# GRAPH SYSTEM REQUIREMENTS SPECIFICATION

# for

# Human-Computer Graph Exploration and Tele-Discovery

**Version 1.0**

**Prepared by Fatima AlSaadeh**
**Supervised by James Abello**

**Rutgers University**

## September 4, 2020

# Contents

# 1 Introduction

## 1.1 Purpose

This Graph system is designed using a collection of the referenced authors' work to visualize various network datasets of a size up to $2^{14}$ edges utilizing a web application using NodeJs, D3, and WebGL, different algorithms, and mechanisms that we will list in detail in this document. In addition to the two and three-dimensional graph visualizing that this project will use, the goal is to make it both interactive and feasible for extracting semantics following different techniques.[3]

## 1.2 Project Scope

The purpose of this graph system is to simplify the large network datasets exploration process by providing a graph visualization and responsive easy-to-use interaction techniques. Above all, this system offers semantic extraction functionalities that will help the user get a summary of their dataset after exploring it.

## 1.3 Intended Audience and Reading Suggestions

Definitions:

- Sparse Net Algorithm [2]:
  - Iteratively finding longest shortest paths in a graph, the diameter of the graph.
    - Find a farthest pair of vertices (x,y) in the graph and a shortest path between them.
    - Find a third vertex (z) of maximum distance from both x and y.
    - Find a shortest path from z to the shortest path from x to y.
  - Iterate.
- Shortest Path: "a path between vertices in a graph such that the total sum of the edges weights is minimum."
- Fixed Point Decomposition [4]:
  - "Edge Partitioned into Maximal Edge Subgraphs of Peel value k."
    - Peel Value: "the largest i  [1, deg(u)] such that u belongs to a subgraph of G of minimum degree i."

    - Fixed Point: "A graph F is a fixed point of degree peeling k if core(F) = V (F) and the peeling value of F is k. Equivalently, a graph F is a fixed point of degree peeling if the vertex peel decomposition of F has only one class and its peeling value is equal to its minimum degree."

- Egonet: "subgraph induced by a vertex neighbors." [5]
- Story Auto Generation Algorithm:
  - start from one of the furthest points in the first sparsenet path. (1)
  - Use it as an opening statement for the story.
  - Discover its neighbors.(2)
  - Use them to continue the previous statement.(3)
  - We have a paragraph now.(4)
  - Find the neighbor of the neighbors that share the most vertices with the previous ones and has the most vertices that are new.(5)
  - Use it to start a new statement that continues the story.(6)
  - Repeat from 3 until reaching the furthest point in the path.(7)
  - Get the next path in the sparsenet Repeat from (3) - (7).(8)

## 1.4 References


[1] *3D Force Graph*. URL: https://bl.ocks.org/vasturiano/02affe306ce445e423f992faeea13521.

[2] James Abello, Daniel Mawhirter, and Kevin Sun. "Taming a Graph Hairball: Local Exploration in a Global Context". In: (2019).

[3] James Abello and Daniel Nakhimovich. "Graph Waves". In: (2020).

[4] James Abello and François Queyroi. "Network decomposition into fixed points of degree peeling". In: (2014).

[5] James Abello et al. "Atlas: Local Graph Exploration in a Global Context". In: (2019).

[6] *Datasets*. URL: https://github.com/benedekrozemberczki/datasets.

[7] *Node.js*. URL: https://nodejs.org/en/.

[8] *Three.js*. URL: https://threejs.org/.

# 2 Overall Description

## 2.1 Product Perspective

The system is designed for network datasets exploration using graph visualization. It is a web-based system implementing client-server model. The Graph System provides different mechanisms for users to discover the data and get summarized semantics from it.

