Comparing Quantitative Linguistics of a Corpus with Vocabulary Networks

Samar Dikshit

I. Introduction

All languages have inherent patterns associated with it. Given a collection of texts, we can use various statistical measures to describe this corpus. These empirical measurements provide information about the general structure of the corpus, along with information about sentence structure, vocabularies, and n-grams used.

A corpus can also be converted into a directed and weighted network, effectively forming a vocabulary network. In such a network, words/phrases (depending on network construction) form the nodes of the graph, while an edge exists if nodes co-occur.

In this project, I've explored a set of quantitative linguistics measurements for a corpus, including Heaps' Law, Zipf's Law, and the brevity law amongst others. This was followed by creating a word network using the corpus and determining its properties and whether it is scale-free, and attempting to co-relate these results with the results obtained from the quantitative analyses. Finally, the network was used as the base for a simple language model for sentence generation.

The motivation for such a comparison arises from the fact at Heaps' and Zipf's Laws are power laws, and scale-free networks are networks with power law degree distributions. This is relationship is discussed further in Section III.2.ii.

II. Data

The corpus used consists of two books by Jules Verne – Twenty Thousand Leagues Under the Sea (1870), and Around the World in Eighty Days (1872). Both books were originally published in French, but have since been widely translated and are available on Project Gutenberg. Henceforth in this report, the term 'corpus' refers to both books combined as one.

The data was pre-processed before it was used for any task. The pre-processing steps were:

1. Replaced newlines with whitespaces
2. Split the text into its constituent sentences
3. Removed all punctuation
4. Case-folding to lower case
5. Added start-of-sentence and end-of-sentence markers (<*s*> and </*s*>) to each sentence

This corpus does not contain labels, as no classification or recognition tasks were carried out.

The following table lists the vocabulary size and unique words of each text and the corpus:

Text	**Vocabulary Size |V|**	**Unique Words (excluding <*s*> and </*s*>)**
Twenty Thousand Leagues	8,669	104,174
Around the Word	6,829	64,343
Corpus i.e. both combined	11,553	168,517

Table 1: Vocabulary sizes and unique words after pre-processing

III. Methodology and Results

III.1. Quantitative Linguistics

III.1.i. Sentence Structure

Quantifying the sentence structure of a text provides simple measurements of the general shape and structure of the text. Table 2 contains the sentence measurements recorded using words per sentence:

Text	Number of Sentences	Mean Length	Median Length	Maximum Length	Minimum Length
Twenty Thousand Leagues	6,587	17.815	16	183	3
Around the World	2,874	24.388	22	178	3
Corpus	9,461	19.812	17	183	3

Table 2: Sentence Structure

III.1.ii. Heaps' Law

Heaps' Law is an empirical power law that describes a relation between the vocabulary size |V| and the number of total words seen N. Mathematically:

$$|V| = KN^{\beta}$$
$$\log|V| = \log K + \beta \log N$$

where K and β are constants (typically $10 < K < 100$, $\beta \approx 0.5$). In its log form, $\log K$ corresponds to the intercept of a line, and β is the slope. Heaps' Law describes the vocabulary growth with every word seen. The law suggests that the vocabulary size continues to grow as more words are seen, although the growth rate decreases but never becomes zero. Hence, there is no limit to |V|. Using observations in the log form, we can determine K and β using linear least-squares regression, and calculate predictions based on Heaps' Law. The following figure describes the result of Heaps' Law for this corpus:

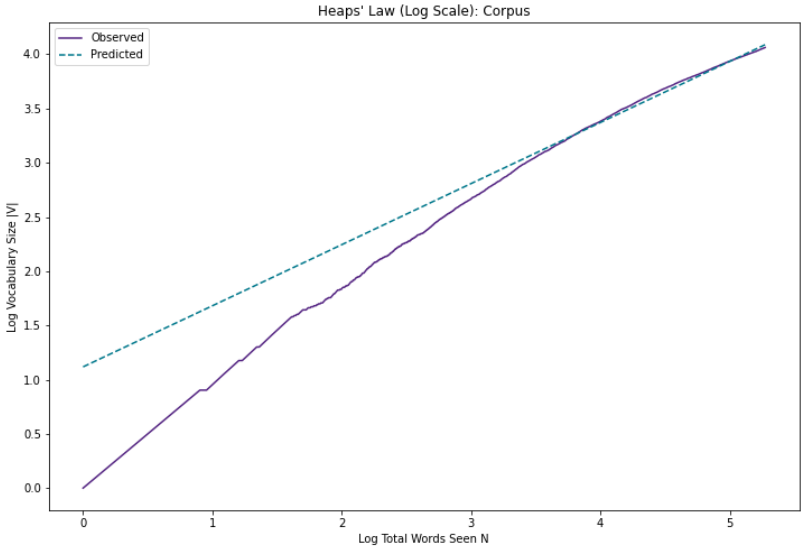


Figure 1: Heaps' Law (log scale) for the corpus (K: 13.119, β: 0.563)

