Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE
Corso di Laurea Magistrale in Matematica

Italian texts as Networks: Topological measurements, Zipf and Heaps' law

Tesi di Laurea in Sistemi Dinamici e Applicazioni

Relatore:
Chiar.mo Prof.
Mirko Degli Esposti

Co-relatore:
Dott.
Giampaolo Cristadoro

Presentata da:
Giulia Tini

Sessione II
Anno accademico 2012/2013

Ma, parlando sul serio,
niente si assomiglia piú
che contare e raccontare.
Entrambi unificano il mondo,
lo districano e lo liberano. [...]
Una storia, puó anche
essere inventata, ma con
l'aritmetica che le é propria
é in grado di scardinare il mondo.

Thomas Vogel,
L'ultima storia di Miguel Torres da Silva.

Contents

Introduction **3**

1 Networks and their properties **5**

1.1 Basic concepts about networks 5

1.2 Topological network measurements 8

1.2.1 Degree and strength distributions 13

1.2.2 Entropy 13

1.3 Scale-Free and Small-World properties 14

1.3.1 Scale-Free Networks 14

1.3.2 Small-World Networks 15

1.4 Linguistic networks 16

1.4.1 Topological measurements in texts 17

2 Zipf and Heaps' Law **21**

2.1 Zipf's Law 21

2.2 Heaps' Law 24

2.3 Relation between Zipf and Heaps' Exponents 26

2.4 Stochastic Models 29

2.4.1 Simon's Model 30

2.4.2 Mandelbrot's Model 31

2.4.3 Modified Simon's Model 34

3 Italian Texts and Networks **37**

3.1 Database 37

3.2 Modeling texts as networks 38

3.2.1 Normal Texts 38

3.2.2 No Stopwords Texts 39

3.2.3 Reduced Texts 41

3.2.4 Shuffled Texts 42

3.2.5 Network formation 42

4 Results and Discussion **47**

4.1 Frequency 49

4.2 Zipf's Law 52

4.3 Heaps' Law 56

4.4 Degree 60

4.4.1 Degree Distribution $P_1(k)$ 62

4.4.2 Conditioned Zipf's law and Degree Distribution 66

4.4.3 Degree Distribution $P_2(k)$ 72

4.5 Entropy 74

4.6 Strength 76

4.6.1 Strength Distribution $P_1(s)$ 77

4.7 Selectivity 77

4.7.1 Selectivity Distribution 83

4.8 Gephi statistics 87

4.8.1 Average Path Length 87

4.8.2 Diameter 88

4.8.3 Average Cluster Coefficient 88

4.8.4 Network Density 89

4.8.5 Betweenness Centrality 90

4.9 A different approach to texts 94

4.9.1 Comparison of results 96

Conclusions **98**

A Gephi 99
A.1 Dynamic Networks 99
A.2 Layout 101
A.3 Statistics 102
B Italian Stopwords List 103
C Python function to create the weight matrix and the GEXF file 105
Acknowledgments 110
References 111

Introduction

In recent years it has been witnessed a gradual extension of the ideas and methods of statistical physics in a vast range of complex phenomena outside the traditional boundaries of physical science: biology, economics, social sciences and linguistics.

In particular, the application of ideas from statistical physics to text analysis has a long tradition, since Shannon's usage of entropy as the central concept in Information Theory, in 1948 [25].

At the same time, even the study of networks has emerged in different disciplines as a mean of analysing complex relational data, for example human language. In fact human language is a natural code, capable of codifying and transmitting highly non trivial information, thus its structures and evolution can be explored with the help of Network Theory [13] [25].

Humans build their language using a very small number of units, words and punctuation, and their co-occurrences in sentences are not trivial. In fact, syntactical and grammatical rules and the use of particular expressions imply that some words co-occur with certain words at higher probability than with others. Both written and spoken language have complex grammatical structures that dictate how information can be communicated from one individual to another: texts are organised in a precise way and they need to be well concatenated [6].

For this reason, in the last decades physicists have proposed new approaches to text analysis, based on concepts from Network Theory. This has been done for a variety of applications, related to fundamental linguistic: to identify literary movements [5], to recognize patterns in poetry and prose [24] [28], to characterize authors' style [3], to study the complexity of a text [4], to generate and evaluate summaries [8].

The use of statistic applied to human language and texts provide a picture of their macroscopic structures. One of the most generic statistical properties of written language is Zipf's law, discovered by the philologist George Zipf in 1936, that describes a scaling law in the distribution of the words frequency. In his *Human Behaviour and the Principle of Least Effort. An Introduction to Human Ecology* (1949), Zipf explained his law by the *principle of least effort*: he tought that the statistic of word usage would depend on the balance between the efforts invested by speaker and hearer in the communication process.

The second important statistical feature of written language is Heaps' law, that usually coexists with Zipf's one. It was discovered by Harold Heaps in 1978 and it predicts the vocabulary size of a document from its length, that is the number of words it contains [14].

These two statistical patterns are universal trends, shared by all spoken languages [27]. However, Zipf and Heaps' laws are not the the only two structures present in all languages: these share universal tendencies at different levels of organization, for example syntactic and semantic categories, or expressed concepts. At the same time there are also deep differences between languages, for example prepositions are not present everywhere.

In this thesis, we will study a set of important Italian books, written between 19th and 20th centuries. Each of them will be analysed even in some different versions: shuffling their words, deleting stopwords and eliminating terms appearing only once. The aim is to understand which measures of Network Theory are able to distinguish masterpieces from their variations.

Texts will be transformed in networks with words as vertices and links created between adjacent nodes, following the reading order, then they will be visualised with the help of the open-source software *Gephi*, that can also

provide some useful measurements.
The methods for text analysis that we will consider, can be divided in two classes: (i) those based on first-order statistics, for example words frequency, Zipf and Heaps' laws; (ii) those based on metrics from Network Theory, obtained by Gephi and by a Python algorithm.
Thinking about similarities and differences between different languages, the study and work to write this thesis was done in collaboration with Filippo Bonora, who chose to analyse an English corpus, in the same way described above. This could be useful in order to compare values obtained in Italian and English measurements, and the role measurements have in these two languages. Since the methods used to analyse texts are the same, the first chapter of the thesis has been written in collaboration with Filippo Bonora, and it is equal to the first chapter of his thesis, *Dynamic Networks, text analysis and Gephi: the art math.*

The thesis is organised as follows. In the first chapter we give a survey of the basic concepts of Network Theory, with particular attention to linguistic graphs.
The second chapter deals with Zipf and Heaps' law, the relation between them and some important stochastic models that can explain these two important features.
The third chapter describes the database, the different versions of the novels that we consider, and the way in which texts are transformed in networks.
In the fourth chapter we show and discuss the results we obtained, displaying especially distributions and average values of the computed measures. We show even that statistical laws hold, and in particular we use Zipf's law to demonstrate the invariance of degree distribution with shuffling.

Chapter 1

Networks and their properties

In this Chapter we will give some definitions and results related to Network Theory, as we will use them to represent and analyse linguistic texts.

1.1 Basic concepts about networks

Networks can be used to model many kinds of relations in physical, biological, social and information systems [19]. Many practical problems can be represented by graphs, thus Network Theory is a common subject of research.

We can start our analysis introducing its basic definitions [12].

Definition 1.1.1. A **weighted directed graph** $\boldsymbol{G}$ is defined by:

- a set $\mathcal{N}(G)$ of **N vertices**, or *nodes*, identified by an integer value i= $1, 2, \ldots ,N$;
- a set $\mathcal{E}(G)$ of **M edges**, or *links*, identified by a pair $(i,\ j)$ that represents a connection starting in vertex i and going to vertex j;
- a mapping $\omega : \mathcal{E}(G) \longrightarrow \mathbb{R}$, that associates to the edge $(i,\ j)$ the value $\omega(i,j)$=ω_{ij} called **weight**.

Definition 1.1.2. A weighted directed graph G can be represented using its **weight matrix** $\boldsymbol{W}$:

$$W = (\omega_{ij}).$$

We can observe that W is a $N \times N$ non symmetric matrix whose elements represent the number of directed links connecting vertex i to vertex j.

In this thesis we will assume that no pair of edges (i_1,j_1) and (i_2,j_2) with $i_1 = i_2$ or $j_1 = j_2$ exists.

We can even define the matrix created by nodes relations without considering their weights:

Definition 1.1.3. The $N\times N$ matrix $\boldsymbol{A}= (a_{ij})$ is the **adjacency matrix** of the graph G if:

$$\forall i,j \qquad a_{ij} = \begin{cases} 1, & \text{if } \omega_{ij} \neq 0 \\ 0, & \text{if } \omega_{ij} = 0. \end{cases}$$

The element a_{ij} tells us whether there is an edge from vertex i to vertex j, independently to the link weight. If $a_{ij} = 0$, such a connection does not exist.

Definition 1.1.4. The **neighbourhood** of a vertex i, $\nu(i)$, is the set of vertices **adjacent** to i:

$$\nu(i) = \{j \in \mathcal{N}(G) | (i,j) \vee (j,i) \in \mathcal{E}(G)\}.$$

Definition 1.1.5. Eventually even two not adjacent vertices i and j can be connected, using a **walk**, that is a sequence of m edges:

$$(i,k_1),(k_1,k_2),\ldots,(k_{m-1},j)$$

where m is the walk **length**.
If all the nodes and the edges composing a walk are distinct, the walk is called **path**.

Definition 1.1.6. A **shortest path** between two nodes is defined as the path whose sum of edge weights is minimum [3].

Definition 1.1.7. A **loop** or *cycle* is a walk starting and ending in the same node i, and passing only once through each vertex k_n composing the walk.

Let us give a simple example showing a weighted directed network:

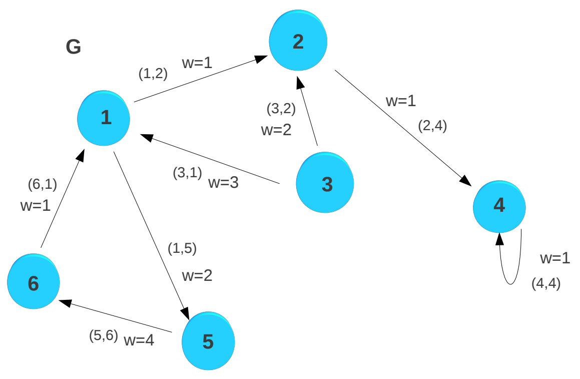


Figure 1.1: A weighted directed graph

Example 1.1.1. The figure 1.1 shows a graph $G = (\mathcal{N}(G), \mathcal{E}(G))$ where:

$$\mathcal{N}(G) = \{1,2,3,4,5,6\}, \qquad N = |\mathcal{N}(G)| = 6,$$

$$\mathcal{E}(G) = \{(1,2),(1,5),(2,4),(3,1),(3,2),(4,4),(5,6),(6,1)\}, \qquad M = |\mathcal{E}(G)| = 8.$$

The values called w are the weights of the edges, so for example $w_{56} = 4$, $w_{31} = 2$, $w_{15} = 2, \ldots$. Using them we can construct the weight matrix W

and the adjacency matrix A:

$$W = \begin{pmatrix} 0 & 1 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 3 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \qquad A = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

The neighbourhoods of some vertices are:

$$\nu(1) = \{2, 3, 5, 6\} \qquad \nu(4) = \{2, 4\}.$$

We can also see some walks connecting non adjacent vertices, for example 4 can be reached by 5 with this sequence of edges:

$$(5, 6), (6, 1), (1, 2), (2, 4) \quad \text{with length } m = 4.$$

There is also a loop for the node 1:

$$(1, 5), (5, 6), (6, 1) \quad \text{with length } m = 3.$$

1.2 Topological network measurements

In this Section we will describe some important measurements that will be useful to analyse and characterize a graph.

The first important quantity is the number of edges connecting vertices, that is called degree. Since we are considering directed graphs we can define two different kinds of degree:

Definition 1.2.1. The **in-degree** k_i^{in} of a vertex i is the number of its predecessors, equal to the number of incoming edges.
Similarly its **out-degree** k_i^{out} is the number of its successors, corresponding to the number of outcoming edges:

$$k_i^{in/out} = \sum_j \Theta\left(\omega_{ji/ij} - \frac{1}{2}\right)$$

where Θ is the Heaviside function

$$\Theta(x) = \begin{cases} 0, & x < 0 \\ 1, & x > 0 \end{cases}.$$

The **average in/out degree** of a network is the average of $k_i^{in/out}$ $\forall$ i:

$$\left\langle k^{in/out} \right\rangle = \frac{1}{N} \sum_i k_i^{in/out}.$$

Proposition 1.2.1.

$$\langle k^{in} \rangle = \langle k^{out} \rangle = \frac{1}{N} \sum_i k_i.$$

In fact:

$$\langle k^{in} \rangle = \frac{1}{N} \sum_j \sum_i \Theta\left(\omega_{ij} - \frac{1}{2}\right) = \frac{1}{N} \sum_i \sum_j \Theta\left(\omega_{ij} - \frac{1}{2}\right) = \langle k^{out} \rangle .$$

We complete the study of the simple network in Example 1.1.1. showing vertices degree:

vertex	k^{in}	k^{out}
1	2	2
2	2	1
3	0	2
4	2	1
5	1	1
6	1	1

$$\langle k^{in} \rangle = \frac{2+2+0+2+1+1}{6} = \frac{4}{3} \qquad \langle k^{out} \rangle = \frac{2+1+2+1+1+1}{6} = \frac{4}{3}.$$

Definition 1.2.2. Given two vertices i and j, if $d_{min}(i,j)$ is the minimum path length that connects them, the **average path length of vertex *i*** is:

$$d_{v(i)} = \frac{1}{N}\sum_{j=1}^{N} d_{min}(i,j).$$

Thus the **average path length** d will be

$$d = \frac{1}{N}\sum_{i=1}^{N} d_{v(i)}.$$

The last quantity indicates the number of steps one needs to make on average in the graph in order to connect two randomly selected nodes.

Definition 1.2.3. The **diameter** of a graph is the longest distance between any two nodes in the network, i.e.

$$\max_{i,j}\ d_{min}(i,j).$$

Hereafter we will consider only a particular kind of networks:

Definition 1.2.4. A **multi-directed Eulerian** network is a directed network where exists a path that passes through all the edges of the network once and only once.

Graphs of this kind are used to describe information networks, such as human language or DNA chains [21] [10].

Coming back to useful network measurements, we give another important quantity:

Definition 1.2.5. The **in-strength** of a vertex i is the sum of the weights of its incoming links:

$$s_i^{in} = \sum_j \omega_{ji}.$$

Similarly the **out-strength** of i is the sum of the weights of its outcoming links:

$$s_i^{out} = \sum_j \omega_{ij}.$$

In the Example 1.1.1. $s_1^{in} = 4$ and $s_1^{out} = 3$, $s_3^{in} = 0$ and $s_3^{out} = 5$.

To analyse in a more completely way the networks that we are going to study in the following Chapters, we introduce some significant measurements that are able to describe the relations between nodes.

Definition 1.2.6. Let $s_i^{in/out}$ be the in and out-strength for the node i, and $k_i^{in/out}$ its in and out-degree. Then the **in/out-selectivity** is [20]:

$$e_i^{in/out} = \frac{s_i^{in/out}}{k_i^{in/out}}.$$

Definition 1.2.7. A **cluster** is a set of connected nodes with similar values of complex networks measurements.

Definition 1.2.8. The **cluster coefficient**:

$$C_i = \frac{\sum_{k>j} a_{ij} a_{ik} a_{jk}}{\sum_{k>j} a_{ij} a_{ik}}$$

is the fraction of triangles involving vertex i among all possible connected sets of three nodes having i as the central vertex. Therefore $0 \leq C_i \leq 1$. This quantity measures the density of connections between the neighbours of node i, i.e. the probability that the neighbours of a given vertex are connected.

We can compute the importance of a vertex or an edge considering the number of paths in which it is involved. Assuming that a vertex is reached using the shortest path, this can be measured by the betweenness centrality.

Definition 1.2.9. The **betweenness centrality** of a vertex or an edge u is defined as

$$B_u = \sum_{i,j \ | i \neq j} \frac{\sigma(i,u,j)}{\sigma(i,j)}$$

where $\sigma(i,u,j)$ is the number of shortest paths between vertices i and j that pass through u, while $\sigma(i,j)$ is the total number of shortest paths between i and j.

Betweenness centrality and cluster coefficient have similar meanings, but the first is based on a global connectivity pattern, while the latter is a local measurement.

Definition 1.2.10. A *potential connection* is an edge that could potentially exists between two nodes. The number of potential connections is calculated as:

$$PC = \frac{N(N-1)}{2}.$$

If *actual connections* (AC) are the edges that actually exist, **network density** Δ is [36]:

$$\Delta = \frac{\text{actual connections}}{\text{potential connections}} = \frac{AC}{PC}.$$

Density is an indicator for the general level of connectedness of the graph. If every node is directly connected to every other node, we have a complete graph. Hence it is a relative measure, with values between 0 and 1.

Example 1.2.1. Let us compute the density of two graph both with three nodes but with different edges:

Figure	**Nodes**	**PC**	**AC**	Δ
1.2	3	3	3	1
1.3	3	3	2	2/3

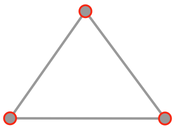


Figure 1.2: Complete graph.

Figure 1.3: Non complete graph.

1.2.1 Degree and strength distributions

The numbers of nodes and edges in graphs are often too high to analyse properties of every vertex or link. For that reason it could be useful to introduce probability distributions to explore degree, strength and selectivity behaviour in the whole network.
We compute two different **degree distributions**:

- The fraction of vertices in the network with degree equal to k:
$$P_1(k) = \frac{\#\{j \mid 1 \leq j \leq N,\ \text{degree}(j) = k\}}{N}$$
- The probability to find a vertex with degree equal to k:
$$P_2(k) = \sum_{j \mid \, degree(j)=k} p(j)$$
with $p(j)$ the probability to find j in the graph.

P_1 can be also used to compute **strength distribution**:
$$P_1(s) = \frac{\#\{j \mid 1 \leq j \leq N,\ \text{strength}(j) = s\}}{N}$$

1.2.2 Entropy

Additional informations about networks are provided by the entropy of the degree distribution [29]:

Definition 1.2.11. The **entropy** of the degree distribution is defined as:

$$H = -\sum_{k=1}^{N-1} P_1(k) \log P_1(k)$$

Its maximum value

$$H_{max} = \ln(N-1)$$

is obtained when $P_1(k) = \dfrac{1}{N-1}$ $\forall k$ (uniform degree distribution).
The minimum value

$$H_{min} = 0$$

is achieved for $P_1(k) = \{0, \dots, 0, 1, 0, \dots, 0\}$, i.e. when all vertices have the same degree.

Entropy is a very important concept in all scientific disciplines and it is related to how much disorder and information are present in a system. Dealing with networks, the entropy of the degree distribution gives an average measure of graphs heterogeneity and it is studied because this is an excellent measure of networks resilience to random failures [29].

1.3 Scale-Free and Small-World properties

1.3.1 Scale-Free Networks

Many graphs show the *Scale-Free Property*, [11]:

$$P_1(k) \approx k^{-\gamma}, \qquad \text{with } 0 < \gamma < 2.$$

This means that some vertices, called **hubs**, are highly connected while others have few connections, with an insignificant degree.

A scale-free graph shows a high stability against perturbations to randomly chosen nodes and a fragility against perturbations to highly connected ones.
In fact the major hubs are usually followed by smaller ones, which are linked to other nodes with an even smaller degree and so on.

Therefore if an attack occurs at random, and there are more small-degree nodes than hubs in the network, the probability that a hub would be affected is almost negligible. Even if a hub-failure takes place, the remaining hubs will keep network connectivity.
On the other hand, if there are several hubs, and we remove them, the network becomes a set of isolated graphs.
Thus, hubs are both a strength and a weakness of scale-free networks [1].

Networks	**Nodes**	**Links**
Hollywood	Actors	Appearance in the same movies
Research collaborations	Scientists	Co-authorship of papers
Protein regulatory network	Proteins that help to regulate a cell's activities	Interactions among proteins

Table 1.1: Examples of Scale-Free Networks [9].

1.3.2 Small-World Networks

Watts and Strogatz noticed that many real world networks, as road maps, food chains, networks of brain neurons, have a small average shortest path length, but a clustering coefficient higher than expected by random chance [30].
This is due to the typical richness of hubs and implies that most vertices can be reached from the others through a small number of edges: this is called *Small-World Property*, [13], [30].

This idea comes from an experiment made by Milgram in 1967: he noticed that two randomly chosen American citizens were connected by an average of six acquaintances.
In linguistic networks the average minimum distance between two vertices is

about three, in spite of the huge number of words existing in human language.

Small-world networks are also more robust to perturbations than other network architectures. In fact the fraction of peripheral nodes in S-W case is higher than the fraction of hubs and so the probability of deleting an important node is very low. Some researchers such as Barabási hypothesized that the prevalence of S-W structures in biological systems implies an advantage against mutation damages or viral infections [9].

1.4 Linguistic networks

Word interactions in human language can be well represented with the help of networks: in particular, we are most interested in written texts as books and novels. This approach to literature has been the subject of research in the last years [3] [4] [5] [6] [20] [23].

A text T can be thought as a sequence of tokens:

$$T = \{v_1, v_2, \ldots, v_N\}, \quad |T| = N$$

where v_i are words and eventually punctuation composing the text.

We call **dictionary** the set of distinct tokens:

$$D = \{l_1, l_2, \ldots, l_d \mid l_i, l_j \in T, l_i \neq l_j \ \forall i \neq j\}, \quad |D| = d.$$

So we can build a network $G = (\mathcal{N}(G), \mathcal{E}(G))$ where:

- $\mathcal{N}(G) = D$;
- $\mathcal{E}(G) = \{(j_1, j_2) \mid v_i = l_{j_1}$ and $v_{i+1} = l_{j_2}$ with $1 \leq i \leq N-1\}$;
- $\omega : \mathcal{E}(G) \to \mathbb{R}$ with $\omega_{ij} = \#\{(i, j) \in \mathcal{E}(G)\}$.