## 2.2 Product Functions

The following are the main functions this system provides:

- Data Preprocessing: The server preprocesses plain csv files, serves the web pages with processed data files, and saves some visualization data (like node layouts) so they can be reused.
- Data Visualization: the client-side provide the system portal where the user can choose the dataset to start the visualization and exploration process.
- Labels on Demand in Different Languages: The system provides on demand labels using the language the user chose if the data provided is mapped to different languages mapping.
- Interactive Data Exploration: once the exploration process starts, the system makes it interactive using :
  - Paths auto exploring on demand.
  - Mouse clicks, mouse hovering, mouse selection and their combinations.
  - Menu driven button and tabs.
  - Basic visual data elements controls : size, color, texture.
  - Zooming: mouse control, elastic window.
  - h-Sliders( one-sided, two-sided).
- Data Filtration and Reduction: the system provides different filtration and reduction techniques to make it easier for the user to understand large datasets using:
  - Fixed point decomposition. [4]
  - Sparsenet.[2]

  - Hotspot Filtering: where hotspots are the vertices with a degree above the average.
  - Nodes coloring by labels, labels ordering and hotspots.

- Data Summary and Annotation: the system provides different summarizing techniques to help the user finish the exploration process with valuable notes and summaries about the data:
  - Generate Story: automatic story generation from the sparsenet path labels and download it on demand.
  - Add Annotation: this technique gives the user the flexibility to add notes on demand and append to these notes interesting vertices automatically.
  - Download Hotspots and Neighbors.

## 2.3 User Classes and Characteristics

This system will support users from different backgrounds like students, researchers, and users who want to achieve network datasets analysis, i.e., social network analysis, internet security, textual analysis, citation analysis. The functionalities provided in the previous section is available for all type of users except for the data preprocessing; **the data files should be sent to the administrators to enter the system and preprocessed**.

## 2.4 Operating Environment

Operating environment for the graph system is as listed below:

- client-server system
- Operating system: Windows, Mac OS, and Linux.
- Data Storage: data2 folder for the datasets in csv format and temp2 folder for preprocessed data generated from the system and the labels mapping in csv format.
- platform: NodeJs [7], THREE.js [8], 3D Force Graph Library [1] and C++

## 2.5 Design and Implementation Constraints

- The labels on demand will be available only if the dataset provided has a labels mapping.
- Every new dataset with new labels will need a code update to perform the color nodes by labels Figure.3.8, hotspots and story generation features.

- The administrator will need to run this command in the bin folder before deployment to make sure, the user will be able to use the sparsenet functionality:

```
g++ -std=c++11 algorithm.cpp -O3 -o sparsenet_approximate
```

- to run this project locally please run :
  in bin folder :

```
g++ -std=c++11 algorithm.cpp -O3 -o sparsenet_approximate
```

  in the root folder:

```
npm install
npm start
```

- This system is tested and verified on chrome web browser.

## 2.6 User Documentation

The following is the user manual for the system, the describing pictures referenced are in section 3:

- The user starts by selecting the dataset from the menu in the left side of the screen.Figure.3.2
  - To explore the whole graph selecting the dataset is enough.
  - To have the fixed point decomposition feature the user need to check the "Using Fixed Point" checkbox in the bottom of the datasets menu before selecting the dataset.
  - Selecting to explore the whole dataset will be interactive only if the dataset with number of edges less than or equal $2^{14}$
  - This step will be followed with "Loading Graph, please wait ..." message prompt in the middle footer of the screen.Figure.3.3
- Whole graph exploration:Figure.3.4
  - Once the graph drawing appears, the user can zoom, pan and rotate.
    - Zooming is using the mouse wheel or touch pad.
    - Rotating the graph you need to click on it using the mouse hold the click and start moving it in the desired direction.
    - Panning the graph it should be moved by mouse click first after that, the user can move it to left, right, top and bottom using the arrow keyboard keys.

