above, but with all words that contained bogus infix punctuation removed. Hyphenated words remain, as do decimal numbers and abbreviations. Unfortunately, this means that there are some bogus words that remain - sentence boundaries were not detected when there was no space after a period, and before the start of the next sentence. These are characteristically words with a period in them, followed by a capital letter. These should have been filtered out, but weren't.

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).

ZEN_PAIRS_THREE Word-pairs for tranche-1 and tranche-2-part-1 of Mandarin novels, segmentation 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 ZH 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.

By contrast, the ZEN_PAIRS dataset, where the Mandarin was pre-segmented into words by 3rd-party tools, behaves much more like English in it's statistics. This is also evident in the table below, where the ZEN dataset behaves like EN, and not like ZH.

Vector Norms

The rows and columns of any matrix can be treated as vectors; any vector can be given an ℓ_p-norm as

$$\ell_p = \left(\sum_k |a_k|^p \right)^{1/p}$$

with $\{a_k\}$ being the vector elements. The following table reports on three for these norms, for $p = 0$, 1 and 2, respectively called the support, count and length.

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
351K	351K	28.8K	27.1K	239	203	22.9	19.8	ZEN_PAIRS_THREE
619K	581K	53.8K	73.1K	282	254	34.7	25.9	EN_PAIRS_CFIVE_MST
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.

English word-pair small dataset July 2017

This report provides a quick sketch of a small dataset containing English word-pairs. This is the EN_PAIRS_RONE dataset described in section above. To recap, its this one:

Size	Pairs	Obs'ns	Obs/pr	Entropy	MI	Dataset
134K x 135K	5.54M	174M	31.4	17.67	1.94	EN_PAIRS_RONE_MST

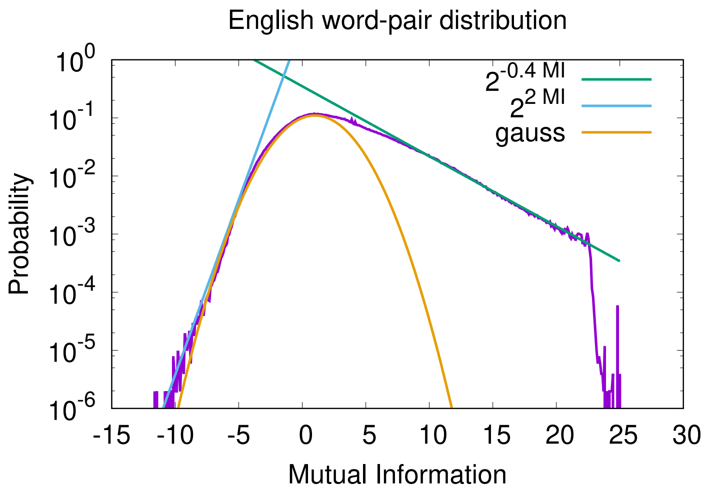
The size and support is recapped here below, just copied from section above. The explanation of the column labels can be found there.

Size		Support		Count		Length		Dataset
L	R	L	R	L	R	L	R	Name
134K	135K	17.4K	17.4K	143	129	16.6	14.0	EN_PAIRS_RONE

The dataset contains 5544578 pairs.

Distribution of Mutual Information

The figure below shows the distribution of the mutual information of English word-pairs.



The peak of the distribution occurs near an MI=1.0. The two straight lines are eyeballed to fit the bimodal distribution. The "meaningful" mode, with positive MI, has a slope of -0.4. The negative-MI mode has a slope of 2.[13] Note that this is quantitatively (and not just qualitatively) similar to the Chinese hanzi pairs distribution, shown below. Both have the same shape, and the slopes are only a little bit different, and the peak is slightly shifted.

Recall that a normal distribution takes the form of a parabola, when graphed on semi-log plot such as this. The parabola, here labeled "gauss", is given by the normal distribution $P = 0.11 \exp -0.1 (MI - 1)^2$. This is an "eyeballed" fit (it looks good, by eye), and not a precision numerical fit. As noted in an earlier experiment[10], a Gaussian distribution is symptomatic of randomly generated word-pairs: the low-MI word-pairs are effectively just noise, the completely random arrangement of nearby words. The "meaningful" distribution - the excess of pairs with large positive MI are due to "true collocations".

The RMS variation is given by σ where $0.1 = 1/2\sigma^2$; solving, one finds that $\sigma = \sqrt{5} \approx 2.2$. The entropy of a normal distribution is given by $H = 0.5 \log_2 \left(2\pi e \sigma^2\right)$. Plugging in, one has $H \approx 3.2$bits for the zero-mode distribution.

Interpretation of the zero-mode

The Gaussian in the above figure is slightly offset to the right, centered on an MI=1 rather than MI=0. As noted in[10], such an offset to positive MI is characteristic of an *incomplete* sampling of random text. That is, if one had a vocabulary of N words, there are N^2 possible word-pairs, each of which one might expect to see equally often, in the long-run. However, if one samples randomly generated sentences, and these sentences contain less than about N^2 words amongst all of them, then some pairs will never be

[13]Graph obtained from the `binned-enpr-mi` data, in the `en-pairs.scm` file.

observed. The resulting MI distribution will be a Gaussian, but it will be centered on a positive value of MI.