Therefore the vertices of the network are the distinct tokens and there is an edge from i to j if the word "j" follows the word "i", according to the natural reading order. The weight matrix W is a square matrix of size d and w_{ij} represents the number of times the relation between i and j appears.

The dictionary and the edges grow while reading the text: at each time step a new vertex and a new edge are introduced, or the weight of an existing edge is updated. This feature makes **dynamic** this kind of network, i.e. the network has a natural dynamical property induced by the order of the reading (i.e. not invariant by shuffling of words).

1.4.1 Topological measurements in texts

Let now see how the properties shown in the previous Sections behave in networks created by texts.
First of all, linguistic networks are Eulerian, that means that every newly introduced vertex will be joined to the last connected one and that to every new in-link corresponds a constrained out-link.
Assuming that the text is circular, this implies:

$$s_i^{in} = s_i^{out} = \frac{s_i}{2} \quad \forall i \in \{1, \dots, d\}$$

If the text is not circular it needs a restriction on i, due to the fact that this equality is not valid for the first and the last word in the text: v_1 has not an incoming link and there is not an outcoming link for v_N.

For what concerns selectivity, its definition can be rewritten using the last equality:

$$e_i^{in/out} = \frac{s_i}{2k_i^{in/out}}.$$

We can also see that

$$e_i^{in/out} \geq 1.$$

A word is strongly selective if it co-occurs with the same adjacent words.

If we consider the average of the degree seen in definition 1.2.1, it can be read in this way: the higher the average degree, the more elaborate the text; a lower value of this quantity could indicates the presence of many repetitions.

From the definition 1.2.7 applied to texts, most clustered words have neighbours also connected to each other; in the same way low values of C_i

imply not related neighbours.
Qualitatively the clustering coefficient tells if words are used in specific contexts: for example "big" is an adjective that can appear in many different fields, while the noun "bistoury" belongs to medical language. For that reason this quantity could be useful in authorship attribution, since it measures the tendency of using specific or generic words.

It is also interesting to see the difference between degree and betweenness centrality: the former shows the variety of a word neighbours, while the latter the variety of contexts joined with it. For example:

- low degree and high B_u indicate a term with the task to connect different clusters and it is not particularly related to one of them.
- high degree and low B_u characterize words that form a specific context but are not very influential nodes within the whole text.

We can also compute the two different degree distributions:

- $P_1(k)$:
$$P_1(k) = \frac{\#\{j \mid 1 \leq j \leq d, \ \text{degree}(l_j) = k\}}{d};$$
- $P_2(k)$:
$$P_2(k) = \sum_{j \ \mid \ degree(l_j)=k} p(l_j);$$

and the strength distribution $P_1(s)$:

$$P_1(s) = \frac{\#\{j \mid 1 \leq j \leq d, \ \text{strength}(l_j) = s\}}{d}.$$

We can even observe that human language graphs follow Scale-Free property. This is due to the fact that linguistic networks are growing graphs following the *rich get richer paradigm*: most connected vertices have greater probability to receive new vertices [12].

Let us give an example showing a weighted directed linguistic network:

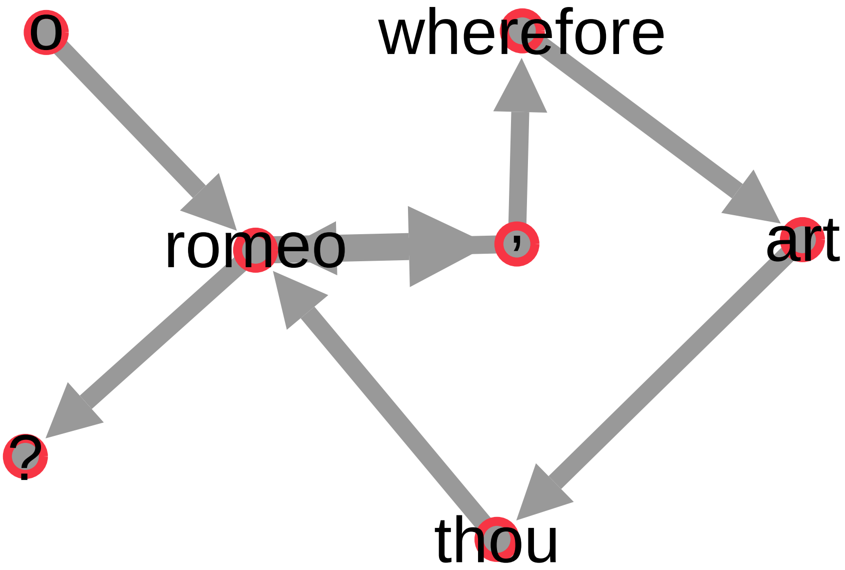


Figure 1.4: A linguistic network.

Example 1.4.1. If we consider the sentence:

"O Romeo, Romeo, wherefore art thou Romeo? "

we can find:

- T={v_1 = O, v_2 = Romeo, v_3 = ',', v_4 = Romeo, v_5 = ',', v_6 = wherefore, v_7 = art, v_8 = thou, v_9 = Romeo, v_{10} = '?'}, N=10;
- D={l_1 = O, l_2 = Romeo, l_3 = ',', l_4 = wherefore, l_5 = art, l_6 = thou, l_7= '?'}, d=7;
- $\mathcal{E}(G)=\{(l_1,l_2), (l_2,l_3), (l_3,l_2), (l_2,l_3), (l_3,l_4), (l_4,l_5), (l_5,l_6), (l_6,l_2), (l_2,l_7)\}$;

$$W = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

$$s_2^{in} = a_{12} + a_{22} + a_{32} + a_{42} + a_{52} + a_{62} + a_{72} = 1+0+1+0+0+1+0 = 3.$$
$$s_2^{out} = a_{21} + a_{22} + a_{23} + a_{24} + a_{25} + a_{26} + a_{27} = 0+0+2+0+0+0+1 = 3.$$

$$s_3^{in} = a_{13} + a_{23} + a_{33} + a_{43} + a_{53} + a_{63} + a_{73} = 0+2+0+0+0+0+0 = 2.$$
$$s_3^{out} = a_{31} + a_{32} + a_{33} + a_{34} + a_{35} + a_{36} + a_{37} = 0+1+0+1+0+0+0 = 2.$$

Chapter 2

Zipf and Heaps' Law

Since in 1940s the philologist George Zipf found a certain relation between the number of words in a text and their rank in the descending order of occurrence frequency [34] [33], many scientists tried to uncover the universal laws that govern complex systems [31] [15].
In this Chapter we will describe two important features shared by several of them, included text creation, and closely related to evolving networks.

2.1 Zipf's Law

The most elementary statistical pattern of human language is probably the frequency with which every different word appears in a written text or a speech.
Its first formulation [34] establishes that:

$$N(n) \approx n^{-\gamma} \tag{2.1}$$

where ***N(n)*** is the number of words occurring exactly n times.
The exponent γ varies from text to text, but it is often observed that $\gamma \approx 2$.

Later Zipf himself gave a different, but equivalent, formulation of his law [33]. We can rank words in a text in decreasing order by the **number of**

their occurrences, *z(r)*, that means $r = 1$ for the most frequent term, $r = 2$ for the second most frequent one and so on. If more words have the same *z(r)*, their ordering in the ranking is arbitrary.
With this operation it can be seen that *z(r)* is inversely proportional to a power of the rank r:

$$z(r) \approx z_{max} r^{-\alpha} \tag{2.2}$$

where usually $\alpha \approx 1$ and z_{max} is the maximum value for *z(r)*.

Equations (2.1) and (2.2) were both shown to be valid in many texts written in existing and extinct languages, so Zipf's law can be considered as a universal feature of human language.

Zipf explained these inverse relations using the *principle of least effort* [33]. He believed that the two agents involved in communication, the speaker (or writer) and the hearer (or reader), have to balance their work in the process.
This leads to a vocabulary where few words are used very frequently, while most of them appear few times.
In fact the speaker tends to use few words that bring different meanings, thus reducing the vocabulary. On the contrary, the hearer would prefer a different word for each meaning to better understand the message, tending in that way to increase lexical diversification.

Definition 2.1.1. The **frequency *f(r)*** of the word of rank r is defined as:

$$f(r) = \frac{z(r)}{\text{text total length}}$$

If we call f_{max} the maximum frequency value, the relation between frequency and rank is equal to the second equation given for Zipf's law:

$$f(r) \approx f_{max} r^{-\alpha} \tag{2.3}$$

As we will see even in the Chapter dealing with our results, if we plot the (2.2) in a loglog scale, we can observe three different behaviours:

- for small r-values, *z(r)* varies very slowly;
- for intermediate values of r, it is clear the power-law decreasing, that is also represented in the plot by a straight line with a slope equal to α;
- for large r, the occurrences number tends to decrease faster, creating *ladder steps*, since there are many words appearing with low frequency.

Example 2.1.1. *Moby Dick*, by Herman Melville, is a novel composed by 215937 words and with a dictionary of 17545 terms.
Its Zipf's plot is:

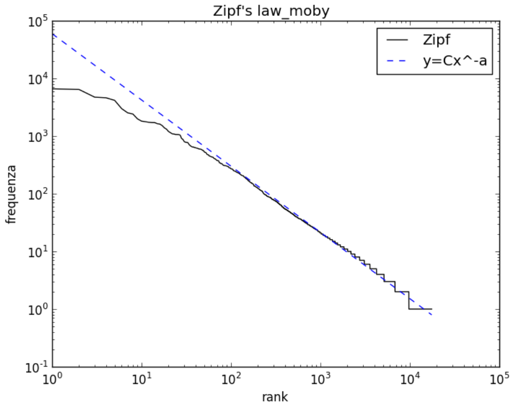


Figure 2.1: Zipf's law for *Moby Dick*, with N=215937 d=17545.

The black continuous line represents the experimental Zipf's law, while the blue dotted one is the power-law (2.2). Its slope, equal to α, has value 1.149.
As we said above, for intermediate r-values, we can see that the black line follows the power-law, while for small values it is slower and for large ones it presents ladder steps.

Zipf's power-law frequency-rank relation implies even a power-law distribution of the frequency [15]:

$$(2.4) \qquad p(f) \approx Af^{-\beta}, \qquad \text{with } \beta = 1 + \frac{1}{\alpha}, \quad A \text{ constant.}$$

Proof. Let us start from Zipf's equation $f(r) \approx r^{-\alpha}$.
We can consider words with ranks between r and $r + \delta r$, where δr is a very small value.
The number of words in the range is δr, and it can be expressed by the probability density function:

$$\delta r = p(f(r))\delta f,$$

where

$$\delta f \approx r^{-\alpha} - (r + \delta r)^{-\alpha} \approx r^{-\alpha-1}\delta r.$$

Thus, we have:

$$p(r^{-\alpha}) \approx r^{-\alpha-1} \approx (r^{-\alpha})^{-\frac{\alpha+1}{\alpha}}.$$

If we call $\beta = \dfrac{\alpha+1}{\alpha}$ we obtain the distribution of the frequency. □

2.2 Heaps' Law

Another experimental law that characterizes natural language processing is Heaps' law. It predicts the vocabulary size of a document starting from the size of the document itself, that is the number of words it contains.

We denote:

- t an instant during the text reading;
- *c(t)* the number of different words appeared until time t.

Since a text is composed by a finite and discrete set of tokens, we can think that only a word is read at each time.
That means that t is equal to the total number of words seen till the instant t itself.

Heaps' law of 1978, [14] is a power-law relation:

$$c(t) \approx t^{\lambda}, \qquad \text{with } \lambda < 1. \tag{2.5}$$

The law expresses the dependence of the total number of distinct words on the text total length: it can be interpreted as the way in which the former quantity grows as increasingly long parts of the text are considered.

Definition 2.2.1. The **average rate** of appearance of new words in a text, ρ, is the probability that the word at any given place in the text occurs there for the first time [31].

Using this definition, Heaps' law expresses the decay of ρ:

$$\rho(t) = \rho_0 t^{\lambda-1}$$

where ρ_0 is a constant lower than one.

The rate at which new words appear is not constant, and that implies the sub-linear growth of the vocabulary given by Heaps' law.

In a loglog plot, it is possible to identify a faster growth of the dictionary size in the first part of the plot, due to the fact that at the beginning of the reading almost every word is new and has to be added to the vocabulary.

Example 2.2.1. Coming back to the novel *Moby Dick*, its Heaps' plot is:

The black continuous line represents the dictionary growth, while the blue dotted one is the power-law.
We can compute the exponent λ, that has value 0.624. This is also the slope of the blue dotted straight line.
We can notice that the experimental line follows the power-law only for large values of t, while for $t < 5000$ the growth is too fast.

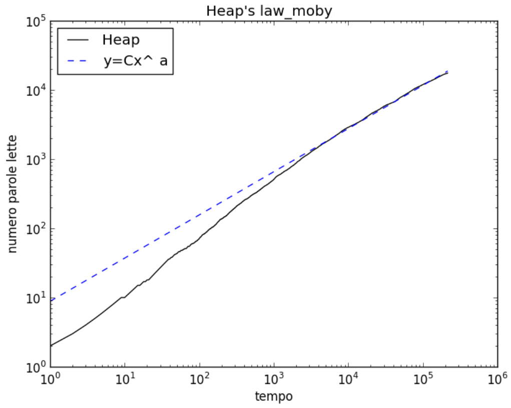


Figure 2.2: Heaps' law for *Moby Dick*, with N=215937, d=17545.

2.3 Relation between Zipf and Heaps' Exponents

The statistical features described above were discovered independently, but later their coexistence became increasingly clear. This led to analyse their relation, using especially stochastic models [26] [18] [32], but also directly with experimental results [31].

First of all we can consider the connection that lies between the exponents, α and λ. A proposition that comes from analytical results, [15], states:

Proposition 2.3.1. For an evolving finite size system with a stable Zipf's exponent:

$$\lambda = \begin{cases} 1/\alpha, & \alpha > 1, \\ 1, & \alpha < 1. \end{cases}$$

Proof. Let t and *c(t)* be the quantities defined to give Heaps' law, *f(r)* the frequency of the word with rank r.

$r - 1$ is the number of different words with frequency bigger than *f(r)*, in fact ranking is done in decreasing frequency order. We have:

$$r - 1 = \int_{f(r)}^{f_{max}} c(t)p(f')\, df' \tag{2.6}$$

where *p(f')* comes from equation (2.4).

Since *p(f)* is a probability distribution:

$$\int_{1}^{f_{max}} p(f)\, df = 1.$$

This implies that, if $\beta > 1$ and $f_{max} >> 1$:

$$A = \frac{\beta - 1}{1 - f_{max}^{1-\beta}} \approx \beta - 1$$

where the last approximation comes from $f_{max}^{1-\beta} \approx 0$.

Now we can rewrite the (2.4) as:

$$p(f') = (\beta - 1)f'^{-\beta}.$$

Substituting it in (2.6), and computing the integral we find:

$$r - 1 = c(t)[f(r)^{1-\beta} - f_{max}^{1-\beta}]. \tag{2.7}$$

Using now the Zipf's law (2.3), and the relation between Zipf's and power-law exponents, $\beta = 1 + \frac{1}{\alpha}$, we can write the second part of the last equation in term of f_{max} and α:

$$\begin{aligned} f(r)^{1-\beta} - f_{max}^{1-\beta} &= f_{max}^{1-\beta}[r^{-\alpha(1-\beta)} - 1] = \\ &= f_{max}^{-1/\alpha}(r - 1). \end{aligned}$$

Then:

$$r - 1 = c(t)f_{max}^{-1/\alpha}(r - 1).$$

Coming back to the equality (2.7), we can now obtain the estimation of f_{max} :

$$f_{max} \approx c(t)^{\alpha}.$$

Since t is the size of the considered text part, it is equal to the sum of all words occurrences:

$$t = \sum_{r=1}^{c(t)} f(r)$$

The sum can be approximated by the integral:

$$\approx \int_1^{c(t)} f(r)\,dr \;=\; \frac{f_{max}(c(t)^{1-\alpha} - 1)}{1-\alpha}$$

If we substitute here the estimation of f_{max}, we obtain:

$$t = \frac{c(t)^{\alpha}(c(t)^{1-\alpha} - 1)}{1-\alpha} \tag{2.8}$$

From this equation we demonstrate the result.
In fact if α is larger than one, $c(t)^{1-\alpha} << 1$ and:

$$c(t) \approx (\alpha - 1)^{1/\alpha} t^{1/\alpha},$$

while if $\alpha << 1$, we have $c(t)^{1-\alpha} >> 1$ and:

$$c(t) \approx (1-\alpha)t.$$

Comparing these approximations with Heaps' law in (2.5), we conclude the proof. □

We can notice that the relation (2.8) obtained in the proof, is a more complex and more accurate formula for Heaps' law. It indicates that the growth of the vocabulary is not exactly described by (2.5), even if the text obeys a perfect Zipf's law. However Heaps' law is an asymptotic approximation of (2.8).

The Proposition 2.3.1 does not consider the case of $\alpha = 1$. The limitation $\alpha \to 1$, implies:

- $c(t)^{\alpha} \approx c(t)$,
- $c(t)^{1-\alpha} \approx 1 + (1-\alpha)\log c(t)$.

Substituting these quantities in equation (2.8) we obtain:

$$c(t)\log c(t) = t,$$

that can be used to compute numerical results for finite systems when $\alpha = 1$ [15].

By the Proposition 2.3.1 it is clear that large values of α corresponds to small values of λ and vice versa:

- if Heaps' exponent λ is large, the number of different words grows fast, and the total number of occurrences is distributed among a great number of distinct terms. This implies that frequency grows slowly with rank, defining a small α;
- if λ is small, Heaps' law has a low value of the slope and a flatter growth. This means that there are few different terms composing the text. High frequencies are produced because of the great number of repetitions, and this implies large values of α.

For example, inflected languages like Italian, present many different words with the same root: they have a richer vocabulary that implies a larger value of λ.
On the contrary, texts written in languages where a single form is used for different persons, for example English, would exhibit smaller values of the Heaps' exponent.

2.4 Stochastic Models

Another way to study the two statistical patterns considered in this Chapter is to use stochastic models. There are several of them explaining the process of text generation, from which is possible to derive Zipf and Heaps' laws.

2.4.1 Simon's Model

The first model we present was proposed by the sociologist Herbert Simon [26]. It simulates the dynamics of text generation as a multiplicative process, specifying how words are added to the text while it is created.
The text generation is thought as a sequence of steps. At each step t a new word is added to the text. It begins with one word at $t = 1$, so at any step the text length is t.
We call $N_t(n)$ the number of different words that appear exactly n times till step t.

The subsequent steps have to follow two dynamical rules:

1. A new word, that has never occurred in the first t steps, is added at time $t + 1$ with a constant probability ρ.
2. With probability $1 - \rho$, the word added at step $t + 1$ is chosen among the terms which have already appeared in the previous t paces. The probability that the $(t + 1)$-th word has already occurred exactly n times, is proportional to $nN_t(n)$, that is the total number of occurrences of all the words appeared exactly n times.

The rules above can be summarized by a recursive equation for $N_t(n)$:

$$N_{t+1}(1) - N_t(1) = \rho - \frac{1-\rho}{t} N_t(1) \qquad \text{for } n = 1; \tag{2.9}$$

and

$$N_{t+1}(n) - N_t(n) = \frac{1-\rho}{t}[(n-1)N_t(n-1) - nN_t(n)] \tag{2.10}$$

for $n > 1$.

These equations do not present a stationary solution: there is not an asymptotic form for $N_t(n)$ non dependent by t. However if we assume that, for large t-values, the relation

$$\frac{N_{t+1}(n)}{N_t(n)} = \frac{(t+1)}{t}$$

holds $\forall n$, we can write the solution:

$$N_t(n) = tP(n)$$

where $P(n)$ makes t-independent (2.9) and (2.10).
For small values of ρ, the solution for $P(n)$ is approximated by a power-law:

$$P(n) \approx \frac{\rho}{1-\rho}\Gamma(\gamma)n^{-\gamma}$$

where $\gamma = 1 + \dfrac{1}{1-\rho}$, and $\Gamma(\gamma)$ is the Gamma function.

Substituting this approximation in $N_t(n)$, it presents the Zipf's law form given by (2.1), or equivalently by (2.2) with $\alpha = 1 - \rho < 1$.

Hence, Simon's model obtains Zipf's law for asymptotically long texts, where r has large value, while it predicts deviations for the higher ranks, in agreement with real texts results.

2.4.2 Mandelbrot's Model

While Simon's model gives Zipf's law an important linguistic significance, Benoit Mandelbrot considered this law as a static property shared by all random arrays of symbols [18]. He supposed to let typewriting monkeys to produce a random text, using an alphabet of $M + 1$ letters including the blank space.

Blank space appears with probability p_0, while all the others M letters occur with probability $p(w) = 1 - p_0$.

If a *word* is a sequence of letters between two consecutive spaces, there are exactly M^l different words composed by l letters. The probability P_l of a word of length l decreases exponentially as l increases:

$$P_l = p_0(1-p_0)^{l-1}$$

Each of the M^l distinct words w of length l, occurs with the same frequency:

$$f(w,l) = \frac{P_l}{M^l} = \frac{p_0}{p(w)}\left(\frac{1-p_0}{M}\right)^l$$

.