III.1.iii. Zipf's Law

Zipf's Law is another empirical power law that describes the relation between word frequency F and its rank R. High frequency words have a lower rank. Mathematically:

$$F = ZR^{-\alpha}$$
$$\log F = \log Z - \alpha \log R$$

where Z and α are constants (typically $\alpha \approx 1$). Similar to Heaps' Law, in its log form, $\log Z$ corresponds to the intercept, and α is the slope. The sign associated with α is negative as lower frequency words must have higher ranks, hence the slope of the line is also negative. Using observations in the log form, we can determine Z and α using linear least-squares regression, and calculate predictions. Figure 2 describes the results of Zipf's Law for this corpus (note the expected error for high frequencies):

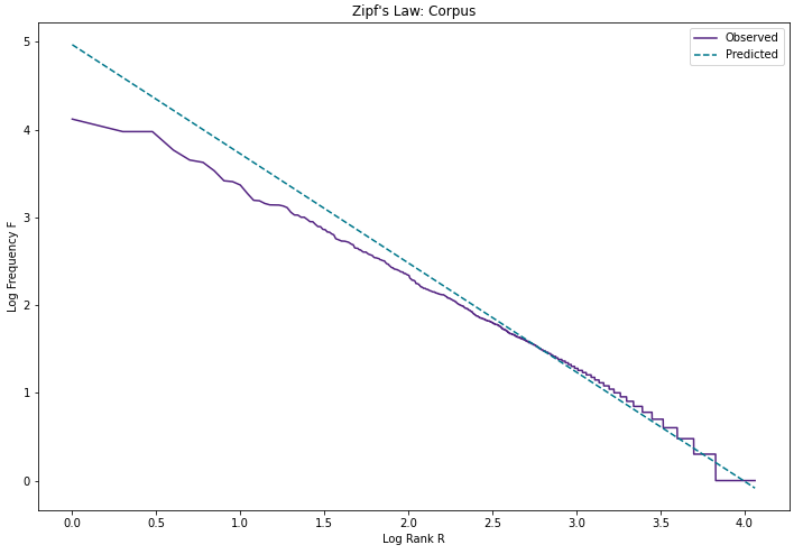


Figure 2: Zipf's Law (log scale) for the corpus (Z: 92,524.164 , α: 1.243)

III.1.iii. Brevity Law

The Brevity Law states that words with higher frequency tend to be shorter, and vice-versa. Unlike Heaps' and Zipf's Laws, the brevity law follows a log-normal distribution. For the corpus, this can be seen in Section A.2. in the appendix. The mean and median word lengths are 7.174 and 7 respectively, with maximum and minimum lengths of 17 and 1. The most frequent word is 'the' (frequency: 13139).

III.1.iv. Word Vectors

Word embeddings of each word are created using skip-grams. Each word vector has a 200 dimensions, and uses a window size of 5 with minimum frequency of 1. Using cosine similarity, the similarity (syntactic and semantic) of two words can be calculated. These similarities will be compared with the length of the shortest path between the two words in an additional experiment in Section A.5.

III.2. Vocabulary Network

A vocabulary network was created using this corpus, where nodes are words in the vocabulary. Each node as an attribute containing its frequency. Edges are based on co-occurrence, and are directed. For example, the phrase 'captain nemo said' contains out-edges from 'captain' to 'nemo' and 'nemo' to 'said'. For 'nemo', the out-edge from 'captain' is an in-edge to it, and similarly for 'said'. Each edge has an attribute containing its count, describing the number of times the word pairs it connects occurs (this is effectively the same as a bigram frequency). This count serves as the edge weight.