This should be compared to the distribution given below, for hanzi pairs. The effective hanzi vocabulary is much much smaller; about N =7K, as noted for ZH_PAIRS_SONE. This vocabulary size allows for N^2 =50M word-pairs to be observed, in principle, if they were randomly arranged; in practice 6M were observed. This is a much "deeper" sample of all possible word-pairs than the English corpora allow; the English vocabulary is orders of magnitude larger.

The experiments with randomly-generated corpora, given in [10] are interesting, but incomplete. They leave open multiple questions:

- How does the MI distribution change when random planar tree parses are used to obtain word-pairs, as opposed to clique-counting?
- How does the MI distribution change as a function of sample size (holding the vocabulary size fixed)?
- How does the MI distribution change as frequency of vocabulary words is altered from a uniform distribution to a Zipfian distribution?

The last question is perhaps the most relevant: it implies that there is no finite-sized vocabulary; that all vocabularies are infinite. This is exactly what is seen, as more and more text is analyzed. A survey of the words in the *en_cfive* dataset shows more than a few (ancient) Greek words, written in the Greek alphabet. These are clearly "not English", but they did appear in some English text, somewhere in the collection of corpora. They are rare; they necessarily can participate in only a small number of word-pairs. Yet most of the words in the dataset are like this - if not Greek, then Latin or Spanish or German (I guess there are travelogues amongst the corpora), and plenty of given names, surnames, geographical place names and the like. Each is increasingly more rare than the last, but the grand-total bulk of them is quite large. How does this large bulk of infrequent, unlikely word-pairs affect the MI distribution? How does it manifest itself?

Chinese character pair dataset July 2017

This report provides a quick sketch of a dataset containing Mandarin Chinese character pairs. This differs from English in two important ways. First, obviously, its not English. Second, there was no word segmentation done: each character (hanzi, ideogram) is treated as being distinct, and so all pairs are between hanzi. The goal/hope here is that word segmentation will appear "naturally", as a by-product of high-MI hanzi pairs. The dataset is the ZH_PAIRS_SONE dataset described in section . To recap, its this one:

Size	Pairs	Obs'ns	Obs/pr	Entropy	MI	Dataset
158K x 159K	5.92M	729M	123	18.45	2.02	zh_pairs_sone

The size and support is recapped here below, just copied from section . The explanation of the column labels can be found there.

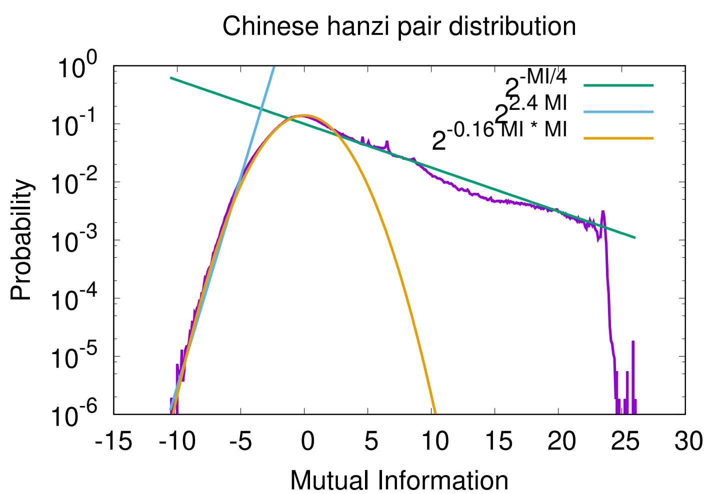
Size		Support		Count		Length		Dataset Name
L	R	L	R	L	R	L	R	
158K	159K	6819	6411	548	487	41.7	37.7	zh_pairs_sone

As mentioned before: the dimensions of the dataset are larger than the number of hanzi, because the dataset treats all Latin-alphabet words as single words. Since this dataset is generated from Wikipedia, we can expect that many of the entries correspond to English-language technical terms and named entities, such as product names, geographical place names and the names of people. There is also likely to be a mixture of simplified and traditional hanzi in the dataset.

The dataset contains exactly 5922477 pairs.

Distribution of Mutual Information

The figure below shows the distribution of the mutual information of the hanzi pairs.



The peak of the distribution occurs near an MI=-0.25. The two straight lines are eyeballed to fit the bimodal distribution. The "meaningful" mode, with positive MI, has a slope of -0.25. The negative-MI mode has a slope of 2.4.[14] Note that this is qualitatively similar to the English word-pairs distribution, shown above, although the slopes are different, and the peak is slightly shifted.

Recall that a Gaussian distribution takes the form of a parabola, when graphed on semi-log plot such as this. In this case, the parabola is centered exactly on zero, with an eye-balled fit as indicated. This implies that the RMS variance σ is given by

[14] Graph obtained from the `binned-hanpr-mi` data, in the `zh-pairs.scm` file.

$0.16 \log 2 = 1/2\sigma^2$. Solving, this gives $\sigma = \sqrt{1/(0.32 \log 2)} \approx \sqrt{4.5} \approx 2.1$. As noted before, the Gaussian centered is symptomatic of a random distribution of pairs: the low-MI hanzi pairs are effectively just noise, completely random groupings.

English CFive Word-pair dataset - July 2019

This section characterizes the **en_pairs_cfive_mi** dataset. This is the primary English word-pair dataset, previously computed in 2017 and described above. Despite having a few problems, it appears to be clean enough to form a foundation for ongoing work, and thus is worth characterizing more closely. This is done in this and the following sections.