We can rewrite this quantity as:

$$\begin{aligned} f(w,l) &= \left(\frac{p_0}{p(w)}\right) e^{-l[\log M - \log p(w)]} = \\ &= \left(\frac{p_0}{p(w)}\right) [e^{l\log M}]^{-[1-\frac{\log p(w)}{\log M}]}. \end{aligned} \tag{2.11}$$

Since we consider an alphabet composed by M letters, there are M different one-letter words, $M + M^2$ terms with length no greater than two, $M + M^2 + M^3$ terms with length no greater than three, and so on. The general expression for the total number of different words shorter than l is:

$$\sum_{i=1}^{l-1} M^i = \frac{M(1-M^{l-1})}{1-M}.$$

Words are ranked with respect to length. This means that, considering the quantity above, one-letter words have all ranks between 1 and M, two-letters words between $M+1$ and $\frac{M(1-M^2)}{1-M}$, etc.
Thus, a word w of length l has an average rank $r(w,l)$ given by the equation:

$$\begin{aligned} r(w,l) &= \frac{\frac{M(1-M^{l-1})}{1-M} + 1 + \frac{M(1-M^l)}{1-M}}{2} = \\ &= M^l\left[\frac{M+1}{2(M-1)}\right] - \left[\frac{M+1}{2(M-1)}\right]. \end{aligned} \tag{2.12}$$

Deriving M from the last equality, we find:

$$e^{l\log M} = \left[\frac{2(M-1)}{M+1}\right]\left[r(w,l) + \frac{(M+1)}{2(M-1)}\right]. \tag{2.13}$$

Combining the equations (2.11) and (2.13), Mandelbrot obtained [16] [17]:

$$f(w,l) = \frac{p_0}{p(w)} \left[\frac{2(M-1)}{(M+1)} \left(r(w,l) + \frac{(M+1)}{2(M-1)} \right) \right]^{-1-\frac{\log p(w)}{\log M}}.$$

Since all the quantities used in the formula are constant, apart $r(w,l)$, it can be written as a relation between frequency and rank, independent from the length l:

$$f(w) = C_1[r(w) + C_2]^{-\alpha}$$

where C_1 and C_2 are constant and the exponent is equal to:

$$\alpha = 1 + \frac{\log(1-p_0)}{\log M} < 1.$$

We can observe that $\alpha \approx 1$ if the alphabet size M is large or p_0 is small.

If we do not consider C_2, we find the power-law relation that gives Zipf's law:

$$f(w) \approx r(w)^{-\alpha}.$$

Every language has specific values for M and p_0. For example, modern European languages have $M \approx 25$ and $p_0 \approx 0.2$: thus is possible to compute C_1 and C_2 [31].

This model implies that even a random text shares Zipf's law, so this feature does not seem to be useful in discerning between a part of a real text and a random sequence of tokens. However some of its predictions are not in agreement with real samples, for example the exponent dependence on the alphabet size or the fact that all letters are used with the same probability.

2.4.3 Modified Simon's Model

Simon and Mandelbrot's models represent the two most important positions about the linguistic significance of Zipf's law. We present another model, derived from Simon's one that is able to catch more language properties and to explain even the relation between Heaps and Zipf's laws.

In fact, studying Simon's model, it is clear that it does not reproduce the faster decay at low frequencies, and explains only exponents smaller than one. However some languages, like English and Spanish, present an exponent bigger than one. Zanette and Montemurro, in 2002, modified this model by linguistically sensible assumptions, in order to correct its lacks [32].

First modification

First of all, Zanette and Montemurro observed that in Simon's model new words are introduced with a constant rate ρ. That would imply a linear vocabulary growth, as $c(t) = \rho t$. However empirical results show a sub-linear growth that can be approximated by what we call Heaps' law:

$$c(t) = \rho t^{\lambda} \qquad 0 < \lambda < 1.$$

Thus, the rate of introduction of new words is given by $\rho_0 t^{\lambda-1}$ where $\rho_0 = \rho\lambda$. This means that a new parameter λ is introduced in the model. Its value depends on author and style, explaining some differences of vocabulary growth in the same language, but it depends mostly on the degree of inflection of each language: those with many inflexions, like Latin, have higher values of λ.

Starting from Simon's equations (2.9) and (2.10), we can replace the discrete variables n and t by the continuous variables y and t respectively, and $N_t(n)$ by $N(y, t)$.
Now we can approximate equation (2.9) by:

$$\partial t \; N + \frac{1-\rho}{t} \partial y \; (yN) = 0, \tag{2.14}$$

and, if $N(1,t) = N_1(t)$, (2.10) by:

$$\dot{N}_1 = \rho - \frac{1-\rho}{t} N_1. \tag{2.15}$$

Considering the new rate, and assuming $\rho << 1 \quad \forall t$, the general solution to equation (2.14), is given by:

$$N(y,t) = \frac{\rho_0}{\lambda+1} t^{\lambda} y^{-1-\lambda}. \tag{2.16}$$

Since $N(y,t)$ and the number of occurrences of word at rank r, $z(r)$, are related by $r = \int_z^\infty N\,dy$, the Zipf's exponent resulting from equation (2.16) is $\alpha = 1/\lambda$, that is bigger than one.

Second modification

The second rule that creates Simon's model reads that the probability that a word is repeated is proportional to the number of its previous occurrences, n_i. Since a newly introduced word has not a clear influence on the context, the probability that it is used again has to be treated apart.

Thus, the second modification made by Zanette and Montemurro is the introduction for every word, of a threshold δ_i, such that the probability to see a new occurrence of word i is proportional to $\max\{n_i, \delta_i\}$.
The set of thresholds is chosen to follow an exponential distribution, so we have to specify only one parameter, the mean δ.

The effect of this modification is that words recently introduced, thus with $n_i < \delta_i$, are favoured to appear again in the text. At the same time, the threshold does not influence words with $n_i > \delta_i$.

Analytically the thresholds δ_i cannot be simply incorporated in Simon's model, but this one has to be simplified.
First of all, the thresholds δ_i are chosen equal to their mean $\delta, \quad \forall i$.

The second simplification involves the events in which a word has to be chosen among those already present in the text. There are two possibilities:

1. with probability γ, it is chosen among those with $n_i < \delta$, with uniform probability;

2. with probability $1 - \gamma$ it belongs to the set of words with $n_i > \delta$, with probability proportional to n_i.

The evolution of $N_t(n)$ for a constant value of ρ, becomes:

$$\begin{aligned} N_{t+1}(n) = {} & N_t(n) + \\ & + \frac{(1-\rho)(1-\gamma)}{t}[(n-1)N_t(n-1) - nN_t(n)] + \\ & + \frac{(1-\rho)\gamma}{t}[N_t(n-1) - N_t(n)]. \end{aligned}$$

This new equation provides a new Zipf's law expression that shows the relation between the number of occurrences *z(r)* and the rank:

$$z(r) = \frac{1}{1-\gamma}\left[\left(\frac{r}{r_0}\right)^{-(1-\gamma)/\lambda} - \gamma\right],$$

where $r_0(t) = \rho_0 t^{\lambda}/(1+\lambda)$.

This distribution presents a cut-off at $r = r_0\gamma^{-\lambda/(1-\gamma)}$, explaining in that way the faster decay of Zipf's law for large ranks.

This last model seems to support the interpretation of Zipf's law as an important language structure, it shows the connection of this one with Heaps' law and provide new information about exponents behaviour among different languages.

Chapter 3

Italian Texts and Networks

In this Chapter we will introduce the corpus we will analyse in Chapter 4. We will consider not only the real texts, but even three modifications of them, in order to extrapolate from novels much information as possible.
Moreover, thanks to a Python algorithm, novels are transformed into networks that can be visualized using the software Gephi (see Appendix A), since they are saved in GEXF format. The same algorithm, both with Gephi commands, will provide us the measures that could describe a graph and that we will study in the next Chapter.

3.1 Database

We analyse a set of six novels, written by Italian authors in a period that range from 1881 to 1923. We choose these six books because of their importance in Italian Literature and because the Italian language used to write them is quite the same we speak today.
Moreover, the fact they were published in a period of 40 years allows us to compare them and their properties, as the grammar, the syntax and the semantic would have not changed too much during this period.

In Table 3.1, a list of the considered novels:

Title	**Author**	**Year of publication**
I Malavoglia	Giovanni Verga	1881
Pinocchio	Carlo Collodi	1881
I Pirati della Malesia	Emilio Salgari	1896
Il fu Mattia Pascal	Luigi Pirandello	1904
Canne al Vento	Grazia Deledda	1913
La Coscienza di Zeno	Italo Svevo	1923

Table 3.1: The books composing our corpus.

3.2 Modeling texts as networks

To better understand the importance of the measures we introduced in Chapter 1, we compute and observe them not only in real texts but even in those obtained modifying them.

3.2.1 Normal Texts

Initially we apply two pre-processing steps:

1. capital letters are transformed in their correspondent lowercase ones. This operation, performed by the Python command

   ```
   string=string.lower
   ```

   avoids to consider different two equal words if one of them comes after a point.

2. punctuation, including the "text return", is removed from the novels in order to study only words occurrences and co-occurrences;

The steps described above produce the first kind of text we have studied, that simply consider the original text without punctuation. We call it **Normal**.

Example 3.2.1. Let us see how a sentence extracted from one of our texts becomes after the pre-processing steps.
From *Pinocchio*, by Carlo Collodi:

Real text	**Normal text**
Cammina, cammina, cammina, alla fine sul far della sera arrivarono stanchi morti all'osteria del Gambero Rosso. - Fermiamoci un po' qui, - disse la Volpe, - tanto per mangiare un boccone e per riposarci qualche ora.	cammina cammina cammina alla fine sul far della sera arrivarono stanchi morti all osteria del gambero rosso fermiamoci un po qui disse la volpe tanto per mangiare un boccone e per riposarci qualche ora

3.2.2 No Stopwords Texts

When we read a novel, a scientific work or any different kind of text, we can notice that there are words appearing very often and carrying little semantic content. These terms such as articles, prepositions and adverbs are called **stopwords**.
There are pre-compiled lists of such words for some languages, including Italian. In addition to the grammar particles already mentioned, in Italian lists we can find the complete conjugations of verbs "essere" (to be), "avere" (to have), "fare" (to make), "stare" (to stay), personal pronouns, possessive adjectives, etc.
Moreover we have to complete the lists adding female and plural forms of all the terms, since they contain only male singular forms.

The complete list we have used in our analysis is reported in Appendix B.

We obtain the second case by removing stopwords from Normal texts: the outputs of this action are called **No Stopwords Texts**. For simplicity we will call them using the abbreviation **NSW**.

This step allows us to consider only words with a significant semantic content and their interrelations. In our Python algorithm, it is performed by a function:

```
def no_stw(text):
    count=0
    inp= open('Italian_stopwordss1.txt','r')
    stw=inp.read()
    while count<2:
        i=0
        j=0
        while i<len(stw):
            j=i
            while (stw[i]!='\n'):
                i=i+1
            f=stw[j:i]
            text=text.replace(f,' ')
            i=i+1
            f=' '
        count=count+1
    inp.close()
    return text
```

The function takes a text as input, and gives the text without stopwords as output. It scans the input two times to be sure to remove all the terms in the list: while reading the texts, if a stopword is found, it is replaced by a blank space, " ".

Example 3.2.2. The NSW version of the sentence from *Pinocchio* already used in example 3.2.1 is:

Real text	**NSW text**
Cammina, cammina, cammina, alla fine sul far della sera arrivarono stanchi morti all'osteria del Gambero Rosso. - Fermiamoci un po' qui, - disse la Volpe, - tanto per mangiare un boccone e per riposarci qualche ora.	cammina cammina cammina far sera arrivarono stanchi morti osteria gambero rosso fermiamoci po volpe mangiare boccone riposarci qualche ora

3.2.3 Reduced Texts

Another way to simplify a novel is removing from it all terms having only an occurrence in the whole text.
This operation is done again with the intent to extract the words which convey more significance. In fact if a term is present only once, probably it is not important within the text and it could be removed without altering the global sense of the considered book.
Applying this step after the pre-processing ones provides us **Reduced Texts**, that are Normal Texts without words appearing only once.

Example 3.2.3. The Reduced sentence from the extract in Example 3.2.1 is simply:

"cammina cammina cammina un per un per".

In fact those are the only three words which appear more than once.

3.2.4 Shuffled Texts

A novel is not merely a collection of words. It is also a set of grammatical and syntactic rules, expressions proper to the writer and interaction between words, essential to create the text structure, style and sense.
If we take a novel and randomly change the order in which terms appear, we get a new text having the same original dictionary but not presenting any grammatical feature or any sense.
We can built this new text model, called **Shuffled**, by the Python command

```
random.shuffle(text).
```

Many different versions of the novel can be built using this instruction, and we study them to catch measures and laws able to distinguish a masterpiece from a casual words sequence.

Example 3.2.4. A shuffled version of the extract from *Pinocchio* is:

Real text	**Shuffled text**
Cammina, cammina, cammina, alla fine sul far della sera arrivarono stanchi morti all'osteria del Gambero Rosso. - Fermiamoci un po' qui, - disse la Volpe, - tanto per mangiare un boccone e per riposarci qualche ora.	tanto fine stanchi ora riposarci un arrivarono mangiare qualche gambero osteria cammina fermiamoci alla la volpe po un e della del rosso cammina disse cammina all sera sul qui morti boccone far per per

3.2.5 Network formation

After transforming original novels in the four text models mentioned above, we build the networks as seen in Chapter 1. It is especially important to remind that nodes are all the distinct words composing texts and that links are created between adjacent vertices.

Numerically, the core of this part consists of three steps:

- to form the dictionary;
- to form the weight matrix;
- to create the Gephi file, in GEXF format, which help to visualize the graph.

The dictionary is built scanning the text and saving every word never seen before. The function doing this is:

```
def create_diz(text):
    DIZ=[]
    DIZ=DIZ+[text[0]]
    for Parola in text:
        count=0
        for k in range(len(DIZ)) :
            if Parola==DIZ[k] :
                count=count+1
        if count==0 :
                DIZ=DIZ+[Parola]
    return DIZ
```

The matrix and the GEXF file are built at the same time in a unique Python function, given in Appendix C. In fact they both need to be updated while reading the text, to catch links between nodes and the exact moments these are created.

Since literary networks are dynamic, the GEXF file representing the novel not only contains nodes and edges, but it is written to show step by step what happens to them: at every time t, if we read a new word, a vertex is created;

otherwise a new link is formed.

Example 3.2.5. Let us consider the first chapter of *I Pirati della Malesia*, by Emilio Salgari. The graph of its Normal version is composed by 919 nodes and 1844 edges, and the total number of words in the text is 2071.

At time 10, i.e. after reading 10 words, the chapter looks like Figure 3.1. Then the network grows, and at times 30, 60, 250, 1000, 1750, 2071 we can see what is shown in Figures 3.2, 3.3 and 3.4.

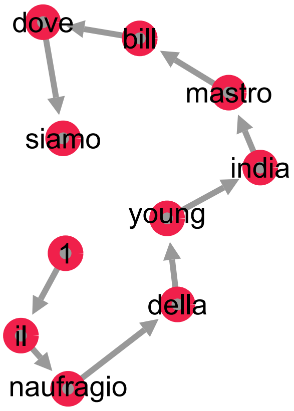
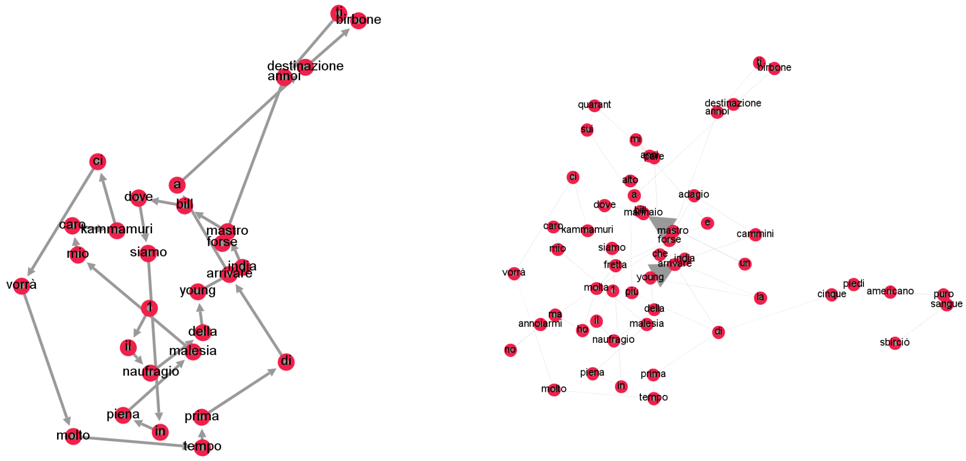
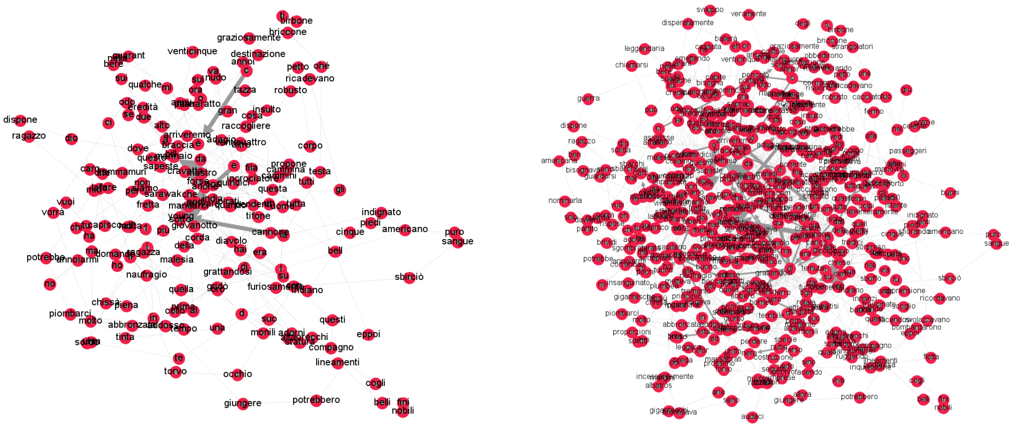


Figure 3.1: First chapter of *I Pirati della Malesia*, t= 10.



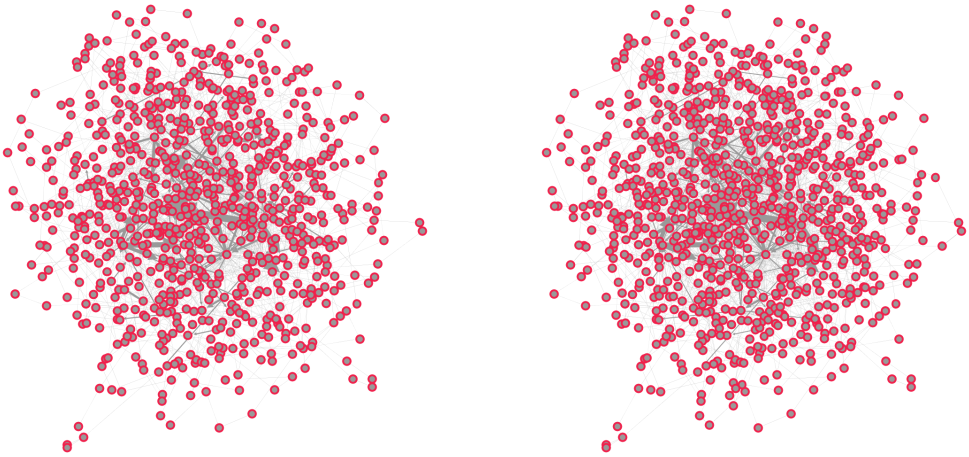
(a) t=30 (b) t=60

Figure 3.2: First chapter of *I Pirati della Malesia*, t= 30 and t= 60.



(a) t= 250 (b) t= 1000

Figure 3.3: First chapter of *I Pirati della Malesia*, t= 250 and t= 1000.



(a) t= 1750

(b) t= 2071

Figure 3.4: First chapter of *I Pirati della Malesia*, t= 1750 and t= 2071.

Chapter 4

Results and Discussion

In this Chapter we will explore how the topological measurements and the statistical patterns that we studied in the previous Chapters behave in the texts composing our corpus.
Each measure will be computed in Normal, Shuffled, No Stopwords and Reduced versions of every text, and then these values and their distributions will be analysed, in order to understand which measurements are able to recognise the real text among the others.
Measurements, laws, average quantities and distributions that could describe a graph are provided by the same algorithm we used in Chapter 3, and by Gephi commands.

First of all, we can consider the reduction in text length and in dictionary size obtained when novels are transformed in one of the different versions above. In Tables 4.1 and 4.2 we can read the results.

It can be noticed that Normal and Shuffled texts share the same values. In fact the latter are created only changing words order.

For what concerns No Stopwords and Reduced novels, they both reduce text and dictionary lengths:

Title	**Normal**	**NSW**	**Shuffled**	**Reduced**
I Malavoglia	88308	40479	88308	84599
Pinocchio	40574	19170	40574	37287
I Pirati della Malesia	59183	30864	59183	54762
Il fu Mattia Pascal	76178	36086	76178	70012
Canne al Vento	60508	29594	60508	56249
La Coscienza di Zeno	145475	65767	145475	138356

Table 4.1: Text length.

Title	**Normal**	**NSW**	**Shuffled**	**Reduced**
I Malavoglia	7631	7285	7631	3922
Pinocchio	5996	5688	5996	2709
I Pirati della Malesia	8398	8047	8398	3977
Il fu Mattia Pascal	10942	10598	10942	4776
Canne al Vento	8115	7796	8115	3856
La Coscienza di Zeno	13820	13464	13820	6701

Table 4.2: Dictionary size.

- NSW sizes are approximately the half of their relative Normal ones, while lengths of Reduced texts decrease but not so much. This implies that stopwords compose a great part of novels, and that words are usually repeated more than once.

- Stopwords deletion does not vary too much the dictionaries size, while Reduced vocabularies are smaller than the others. We can notice that, in the case of Reduced texts, the reduction in dictionaries length is exactly the same in texts length.

4.1 Frequency

We can now explore the frequency of the words composing our texts. In this Section, with word frequency we mean the number of times a word appears in the novel.
Let us start with a table that gives texts average frequency, Table 4.3.

Title	**Normal**	**NSW**	**Shuffled**	**Reduced**
I Malavoglia	11.572	5.556	11.572	21.570
Pinocchio	6.767	3.370	6.767	13.764
I Pirati della Malesia	7.047	3.835	7.047	13.770
Il fu Mattia Pascal	6.962	3.405	6.962	14.659
Canne al Vento	7.456	3.796	7.456	14.587
La Coscienza di Zeno	10.526	4.885	10.526	20.647

Table 4.3: Average frequency.