For the corpus, the following network is created:

- Nodes: 11,553
- Edges: 79,604 (an in-edge for one node is an out-edge for another, hence 79,604 is the total)
- Density: 0.00059

The following figure shows a vocabulary network for only 4 sentences of the corpus:

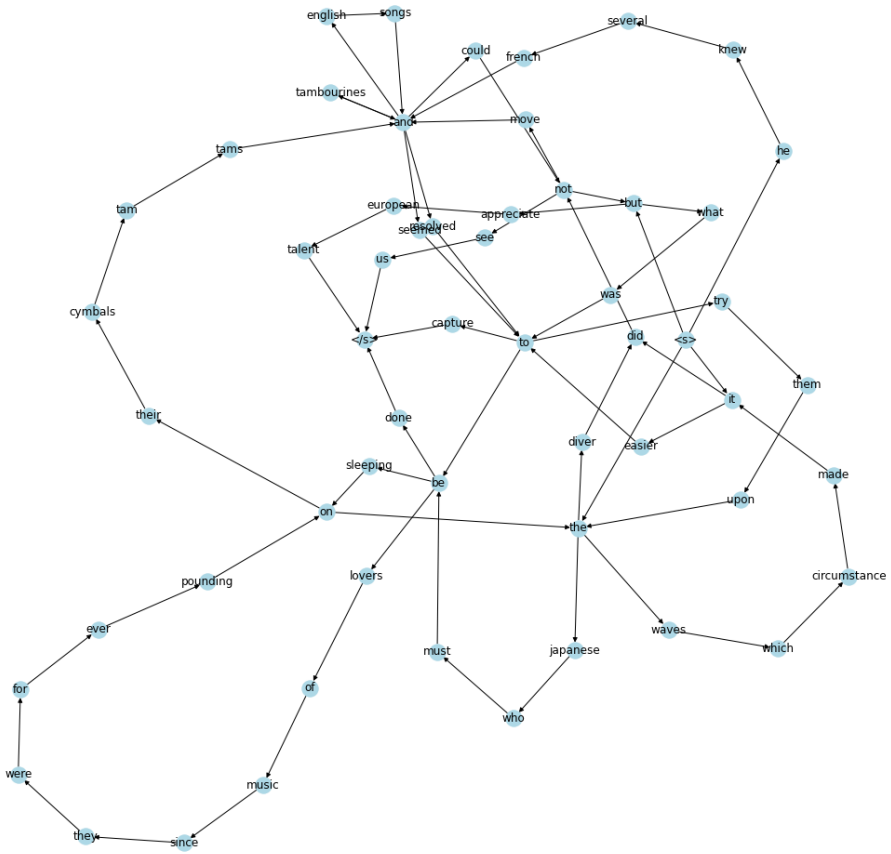


Figure 3: Vocabulary network for 4 sentences of the corpus

III.2.i. Degree Distribution and Scale-free Nature

The degree distribution of a directed network is given by the following equations:

$$p(k_{in}) \sim k^{-\gamma_{in}}$$
$$p(k_{out}) \sim k^{-\gamma_{out}}$$

where γ_{in} and γ_{out} are degree exponents for the in-degrees and out-degrees respectively. The equations above are power law distributions, similar to Heaps' and Zipf's Laws from sections III.1.ii. and III.1.iii. respectively. If the values of these exponents are between 2 and 3, the degree distribution is heavy tailed, and the network is scale-free.

These heavy tailed distributions occur due to the presence of 'hubs' i.e. a node with a high degree. In a vocabulary network, nodes with high degrees tend to be stop-words, since they are used frequently. For example, in Figure 3, we can see words such as 'and', 'to', and 'the' have a larger number of in and out-degrees than less common words such as 'cymbals', 'waves', or 'tambourines'.

The values that fit γ_{in} and γ_{out} are calculated empirically, after plotting the degree distribution for the in and out-degrees of the network. For this corpus' vocabulary network, the degree exponents γ_{in} and γ_{out} have values 2.116 and 2.192. Hence, the network is scale-free.

One of important consequence of the network being scale-free is that the average path length of the network is given by $ln\, ln|nodes|$. For this network, this results in an average path length of 2.236. This result will be used comparing vector similarity to the shortest path between two nodes in Section A.5.