The starting point is the **en_pairs_cfive_mst** dataset with the disjuncts (Sections) deleted, and word-pair MI and marginals freshly recomputed.[15] The dataset is summarized again, in the following table.

Size	Pairs	Obs'ns	Obs/pr	Entropy	MI	Dataset
619K x 581K	27.9M	1.25G	44.7	18.65	1.80	en_pairs_cfive

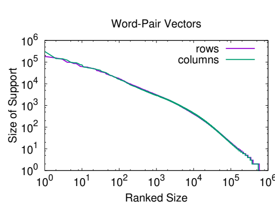
Vector Norms

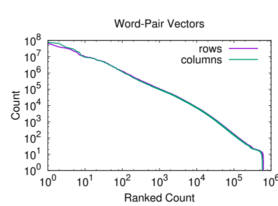
The rows and columns of the matrix can be treated as vectors; the norms are given in the next table. Definitions are as before.

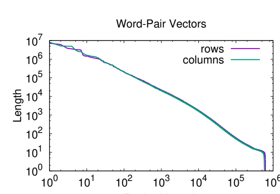
Size		Support		Count		Length		Dataset
L	R	L	R	L	R	L	R	Name
619K	581K	73.1K	53.8K	254	282	25.9	34.7	en_pairs_cfive

One impediment to properly understanding these numbers is that the distribution of each of them is approximately Zipfian. That is, when the rows (or columns) are treated as vectors, and are then ranked according to the size of the support, these follow a Zipfian distribution. This can be clearly seen in the figures below. The primary point of confusion is that the notion of "average" is delicate ... one should say "inappropriate", for true Zipfian distributions. The exact meaning of the word "average" is explored in the following graphs.

First, one can observe the explicitly Zipfian distribution in each of these three vector norms.

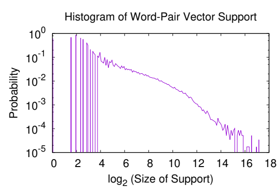


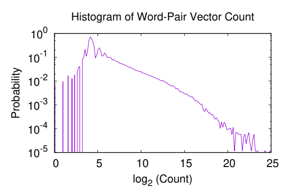


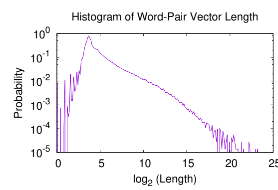


[15]There was no particular need to recompute the MI and the marginals; mostly this was done to verify that the earlier computations were still good. They were; results seem to be identical.

The support, count and length are the ℓ_0, ℓ_1 and ℓ_2 vector norms, for the columns and rows of the pair-matrix (these norms are exactly as defined in an earlier table above). All of the graphs have a bit of a hump in the middle, with the hump being most prominent in the support distribution. The corresponding histograms are shown below. These are to be understood as follows. There are precisely 580804 columns, and thus that many column vectors. The log (base 2) of the support (respectively, count and length) of each vector is taken; the vector is then assigned to one of 200 bins, according to that log-value. The final tally of the number of vectors in each bin is shown in the graphs below. Properly, this can be labeled as a probability; simply by multiplying by the bin-width, and dividing by the total number of vectors. Thus, the total area under the curve is exactly 1.0. Do not be deceived by the spiky areas on the left: these are very full bins next to empty bins. With a bit of smoothing, these would get smoothed down; but no smoothing was applied (other than the natural smoothing of the finite bin-width.)

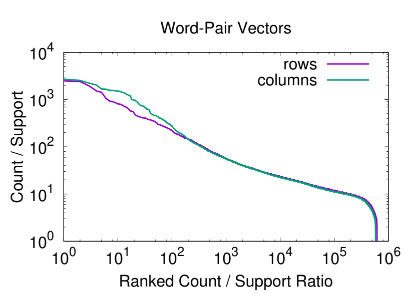


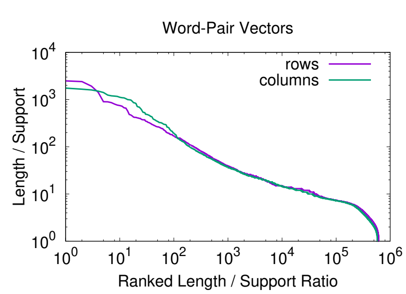




These histograms help "explain" the not-quite straight-line Zipfian dependence in the earlier graphs. A true Zipf distribution would not demonstrate peaks at a finite size. Consider, for example, the vector length distribution. Eyeballing the rightmost graph, it appears that there is a peak at about $2^4 = 16$, so that "most" vectors have a length of about 16. There are many vectors longer than this, but not so many shorter than this. The correct interpretation for this is that most words have $2^4 = 16$ nearby neighbors, or more; very few words are so strongly tied into idioms or collocations that they have a paucity of neighbors, a paucity of context. This should be contrasted with vectors having very large support or length: these correspond to words that appear next to just about any other word. That is, for a fixed word w_r, the support counts the number of distinct pairs (w_l, w_r); a large support means that there are many words w_l the appeared near w_r; a small support means that very few were. The count (the ℓ_1-norm) just counts how many such word-pair observations were made (*i.e.* with multiplicity).