Normal and Shuffled texts have the same average frequency, since they are composed exactly of the same words.
NSW novels have a lower frequency, above the half. In fact we delete stopwords that, as shown in the following tables, are very frequent, thus the average decreases.
Reduced values, on the contrary, are higher, quite twice of Normal values. This is obvious, since less frequent words are deleted and then the average increases.

It is also interesting seeing how most frequent words change as we transform the texts. We computed the five most frequent words for every novel, and for all the versions.

Observing Tables 4.4, 4.5, 4.6, 4.7, 4.8 and 4.9, we can extract some information.

I Malavoglia	**1**	**2**	**3**	**4**	**5**
Normal	e	che	la	a	di
NSW	ntoni	don	casa	padron	mena
Shuffled	e	che	la	a	di
Reduced	e	che	la	a	di

Table 4.4: Five most frequent words, *I Malavoglia.*

Pinocchio	**1**	**2**	**3**	**4**	**5**
Normal	e	di	che	a	il
NSW	pinocchio	burattino	povero	sempre	casa
Shuffled	e	di	che	a	il
Reduced	e	di	che	a	il

Table 4.5: Five most frequent words, *Pinocchio.*

I Pirati della Malesia	**1**	**2**	**3**	**4**	**5**
Normal	di	e	il	che	la
NSW	yanez	sandokan	rajah	tigre	kammamuri
Shuffled	di	e	il	che	la
Reduced	di	e	il	che	la

Table 4.6: Five most frequent words, *I Pirati della Malesia.*

Il fu Mattia Pascal	**1**	**2**	**3**	**4**	**5**
Normal	e	di	che	la	a
NSW	adriana	casa	via	signor	forse
Shuffled	e	di	che	la	a
Reduced	e	di	che	la	a

Table 4.7: Five most frequent words, *Il fu Mattia Pascal.*

Canne al Vento	**1**	**2**	**3**	**4**	**5**
Normal	e	di	la	il	che
NSW	efix	donna	noemi	don	giacinto
Shuffled	e	di	la	il	che
Reduced	e	di	la	il	che

Table 4.8: Five most frequent words, *Canne al Vento.*

La Coscienza di Zeno	**1**	**2**	**3**	**4**	**5**
Normal	di	che	e	non	la
NSW	essa	guido	ada	augusta	prima
Shuffled	di	che	e	non	la
Reduced	di	che	e	non	la

Table 4.9: Five most frequent words, *La Coscienza di Zeno.*

- Normal, Shuffled and Reduced texts present the same most frequent words, and they all are stopwords. We can notice that "e", "che" and "di" always appear in the lists. The other two words are definite articles or the preposition "a".
  A different word is present in *La Coscienza di Zeno*: it is "non". This presence can be explained by the role this word has in the novel, that is built on denial.

- Regarding Shuffled versions, the equality of their lists with Normal ones can be explained by the fact that shuffling only reorder words, so the number of times they appear does not change.

- At the same way, since in Reduced texts stopwords are not deleted, they maintain their frequency. Thus, the most frequent words would be the same as in Normal texts.

- A different behaviour can be notice for NSW novels. In fact, most frequent words cannot be the ones of the other cases, since here they do not appear.

It is interesting to study the words we find in this last case: often they are important terms for the novels. In fact they often correspond to characters' names, for example "pinocchio", or "yanez" and "sandokan", or "noemi". Moreover in those lists we can read words carrying with them some important significance of the novel they belong to. For example, in *Il fu Mattia Pascal* the fifth most frequent word is "forse", that can symbolize the uncertainty typical of this book. At the same way in *Pinocchio* we can find "povero" and "casa", representing respectively the condition in which characters live and the place from which Pinocchio gets away and at the same time wants to come back to.

We can continue the analysis of our Italian corpus with the study of the two statistical laws presented in Chapter 2.

4.2 Zipf's Law

Let us now consider Zipf's law, that is the relation between frequency and rank. Since in the previous Section we called *frequency* the number of occurrences, we will use the Zipf's formula 2.2.

We find that all the texts in our corpus share this statistical pattern.
We present the law obtained for *I pirati della Malesia* in Figure 4.1. The loglog plot contains both the experimental law and the power-law that approximates it.

It is interesting to see that not only Normal texts, but even the other versions follow the Zipf's law.
Clearly, it is valid in Shuffled texts. In fact, as we seen in the previous Section, frequency is not altered by shuffling process. The result for *I Pirati della Malesia* is shown in Figure 4.2.

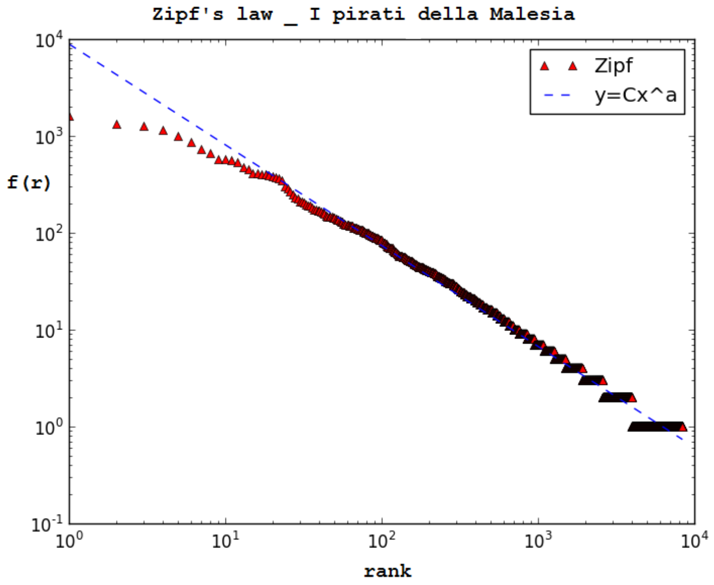


Figure 4.1: Zipf's law in *I Pirati della Malesia*, Normal text.

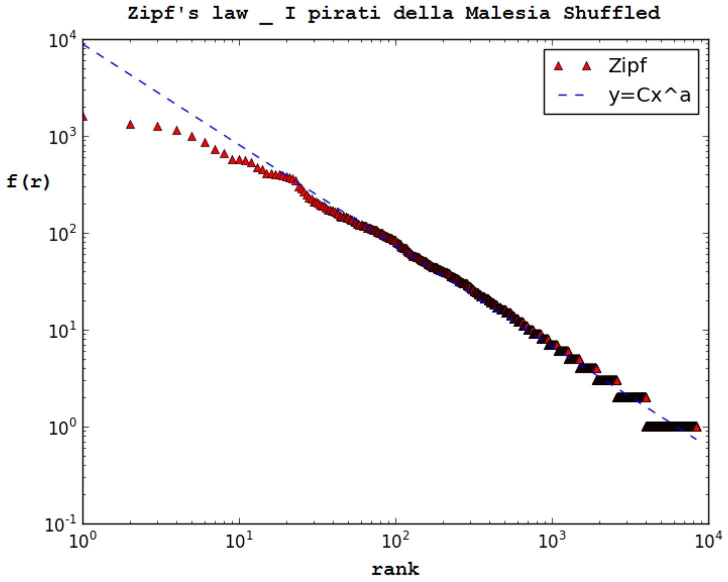


Figure 4.2: Zipf's law in *I Pirati della Malesia*, Shuffled text.

Concerning NSW and Reduced *I Pirati della Malesia* versions, we can see their Zipf's law in Figure 4.3 and 4.4.

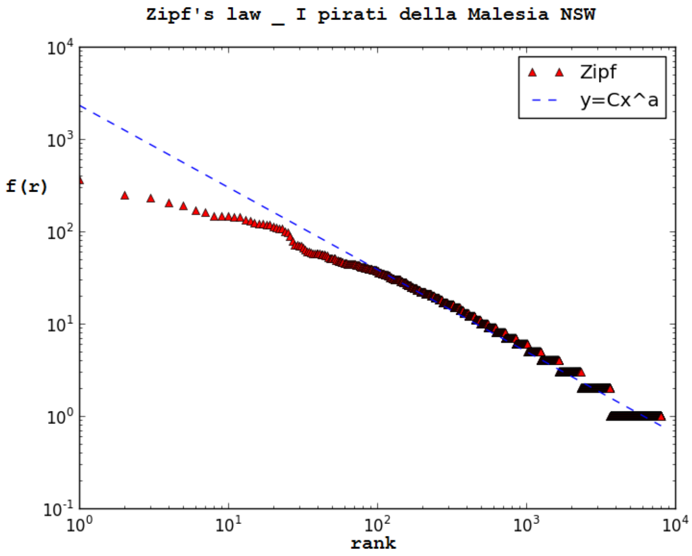


Figure 4.3: Zipf's law in *I Pirati della Malesia*, NSW text.

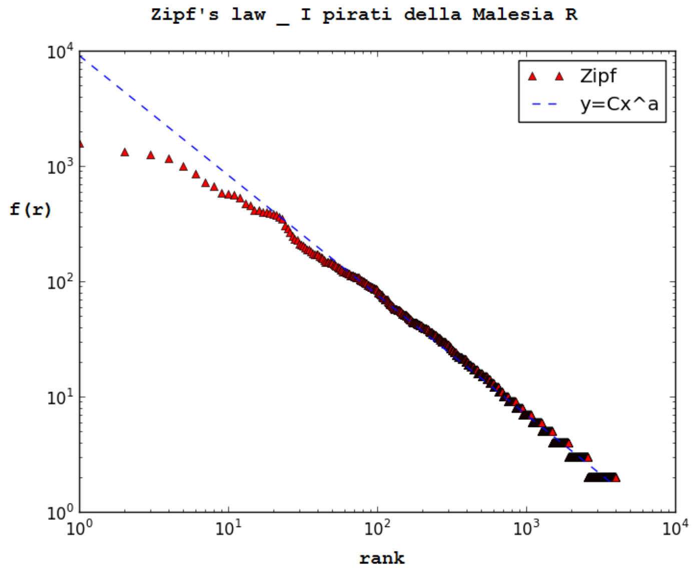


Figure 4.4: Zipf's law in *I Pirati della Malesia*, Reduced text.

Now we can explore the Zipf's exponents in our corpus, that are the exponents of the power-laws.

In Table 4.10 we can find the results, and extract by them some information.

Title	**Normal**	**NSW**	**Shuffled**	**Reduced**
I Malavoglia	1.216	1.044	1.216	1.196
Pinocchio	1.037	0.850	1.037	1.047
I Pirati della Malesia	1.040	0.889	1.040	1.038
Il fu Mattia Pascal	1.005	0.849	1.005	1.050
Canne al Vento	1.055	0.892	1.055	1.052
La Coscienza di Zeno	1.126	0.985	1.126	1.132

Table 4.10: Zipf's Exponents

- First of all we can notice that all the values in the table are close to one, in agreement with what we said in Chapter 2.

- Shuffled texts not only follow Zipf's law, but we can add that their power-law exponents are exactly the same of Normal power-law exponents, for the same reason explained above.

- NSW Zipf's exponent, on the contrary, is always lower than the others, and that can be explained with the small values of the average frequency.

- Reduced texts present two different behaviours. The exponent value computed in this case is always very close to the value of Normal case, but sometimes it is lower and sometimes higher, even if Reduced average frequency is always larger than Normal one.

Since large values of the exponent mean the presence of repetitions in the text, NSW smaller values imply that they are less repetitive. This is obvious, since stopwords are the most frequent words.
At the same time, when Reduced texts have an higher exponent, it means that deleting once-appearing words we increase the repetitions, while when the exponent is lower this operation implies a considerable presence of once-appearing terms in the text.

4.3 Heaps' Law

The second statistical law that we have studied in Chapter 2, is Heaps' law, described by equation 2.5. We remind that it gives the growth of the dictionary while reading a text. As we did for Zipf's law, we plot both the experimental results and the power-law that approximates Heaps' law in the tail.
In Figure 4.5 we can see this law for the novel *Pinocchio*.

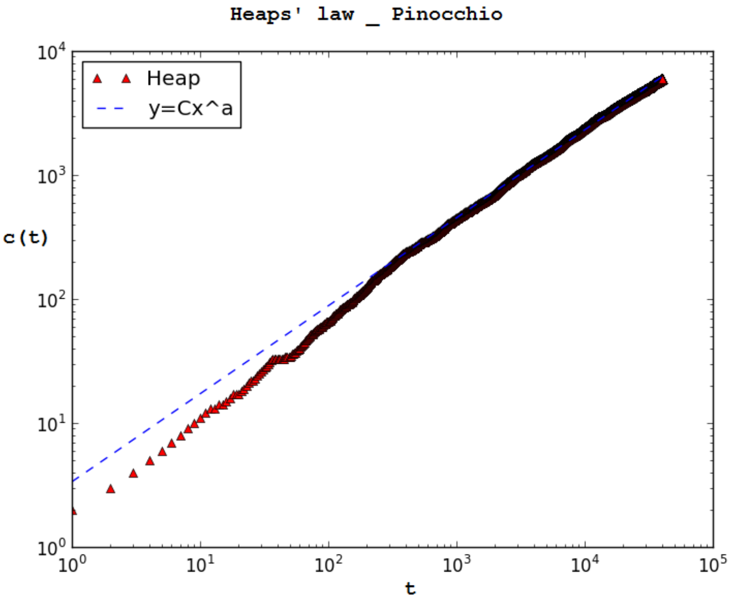


Figure 4.5: Heaps' law in *Pinocchio*, Normal text.

Again we can explore what happens when we shuffle the words composing the text. In Figure 4.6, it is presented Heaps' law in Shuffled *Pinocchio*.

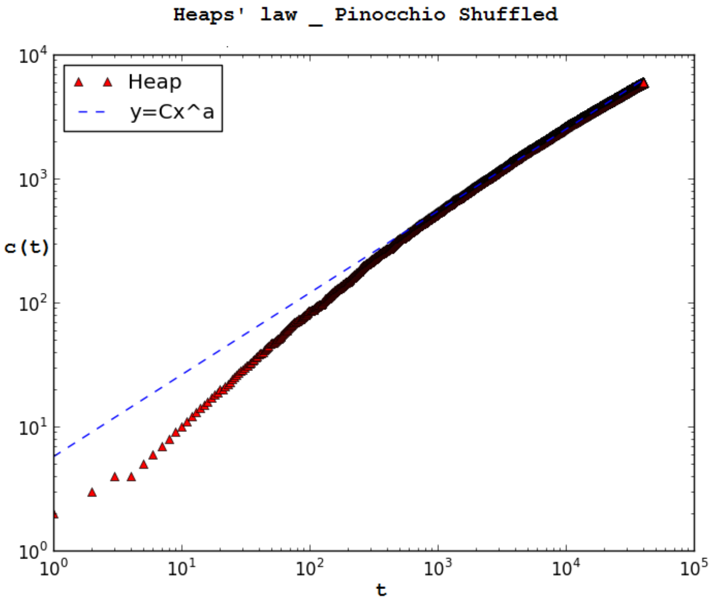


Figure 4.6: Heaps' law in *Pinocchio*, Shuffled text.

Even in texts without stopwords and in Reduced ones Heaps' law can be seen. In Figures 4.7 and 4.8 we show the plot obtained for *Pinocchio*.

In Reduced version we can notice a little shoulder at the end of the tail. It seems that the growth in the dictionary stabilizes at the end of the reading, while in Normal, Shuffled and NSW texts the growth does not stop.

We can now study the Heaps' exponent, λ, in our corpus. In fact, all the six Italian novels that we are analysing present this statistical pattern. The exponents are the slopes of the straight lines defining the approximating power-laws.
In Table 4.11 we collect the results.

First, we notice that all the exponents are lower than 1, in agreement

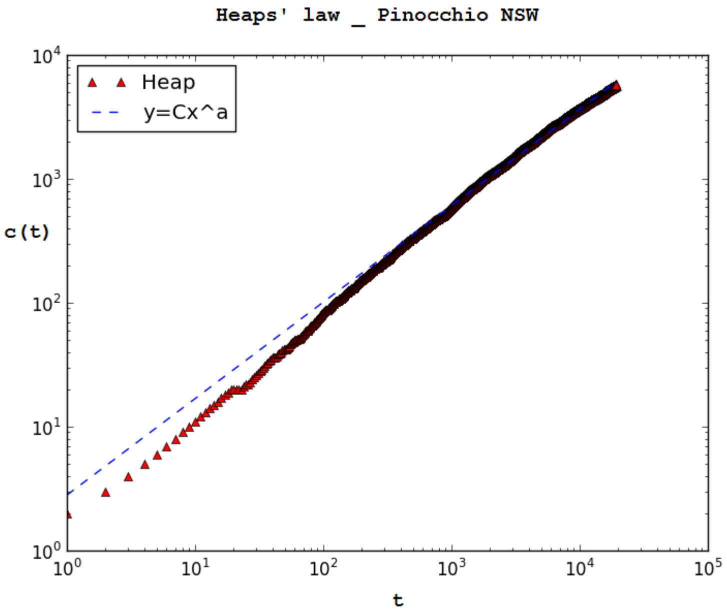


Figure 4.7: Heaps' law in *Pinocchio*, NSW text.

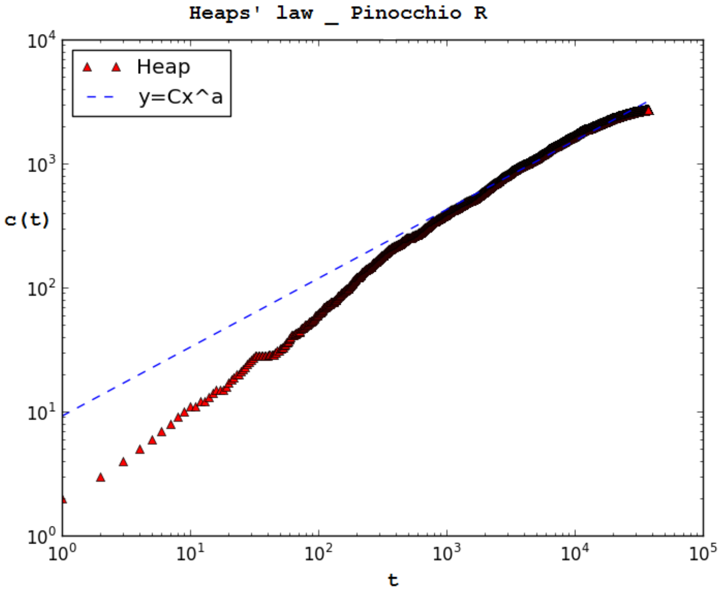


Figure 4.8: Heaps' law in *Pinocchio*, Reduced text.

Title	**Normal**	**NSW**	**Shuffled**	**Reduced**
I Malavoglia	0.572	0.629	0.589	0.469
Pinocchio	0.709	0.779	0.668	0.555
I Pirati della Malesia	0.670	0.729	0.665	0.536
Il fu Mattia Pascal	0.698	0.751	0.684	0.547
Canne al Vento	0.671	0.723	0.658	0.528
La Coscienza di Zeno	0.638	0.683	0.631	0.499

Table 4.11: Heaps' exponents.

with what we said in Chapter 2.

In contrast to what happens for Zipf's law, Heaps' law changes in Shuffled text. In fact, while the former accounts fixed quantities, the latter studies a dynamic process that is altered by shuffling. This implies that Shuffled Heaps' exponent would be different from Normal one.
Moreover, it is not possible to determine if Shuffled exponent is larger than Normal one or vice versa. In fact it depends on how the words are sorted: if there are many repetitions at the beginning of the text, later the dictionary would grow faster, causing an higher exponent. Otherwise, if many different terms appear in the firs part of the novel, then the Heaps' exponent would be lower.

We can observe that NSW exponent is always greater than Normal one. This implies that the NSW dictionary increases faster. In fact stopwords are often repeated, and appear among the other terms, lengthening the time of appearance of new words.
On the contrary, Reduced novels present lower values than real texts. The dictionary of that kind of text, in fact, contains stopwords and terms that appear more than once, favouring in such a way repetitions and slowing its own growth.

We can now consider the topological measurements seen in Chapter 1 and analyse their behaviour in our Italian corpus.

4.4 Degree

Concerning the degree, we study both the -in and the -out versions of this quantity.
Let us start with Table 4.12 and Table 4.13. They give respectively the average in-degree, $< k^{in} >$, and the average out-degree, $< k^{out} >$.

Title	**Normal**	**NSW**	**Shuffled**	**Reduced**
I Malavoglia	5.923	4.672	8.472	10.215
Pinocchio	4.239	3.011	5.489	7.748
I Pirati della Malesia	4.449	3.358	5.806	7.957
Il fu Mattia Pascal	4.526	3.166	5.559	8.643
Canne al Vento	4.767	3.467	5.988	8.573
La Coscienza di Zeno	5.766	4.487	7.520	10.398

Table 4.12: Average in-degree, $< k^{in} >$.

Title	**Normal**	**NSW**	**Shuffled**	**Reduced**
I Malavoglia	5.923	4.672	8.472	10.215
Pinocchio	4.239	3.011	5.489	7.748
I Pirati della Malesia	4.449	3.358	5.806	7.957
Il fu Mattia Pascal	4.526	3.166	5.559	8.643
Canne al Vento	4.767	3.467	5.988	8.573
La Coscienza di Zeno	5.766	4.487	7.520	10.398

Table 4.13: Average out-degree, $< k^{out} >$.

As we demonstrated in Proposition 1.2.1, in linguistic networks, $< k^{out} >$ is equal to $< k^{in} >$.

It is clear that, starting from Normal texts, degree increases in Shuffled and Reduced versions, while it decreases erasing stopwords.
In fact, shuffling process deletes semantic connections, increasing the number of different neighbours for every nodes. The same effect is obtained with the removal of words appearing only once.
NSW texts, on the contrary, have lower degree on average because stopwords are neighbours of quite all other terms.