Figure 4 shows the degree distribution for this network. Other important degree statistics are:

- Maximum in and out-degree: 3,222 and 3,039 respectively
- Average in and out-degree: 6.89

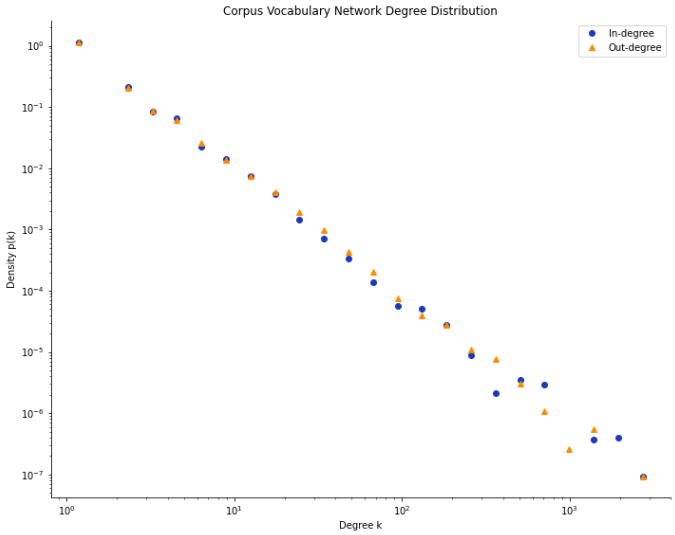


Figure 4: Corpus vocabulary network degree distribution (log-log scale). A heavy tail distribution can be seen.

III.2.ii. Network Metrics and the Relationship with Quantitative Linguistics

To establish a relationship between quantitative linguistics and network metrics, we can start with Zipf's Law and the degree distribution of the network. Both are power laws with negative exponents, and are related to frequency. We can plot the term frequencies of terms against their in and out-degrees. We have the following equations:

$$k_{in} = BF^{x_{in}} \text{ and } k_{out} = CF^{x_{out}}$$
$$\log k_{in} = \log B + x_{in} \log F$$
$$\log k_{out} = \log C + x_{out} \log F$$

where $k_{in}$is the in-degree, k_{out} is the out-degree, B and C are the in and out-degree constants respectively, $x_{in}$and x_{out} are the in and out-degree exponents, and F is the term frequency. Similar to what was done for Heap's and Zipf's Laws, we can use linear least-squares to estimate B, C, x_{in} and x_{out} given the observations, and then calculated predicted degrees given the frequencies. The results for term frequency vs in-degree is shown in Figure 5:

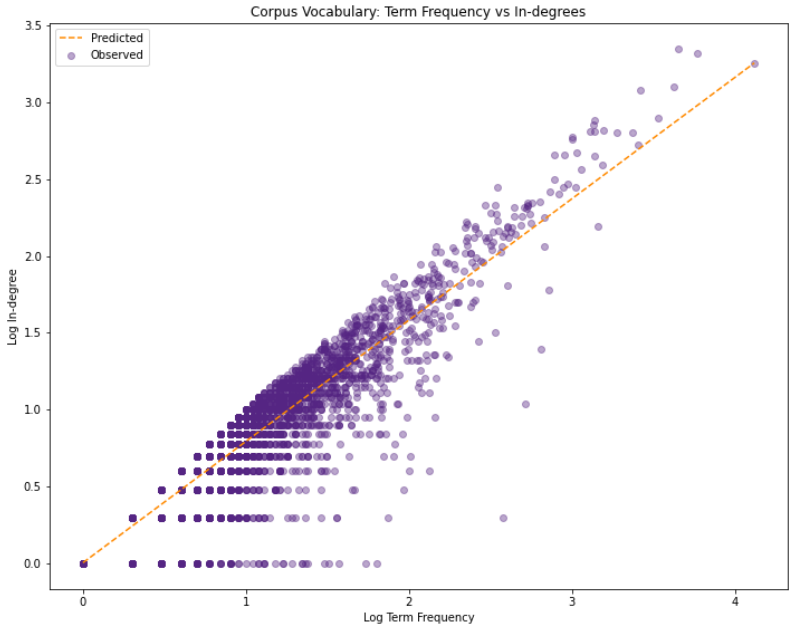


Figure 5: Term frequencies vs in-degrees (B: 1.013, x_{in}: 0.789)

For term frequency vs out-degree, we get C = 1.009 and x_{out} = 0.831 (Section A.3).