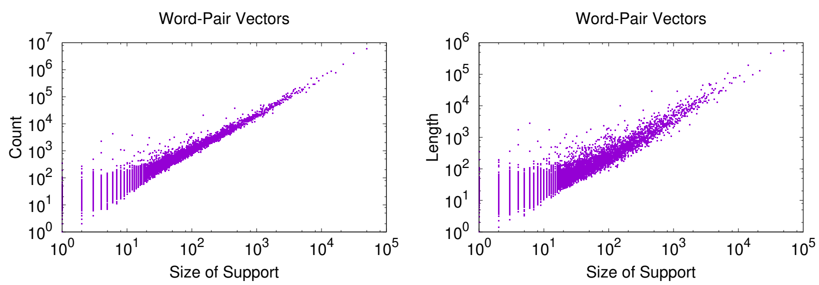
One might wonder if the ratio of count and length to support is Zipfian; it is; this is shown below.





Presumably, the prominent dip in the middle is due to the hump visible in the support distribution. One can still see that the distribution is roughly Zipfian.

One might wonder just how strongly the ℓ_0, ℓ_1 and ℓ_2 vector norms correlate with each other; that is, whether the ratio taken above was justified. It does seem to be, as the following graphs show.

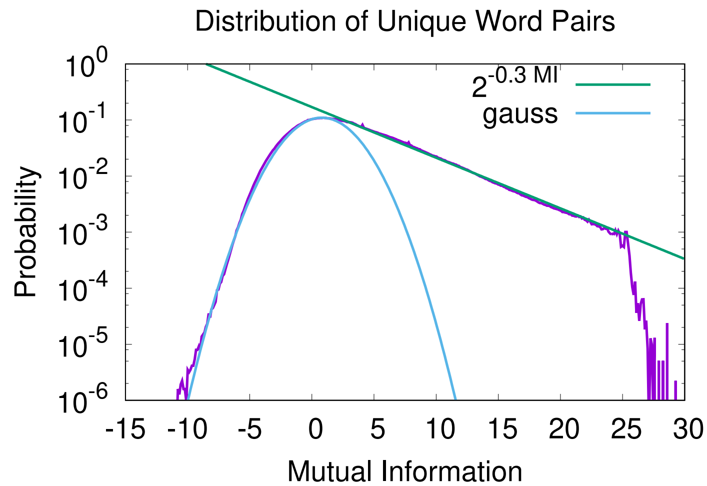


These are scatter-plots of 581K column vectors, showing the scatter of ℓ_0 vs. ℓ_1 and of ℓ_0 vs. ℓ_2, respectively. To keep the graphs relatively small and not overly busy, only every 25th point is graphed (so only 581K/25= approx 23K points are actually shown). Keep in mind that the projection of these scatter-plots along the horizontal axis gives rise to the histograms presented earlier. The linear lower bound apparent in both figures is simply a statement that the ℓ_1 and ℓ_2 norms cannot be less than the ℓ_0-norm. That is, a given word-pair cannot be observed less than once. The large variation at low support is an alternative indication for a fair number of collocations in the data. When a column-vector (the vector for the fixed word w_r) has low support, that means that only a few distinct pairs (w_l, w_r) were seen. But if that same vector has a large length, that means that this handful of pairs was seen many, many times. These are collocations: words that appear near each other. Of course, one expects collocations in textual data. These graphs characterize just how many actually show up, when one does unbiased, random counting.

MI Distributions

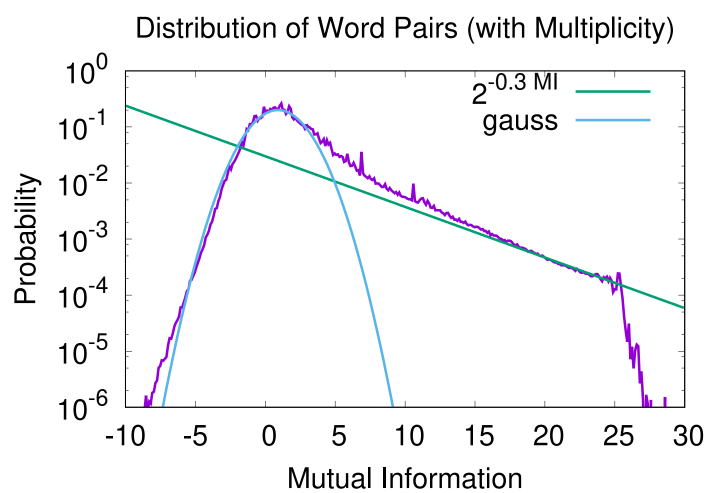
More interesting than the vector norms is the MI distribution, as this has the most immediate effect on the results of MST parsing. The next graph shows bin-counts of the 28M unique word-pairs, sorted into 400 bins.[16] The distribution is both qualitatively and quantitatively similar to the distributions previously shown, for other datasets.

[16] Scripts are in '`word-pairs.scm`' and '`cfive.gplot`'.