We can explore the maximum values of out and in-degree and the words presenting them, in order to understand which terms are hubs of our networks. In Table 4.14 and Table 4.15 we collect those words and their degree.

Title	Normal	NSW	Shuffled	Reduced
I Malavoglia	e, 1547	don, 322	e, 1209	e, 1349
Pinocchio	e, 1061	pinocchio, 287	e, 811	e, 828
I Pirati della Malesia	e, 966	sandokan, 231	di, 920	e, 860
Il fu Mattia Pascal	e, 1455	adriana, 136	e, 1022	e, 1193
Canne al Vento	e, 1339	efix, 402	e, 965	e, 1159
La Coscienza di Zeno	di, 2045	guido, 445	di, 1742	di, 1769

Table 4.14: Maximum In-Degree Words.

We can notice that, apart in NSW case, the hubs are stopwords, "e" and "di", that are the conjunction and the preposition most used in Italian language. In all cases hubs belong to the list of the most frequent words, and are the first or the second element of the list. It is obvious, since words occurring many times need also many links.

Title	Normal	NSW	Shuffled	Reduced
I Malavoglia	e, 811	ntoni, 370	e, 1189	e, 681
Pinocchio	di, 646	pinocchio, 316	e, 791	di, 513
I Pirati della Malesia	di, 749	yanez, 287	di, 880	di, 632
Il fu Mattia Pascal	e, 816	adriana, 152	e, 1027	di, 670
Canne al Vento	di, 760	efix, 354	e, 986	di, 643
La Coscienza di Zeno	di, 1517	guido, 446	di, 1721	di, 1284

Table 4.15: Maximum Out-Degree Words.

Moreover, when Normal, Shuffled and Reduced versions share the same maximum in- (or out-) degree word, we can notice that:

$$k^{in}_{Sh} < k^{in}_{Red} < k^{in}_{Normal}$$

and

$$k^{out}_{Red} < k^{out}_{Normal} < k^{out}_{Sh}.$$

It is obvious that Reduced maximum is lower than Normal maximum, since we delete links. If we consider even NSW maximum, this is the lowest among all: in fact removing stopwords produces a network with few connections.

4.4.1 Degree Distribution $P_1(k)$

We can now compare the distributions of k^{in} and k^{out}, showing the $P_1(k)$ graph for the novel *I Malavoglia* in Figure 4.9 .

We can notice that the trend of the two quantities is the same, and this is true for every text in our corpus.

In fact:

- $P_1(k^{out})$: there are many words with a low value of k^{out}. This means that they are followed by a small number of other tokens. Those words

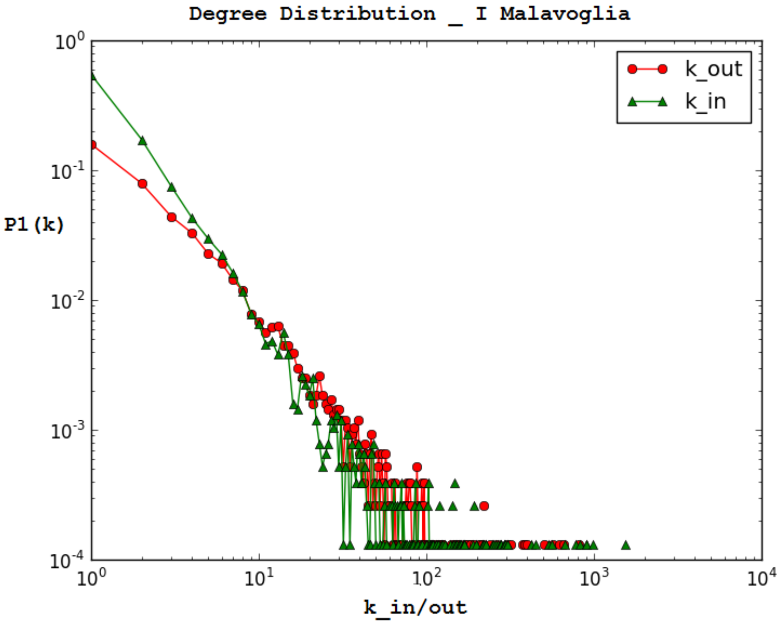


Figure 4.9: Degree distribution $P_1(k)$ in *I Malavoglia*, Normal text.

have low frequencies: they are a large number but they appear few times. Then, $P_1(k^{out})$ decreases: there are few words with a high value of k^{out}, in fact they are the most frequent ones, implying a large number of different terms following them;

- $P_1(k^{in})$: at the same way there are many words with a small number of preceding vertices. If a word appears few times, it has a low k^{in}, while when frequency increases, even the number of preceding terms grows. However there are few high-frequency words.

Moreover, the degree distribution $P_1(k)$ presents a power-law decrease, implying the Scale-Free Property for Italian literary texts. In Figures 4.10 and 4.11 we can observe the trend of in-degree and out-degree for *I Malavoglia*. The exponents of the power-law approximating the degree distributions are very close:

$$k^{in}: \ \gamma = 1.086 \qquad k^{out}: \ \gamma = 1.227.$$

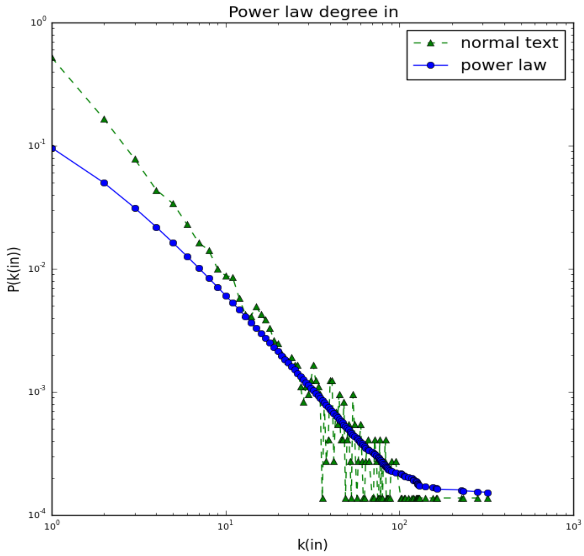


Figure 4.10: In-degree distribution $P_1(k)$ in *I Malavoglia*.

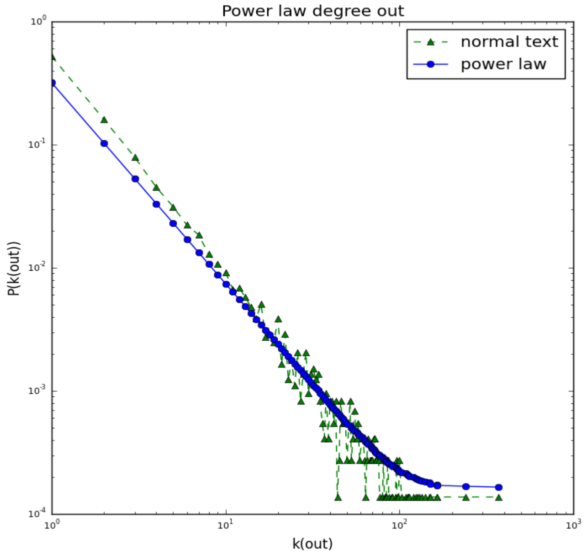


Figure 4.11: Out-degree distribution $P_1(k)$ in *I Malavoglia*.

The same behaviour can be observed in NSW, Shuffled and Reduced texts, as shown in Figures 4.12, 4.13, 4.14, that refer again to *I Malavoglia*.

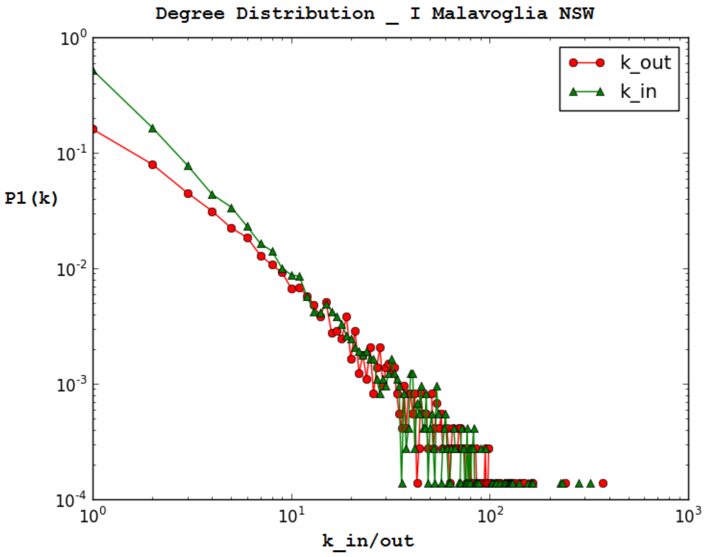


Figure 4.12: Degree distribution $P_1(k)$ in *I Malavoglia*, NSW.

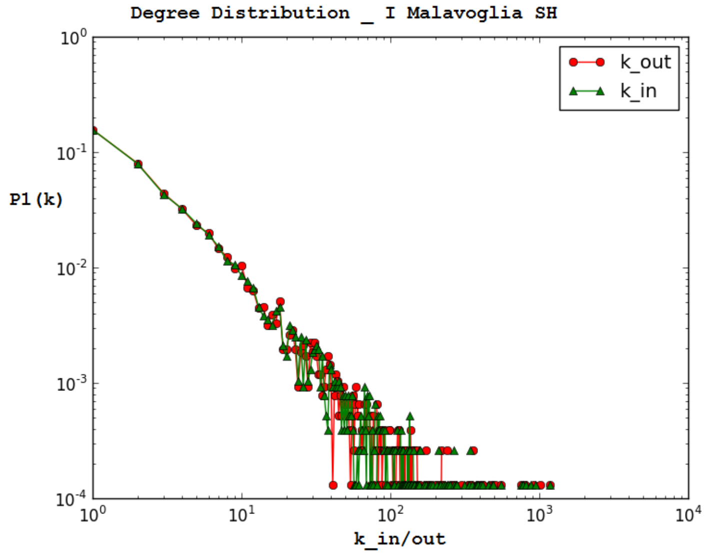


Figure 4.13: Degree distribution $P_1(k)$ in *I Malavoglia*, Shuffled.

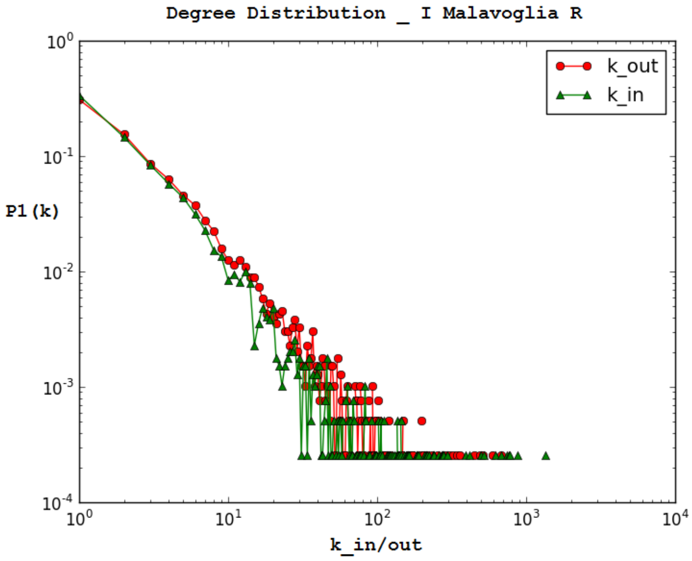


Figure 4.14: Degree distribution $P_1(k)$ in *I Malavoglia*, Reduced.

We can even compare the degree distribution of Normal and Shuffled texts.
In Figure 4.15 we can see the result obtained for the in-degree of *La Coscienza di Zeno*, but we find the same trend in all the other novels and even when dealing with out-degree.

The degree distribution is not altered by the shuffling process: even if it redistributes the numbers that occupy the rows of the weight matrix, changing the vertices degree, it preserves the average degree distribution.
This means that degree cannot distinguish between a real novel and a random sequence of words.

4.4.2 Conditioned Zipf's law and Degree Distribution

We can demonstrate the invariance of degree distribution in Shuffled texts, as we saw in Figure 4.15, using Zipf's law.
We build a particular version of this structure:

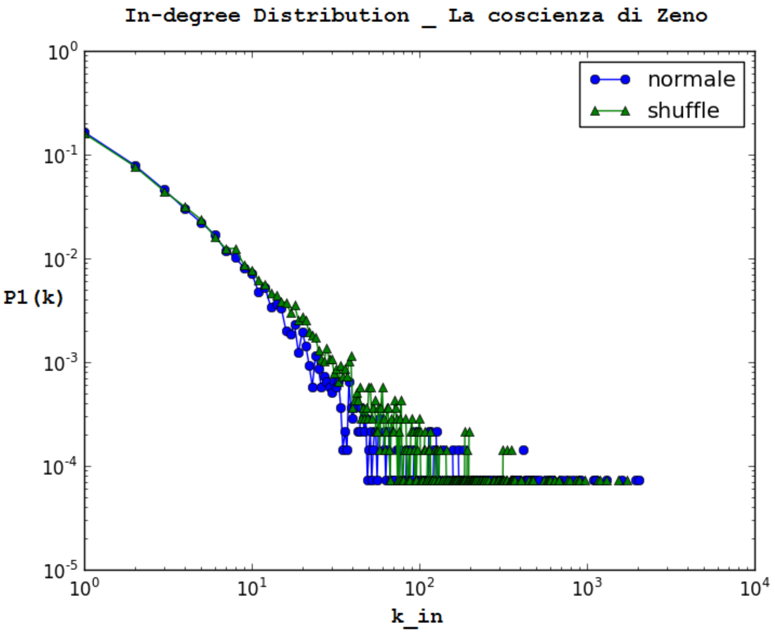


Figure 4.15: Degree distribution $P_1(k)$ in *La Coscienza di Zeno*: Normal and Shuffled.

1. We can consider the most frequent words for every text in our corpus, as we did in Section 4.1;

2. then, for every word in that list, we collect the terms following it in the considered text;

3. we compute the relation between frequency and rank in those set of words.

In such a way we can see that even the words near another term follow Zipf's law, and we call the result **Conditioned Zipf's law**, since it is achieved by fixing tokens.

Since in Figure 4.15 we present degree distribution for *La Coscienza di Zeno*, both for Normal and Shuffled versions, we can now see what happens

for its five most frequent words.
We remind that those are:

1. di;
2. che;
3. e;
4. non;
5. la.

Let us now show Conditioned Zipf's law for those five terms. First of all, in Table 4.16 we display the Zipf's exponents in Normal and Shuffled versions, in order to see their relation.

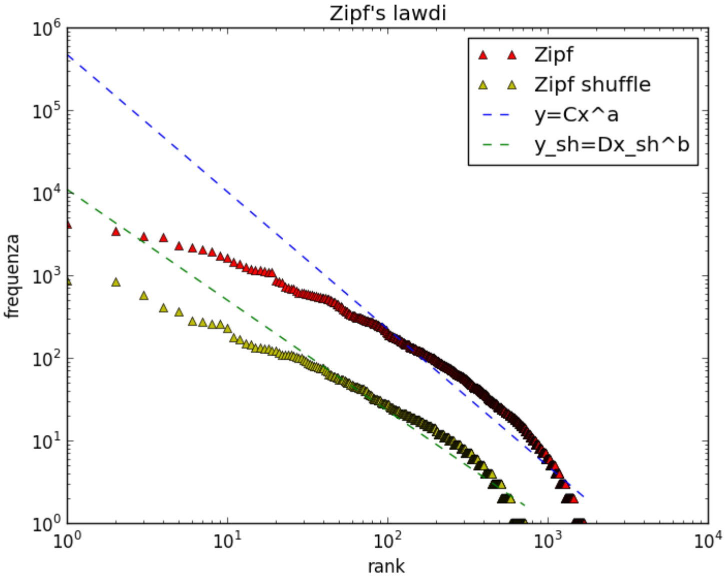
	"di"	"che"	"e"	"non"	"la"
Normal	1.507	1.921	1.800	1.798	1.340
Shuffled	1.661	1.650	1.698	1.689	1.723

Table 4.16: Conditioned Zipf's exponents for *La Coscienza di Zeno.*

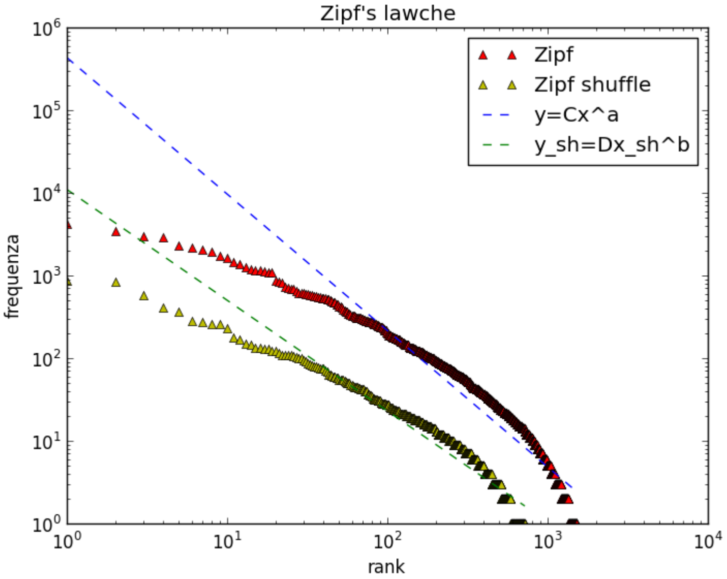
The exponents seem to be very close, and that is better shown in Figure 4.16. We plot the results obtained for the Normal and the Shuffled texts and their respective power-laws, represented as straight lines.

We can observe that Zipf's law is respected for each word. Since the terms following them are less than those composing the whole text, the trend is not so clear as for classic Zipf's law.
However, the tail is again a power-law even in the Conditioned Zipf's law.

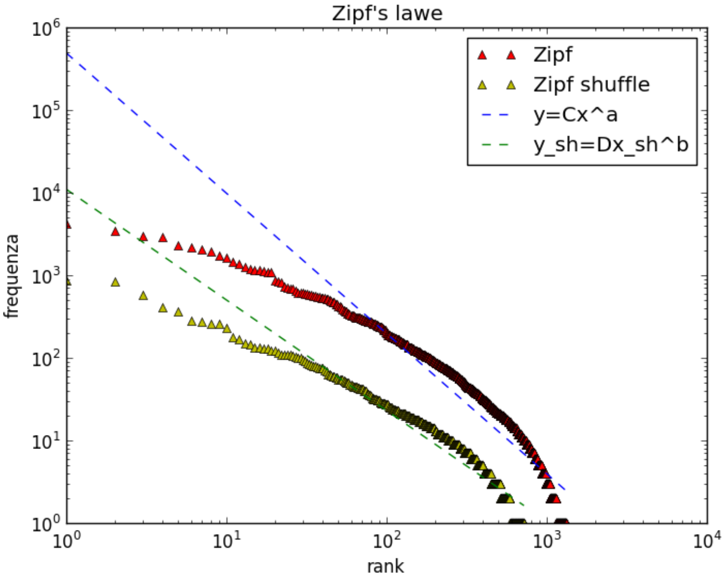
Obserivng Figure 4.16 and Table 4.16 we notice that Conditioned Zipf's law is valid both in Normal and in Shuffled texts, and that they have a similar trend. Moreover Conditioned Zipf's law is valid not only for the most



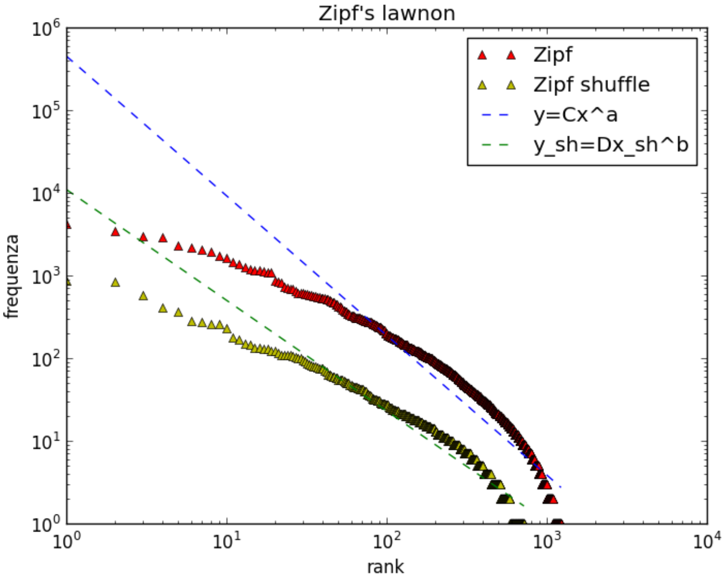
(a) Fixed Word: "di".



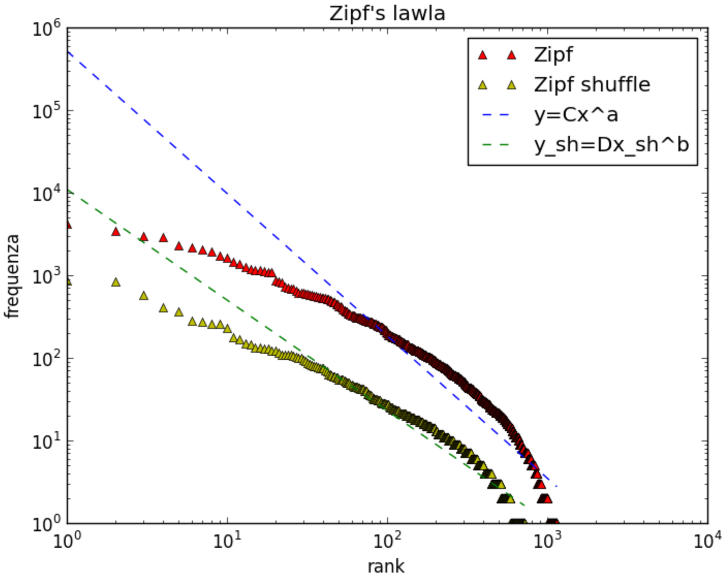
(b) Fixed Word: "che".



