

Customized software for the graph analysis of text (*SpeechGraphs*)

The *SpeechGraphs* software was developed at the Brain Institute of the Federal University of Rio Grande do Norte (Natal, Brazil), by R. Furtado, P.P.C. Maia, N.B. Mota, S. Ribeiro, M. Copelli, and D.F. Slezak. Software logo by G.M da Silva.

1. Introduction

This document is a guide for the use of the free *SpeechGraphs* software, a graph-theoretical analysis tool that uses text as input and graph features as output. *SpeechGraphs* can run on many different platforms, such as Linux, Windows and OSX. This guide will be updated each time we release a new version of *SpeechGraphs*. You should receive a copy of this guide along with the software; alternatively, this guide can be directly downloaded from the project website:

<http://neuro.ufrn.br/research/software/speechgraphs>

2. Dependencies

SpeechGraphs uses the following third-party libraries:

- **JUNG** - Java Universal Network/Graph Framework (<http://jung.sourceforge.net/>).
- **Tarjan Algorithm** for the calculation of the strongly connected components of a graph (http://en.wikipedia.org/wiki/Tarjan_%27s_strongly_connected_components_algorithm).

3. Getting Started

Here we describe how to set up the *SpeechGraphs* environment and get it running.

3.1 Requirements

- Java Virtual Machine (JVM) 1.2 or later.

3.2 Get it running

After downloading *SpeechGraphs*, double-click the icon of the application.

4. Approaches

4.1. Visualization

4.1.1 Visualization File Format

The Visualization file format is a simple text file, as shown below:

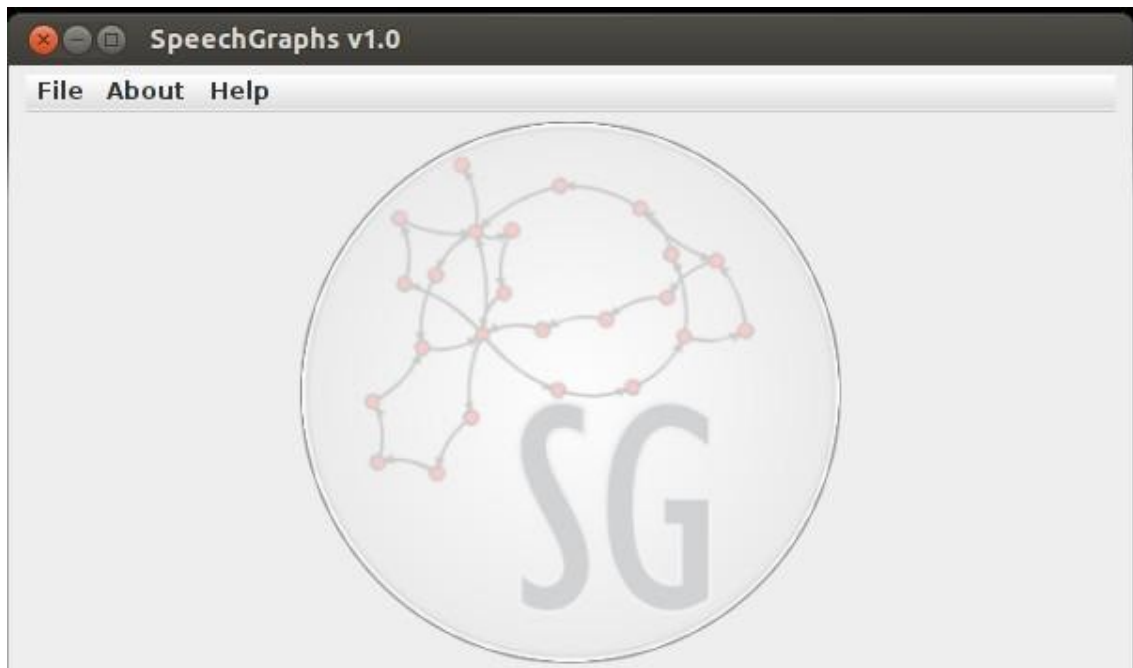
... *The faculty by which... we are subject to illusion when affected by disease, is identical with that which produces illusory effects in sleep*” from “On Dreams” by Aristotle.