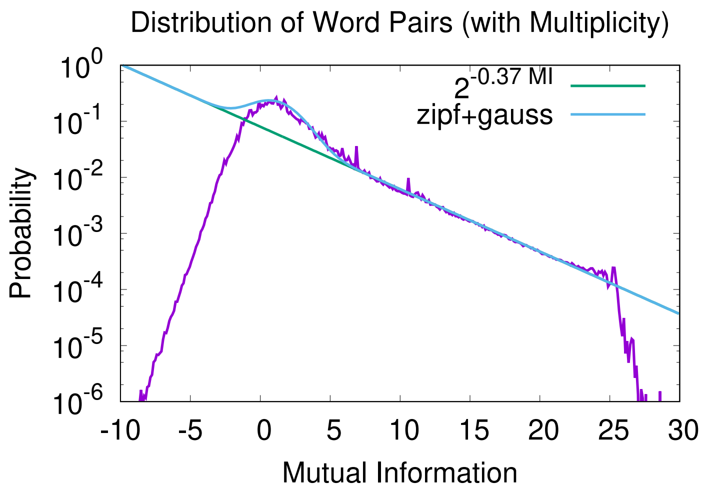


The normal distribution in the graph above is an eyeballed-fit, and is given by $P(MI) = 0.11\exp\left(-0.1(MI-0.8)^2\right)$ which is very similar to the distributions given earlier. The RMS variation is given by σ where $0.1 = 1/2\sigma^2$; solving, one finds that $\sigma = \sqrt{5} \approx 2.2$, exactly the same as that given before, for the EN_PAIRS_RONE dataset, and only a little bit different from the $\sigma \approx 2.1$ seen in the Mandarin hanzi-pair dataset. This Gaussian appears to be a generic feature in natural-language word-pairs.

The above was a histogram of the unique word-pairs; below is the same histogram, this time counted with multiplicity. That is, if a word-pair was observed N times, then it contributes N to the histogram bin to which it is assigned.



The Zipf distribution is drawn with the same slope as before; this is done to allow direct comparison. A Gaussian is drawn, but this time, the left edge appears more linear than parabolic. The Gaussian is, again, an eye-ball fit, given by $P(MI) = 0.2\exp\left(-0.18\,(MI-0.9)^2\right)$. Again, this is an eyeball fit; the difference in the mean should be considered as an indicator of the uncertainty in the mean. The difference in the variance is quite real. This time, the distribution appears to be more obviously a sum of two distinct distributions. This is shown more clearly in the figure below.



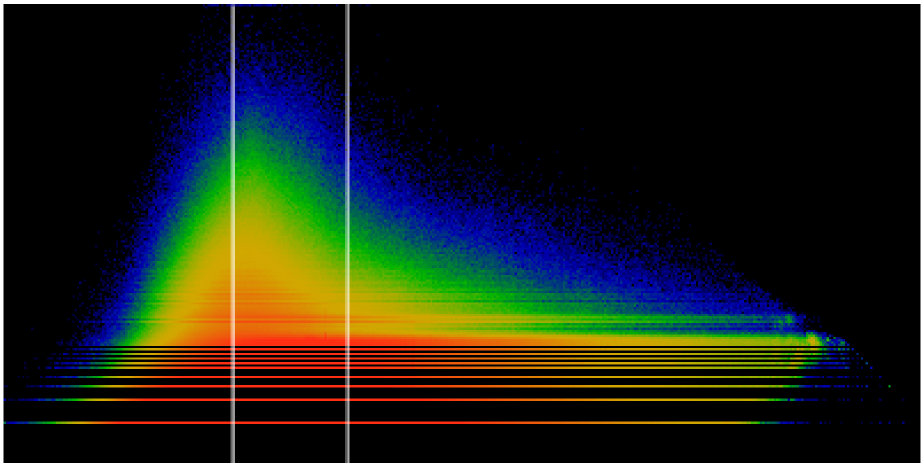
The curve labeled "zipf+gauss" is a sum of the two distributions; note that the slope of the Zipfian has been adjusted to provide a better fit. Again, the parameters are obtained by eyeballing; the curve is given by

$$0.08\times 2^{-0.37MI}+0.17\exp\left(-0.18\,(MI-0.9)^2\right)$$

The Gaussian has the same shape as before; it's strength was reduced to better fit the data. Again, the left edge of the data appears more linear than parabolic; perhaps the correct fit would be some blending of of two semi-log-linear distributions. Without a robust theoretical foundation, its uncertain just how far these fits can be pushed.

MI vs. Frequency

A final bit of insight can be gained by examining the MI vs. frequency scatter-plot.



This image, 400 pixels by 200 pixels, is a sequence of histograms. The image extends, left to right, from MI=-10 to MI=+30, that is, spanning exactly the same MI range as the earlier histograms, as well as having the same bin-width. The bottom-most horizontal line corresponds to (unique) word-pairs that were observed only once; the second-from-the-bottom horizontal line word-pairs that were observed twice, and so on by frequency. The vertical axis is logarithmic, so the lines soon run together, once the log-frequency is less than a pixel in size (with some aliasing effects at first).[17] The color coding, from black to red, indicates how many word-pairs were observed that can be assigned to a single pixel. The color scale is logarithmic, so that red-pixels have bin-counts that are orders of magnitude larger than blue pixels. The integral along a vertical line then yields a single bin-count for single MI-range, independent of the observation frequency. The counts are for unique word-pairs, and not word-pairs with multiplicity.

The scatter-plot provides a bit more insight into the bimodal distribution. As noted before, many of the most-frequent word-pairs (pixels in the upper half of the image) are also the ones with the lowest MI (pixels to the left of the image). Indeed, the prominent peak is located at just above MI=0; the left-most vertical white line is positioned on MI=0; the one on the right is at MI=5.