(c) Fixed Word: "e".



(d) Fixed Word: "non".



(e) Fixed Word: "la".

Figure 4.16: Conditioned Zipf's Law, *La Coscienza di Zeno.*

frequent word, but for more terms, shaping the whole text. Since frequencies do not change with shuffling, and, fixing a word, neither the behaviour of its neighbours changes, then degree would have the same distribution in Normal and Shuffled texts.

To complete the study, in Table 4.17 we show the Conditioned Zipf's exponents obtained for the other texts, in order to explain that the result is valid throughout the corpus.

Table 4.17: Conditioned Zipf's exponents.

(a) *Malavoglia.*

	"e"	"che"	"la"	"a"	"di"
Normal	1.760	1.815	1.352	1.559	1.543
Shuffled	1.572	1.589	1.591	1.581	1.620

(b) *Pinocchio.*

	"e"	"di"	"che"	"a"	"il"
Normal	1.680	1.374	1.718	1.450	1.241
Shuffled	1.515	1.559	1.570	1.582	1.605

(c) *I Pirati della Malesia.*

	"di"	"e"	"il"	"che"	"a"
Normal	1.419	1.724	1.293	1.766	1.255
Shuffled	1.570	1.589	1.632	1.634	1.634

(d) *Canne al Vento.*

	"e"	"di"	"la"	"il"	"che"
Normal	1.625	1.437	1.237	1.192	1.792
Shuffled	1.548	1.600	1.635	1.651	1.664

(e) *Il fu Mattia Pascal.*

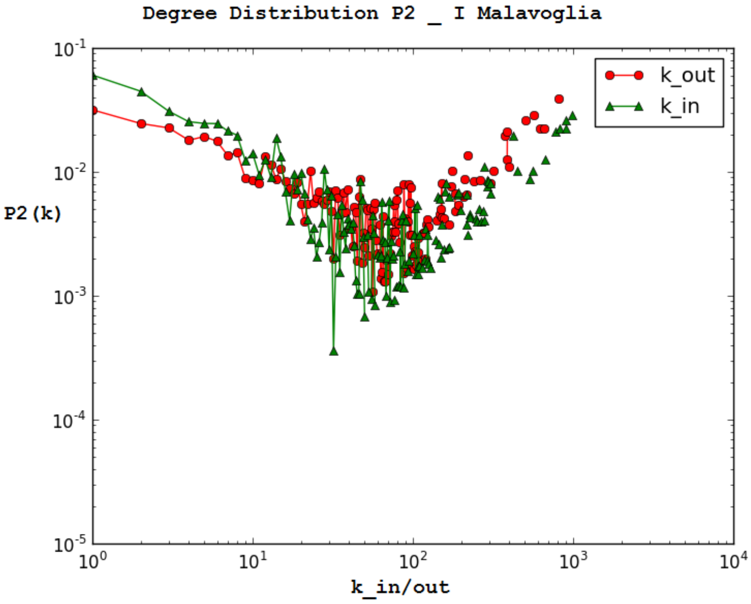
	"e"	"di"	"che"	"la"	"a"
Normal	1.763	1.552	1.834	1.296	1.630
Shuffled	1.642	1.677	1.638	1.683	1.694

4.4.3 Degree Distribution $P_2(k)$

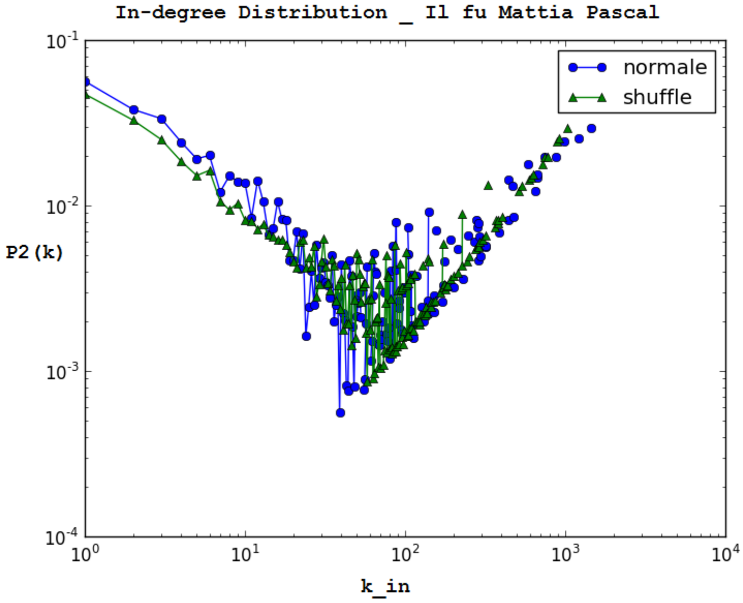
Let us consider now the degree distribution $P_2(k)$, that describes the probability to find a word with degree equal to k. Its behaviour is different from that of $P_1(k)$, as shown in Figure 4.17.

We can explain this trend studying what happens for different values of degree.

1. When we calculate the distribution for small k-values, we obtain high $P_2(k)$-values. In fact, there are many words with a low degree, thus



(a) *I Malavoglia*, Normal text.



(b) *Il fu Mattia Pascal*: Normal and Shuffled.

Figure 4.17: Degree distribution $P_2(k)$.

their sum is high, even if they appear few times.

2. For high k-values, large $P_2(k)$-values are due to the fact that we sum a small number of very frequent words. In fact, to have high degree, words need many distinct neighbours, thus they have to appear often. This implies results similar to those obtained in the case above.

3. At last, for every intermediate k-values, the number of words with degree equal to k is small while their frequency is intermediate. Thus, the summation is done with a small number of words whose frequency is not enough to reach high $P_2(k)$-values.

As in the case of $P_1(k)$, even $P_2(k)$ has the same behaviour in Normal, Shuffled, NSW and Reduced texts. Thus, neither this distribution can distinguish between real texts and their alterations.

4.5 Entropy

We can now explore the entropy of degree distribution, and consider if it can distinguish masterpieces from their modifications. We can give the values of out-entropy and in-entropy, in Tables 4.18 and 4.19, and then studying them.

Title	Normal	NSW	Shuffled	Reduced
I Malavoglia	1.806	1.932	2.186	2.529
Pinocchio	1.566	1.577	1.855	2.367
I Pirati della Malesia	1.632	1.687	1.909	2.335
Il fu Mattia Pascal	1.561	1.595	1.775	2.341
Canne al Vento	1.657	1.718	1.928	2.363
La Coscienza di Zeno	1.782	1.874	2.020	2.503

Table 4.18: In-entropy.

Title	Normal	NSW	Shuffled	Reduced
I Malavoglia	2.009	1.939	2.190	2.755
Pinocchio	1.686	1.580	1.858	2.494
I Pirati della Malesia	1.760	1.683	1.908	2.489
Il fu Mattia Pascal	1.677	1.592	1.775	2.456
Canne al Vento	1.812	1.725	1.927	2.507
La Coscienza di Zeno	1.902	1.880	2.019	2.643

Table 4.19: Out-entropy.

First of all we can notice that in Normal texts, out-entropy is larger than in-entropy. This can be explained with the help of degree distributions. In fact, looking at Figure 4.9, we can see that k_{out} has values a little higher than k_{in}.

Since entropy describes information and disorder of a network, we can deduce that:

- Shuffled texts bring less information, in fact both in-entropy and out-entropy are higher than in Normal case. This is in agreement with the definition of shuffling, that messes the terms of real novels;

- Reduced texts have higher entropies than Normal ones. This implies that removing once-appearing words deletes information and brings more disorder in texts. Actually, even if a word appears only once, it could be important to better understand a sentence, and moreover its removing produces a lack in the text structure, explaining in such way the disorder;

- No Stopwords texts present different behaviours. In-entropy has a NSW-value higher than Normal one, while out-entropy decreases with stopwords deletion. This means that in-degree carries more information in Normal texts than in NSW ones. On the contrary, out-degree brings more information in the latter case.

Thus, concerning in-entropy:

$$H_{Normal} < H_{NSW} < H_{Sh} < H_{Red},$$

while for out-entropy:

$$H_{NSW} < H_{Normal} < H_{Sh} < H_{Red}.$$

4.6 Strength

We can now study another important measure in Network Theory: strength. First of all we can consider the average strength in the four kinds of text that we are analysing.
In this Section, the strength of a vertex is given by:

$$s_i = s_i^{in} + s_i^{out} \qquad \forall i = 1 \dots N$$

where s_i^{in} is the vertex in-strength and s_i^{out} its out-strength.

The results are in Table 4.20.

Title	**Normal**	**NSW**	**Shuffled**	**Reduced**
I Malavoglia	23.144	11.113	23.144	43.140
Pinocchio	13.533	6.740	13.533	27.528
I Pirati della Malesia	14.094	7.671	14.094	27.539
Il fu Mattia Pascal	13.924	6.810	13.924	29.318
Canne al Vento	14.912	7.592	14.912	29.174
La Coscienza di Zeno	21.053	9.769	21.053	41.294

Table 4.20: Average Strength.

If we compare Table 4.20 with Table 4.3 that contains average frequencies, we can observe that strength is always twice the frequency. In fact frequency counts words occurrences, while strength is the number of connections present in the text. Clearly, every node as an in-coming and an out-coming link, thus for every occurrence two links are added.

Observing the results, we can also notice that shuffling preserves strength. In fact it is computed as the sum of the elements in W matrix, and shuffling only redistributes the numbers in the matrix rows, thus preserving the summation.

The NSW texts present a lower strength, about the half, since stopwords are neighbours of many words. Deleting them, the remaining terms lose lots of connections.

In Reduced texts, strength becomes higher. This can be explained by the fact that, without words appearing only once, the others have an higher probability to have different adjacent terms.

4.6.1 Strength Distribution $P_1(s)$

Not only the average strength is preserved with shuffling, but even its distribution, as shown in Figure 4.18 for *Canne al Vento* case.
Again, since this is valid for the whole corpus, it means that neither strength is able to distinguish between a masterpiece and a casual sequence of tokens.

At the same way, the power-law trend of the distribution is maintained even in Reduced and NSW cases, as shown in Figure 4.19.

4.7 Selectivity

Selectivity has been created to distinguish shuffled texts from real ones. It can capture the effective distribution of numbers in the weight matrix.

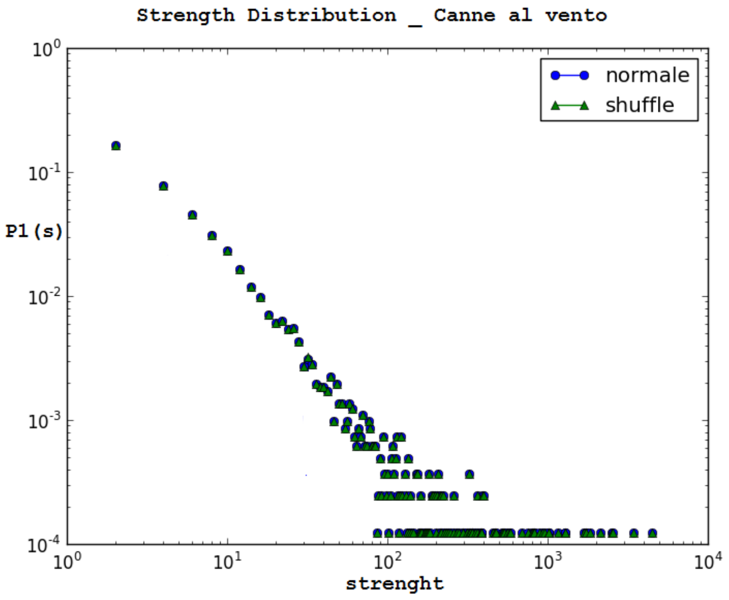
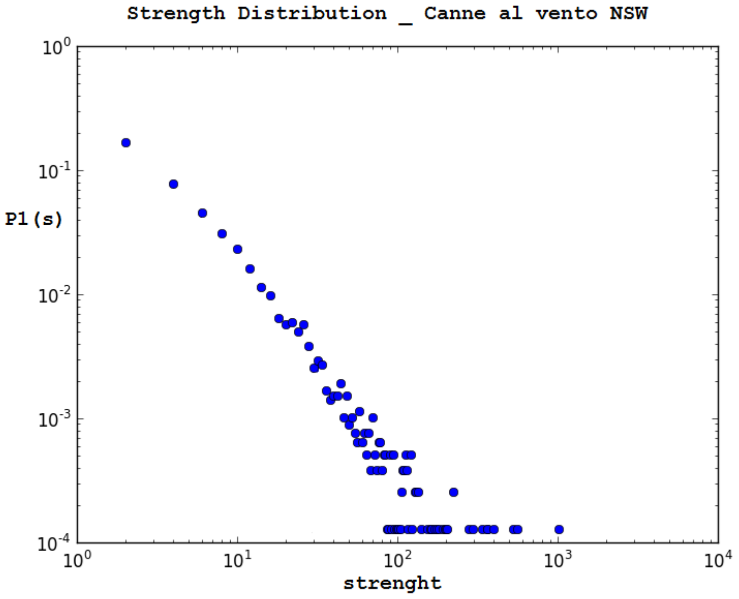
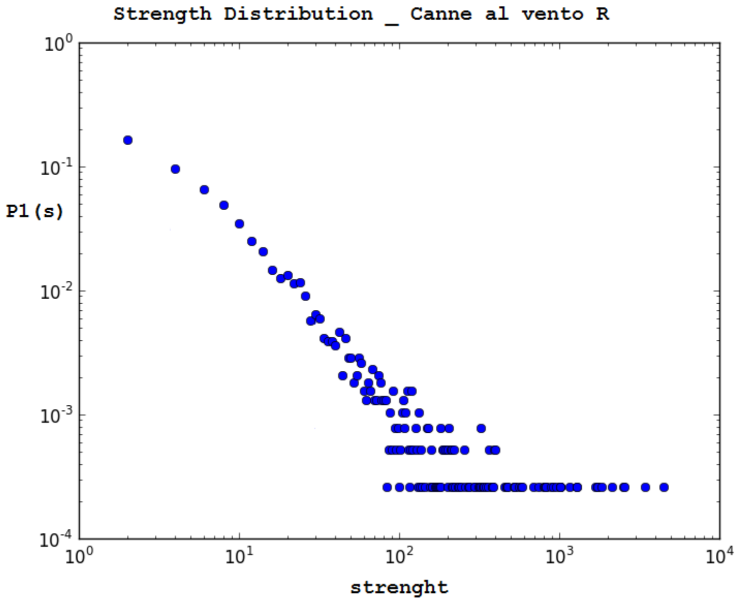


Figure 4.18: Strength distribution in *Canne al Vento*, Normal and Shuffled text.



(a) NSW text.



(b) Reduced text.

Figure 4.19: Strength distribution in *Canne al Vento*.

In fact, words with high values of this quantity, are very selective in the choice of their neighbours, usually forming *morphologic structures*.
On the contrary, tokens with small selectivity are terms that appear just few times in the text, or terms connecting with a different token each time.

For example, in Normal *Pinocchio*, the most out-selective word is "c", with $e_c^{out} = 21$. In fact it is always followed by the words "è", thus creating the structure "c' è", used very often in Italian language.

Studying minimum values of selectivity, we have:

$$\min_i e_i^{out} = \min_i e_i^{in} = 1.$$

This means that:

$$k_j^{in/out} = s_j^{in/out} \qquad \forall\, j: \quad e_j^{in/out} = 1.$$

This implies that, in this cases, the weight of the links involving j is always equal to one: these words connect with a different term each time they appear.

Regarding selectivity maximum value, it changes with text transformations.
In Tables 4.21 and 4.22, we can read the results concerning out-selectivity and try to achieve some observations.

Title	**Normal**	**NSW**	**Shuffled**	**Reduced**
I Malavoglia	83.25	83.25	2.93	83.25
Pinocchio	21.0	9.0	2.23	21.0
I Pirati della Malesia	146.0	146.0	2.09	146.0
Il fu Mattia Pascal	19.0	19.0	2.17	19.0
Canne al Vento	33.33	33.33	2.28	33.33
La Coscienza di Zeno	26.65	23.0	2.90	26.65

Table 4.21: Maximum Out-Selectivity Value.

Title	**Normal**	**NSW**	**Shuffled**	**Reduced**
I Malavoglia	padron	padron	umili	padron
Pinocchio	c	mastr	carlo	c
I Pirati della Malesia	tremal	tremal	ma	tremal
Il fu Mattia Pascal	mila	mila	cose	mila
Canne al Vento	don	don	tutto	don
La Coscienza di Zeno	ch	psico	di	ch

Table 4.22: Maximum Out-Selectivity Words.

First of all we can notice that the range of the vertex out-selectivity is much smaller in Shuffled Texts than in Normal ones, and this difference is one order of magnitude. This can be explained by the fact that, in real texts, tokens are selective in choosing their neighbours, forming specialised local structures. The lack of those structures determines the small values of selectivity in Shuffled Texts.

Concerning the other two cases, sometimes they both present the same value of Normal version, but this is not observed for all the texts. However, we can explain this behaviour with some examples:

Example 4.7.1. In *I Pirati della Malesia*, the words with higher selectivity is "tremal", $e^{out}_{tremal} = 146.0$, that composes the structure "tremal naik", the name of one of the protagonists of the novel. Neither "tremal" or "naik" are stopwords, and they clearly appear more than once, thus this structure remains unchanged in NSW and Reduced texts. This implies:

$$\max_{Normal} e^{out} = \max_{NSW} e^{out} = \max_{Reduced} e^{out}.$$

Example 4.7.2. If we now consider *Pinocchio* by Carlo Collodi, we can see that:

$$\max_{Normal} e^{out} = \max_{Reduced} e^{out} \neq \max_{NSW} e^{out}.$$

In fact, as we said above, in Normal text the most out-selective word is "c". However this is a stopword, and it is deleted in NSW version, where the most selective word is "mastr", that appear near "Antonio" at the beginning of the novel.
At the same time, "c" appears more than once, becoming the most selective word even in the Reduced *Pinocchio.*

Thus, we can deduce the following rules:

- if the most selective term is neither a stopword or a once-appearing word, then the out-selectivity range is the same for Normal, NSW and Reduced texts;
- if it is a stopword, Normal and Reduced versions share the same out-selectivity maximum value.

We can even consider in-selectivity, with values in Table 4.23 and words in Table 4.24.

Title	**Normal**	**NSW**	**Shuffled**	**Reduced**
I Malavoglia	138.0	138.0	2.88	138.0
Pinocchio	30.0	17.0	2.17	30.0
I Pirati della Malesia	146.0	146.0	2.02	146.0
Il fu Mattia Pascal	19.0	19.0	2.18	19.0
Canne al Vento	26.0	26.0	2.33	26.0
La Coscienza di Zeno	41.0	14.0	2.86	41.0

Table 4.23: Maximum In-Selectivity Value.

Title	**Normal**	**NSW**	**Shuffled**	**Reduced**
I Malavoglia	comunale	comunale	più	comunale
Pinocchio	argento	grillo	di	argento
I Pirati della Malesia	tremal	tremal	no	tremal
Il fu Mattia Pascal	don	don	poche	don
Canne al Vento	imbarcata	imbarcata	primo	andata
La Coscienza di Zeno	prima	solfato	dottore	prima

Table 4.24: Maximum In-Selectivity Words.

Again, Shuffled in-selectivity range is smaller than Normal one, with the same explanation given for out-selectivity.

Observing the values in Table 4.23 and the words in Table 4.24, we can see that Normal and Reduced in-selectivity are always equal. In *Canne al Vento* case, however two different words represent the maximum value. This can be explained by the fact that many terms can share the same in-selectivity value, and in these cases Python chooses at random the element to show.

NSW texts share the same maximum in-selectivity value of Normal and Reduced texts if their most in-selective word is not a stopword.

Thus, the rules given above for out-selectivity remain valid even for in-selectivity. However we can add a rule, observing *Pinocchio*'s behaviour:

- if the most in-selective word is not a stopword, but always comes after a stopword, then NSW would change its most selective word and decreases the in-selectivity range.

In fact, the word "argento" in *Pinocchio* is always preceded by the term "d". Obviously, in NSW version, the latter is deleted, destroying the structure "d' argento" and reducing "argento" selectivity. This implies a new most selective word.

At last, we can notice that only few stopwords are present in Table 4.24, all concerning Shuffled texts. This tells us that stopwords are more selective in choosing their successors than in choosing their predecessors. Thus, they create more morphologic structures if they come before other terms.

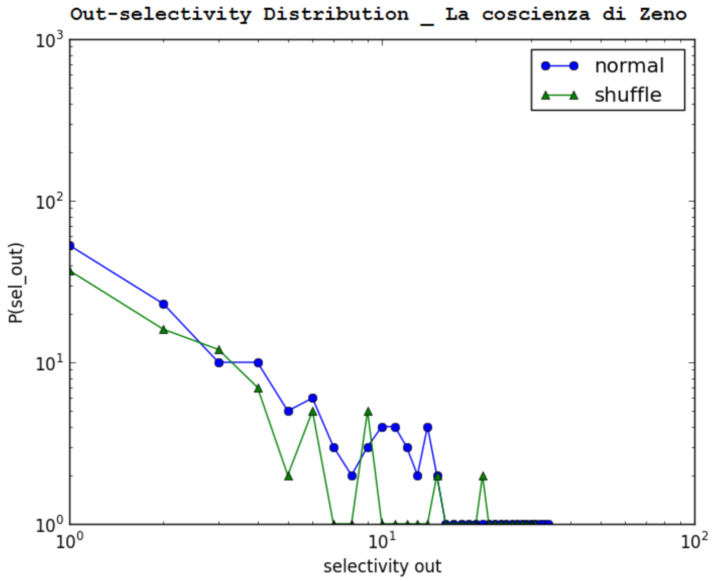
4.7.1 Selectivity Distribution

Studying the distribution of this quantity, again we can notice the same trend in Normal and Shuffled texts. This means that neither this quantity is able to distinguish between the two versions. Even the behaviour of NSW and Reduced texts is similar to that of Normal ones, implying the inability of this measure.