The program represents the loaded text as a graph, i.e. a mathematical representation of a network $G = (N, E)$, with $N = \{w_1, w_2, w_3, \dots\}$ as the set of nodes and $E = \{(w_i, w_j)\}$ as the set of edges between words w_i and w_j in N . Words are therefore represented as nodes and the temporal links between words are represented as edges. *SpeechGraphs* generates both a directed graph (digraph) and an undirected graph without parallel or repeated edges, or self-loops (an edge linking a node with itself).

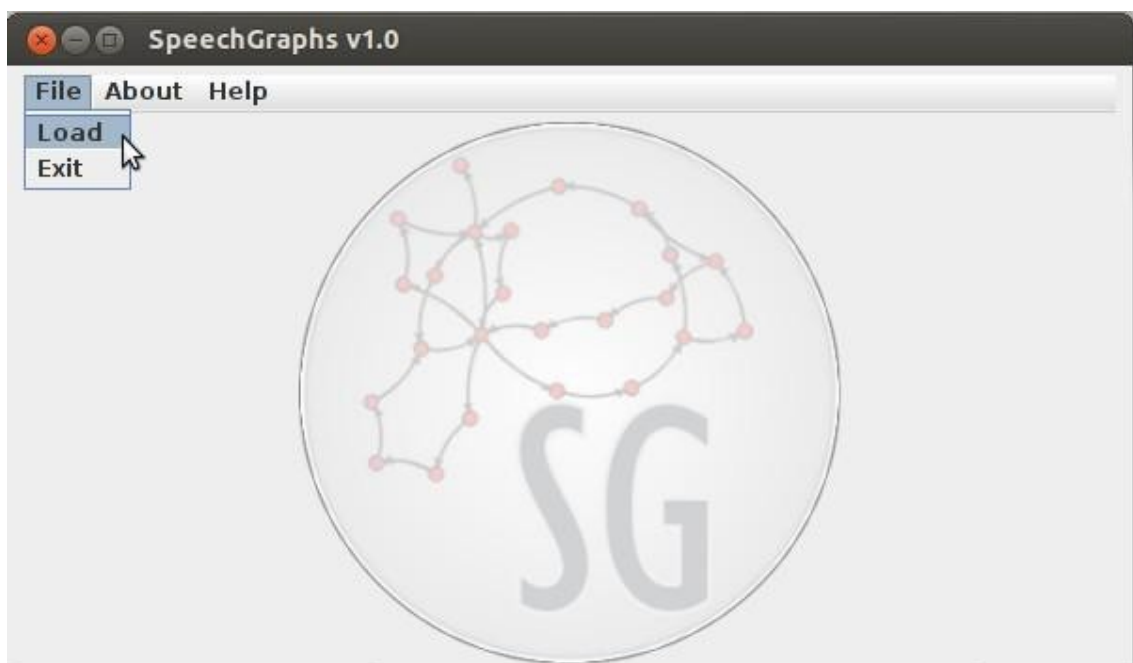
4.1.2 Visualization of graph parameters:

To visualize a set of graph parameters as Graph Plot, Graph Histogram Plot and Degree Distribution Plot for both directed and undirected graph, follow the steps:

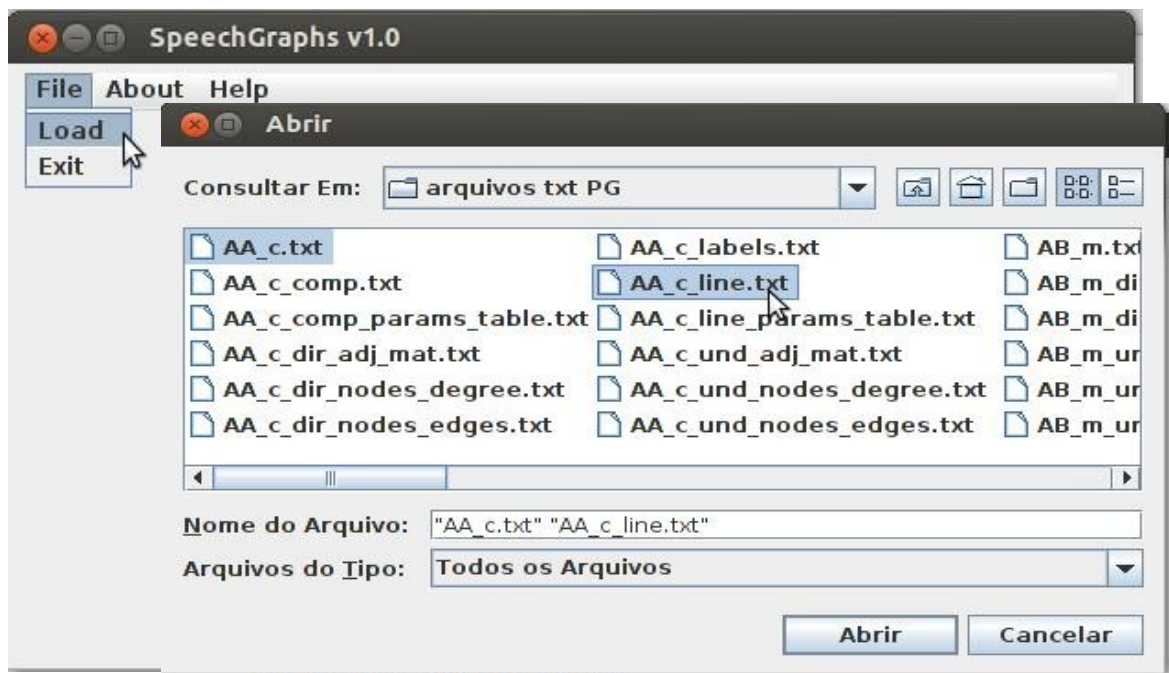
1. Open *SpeechGraphs* by clicking the icon on your desktop.
2. The main screen of the application will appear, as shown below:



3. To load a file into *SpeechGraphs*, click on the **File Menu**, Option **Load**, as shown in the picture below:



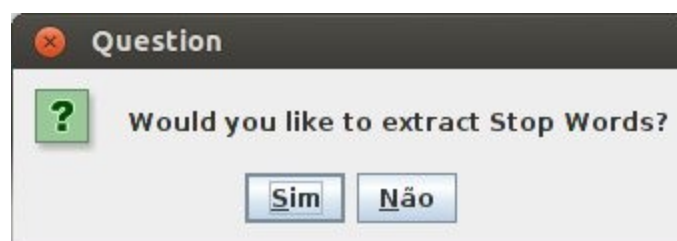
4. A file chooser window will open. Select the file(s) that you want to analyse:



5. Click on the Visualization button:



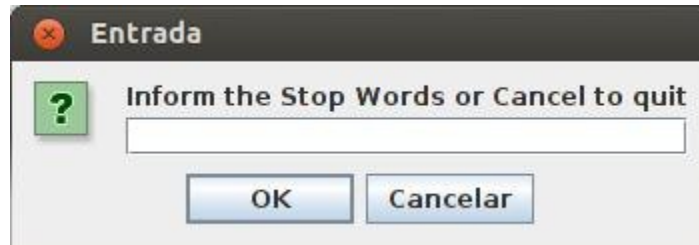
6. You must inform if you want to extract Stop Words, as shown below:



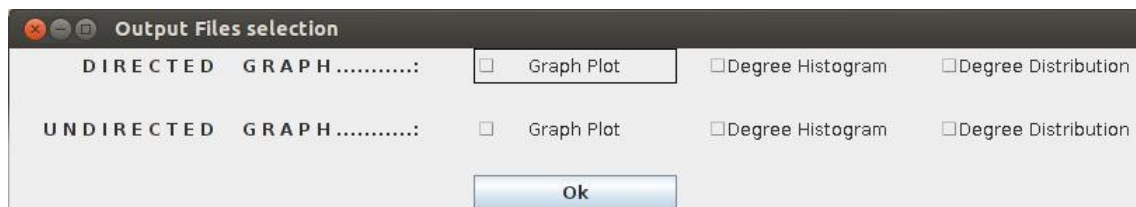
- **What are Stop Words?**

- Stop Words are any words in the text file that you want to exclude from your analysis.

If you chose to extract Stop Words, the screen below will appear. Type the list of words that you want to exclude from your analysis.