Cutting distributions - July 2019

That there is a bimodal distribution seems clearly established, no matter how one looks at the data. It also seems clear that one of the modes, termed the zero-mode, corresponds to completely random word-pair choices. Since MST parses and the extracted disjuncts are based on the MI of individual word-pairs, it seems reasonable that their quality might be improved by cutting away those word-pairs corresponding to random noise, and keeping only those arising from "true" word collocations.

[17] The vertical axis runs from $-\log p = 32$ at the bottom to $-\log_2 p = 12$ at the top. The pixels are effectively "square". The first horizontal lines is at $-\log_2 8.016 \times 10^{-10} = 30.21$.

There are several ways in which such cuts might be made. One can trim away all word pairs having an MI below some threshold. One can trim away word pairs having support above some given threshold. Perhaps a cut combining length and support would be appropriate. Each different cut will generate a different word-pair dataset; each different dataset will result in different parses and disjuncts.

The variety of different cuts means that the parameter space needs to be explored. Cutting before parsing means that a lot of CPU time must be devoted to re-parsing the corpora to obtain disjuncts. Fortunately, there is a trick that can be used to reduce the amount of required computations: there is a way to apply the cuts after parsing, rather than before. The trick works as follows.

Rather than parsing to obtain a Maximum Spanning Tree (MST), one can parse to obtain a Maximum Planar Graph (MPG). The MPG parse starts with the MST parse, and then adds edges, one at a time, each having the highest MI of all the others, to obtain the largest possible planar graph. This planar graph can be cut, just as before, and used to obtain disjuncts. The resulting disjuncts are necessarily "bushier" than the MST-derived disjuncts, as there are more edges in the graph.

The utility of the MPG-derived disjuncts is that the data cuts can be applied to these, rather than at an earlier stage. Given an MPG-derived disjunct, one can examine each individual connector, and then decide whether it should be kept or not, based on whether the word-connector MI is too low, or the support is too high, or whichever cut one might wish to explore. This enables just a single computational pass to create the disjunct dataset, and the exploration of different data cuts delayed until afterwards, closer to the later stages used for quality evaluation.

Linguistically, connectors with a low MI value to their root might be treated as optional connectors: if present in a parse, they should be used, but, if absent, that's OK too, as the low MI indicates that no connection is required.

Cutting and recomputing the MI

It is not entirely clear what happens to the data when low-MI word-pairs are discarded. This section takes a closer look at this. In particular, it looks to see just how close the marginal and MI computations are to being idempotent. Thus, the dataset with cuts applied is compared to the same cut dataset, but with the marginals and MI recomputed from scratch, cleanly, to see just how much these might change.

The starting point is the **EN_PAIRS_CFIVE_MI** dataset; this is the **EN_PAIRS_CFIVE_MST** dataset with the disjuncts (Sections) deleted. Starting with this dataset, all word-pairs with an MI of less than 1.8 are discarded, then the MI is recomputed, from scratch, with the remaining pairs. The **EN_PAIRS_MI_1.8CUT** dataset contains both the old, and the new MI values.[18]

The table below summarizes these two datasets. The first row is the same as before.

[18]See the ‘`word-pairs.scm`‘ file for the scripts. The new values can be accessed by using the filter `(add-fmi-filter wps 1.8 #t)` where the `#t` generates the filter-label under which the recomputed marginals can be found.

Size	Pairs	Obs'ns	Obs/pr	Entropy	MI	Dataset
619K x 581K	27.9M	1.25G	44.7	18.65	1.80	EN_PAIRS_CFIVE
619K x 581K	15.8M	470M	29.8	18.59	5.37	EN_PAIRS_MI_1.8CUT

The cut was positioned so that it was located at the average MI of the original dataset. This eliminated slightly less than half of all word-pairs. However, more than half of all observations were eliminated; apparently, as suggested in the Rohit data, some of the very high-frequency pairs have a low MI. Thus, eliminating low-MI pairs appears to eliminate some of the high-frequency pairs. Although the total number of observations drops by more than one-half, the total observations-per-pair drops only by a third. At the same time, the MI goes up, as the location of the peak shifts over, as shown in the graphs below.

The vector norms of the rows and columns are given in the table below.

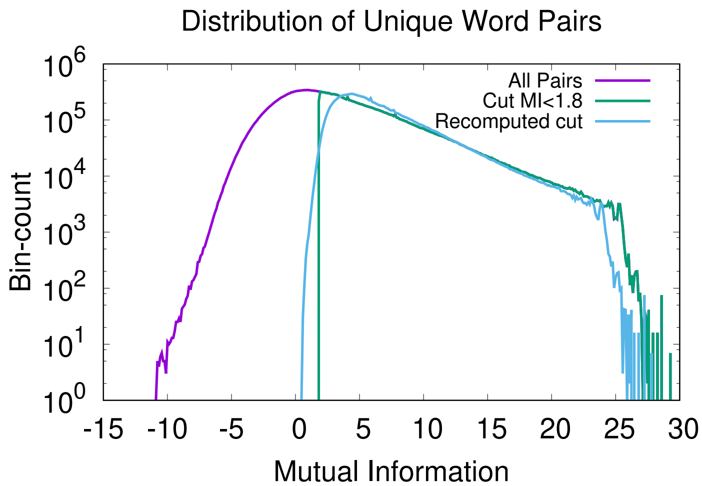
Size		Support		Count		Length		Dataset
L	R	L	R	L	R	L	R	Name
619K	581K	73.1K	53.8K	254	282	25.9	34.7	EN_PAIRS_CFIVE
619K	581K	10.0K	12.5K	138	210	24.2	26.3	EN_PAIRS_MI_1.8CUT