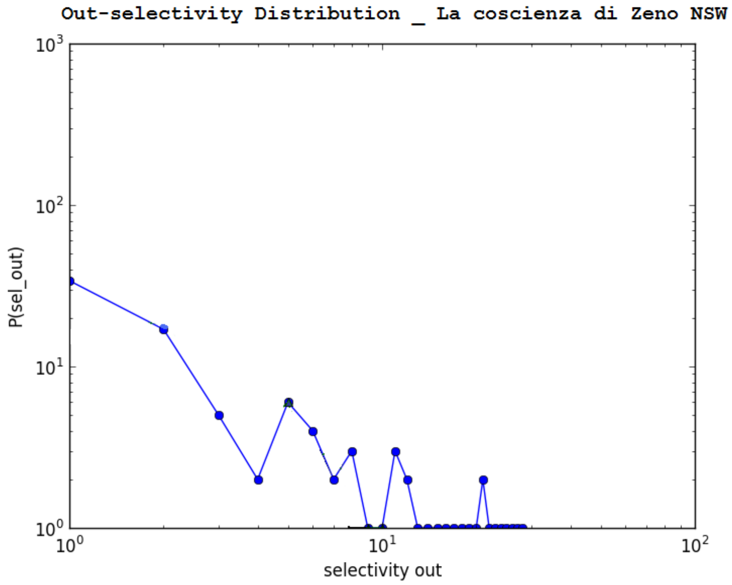
The distributions are obtained using the Python command

`numpy.bincount`.

In Figure 4.20, we can see the results achieved for out-selectivity of *La Coscienza di Zeno*.

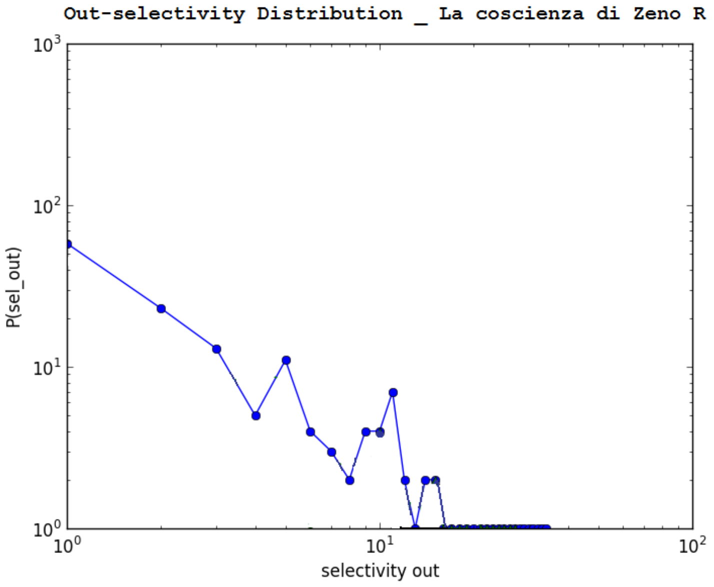


(a) Normal and Shuffled



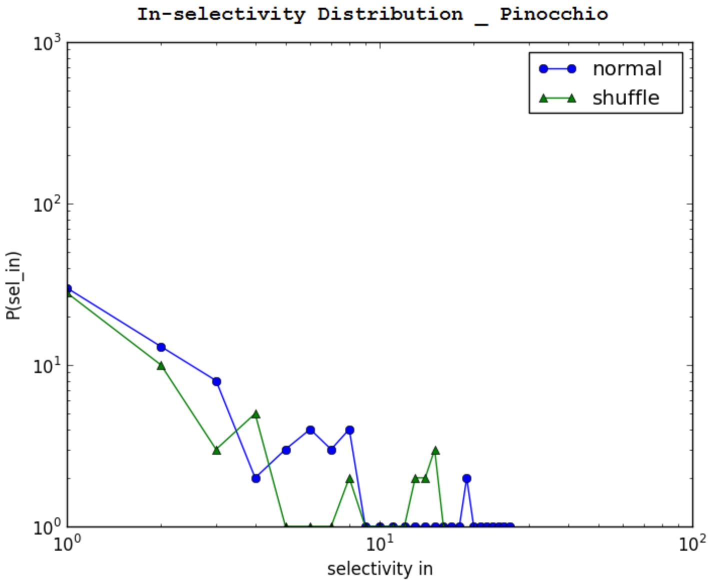
(b) No Stopwords

In Figure 4.21, are presented in-selectivity distributions for the different versions of *Pinocchio*.

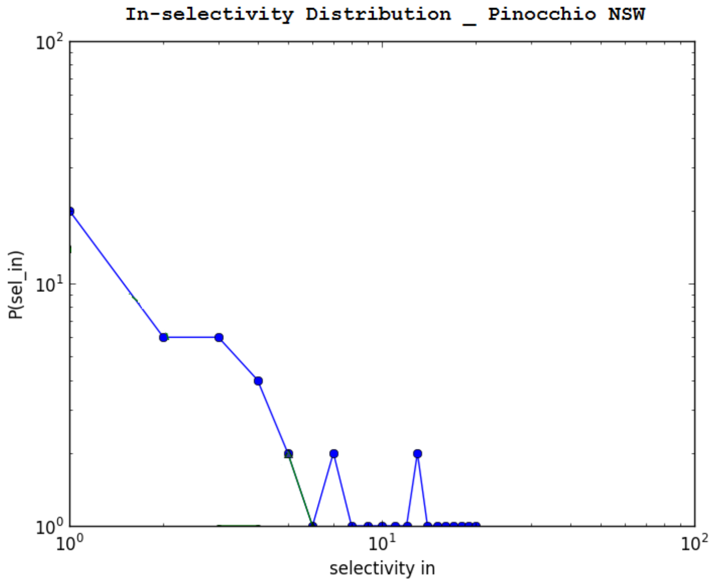


(c) Reduced

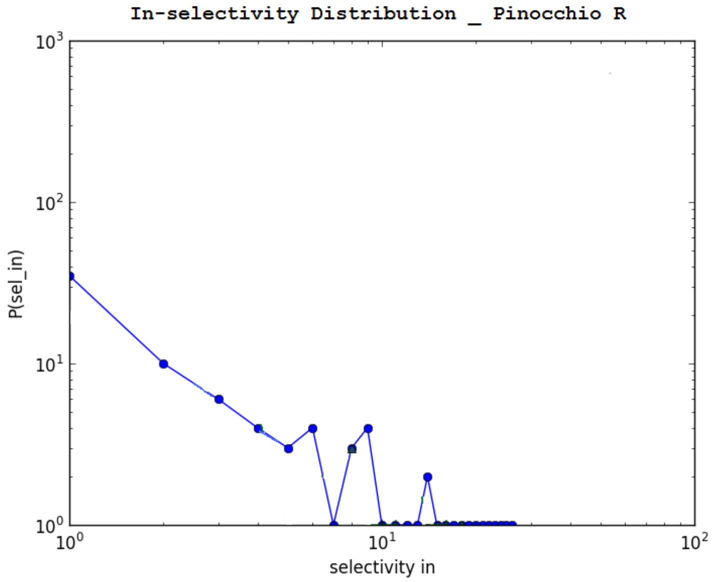
Figure 4.20: Out-selectivity distribution in *La Coscienza di Zeno*.



(a) Normal and Shuffled



(b) No Stopwords



(c) Reduced

Figure 4.21: In-selectivity distribution in *Pinocchio*.

4.8 Gephi statistics

Importing in Gephi the GEXF files created with the Python function described in Appendix C, we are able to compute some of the quantities described in Chapter 1.

4.8.1 Average Path Length

First of all we can compute measures concerning the distance between different nodes. In Table 4.25 we present the results obtained for Average Path Length.

Title	Normal	NSW	Shuffled	Reduced
I Malavoglia	3.463	4.446	3.263	2.895
Pinocchio	3.660	5.201	3.512	2.971
I Pirati della Malesia	3.764	5.007	3.529	3.055
Il fu Mattia Pascal	3.692	5.238	3.543	2.988
Canne al Vento	3.664	4.928	3.490	2.957
La Coscienza di Zeno	3.483	4.432	3.314	2.897

Table 4.25: Average Path Length.

Comparing NSW texts with their Normal versions, we notice that this quantity increases. In fact the stopwords absence causes the deletion of many edges in paths joining two different tokens.

Even in Reduced texts we remove words, reducing in that way the vocabulary. This implies a smaller number of edges, and thus a smaller value of average path length.

At last, shuffling makes texts lose their semantic connections, implying a lower value of this measure.

4.8.2 Diameter

Naturally the behaviour of diameter follows the average path length trend, since it is the longest distance between two vertices.
Thus, it becomes smaller in Shuffled and Reduced texts but higher in texts without stopwords.
The diameters of the novels we analysed are shown in Table 4.26.

Title	**Normal**	**NSW**	**Shuffled**	**Reduced**
I Malavoglia	9	19	9	7
Pinocchio	12	20	11	7
I Pirati della Malesia	13	17	14	7
Il fu Mattia Pascal	15	21	11	8
Canne al Vento	12	20	11	7
La Coscienza di Zeno	15	19	10	7

Table 4.26: Diameter.

4.8.3 Average Cluster Coefficient

Stopwords, words appearing only once and randomization make higher the probability that neighbours of a vertex are connected.
This implies:

$$C_{Red} > C_{Sh} > C_{Norm} > C_{NSW}.$$

In Table 4.27 we present the results we obtain for our six texts.

Observing these values, we can say that the texts composing our corpus are not Small-World networks.
In fact the Shuffled cluster coefficient is always bigger than Normal one, while in S-W graphs the real cluster coefficient has to be greater than random one.

Title	Normal	NSW	Shuffled	Reduced
I Malavoglia	0.208	0.041	0.333	0.338
Pinocchio	0.177	0.031	0.244	0.304
I Pirati della Malesia	0.158	0.033	0.236	0.259
Il fu Mattia Pascal	0.196	0.019	0.261	0.315
Canne al Vento	0.172	0.030	0.258	0.307
La Coscienza di Zeno	0.236	0.039	0.359	0.367

Table 4.27: Average Cluster Coefficient.

4.8.4 Network Density

Analysed texts are very sparse, in fact they all present few connections in respect to their number of nodes. Thus, Δ is negligible, showing often a value close to 0.001.
We can notice only a little increase of Δ in Reduced cases, when it assumes a value of 0.002 or 0.003, thus remaining very small.
In Table 4.28 we give the computed values.

Title	Normal	NSW	Shuffled	Reduced
I Malavoglia	0.001	0.001	0.001	0.003
Pinocchio	0.001	0.001	0.001	0.003
I Pirati della Malesia	0.001	0	0.001	0.002
Il fu Mattia Pascal	0	0	0.001	0.002
Canne al Vento	0.001	0.001	0	0.002
La Coscienza di Zeno	0	0	0.001	0.003

Table 4.28: Network density.

4.8.5 Betweenness Centrality

With Gephi we can also compute betweenness centrality, defined in Chapter 1. Thanks to its internal algorithms, Gephi provides the betweenness value of every node in the network.

We can study the words with the higher values of betweenness centrality. In Tables 4.29, 4.30, 4.31, 4.32, 4.33, 4.34, we display for every version of our six texts their ten most central words.

Since the values of betweenness centrality computed by Gephi are very high, we write them using the exponential notation.

	Normal	**Shuffled**	**NSW**	**Reduced**
1	e, 1.041e7	e, 7.637e6	ntoni, 5.426e6	e, 2.937e6
2	che, 6.134e6	che, 5.736e6	casa, 4.503e6	di, 1.658e6
3	di, 5.843e6	la, 4.903e6	malavoglia, 1.697e6	che, 1.623e6
4	a, 5.336e6	a, 4.434e6	mena, 1.685e6	a, 1.510e6
5	la, 5.022e6	di, 4.312 e6	andava, 1.685e6	la, 1.370e6
6	il, 3.930e6	il, 4.079e6	nulla, 1.678e6	il, 1.121e6
7	per, 3.288e6	non, 3.559e6	fatto, 1.591e6	per, 8.354e5
8	le, 3.147e6	per, 2.362e6	nonno, 1.481e6	le, 8.088e5
9	non, 2.365e6	si, 1.992e6	piedipapera, 1.440e6	non, 5.923e5
10	si, 2.333e6	le, 1.799e6	sempre, 1.415e6	come, 5.214e5

Table 4.29: Ten higher Betweenness Centrality words for *I Malavoglia*.

	Normal	**Shuffled**	**NSW**	**Reduced**
1	e, 8.088e6	e, 6.031e6	pinocchio, 8.963e6	e, 1.600e6
2	di, 6.362e6	di, 4.694e6	burattino, 4.049e6	di, 1.350e6
3	che, 3.760e6	che, 3.492e6	sempre, 1.407e6	a, 7.396e5
4	a, 3.458e6	a, 3.168e6	dopo, 1.284e6	che, 7.395e5
5	un, 2.557e6	il, 3.062e6	fatto, 1.233e6	un, 5.232e5
6	il, 2.115e6	un, 2.351e6	casa, 1.025e6	il, 4.867e5
7	la, 1.959e6	la, 2.221e6	povero, 1.003e6	la, 4.511e5
8	per, 1.689e6	non, 1.641e6	ragazzi, 9.946e5	per, 3.350e5
9	una, 1.469e6	per, 1.516e6	fata, 9.879e5	in, 2.842e5
10	si, 1.463e6	in, 1.467e6	mai, 9.056e5	si, 2.758e5

Table 4.30: Ten higher Betweenness Centrality words for *Pinocchio*.

	Normal	**Shuffled**	**NSW**	**Reduced**
1	di, 1.108e7	di, 6.155e6	sandokan, 7.905e6	di, 2.658e6
2	e, 1.016e7	e, 5.442e6	yanez, 7.701e6	e, 2.416e6
3	che, 7.387e6	il, 4.665e6	rajah, 4.662e6	che, 1.607e6
4	a, 5.289e6	che, 4.327e6	kammamuri, 3.773e6	la, 1.215e6
5	la, 5.242e6	la, 3.824e6	pirati, 3.040e6	a, 1.209e6
6	un, 4.839e6	un, 3.288e6	tigre, 2.820e6	il, 1.137e6
7	il, 4.721e6	a, 2.757e6	verso, 2.717e6	un, 1.118e6
8	una, 3.548e6	si, 2.275e6	capitano, 2.601e6	si, 7.186e5
9	non, 3.337e6	non, 2.238e6	uomo, 2.299e6	una, 6.902e5
10	si, 3.179e6	le 1.894e6	uomini, 2.179e6	non, 6.692e5

Table 4.31: Ten higher Betweenness Centrality words for *I Pirati della Malesia*.

	Normal	**Shuffled**	**NSW**	**Reduced**
1	e, 1.857e7	e, 1.306e7	adriana, 6.038e6	e, 3.626e6
2	di, 1.551e7	di, 1.184e7	forse, 5.541e6	di, 3.163e6
3	che, 1.213e7	che, 1.110e7	casa, 4.978e6	che, 2.212e6
4	a, 1.020e7	la, 9.116e6	via, 4.553e6	a, 1.988e6
5	la, 9.004e6	a, 8.696e6	occhi, 4.371e6	la, 1.854e6
6	non, 6.798e6	non, 7.963e6	vita, 4.370e6	il, 1.206e6
7	per, 6.431e6	il, 6.782e6	prima, 4.255e6	un, 1.202e6
8	il, 6.246e6	per, 6.662e6	giá, 4.192e6	per, 1.181e6
9	un, 6.189e6	un, 6.408e6	qualche, 3.642e6	non, 1.160e6
10	mi, 5.282e6	mi, 5.607e6	fatto, 3.480e6	mi, 1.117e6

Table 4.32: Ten higher Betweenness Centrality words for *Il fu Mattia Pascal*.

	Normal	**Shuffled**	**NSW**	**Reduced**
1	e, 1.215e7	e, 9.102e6	efix, 1.332e7	e, 2.920e6
2	di, 9.764e6	di, 7.365e6	noemi, 4.182e6	di, 2.411e6
3	che, 4.684e6	la, 5.185e6	giacinto, 3.849e6	la, 1.098e6
4	la, 4.482e6	il, 5.122e6	occhi, 3.419e6	il, 1.064e6
5	il, 4.428e6	che, 4.270e6	casa, 2.758e6	che, 1.056e6
6	a, 4.188e6	a, 3.595e6	donna, 2.480e6	a, 9.507e5
7	le, 3.901e6	le, 3.367e6	bene, 1.800e6	le, 8.837e5
8	un, 3.714e6	non, 3.362e6	pareva, 1.771e6	un, 7.660e5
9	come, 3.426e6	un, 3.259e6	viso, 1.760e6	come, 6.931e5
10	non, 2.917e6	si, 2.613e6	sempre, 1.667e6	si, 6.786e5

Table 4.33: Ten higher Betweenness Centrality words for *Canne al Vento*.

	Normal	Shuffled	NSW	Reduced
1	di, 2.788e7	di, 2.140e7	guido, 1.553e7	di, 7.433e6
2	e, 2.299e7	che, 1.759e7	ada, 1.376e7	e, 5.602e6
3	che, 1.809e7	e, 1.492e7	essa, 1.354e7	che, 4.153e6
4	la, 1.239e7	non, 1.279e7	augusta, 9.021e6	la, 3.132e6
5	non, 1.191e7	la, 1.224e7	prima, 7.955e6	per, 2.616e6
6	a, 1.097e7	il, 9.492e6	sempre, 6.671e6	a, 2.544e6
7	per, 1.091e7	per, 8.750e6	carla, 6.007e6	non, 2.539e6
8	il, 8.761e6	a, 8.558e6	qualche, 5.277e6	il, 2.116e6
9	un, 8.704e6	un, 7.857e6	grande, 5.218e6	un, 2.038e6
10	mi, 7.197e6	mi, 7.200e6	giorno, 5.036e6	mi, 1.772e6

Table 4.34: Ten higher Betweenness Centrality words for *La Coscienza di Zeno*.

We can notice that, in Normal, NSW and Reduced texts, the words with higher values of betweenness centrality are quite the same, apart few cases. Moreover, they are all stopwords.
In fact, these are the most influential nodes in the texts, since they are junctions for meaning circulation within the novels. Moreover, they can appear in all the contexts present in the books. For these reasons they do not change too much with shuffling and neither in Reduced texts.

In NSW case, the most central words appear often even in the NSW-list of the most frequent words. In fact, the most frequent words are often name of characters or terms that appear throughout the whole text, thus it is possible that they belong to different contexts and connect them, having in such a way high betweenness centrality.
We can even observe that in this case, the words with higher betweenness represent important terms for the novels, and not only if we think to characters' names.
For example, in *I Malavoglia* at the second place of the list we can find the

word "casa", that is the core of Verga's poetic, representing the theme of family. In *Pinocchio*, the second most central word is "burattino", that is Pinocchio condition; while in the list of *Il fu Mattia Pascal* we can see the words "forse" and "vita", that represent respectively the uncertainty present in the book and its central topic.

Since NSW and Reduced texts have less edges than Normal ones, we can notice that their values of betweenness centrality are lower than Normal and shuffled values. Often the difference between NSW values and Normal ones is one order magnitude, especially at the top of the lists, while with Reduced texts the difference is not so large.
At last, we can notice that Normal and Shuffled values are similar, in fact they are composed by the same words. However shuffling changes links and their weights: sometimes Shuffled between centrality is higher than Normal one, sometimes it is lower.

4.9 A different approach to texts

In our study, we consider texts as networks. However this is not the only method to analyse sequences of words. For example a literary text can be considered as the output of a stationary and ergodic source, that takes values in a finite alphabet. This is the method used by our colleague Tommaso Pola, in his thesis *Statistical Analysis of written languages*, where he searches information about the source through a statistical analysis of texts.

In particular he focused on two measures, **burstiness** and **long-range correlations** [2].
Let us see how to construct and use these quantities.

If s is a sequence of symbols, we denote by s_k the symbol in the kth position and, if $m \geq n$, we denote by s_n^m the subsequence $(s_n, s_{n+1}, \dots, s_m)$.

Definition 4.9.1. An **observable** is a function f that maps a symbolic sequence s into a number sequence x.
In particular, we can focus in local mappings, where:

$$x_k = f(s_k^{k+r}) \qquad \forall\, k, \quad r \geq 0.$$

The observable that we will use is defined as:

$$x_k = f_\alpha(s_k) = \begin{cases} 1, & \text{if condition } \alpha \text{ is verified;} \\ 0, & \text{if condition } \alpha \text{ is not verified.} \end{cases}$$

Thus, x will be a binary sequence associated to the chosen condition, α. If we study a novel, s represents the whole text, while α could represents a letter or a word.
In that way, x shows when α appears in the text: we can call τ the sequence of the inter-event times τ_i, that are the number of zeros between two consecutive ones.

Definition 4.9.2. We define the **average** of a sequence x obtained from a text of length N, for each fixed t, as:

$$\langle x \rangle = \frac{1}{N-t} \sum_{j=1}^{N-t} x_j.$$

Definition 4.9.3. Now we can define the **long-range correlation** of the sequence x as:

$$C_x(t) = \langle x_j x_{j+t} \rangle - \langle x_j \rangle \langle x_{j+t} \rangle .$$

This long-range correlation can be easily studied as:

$$\sigma_X^2(t) = \left\langle X(t)^2 \right\rangle - \left\langle X(t) \right\rangle^2 \propto t^\gamma,$$

where

$$X(t) = \sum_{j=0}^{t} x_j \qquad \text{and} 1 < \gamma < 2.$$

Definition 4.9.4. We define the **burstiness** of a condition α as:

$$b_\alpha = \frac{\sigma_\tau}{\langle \tau \rangle}.$$

4.9.1 Comparison of results

We can compare the words found with our Betweenness Centrality in NSW texts and the words found by Tommaso Pola with his statistical approach.