III.2.iii. Vocabulary Network as a Language Model

This vocabulary network can be used as a simple language model with two different techniques for sentence generation by traversing edges in the network:

1. Network Shannon's method: Starting from any selected node, follow out-edges by sampling probabilities of out-edges to select the next word until the maximum length condition is hit or we reach the </s> node. Each edge has a count attribute describing how often that node pair i.e. word pair occurs. Hence the probability of edge occurrence is the count of the edge divided by count of all out-edges from the node. It performs as well as Naïve Bayes-based model.
2. Inside-out generation: Starting from a node, we can define the maximum number of words required both before and after the selected node (not including <s> or </s>). For the words that need to be generated after the starting node, the technique described in point 1 is used. For the words required before the selected word, we traverse out-edges backwards i.e. follow the procedure in point 1 but using in-edges to a node until we hit the length condition or the <s> node. As we generate a sentence from some word in between, this technique is termed inside-out generation. Compared to standard Shannon's method, inside-out generation tends to give slightly more meaningful sentences, but will be outperformed by a neural model.

```
1: <s> bed not to nothing say the captain nemo appeared really going away echoed with conseil to chaff fix who was
</s>
2: <s> as mr fogg sir replied captain nemo i heard what it was never trod along the man said if </s>
3: <s> captain nemo spoke little schooners coasting about eight hours slow hours and mr fogg </s>
4: <s> entering her boilers and leaving the captain nemo soon large number </s>
5: <s> would declare to the wind was captain nemo employed my imagination of the party only had left him to deprive
</s>
```

Figure 6: Five inside-out sentences using 'nemo' as the seed (max 7 words before and max 8 after)

IV. Conclusion and Future Work

This corpus follows Heaps', Zipf's, and the brevity law. From Table 2, we can also conclude that Verne's general writing style is the same across both books. This project has also been able to successfully model the corpus as a vocabulary network that follows a scale-free distribution, and find a relationship between frequencies and degrees. The network can even be used as a language model. However, there are still opportunities for future work:

- Does the scale-free nature of the network create any unique properties related to language? In this project, only one short experiment was conducted focusing on a property of scale-free nature (using average path length, Section A.5.). More experiments need to be run to determine the relationship between scale-free nature and language.
- Language structure: If words are tagged using part-of-speech tagging, the network will reflect the structure of the language e.g. 'the nautilus does not sink and nemo survives' will have underlying tags DT → NNP → VBZ → RB → VB → CC → NNP → VBZ for the sentence. I believe that this has applications in named entity recognition and machine translation as we can use a network of tags and frequencies to determine language rules such as adjective-verb-noun orders.

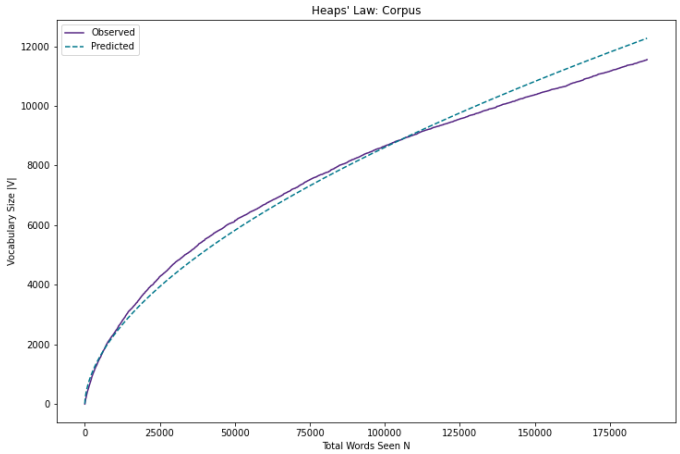
V. Sources:

1. Jules Verne on Project Gutenberg
2. A standardized Project Gutenberg corpus for statistical analysis of natural language and quantitative linguistics – Gerlach et al.
3. Toplogy of the conceptual network of language – Motter et al.
4. Power laws, Pareto Distributions and Zipf's law – Newman
5. The brevity law as a scaling law, and a possible origin of Zipf's law for word frequencies – Corral et al.
6. Network Science – Barabasi

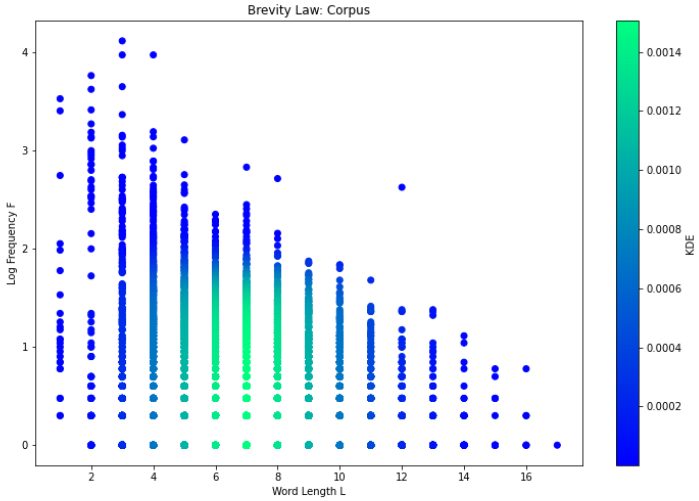
The code for this project is available on my GitHub.