7. Select the set of features that you want to generate for directed graph, undirected graph or both, as shown in the picture below:



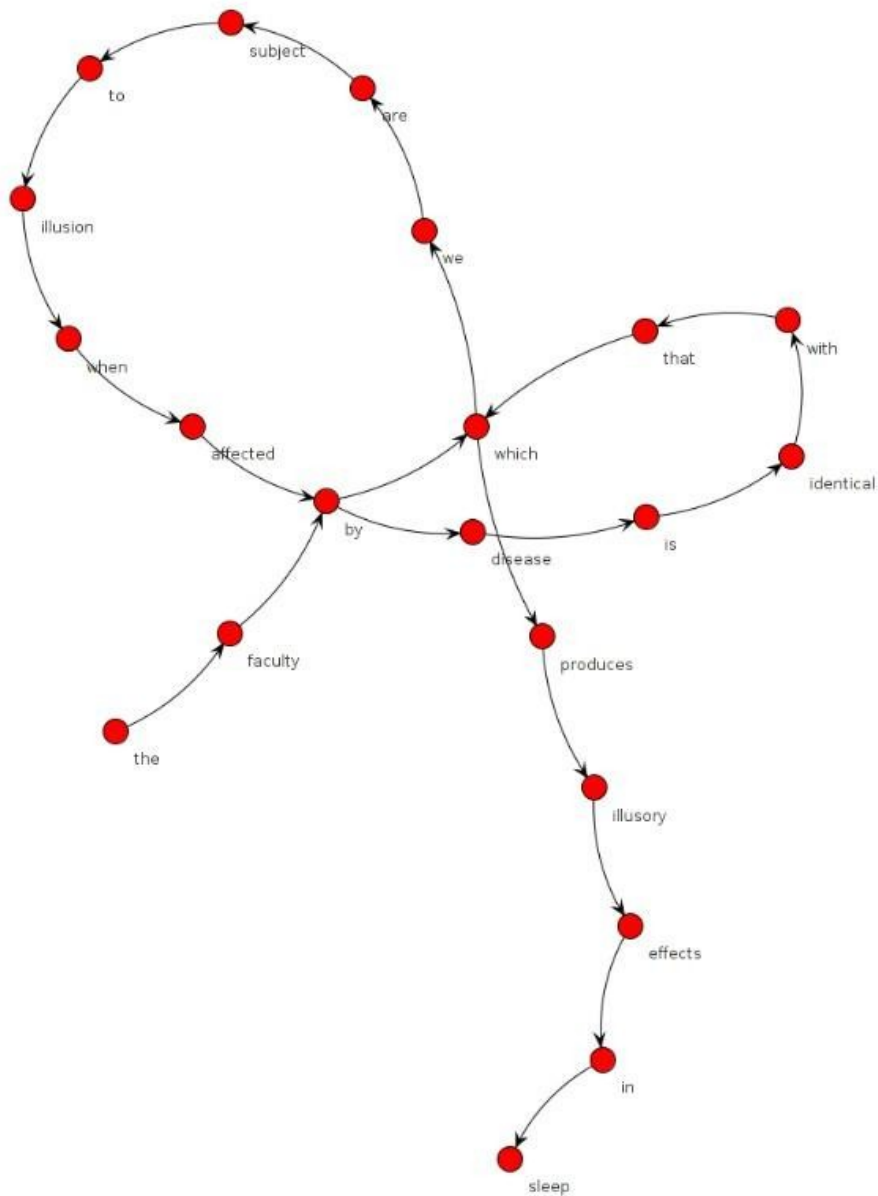
4.1.3 Visualization options:

- **Graph Plot** – Shows the plot of the graph(s) selected with its/their parameters. There is an option to export the graph plot to PNG or EPS format (saved on program directory), then click on 'File' (top right corner) and then click 'Export'. There is another option called 'Visualization' that allows the selection of parameters to be visualized on the left window.

- **Node Degree Histogram** – Shows a histogram with the degree of each node in sequence.

- **Degree Distribution** – Shows the number of nodes for each degree value.