Particularly interesting is the change in the support. The drop is dramatic: something like 5/6ths of the support disappears. This corresponds quite well with the earlier analysis: vectors with a large support correspond to effectively random pairings. Random pairings have a low MI. Thus, when low-MI pairs are cut, then so are the vectors with large support. The count drops by only a third, and the length is very nearly unchanged. This can be understood as saying that the cut eliminates primarily low-frequency pairs. This is particularly dramatic in the length: the dominant contribution to the length comes from high-frequency pairs (sum-of-squares amplifies large values, and makes small values irrelevant). This, the MI cut appears to do not just one, but two desirable things:

- It removes support from high-support vectors, the support of which obviously comes from random associations.
- It removes low-frequency tails, i.e. those word-pairs for which there is little evidence of any meaningful association. Word pairs that just happened to be seen, but not seen often enough to be interpreted as evidence of association.

MI distribution

One might wonder: by discarding the low-MI pairs, how much is the dataset truly altered? The change in support was quite dramatic. The change in the average MI was large. What about the MI distribution - how is that affected? The answer appears to be "not much". This is explicitly shown in the graph below.



The *all-pairs* curve just reproduces the bin-counts previously shown; this time the counts are shown unnormalized.[19] The *cut-mi* curve is exactly the same data, but with the cut plain to see. Lower-MI pairs are just gone. The *recomputed-mi* curve is the interesting one: this takes the cut data, and recomputes the MI from scratch. Interestingly, this re-computation does not alter the distribution very much.

This should not be much of a surprise, perhaps. In an earlier section, the zero-mode was fit with a parabola, centered at MI=0.8 and a standard deviation of $\sigma = \sqrt{5} \approx 2.2$. The cut at MI=1.8 thus corresponds to a cut at $(1.8 - 0.8) \approx 0.45\sigma$. At this cut, the fraction of "random" pairs that were removed is

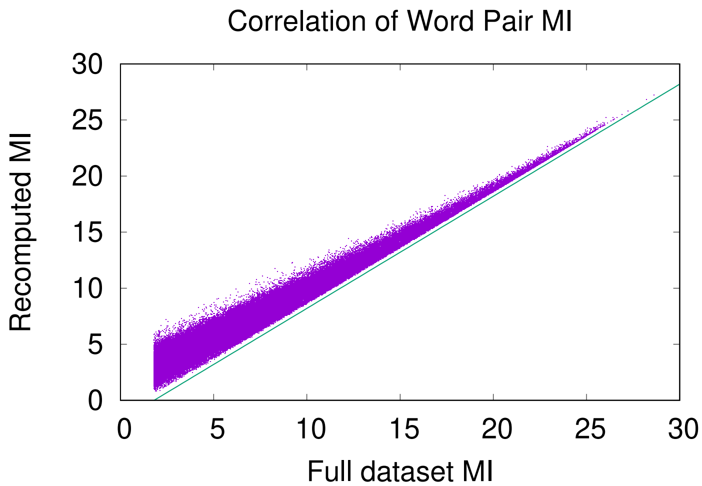
$$CDF = \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{0.45\sigma} e^{-x^2/2\sigma^2} dx \approx 67\%$$

Placing the cut at 2σ would remove 98% of the zero-mode; this cut would be located at MI=5.3 – its well up there, but not ineffectively so. This deeper cut is examined in the next section.

When the MI is recomputed, how are existing pairs actually affected? The figure below is a scatter-plot of old vs. new MI values. The old values are plotted along the x-axis; the sharp cutoff at MI=1.8 is clearly visible.[20] The straight line in the graph is drawn at $(oldMI - 1.8)$. It appears to provide the lower bound for the reassigned MI's. It's clear that pairs that used to have a high MI still have almost the same high MI as before. There is some reshuffling at the lower scores, but not all that much. This provides proof that the reshuffling is local, that the changes are limited in scope.

[19]The bin-count depends not only on the total number of observed pairs, but also on the width of each bin. Here, the interval from -15 to +30 is divided into 400 bins; thus, the width of each bin is 45/400=0.1125. Thus, at the peak there are more than 100K word-pairs having an MI that is within 0.1125 of each-other.

[20]To make the size of the figure manageable, only 200K of the nearly 16M pairs are shown.There's no qualitative change by doing so.



There seems to be a slight gap between lower bound, and the distribution. The explanation for this gap is unknown.

A deeper cut

The above process was repeated, taking a deeper cut, located at MI=5.3 in the original dataset. This dataset, EN_PAIRS_5.3_REMI contains only those pairs which originally had an MI>5.3. After discarding those pairs, the MI and marginals are recomputed. Results are presented below; the earlier datasets are repeated for easier comparison.