It is important to underline that the preprocessing steps to prepare texts for studying them are the same. However, during the analysis we consider the whole texts every time we want to extract a measure, while Tommaso chooses the words to study: he takes the seven most frequent terms, the seven most frequent nouns and the seven words with frequency similar to frequency of the seven nouns.

In spite of these differences in the approach to novels, we can notice some similarities in the results. In Table 4.35 we can see the keywords obtained with the statistical method.

Title	**Keywords**
I Malavoglia	don, zio, ntoni, padron, compare
Pinocchio	geppetto, babbo, fata, ragazzi, casa
I Pirati della Malesia	sandokan, yanez, rajah, tigre, kammamuri, malesia, pirati
Il fu Mattia Pascal	signor, papiano, adriana, vita
Canne al vento	noemi, predu, giacinto, donna, don, é, efix
La Coscienza di Zeno	carla, guido, ada, augusta

Table 4.35: Keywords found with the statistical approach.

We can see analogies with our lists of most central words for *I Pirati della Malesia*, *Canne al vento* and *La Coscienza di Zeno*.

The maximum analogy is obtained for the longest novel, *La Coscienza di Zeno*: every keyword appears in Table 4.34. This is in according with the fact that the statistical analysis studies asymptotic behaviours, however this aspect has to be deepened.

Conclusions

In this thesis we presented the most important topological measures from Network Theory and the most important statistical patterns shared by linguistic networks.

We applied them to a corpus composed by six Italian novels, each of them studied in four different versions: Normal, Shuffled, No Stopwords and Reduced.

The comparison of such versions led to analyse every measurement (described often by average values and probability distributions) in order to understand which are able to distinguish the real novel by the other texts.

We noticed that the distributions of the most common measures, as degree and strength are not able in doing this, but neither selectivity is useful. Moreover, we demonstrated the invariance of degree distribution with shuffling, using Conditioned Zipf's law.

Average and maximum values sometimes change in different versions, giving in such a way information about specific characteristics of texts. For example, we noticed that deleting stopwords provides words, representing some maximum values (frequency, degree, betweenness centrality), related to the themes of the novels.

The results we obtained could be useful in searching methods for distinguish masterpieces from random sequences of words and also in individualising measures that can extract information by texts.

In addition to this, the same methods described in this thesis has been used by Filippo Bonora on an English corpus composed of five books. He found similar results, in particular the validity of Conditioned Zipf's law, and the general trends of measurements.

This can lead to a deeper study of the differences and similarity between languages.

Appendix A

Gephi

Gephi is an open-source software for visualizing and analysing large graphs, available for Linux, Mac and Windows.
Using this software, it is possible to view and manipulate networks in according with our needs. It is also possible to compute some of the measurements we study in this thesis [35].

A.1 Dynamic Networks

Gephi is able to read some different file formats, but we decide to save our literary networks in *.gexf* format since it allows to create dynamic graphs with edges weight and values of vertices changing in time. When Gephi loads this kind of format, it creates a timeline useful to visualize how texts grow while reading them.

Example A.1.1. We can give an example of a simple *.gexf* file.
It describes the sentence:

O Romeo, Romeo, wherefore art thou Romeo?

The file is composed by four parts. The first and the last ones respectively open and close the network. The second part is the list of nodes, while the third contains the edges, their weights and evolution.

Thus, the file is written as:

```
<gexf version="0.8.2 beta">
 <meta lastmodifieddate="2013-09-20">
  <graph mode="dynamic" defaultedgetype="directed" timeformat="double">

 <node id="0" label="o" start="1.0"/>
 <node id="1" label="romeo" start="2.0"/>
 <node id="2" label="," start="3.0"/>
 <node id="3" label="wherefore" start="6.0"/>
 <node id="4" label="art" start="7.0"/>
 <node id="5" label="thou" start="8.0"/>
 <node id="6" label="?" start="10.0"/>

 <edge id="0" source="0" target="1" start="2.0" end="11.0">
   <attvalue for="weight" value="1.0" start="2.0" end="11.0">
 <edge id="1" source="1" target="2" start="3.0" end="11.0">
   <attvalue for="weight" value="1.0" start="3.0" end="5.0">
   <attvalue for="weight" value="2.0" start="5.0" end="11.0">
 <edge id="2" source="1" target="6" start="10.0" end="11.0">
   <attvalue for="weight" value="1.0" start="10.0" end="11.0">
 <edge id="3" source="2" target="1" start="4.0" end="11.0">
   <attvalue for="weight" value="1.0" start="4.0" end="11.0">
 <edge id="4" source="2" target="3" start="6.0" end="11.0">
   <attvalue for="weight" value="1.0" start="6.0" end="11.0">
 <edge id="5" source="3" target="4" start="7.0" end="11.0">
   <attvalue for="weight" value="1.0" start="7.0" end="11.0">
 <edge id="6" source="4" target="5" start="8.0" end="11.0">
   <attvalue for="weight" value="1.0" start="8.0" end="11.0">
 <edge id="7" source="5" target="1" start="9.0" end="11.0">
```

```
   <attvalue for="weight" value="1.0" start="9.0" end="11.0">

 </graph>
</gexf>
```

A.2 Layout

With Gephi we can modify graphs in different ways, thanks to some internal algorithms. Applying these ones we obtain different layouts, each of them with specific properties.
For example, *Force Atlas* is a layout often used to visualize graphs with a high number of nodes connected together and Small World networks. However it is a very slow algorithm, *O(n^2)*. *Yifan Hu Multilevel* is a very fast algorithm, *O($n * \log(n)$)*, and organizes the graph in clusters.

In Figure A.1 and A.2 we can see the difference between the two layouts described above, applied to the song "Romeo and Juliet", by Dire Straits.

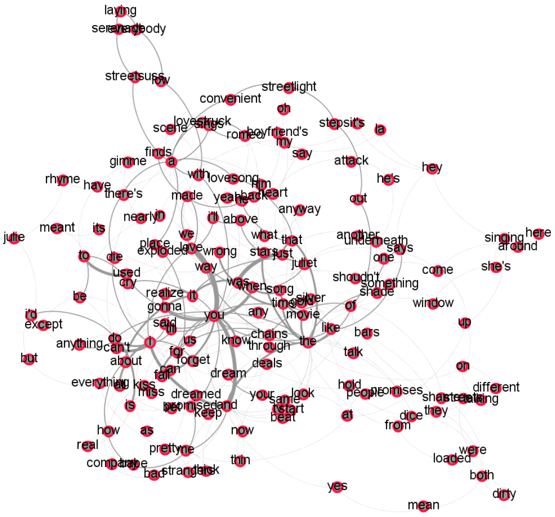


Figure A.1: *Romeo and Juliet*, with Force Atlas

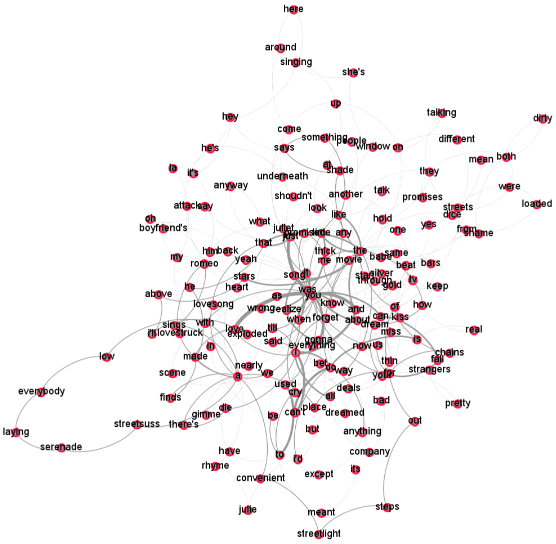


Figure A.2: *Romeo and Juliet*, with Yifan Hu Multilevel

A.3 Statistics

Gephi is useful not only in graph visualization, but also in computing networks measures and statistical quantities, such as:

- *Average path length*;
- *Network diameter*;
- *Average Cluster Coefficient*;
- *Network Density*;
- *Node Betweenness Centrality*.

This is done using again Gephi internal algorithms that produce values and graphics.

Appendix B

Italian Stopwords List

The list of stopwords used in studying Italian texts:

1, 2, 3, 4, 5, 6, 7, 8, 9, 0, a, abbiamo, abbia, abbiate, abbiano, ad, adesso, agli, ai, al, alla, alle, allo, agli, agl, all, allora, altra, altre, altri, altro, anche, ancora, avere, avevo, avevi, avete, aveva, avevamo, avevate, avevano, avesti, avemmo, aveste, avessi, avesse, avessimo, avessero, avendo, avuto, avuti, avute, avuta, avrò, avrai, avrà, avremo, avrete, avranno, avrei, avresti, avrebbe, avremmo, avreste, avrebbero, ben, buono, c, ch, che, chi, ci, cinque, come, comprare, con, contro, col, colla, coi, consecutivi, consecutivo, cosa, così, cui, d, da, dal, dallo, dalla, dalle, dagli, dall, dagl, dai, dei, del, dell, degl, della, delle, dello, degli, dentro, deve, devo, di, disse, diceva, dice, doppio, dov, dove, due, e, è, ebbe, ebbero, ebbi, egli, ed, ecco, ero, eri, era, eravamo, eravate, erano, essendo, facevo, facevi, faceva, facevamo, facevate, facevano, feci, facesti, fece, facemmo, faceste, fecero, facessi, facesse, facessimo, facessero, facendo, faccio, fai, fanno, facciamo, faccia, facciate, facciano, farò, farai, farà, faremo, farete, faranno, farei, faresti, farebbe, faremmo, fareste, farebbero, fare, fine, fino, fra, fui, fosti, fu, fummo, foste, furono, fossi, fosse, fossimo, fossero, gente, gli, giù, ha, hai, hanno, ho, i, il, in, indietro, invece, io, l, lí, là, la, lavoro, le, lei, li, lo, loro, lui, lungo, m, ma, me, meglio, mi, mio, mia, miei, mie, molta, molti, molto, n, ne, nei, nel, nello, nelle, nella, nell, negl, negli, no, noi, nome, nostro, nostra, nostre, nostri, nove, nuovi, nuovo, non, o, oltre, ora, otto, peggio, per, perché, però, persone, più, poco, poi, primo, promesso, qua, quale, qual, quando, quanto, quanti, quanta, quante, quarto, quasi, quattro, quel, quello, quella, quelle, quegli, quelli, quei, questo, questa, queste, questi, qui, quindi, quinto, rispetto, s, saró, sarai, sará, saremo, sarete, saranno, sarei, saresti, sarebbe, saremmo, sareste, sarebbero, se, secondo, sei, sembra, sembrava, senza, sette, si, sì, sia, siamo, siete, siate, siano, solo, sono, sopra, soprattutto, sotto, sto, stai, sta, stiamo, state, stanno, stati, stato, starò, starai, starà, staremo, starete, staranno, starei, staresti, starebbe, staremmo, stareste, starebbero, stavo, stavi, stava, stavamo, stavate, stavano, stando, stesso, stetti, stesti, stette, stemmo, steste, stettero, stessi, stesse, stessimo, stessero, stia, stiate, stiano, su, suo, sua, suoi, sue, subito, sui, sul, sull, sulla, sullo, sugl, sugli, sulle, tanto, t, te, tempo, terzo, ti, tra, tre, triplo, tu, tuo, tua, tuoi, tue, tutto, tutti, tutte, tutta, ultimo, un, una, uno, v, va, vai, vi, voi, volte, vostra, vostri, vostre, vostro.

Appendix C

Python function to create the weight matrix and the GEXF file

The core of the Python algorithm created to study literary networks is the function that build their weight matrices. It also creates the GEXF file that imported in Gephi allows us to visualize the networks and to compute some measurements.
In this Appendix we display the function we created.

```
def matrice_gephi(name,text,diz):

output=open(name+'.gexf','w')
output.write('<?xml version="1.0" encoding="UFT-8"?>\n')
output.write('<gexf xmlns="http://www.gexf.net/1.2draft" version="1.2">\n')
output.write('<meta lastmodifieddate="2009-03-20">\n')
output.write('<creator>Giulia</creator>\n')
output.write('<description>A hello world! file</description>\n')
output.write('</meta>\n')
output.write('<graph mode="dynamic" defaultedgetype="directed"
          timeformat="double">\n')
output.write('<attributes class="node" mode="dynamic">\n')
output.write('</attributes>\n')
output.write('<attributes class="edge" mode="dynamic">\n')
output.write('<attribute id="weight" title="weight" type="float">\n')
output.write('</attribute>\n')
output.write('</attributes>\n')
output.write('<nodes>\n')
ii=0
p=len(diz)
while ii<p:
output.write('<node id="')
output.write(str(ii))
output.write('" label="')
output.write(diz[ii])
output.write('" start="')
i=0
while i<len(text):
if diz[ii]==text[i]:
output.write(str(float(i+1)))
i=len(text)+1
i=i+1
output.write('" />\n')
ii=ii+1

output.write('</nodes>\n')
output.write('<edges>\n')

A=scipy.sparse.dok_matrix((p,p))
i=0
j=0
k=0
while i<len(diz):
j=diz[i]
```

```
D=scipy.sparse.dok_matrix((p,p))
a=0
while a<len(text)-1:
if j==text[a]:
b=0
jj=text[a+1]
while b<len(diz):
if jj==diz[b]:
A[i,b]=A.get((i,b),0)+1
c=0
while D.get((b,c),0)!=0:
c=c+1
D[b,c]=D.get((b,c),0)+a+1
b=len(diz)
else:
b=b+1
a=a+1
else:
a=a+1
sa=0
while sa<len(diz):
if A.get((i,sa),0)!=0:
output.write('<edge id="')
output.write(str(k))
k=k+1
output.write('" source="')
output.write(str(i))
output.write('" target="')
output.write(str(sa))
output.write('" start="')
output.write(str(float(D[sa,0]+1)))
output.write('" end="')
output.write(str(float(len(text)+1)))
output.write('" >\n')
output.write('<attvalues>\n')
sd=0
while D.get((sa,sd),0)!=0:
output.write('<attvalue for="weight" value="')
output.write(str(float(sd+1)))
output.write('" start="')
output.write(str(float(D[sa,sd]+1)))
output.write('" end="')
if D.get((sa,sd+1))!=0:
output.write(str(float(D[sa,sd+1]+1)))
else:
output.write(str(float(len(l)+1)))
output.write('" >\n')
output.write('</attvalue>\n')
sd=sd+1
output.write('</attvalues>\n')
output.write('</edge>\n')
sa=sa+1
else:
sa=sa+1

i=i+1

output.write('</edges>\n')
output.write('</graph>\n')
output.write('</gexf>')
output.close()
return A
```

Acknowledgments

I want to thank Mirko Degli Esposti and Giampaolo Cristadoro, for their patience and their precious teachings and advices, and because they gave me the chance to join two of my passions: mathematics and books.

Thanks to Filippo, who shared with me these last months of study and work. Thanks for your patience, your great competence and the support both in funny and difficult moments.

An infinite thanks to my family: all of you taught me the importance of studying, that honesty, diligence and passion are necessary whatever I do, and that they always are rewarded. In particular thanks to my mummy, Emma, who supports me in all my choices and always gives me the possibility to follow my path.
Thanks even to Mario, Rita, Silvia, Luca, Valeria, Matteo and Riccardo, because family is much more than relationship.
Daddy, zio Paolo, zio Davide and Giovanni, I hope you are proud of me.

Thanks to the friends I met in these five years of University, they made studying simpler. In particular I want to thank Andrea, Diego, Elena, Elisa, Erika, Laura and Margherita: we shared chat, laughs, travels and strange moments.

"There are a mathematician, a physicist and two engineers"...thanks to

Federica, Martina and Monica. Thank you all for our friendship, for the support and the advices you give to me, and for the fantastic moments of madness we sometimes live together.
Thanks to Chiara, for your optimism and for giving me often different and stimulant points of view.

Thanks to Monica, an irreplaceable friend. Thank you for listening to me at any time, always without judging my choices. Thank you because you always believe in me and in my abilities, more than I do.

Thanks to Lorenzo, the best reason to wake up at 6.15 every morning during the last five years: you understand me and my wishes better than I do. Thank you for giving me always the right advices, for your patience and your support in difficulties. Thanks for living with me lots of funny experiences, for studying together and for loving me exactly as I am.

Bibliography

[1] R.Albert, H.Jeong, A.-L.Barabási *Error and attack tolerance of complex networks*, 2000, Nature **406**, Nature Publishing Group, pp. 378-382.

[2] E.G.Altmann, G.Cristadoro, M.Degli Esposti, *On the origin of long-range correlations in texts*, 2012, PNAS **109**, pp. 11582-11587.

[3] D.R.Amancio, E.G.Altmann, O.N.Oliveira Jr., L.d.F Costa, *Comparing intermittency and network measurements of words and their dependency on authorship*, 2011, NJPH **13**, pp. 123024-123040.

[4] D.R.Amancio, S.M.Alvisio, O.N.Oliveira Jr, L.da.F.Costa, *Complex networks analysis of language complexity*, 2012, EPL **100**, p. 58002.

[5] D.R.Amancio, O.N.Oliveira Jr, L.da.F.Costa, *Identification of literary movements using complex networks to represent texts*, 2012, NJPH **14**, p. 43029.

[6] D.R.Amancio, O.N.Oliveira Jr., L.da.F.Costa, *Using complex networks to quantify consistency in the use of words*, 2012, J. Stat. Mech. p. 1004.

[7] D.R.Amancio, O.N.Oliveira Jr, L.da.F.Costa, E.G.Altmann, D.Rybski *Probing the statistical properties of unknown texts: application to the Voynich manuscript*, 2013, PLoS ONE **8**, p.e67310.

[8] L.Antiqueira, O.N.Oliveira Jr., L.F.Costa, M.G.V.Nunes, *A complex network approach to text summarization*, 2009, Inf. Sci. **179**, pp. 584-599.

[9] A.-L.Barabási, E. Bonabeau, *Scale-Free Networks*, 2003, Sci.Am, pp. 50-59.

[10] A.Barberán, S.T.Bates, E.0.Casamayor, N.Fierer *Using network analysis to explore co-occurrence patterns in soil microbial communities*, 2011, ISME J. **6**, pp. 343-351.

[11] R.G. Clegg, *Power laws in networks*, 2006, University of York.

[12] L.da.F.Costa, F.A.Rodrigues, G.Travieso, P.R. Villas Boas, *Characterization of Complex Networks: A Survey of measurements*, 2007, Adv. Phys. **56**, pp. 167-242.

[13] R.Ferrer i Cancho, R.V.Solé, *The small world of human language*, 2001, Proc. R. Soc. B **268**, pp. 2261-2265.

[14] H.S.Heaps, *Information Retrieval: Computational and Theoretical Aspects*, 1978, Academic Press, Orlando.

[15] L. Lü, Z.Zhang, T.Zhou, *Zipf's Law leads to Heaps' Law: analyzing their relation in finite-size systems*, 2010, PLoS ONE **5**, p.e14139.

[16] B.Mandelbrot, *Jeux de communication*, 1953, Publ. Ins. Stat. Univ. Paris, **2**, pp. 1-124.

[17] B.Mandelbrot, *Simple games of strategy occurring in communication through natural languages*, 1954, IEEE Trans. Inf. Theory **3**, pp. 124-137.

[18] B.Mandelbrot, *The Fractal Structure of Nature*, 1983, Freeman, New York.

[19] A.R.Mashaghi, A.Ramenzanpour, V.Karimipour, *Investigation of a Protein Complex Network*, 2004, EPJB **41**, pp. 113-121.

[20] A.P.Masucci, G.J.Rodgers, *Differences between normal and shuffled texts: structural properties of weighted networks*, 2009, Advs. Complex Syst. **12**, pp. 113-129.

[21] A.P.Masucci, G.J.Rodgers, *Multi-directed Eulerian growing networks*, 2007, Phys A **386**, pp. 557-563.

[22] E.Otte, R.Rosseau, *Social network analysis: a powerful strategy, also for the information sciences*, 2002, JIS **28**, pp. 441-454.

[23] D.Paranyushkin, *Identifying the pathways for meaning circulation using text network analysis*, 2011, http://noduslabs.com/research/pathways-meaning-circulation-text-network-analysis/.

[24] R.M.Roxas, G.Tapang, *Prose and Poetry Classification and Boundary Detection Using Word Adjacency Network Analysis*, 2204, IJMPS C **21**, pp. 503-512.

[25] C.E.Shannon, *A Mathematical Theory of Communication*, 1948, Bell Syst. Tech.J. **27**, pp. 379-423.

[26] H.Simon, *On a Class of Skew Distribution Functions*, 1955, Biometrika **42**, pp.425-440.

[27] R.V.Solé, B.Corominas Murtra, S.Valverde, L.Steels, *Language Networks: their structure, function and evolution*, 2010, Complexity **15**, pp.20-26.

[28] J.T.Stevanak , D.M.Larue, D.C.Lincoln *Distinguishing Fact from Fiction: Pattern Recognition in Texts Using Complex Networks*, 2010, arXiv: 1007.3254.

[29] B.Wang, H.Tang, C.Guo, Z.Xiu *Entropy optimization of scale free networks robustness to random failures*, 2006, Phys A **363**, pp. 591-596.

[30] D.J.Watts, S.H.Strogatz, *Collective dynamics of "small-world" networks*, 1998, Nature **393**, Nature Publishing Group, pp. 440-442.

[31] D.H.Zanette, *Statistical Patterns in written language*, 2012, http://fisica.cab.cnea.gov.ar/estadistica/zanette/papers/lang-patterns.pdf.

[32] D.H.Zanette, M.A.Montemurro, *Dynamics of text generation with realistic Zipf distribution*, 2005, J. Quant. Ling. **12**, pp. 29-40.

[33] G. K.Zipf, *Human Behaviour and the Principle of Least Effort. An Introduction to Human Ecology*, 1949, Addison-Wesley, Cambridge.

[34] G. K.Zipf, *The Psycho-Biology of Language. An Introduction to Dynamic Philology*, 1936, Routledge, London.

[35] https://gephi.org

[36] http://www.the-vital-edge.com/what-is-network-density/