Appendix

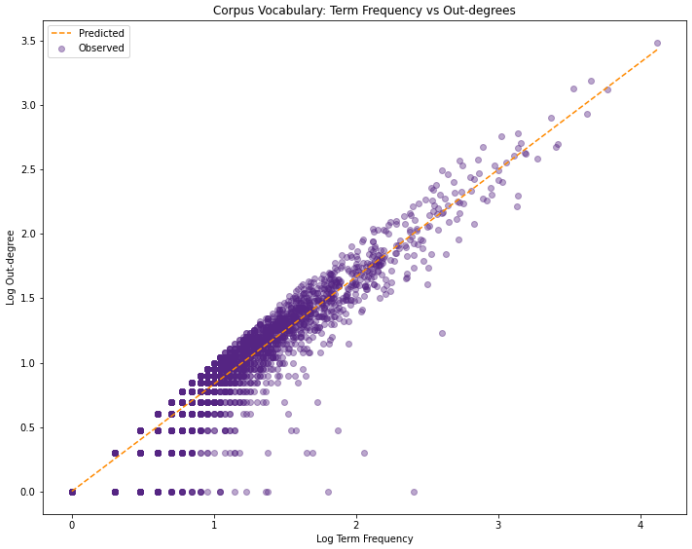
A.1.: Heaps' Law for the corpus in the linear scale



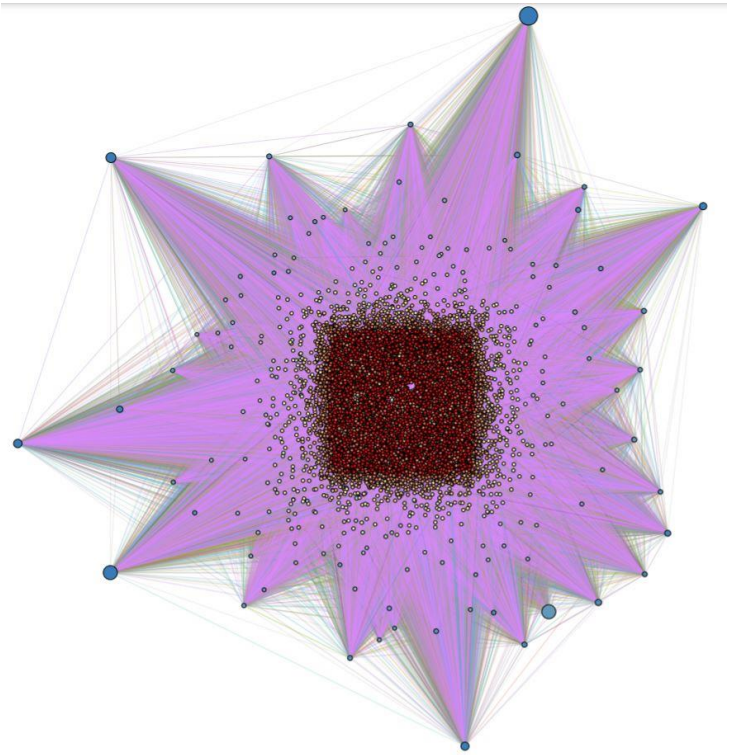
A.2.: Brevity Law for the corpus (log-normal distribution with KDE superimposed)



A.3. Term frequencies vs out-degrees



A.4.: The entire vocabulary network visualized



Node size corresponds to word frequency, node colour corresponds to total degree (blue: high degree, pale and red shades: low degree), edge colour corresponds to edge count (pink: lower count)

A.5.: Average Vector Similarity and Average Shortest Path Length

Taking four characters as sources (‘fogg’, ‘aouda’, ‘nemo’, ‘aronnax’) in the corpus as an example, we can find the mean vector similarity for the *n* most similar nodes to each source using word embeddings from Section III.1.iv., and compare it with the average of the shortest path lengths been the source and each of the *n* most similar nodes. For the following experiment, *n* = 20 was chosen.

Source	**Avg. Vector Similarity**	**Avg. Shortest Path Length**
fogg	0.845	1.263
aouda	0.959	1.684
nemo	0.839	1.6
aronnax	0.971	1.75

Table A.1: Avg. vector cosine similarity and avg. shortest path length from the source to the 20 most similar words