4.1.3.1 Graph Plot window: If you chose to view the graph plot, you will see the following window:
 On the right side of the window, you will have the following graph parameters:



- **Graph nodes list:** describes all the nodes (words) in the text.

```

..... DIRECTED GRAPH .....
*Nodes 21
id*int label*string
1 "the"
2 "faculty"
3 "by"
4 "whichwe"
5 "are"
6 "subject"
7 "to"
8 "illusion"
9 "when"
10 "affected"
11 "disease"
12 "is"
13 "identical"
14 "with"
15 "that"
16 "which"
17 "produces"
18 "illuory"
19 "effects"
20 "in"
21 "sleep"
  
```

- **Graph edges list:** lists all the edges found between words in the text.

```
*DirectedEdges 19
source*int target*int
1 2
2 3
3 4
4 5
5 6
6 7
7 8
8 9
10 3
3 11
11 12
12 13
13 14
14 15
15 16
16 17
17 18
18 19
19 20
```

- **Graph adjacency matrix:** for a graph G with n nodes, corresponds to a n x n matrix, considering a_{ij} the number of edges from node i to node j.

```
----- Adjacency Matrix -----
0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
```

- **Graph hub node:** plots the most connected node (hub); then, lists the nodes according to a decreasing degree sequence; for directed graphs, also plots the number of “in and out” edges of each node.

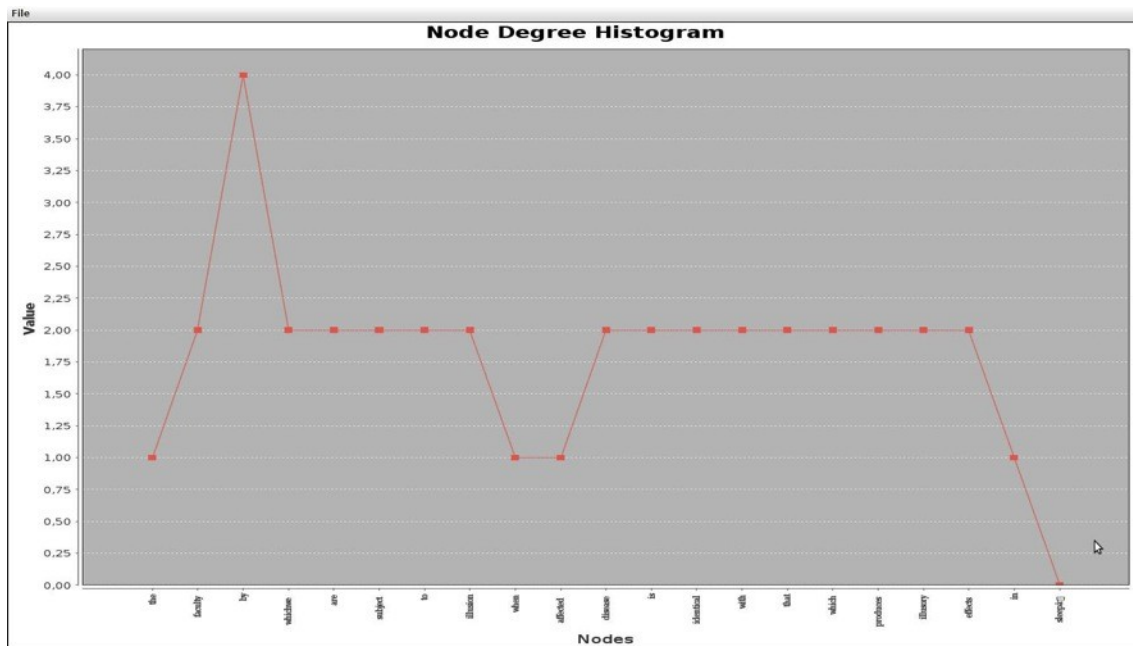
```
----- Nodes and Edges -----  
  
Hub node: by  
Number of in edges of hub node: 2  
Number of out edges of hub node: 2  
  
Node: by  
Number of in edges: 2  
Number of out edges: 2  
  
Node: faculty  
Number of in edges: 1  
Number of out edges: 1  
  
Node: whichwe  
Number of in edges: 1  
Number of out edges: 1  
  
Node: are  
Number of in edges: 1  
Number of out edges: 1  
  
Node: subject  
Number of in edges: 1  
Number of out edges: 1  
  
Node: to  
Number of in edges: 1  
Number of out edges: 1  
  
Node: illusion  
Number of in edges: 1  
Number of out edges: 1
```