Size	Pairs	Obs'ns	Obs/pr	Entropy	MI	Dataset
619K x 581K	27.9M	1.25G	44.7	18.65	1.80	EN_PAIRS_CFIVE
619K x 581K	15.8M	470M	29.8	18.59	5.37	EN_PAIRS_MI_1.8CUT
616K x 573K	8.95M	166M	18.6	18.65	8.42	EN_PAIRS_5.3_REMI

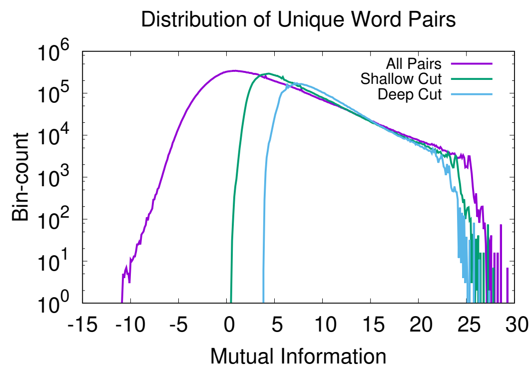
As before, the cut to the number of observations is proportionately greater than the cut to the number of pairs: the number of pairs is about a third of the original dataset; the number of observations dropped by almost a factor of 8.

The vector norms of the rows and columns are given in the table below.

Size		Support		Count		Length		Dataset
L	R	L	R	L	R	L	R	Name
619K	581K	73.1K	53.8K	254	282	25.9	34.7	EN_PAIRS_CFIVE
619K	581K	10.0K	12.5K	138	210	24.2	26.3	EN_PAIRS_MI_1.8CUT
616K	573K	2915	1840	109	137	55.4	81.8	EN_PAIRS_5.3_REMI

With the deeper cut, the support continues to drop dramatically. The count/support ratio drops mildly. The length/support ratio actually increases! In effect, huge vectors, with lots of support, but otherwise small amounts of observations are sharply cut down to size. What remains in those vectors are matrix elements with a large amount of observations: thus the length (per basis element) increases! Again, this is a confirmation of the behavior reported earlier.

If the MI is recomputed on the smaller dataset, the effect of the re-computation is much as before: a general shift downwards of all MI, and a rounding-off of the sharp corner in the cut.



In the above figure, the distribution labeled "Shallow Cut" just reproduces the earlier recomputed 1.8<MI distribution. The "Deep Cut" is the distribution after recomputing the 5.3<MI cut.

Conclusions

The above analysis has lead to several insights and proposals:

- The distribution of word-pairs observed in natural language corpora have a clear bimodal distribution. One mode, termed the zero-mode, appears to consist of effectively random word-pairs, having little correlation or affinity to one-another. This mode reveals itself in several ways: the word pairs have a low MI value, and they are paired with a large variety of other words.
- The zero-mode accounts for the majority of word-pair observations. However, it seems that it can be identified fairly cleanly, and thus removed or cut away.
- The other mode consists of what might be termed "true collocations" or the "true mode".
- Based on this observation, it is relatively straight-forward to pose a hypothesis: if MST parsing is limited to pairs that are in the "true mode", will this lead to higher-quality disjuncts and higher-quality grammars?

- The Maximal Planar Graph parsing strategy seems to offer a CPU-efficient mechanism for exploring the effect of different data cuts on disjunct quality, and should probably adopted as the primary pipeline parser.

- The relationship between these two modes, natural language, and randomly-generated pseudo-language remains a bit murky, despite an earlier investigation. In particular, the earlier experiments did not emulate the Zipfian distribution of natural language, and did not make any clear conclusions relating corpus size to vocabulary size. This omissions mean that it remains unclear which statistical properties are particular to natural language, and which statistical properties are typical of any random text. Understanding the differences between random corpora and natural text remains an important, unfinished task.

The End

References

[1] Ben Goertzel and Linas Vepstas, "Langauge Learning", , 2014, URL https://arxiv.org/abs/1401.3372, arXiv abs/1401.3372.

[2] Linas Vepstas, "Unsupervised Learning Project GitHub Repository", , 2014, URL https://github.com/opencog/learn/raw/master/, gitHub Repo.

[3] Linas Vepstas, "Gradient Decent vs. Graphical Models", , 2018, URL https://github.com/opencog/learn/learn-lang-diary/skippy.pdf.

[4] J. Lambek, "On the calculus of syntactic types", in *Structure of Language and its Mathematical Aspects*, America Methematical Society, 1961, pp. 166–178.

[5] Solomon Marcus, *Algebraic Linguistics; Analytical Models*, Elsevier, 1967, URL https://monoskop.org/images/2/26/Marcus_Solomon_editor_Algebraic_Linguistics_Analytical_Models_1967.pdf.

[6] Thierry Mora, et al., "Maximum entropy models for antibody diversity", *Proceedings of the National Academy of Sciences*, 107, 2010, pp. 5405–5410, URL http://www.pnas.org/content/107/12/5405.

[7] Deniz Yuret, *Discovery of Linguistic Relations Using Lexical Attraction*, PhD thesis, MIT, 1998, URL http://www2.denizyuret.com/pub/yuretphd.html.

[8] Thierry Mora and William Bialek, "Are biological systems poised at criticality?", *Journal of Statistical Physics*, 144, 2011, pp. 268–302, URL https://arxiv.org/abs/1012.2242.

[9] Joakim Nivre, *Inductive Dependency Parsing (Text, Speech and Language Technology)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[10] Linas Vepstas, "The Distribution of English Language Word Pairs", , 2009, URL https://github.com/opencog/learn/raw/master/learn-lang-diary/word-pairs-2009/word-pairs.pdf.