- **Graph parameters:** description on section: 4.2. Output Files Approach, on Parameters table file.

```
----- Graph Parameters -----  
  
N:21  
E:19  
ATD:1.8095238  
LCC:20  
LSC:1  
PE:0  
RE:0  
L1: 0  
L2: 0  
L3: 0
```

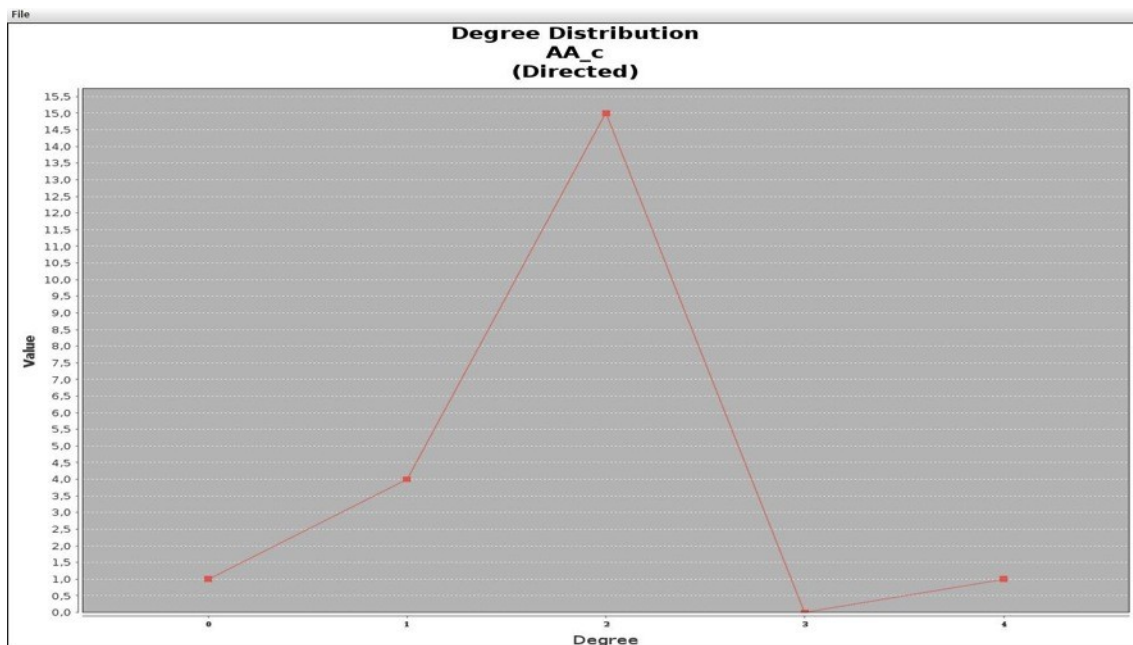
4.1.3.2 Degree Histogram:

If you chose to view the Degree Histogram, you will see the following window:



4.1.3.4 Degree Distribution:

If you chose to view the Degree Distribution, you will see the following window (X axis you have the degree value and on Y axis you have the number of nodes):



4.2. Output Files Approach

The Output File approach will generate the following files:

- **Nodes and Edges File** – Contains all nodes and all edges of the graph.

Name : <text file name>_<dir/und>_nodes_and_edges.txt

Format:

**Nodes 21*

*id*int label*string*

1 "the"

2 "faculty"

3 "by"

4 "whichwe"

5 "are"

6 "subject"

7 "to"

8 "illusion"

9 "when"

10 "affected"

11 "disease"

...

**DirectedEdges 19*

*source*int target*int*

1 2

2 3

3 4

4 5

5 6

6 7

7 8

...

- **Adjacency Matrix File** – Contains the adjacency matrix of the graph.

Name : <text file name>_<dir/und>_adj_mat.txt

Format:

```

----- Adjacency Matrix -----
0 1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1
0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0

```

- **Nodes and Degree File** – Contains a list of nodes and their degrees.

Name : <text file name>_<dir/und>_nodes_degree.txt

Format:

Node Degree

7 22

6 16

42 10

24 8

3 6

4 6

25 6

30 6

35 6

...

- **Parameters Table File** – Contains the table of graph parameters.

Name : parameters_table.txt

Format:

File, WC, Nodes, Edges, RE, PE, L1, L2, L3, LCC, LSC, ATD, Density, Diameter, ASP, CC

AA_c, 117, 67, 116, 13, 14, 0, 1, 3, 67, 65, 3.4626, 0.0454, 11, 3.9090, 0.0724

Parameters Description:

File – Name of the text file.

Word Count (WC) – Number of words in the text.

N: Number of nodes.

E: Number of edges.

RE (Repeated Edges): sum of all edges linking the same pair of nodes.

PE (Parallel Edges): sum of all parallel edges linking the same pair of nodes given that the source node of an edge can be the target node of the parallel edge.

L1 (Loop of one node): sum of all edges linking a node with itself, calculated as the trace of the adjacency matrix.

L2 (Loop of two nodes): sum of all loops containing two nodes, calculated by the trace of the squared adjacency matrix divided by two.

L3 (Loop of three nodes): sum of all loops containing three nodes (triangles), calculated by the trace of the cubed adjacency matrix divided by three.

LCC (Largest Connected Component): Number of nodes in the maximal subgraph in which all pairs of nodes are reachable from one another in the underlying undirected subgraph.

LSC (Largest Strongly Connected Component): Number of nodes in the maximal subgraph in which all pairs of nodes are reachable from one another in the directed subgraph (node a reaches node b, and b reaches a).

ATD (Average Total Degree): Given a node n, the Total Degree is the sum of “in and out” edges. Average Total Degree is the sum of Total Degree of all nodes divided by the number of nodes.

Density: Number of edges divided by possible edges. ($D = 2 * E / N * (N - 1)$), where E is the number of edges and N is the number of nodes.

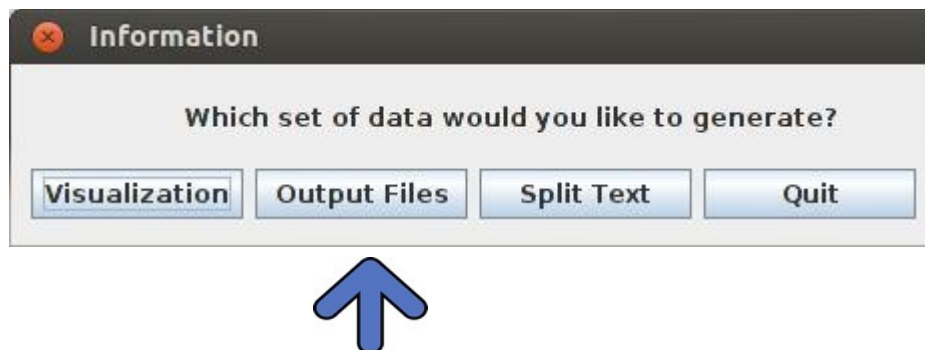
Diameter: Length of the longest shortest path between the node pairs of a network.

Average Shortest Path (ASP): Average length of the shortest path between pairs of nodes of a network.

CC (Average Clustering Coefficient): Given a node n, the Clustering Coefficient Map (CCMap) is the set of fractions of all n neighbours that are also neighbours of each other. Average CC is the sum of the Clustering Coefficients of all nodes in the CCMap divided by number of elements in the CCMap.

4.2.1 Generating Output Files

On the approach menu window, click on the button **Output Files**, as shown below:

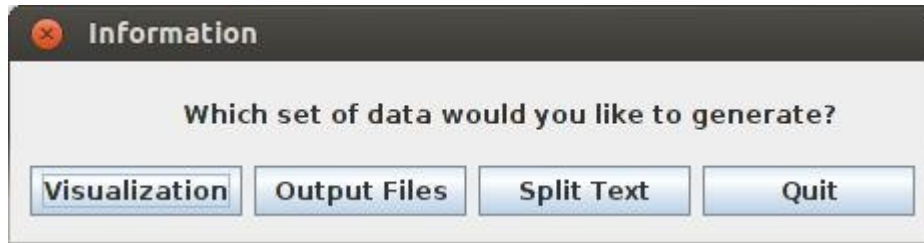


The output files will be generated in the same directory of the text file(s).

4.3 Split Text Approach

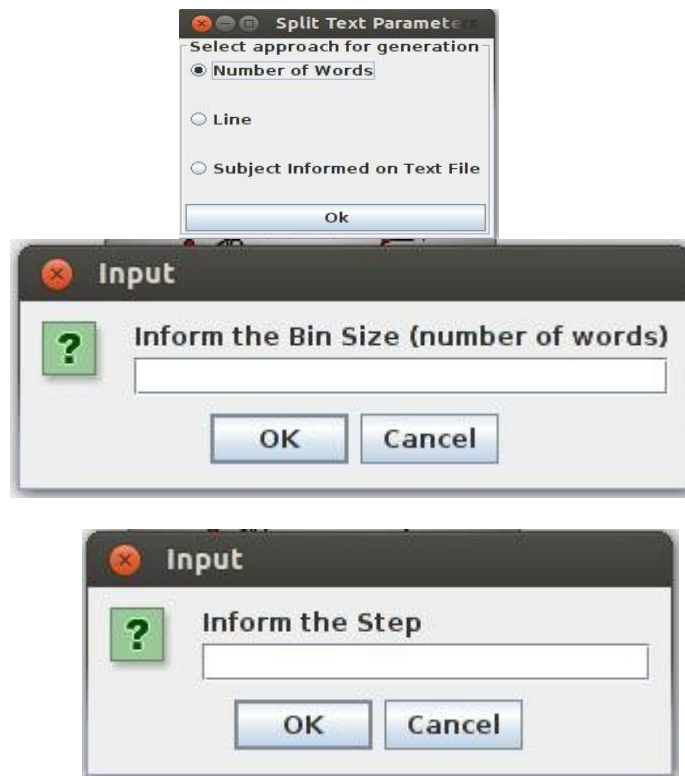
This approach can be used to split the text by a fixed number of words, by text line or by a given subject informed on the text file. As a result, different graphs are created from the same text file.

On the main menu, select the **Split Text** Button:



The Split Text approach is divided into three methods:

1. **Number of Words:** This option allows for a moving window analysis. You must inform the bin size (number of words per graph) and the step size (numbers of words to be skipped to create the next graph), as shown in the picture below:



How do bin size and step work?

This approach implements a sliding window through the whole text. The window will advance depending on the bin size and step size. This is illustrated below, using a bin size of 4 words and step of 3 words.

EXAMPLE: *“The faculty by which... we are subject to illusion when affected by disease, is identical with that which produces illusory effects in sleep.”*

1st iteration

— **The faculty by which**..we are subject to illusion when
affected by disease, is identical with that which produces illusory
effects in

sleep.

2nd iteration

— The faculty by **which...we are subject** to illusion when
affected by disease, is identical with that which produces illusory
effects in

sleep.

3rd iteration

— The faculty by which...we are **subject to illusion when**
affected by disease, is identical with that which produces illusory
effects in

sleep.

4th iteration

— The faculty by which...we are subject to illusion **when**
affected by disease is identical with that which produces illusory
effects in

sleep.

5th iteration

— **The faculty by which...we are subject to illusion when affected by disease, is identical with that which produces illusory effects in sleep.**

6th iteration

— **The faculty by which...we are subject to illusion when affected by disease, is identical with that which produces illusory effects in sleep.**

The Split Text approach returns two files:

- Parameters table: contains the parameters for each graph generated by a text file.
- Mean Parameters table: contains the mean parameters of all graphs generated by a text file.

2. **Line:** For this option, one graph is generated for each text line. The parameters table contains a line of graph parameters for each line of the text file(s).

3. **Subject Informed on Text File** – For this option, the text file(s) must have the subjects delimited by the tags `#{ }`, as shown below:

`#{Dream}`

'A dream is a short-lasting psychosis, and a psychosis is a long-lasting dream.'

`#}`

With this option you will also have the opportunity to analyse a given subject graph as if all its separate components were together in a single text line, as shown below:



The parameters table contains a line of parameters for each subject graph of the text file(s).