- Labels on demand: When the user hover on vertex by mouse.
- Vertex select and drag: click on the vertex to select it and drag it by changing the mouse location to the required position while selecting, Selected vertex will be marked by (+) marker as long as it's selected.
- Vertical spread, horizontal spread, node size and link thickness can be changed by dragging the sliders in the footer of the screen.
- More statistics about the graph can be found in the header of the screen or in the third tab in the right of the screen, and statistics plots in the second tab Figure.3.4, the fourth tab gives the user the connected components and layers ribbons, while in the fifth tab the user can choose various search, filtering and language change options.Figure.3.1
- In the right footer of the screen buttons can be found to add annotation, and generate story.
- In the left footer of the screen buttons can be found to save the layout to see the same graph when the user come back to same dataset, save the drawing as image, (un)pause the drawing and other camera control options.
- The user can explore the egonet of a vertex in a separate popup window by clicking on the keyboard key 'e' and hover by mouse on the vertex, if the vertex has an egonet with number of edges more than $2^9$ this new window will appear, having the hovered vertex highlighted in the background. In this new window the user can show the labels on demand once hover on the vertex, rotate, zoom and change the edges length. This window will appear if the hovered vertex. Figure.Figure.3.6
- The user can explore the egonet of a vertex in a separate popup window by pressing on the keyboard key 'e' and hover by mouse on the vertex, if the vertex has an egonet with number of edges more than $2^9$ this new window will appear, having the hovered vertex highlighted in the background. In this new window the user can show the labels on demand once hover on the vertex, rotate, zoom and change the edges length. This window will appear if the hovered vertex. Press the keyboard key 'e' again to disable this feature.
- The user can start the sparsenet exploration by pressing the keyboard key 's', or choosing sparsenet from the menu on the right of the screen under subview tools: Figure.3.7
  * The first sparsenet path will appear in red color.
  * Hovering on a vertex in the sparsenet will give the user the label and an option to show the neighbors labels.
  * The user can press keyboard key 'a' to show the labels of the vertices with degree above the average. Figure.3.7
  * a new slider will appear in the footer of the screen with Path Sequence label where the user can step to next path, step back or move the slider

to the end show the whole sparsenet graph.

    * Statistics about the visible number of nodes and edges will appear in the header of the screen and in the third info tab.
    * The user can show the vertices that are at distance one from the sparsenet graph by checking the "Show Neighbors" checkbox from the menu on the right of the screen unders sparsenet.
    * Two-sided slider to filter the sparsenet paths appearing can be found in the fifth settings tab "sparsenet paths".
    * Auto exploring shortest path : The user can select any two vertices in the sparsenet graph, Go to the fifth setting tab and check the checkbox "Auto Exploring Path" and choose how the labels should appear in the auto exploring. After that right click on the white space and choose shortest path. The shortest path will be highlighted with labels auto exploring.

  – Click on Add Annotations button a small box will appear "User Annotation" box with text editor the user can start typing notes there after clicking on the first button to the left, save and download.

- Fixed points exploration: Figure.3.14
  – An overall structure of the fixed points will appear.
  – In the fourth tab, the fixed points ribbons will appear:
    * Color coded where each color of the ribbon match the layer color in the overall structure.
    * The outer rectangle width of the ribbon is proportional to the number of edges in the fixed point.
    * The inner rectangle width is proportional to the number of vertices in the fixed point.
    * Numbers to the left represent the peeling value of the fixed point.
  – Click on the ribbon to take you to the desired fixed point exploration. Please read the "Whole graph exploration" instructions above as it applies to exploring each fixed point
  – Hovering on a vertex will give the label and extra information about what other fixed points this vertex appears in.

- Second exploration option - Show vertices and its neighbors on hover:
  – To use this the user should check using fixed points and show vertices and edges on hover checkbox.
  – Another exploration option which involves showing vertex and its neighbors on hover and hide the rest of the topology, keep the previously hovered vertices

and its neighbors and allow the user to save the layout so they can come back to where they left in the future.

  - If a graph with more than one connected component, the user should select a vertex in one the desired connected component, right click on the white space and choose "Explore on hover", this connected component will be explored in the above way the other connected component will remain visible. In this way in the future to catch up on where the user left, the previous cached layout will appear. Figure.3.15
  - To cache the current exploration, click on view options in the bottom left of the screen and click save layout.
  - When you navigate back to the saved layout, it will show where the user left.

- Special Features for "Fabula" dataset and fixed point exploration that will be expanded to other datasets in the future: Note: these features are customized for this dataset, to maek it work for another dataset, some changes should be made on the code for the keywords and tokens customization.
  - Annotation Features: Figure.3.11
    * While exploring one fixed point the user can select a vertex by clicking on it, move the mouse to the white space of the screen and right click, select add label to annotation:
      · The user can find the added label in the User Annotation box, the most recent annotated vertex label will appear on the top.
      · After a vertex was annotated it will be marked by (x) in the graph and that will be cached so whenever the user come back to the fixed point will see it.
      · If the same vertex appears in another fixed point and was selected and annotated, in the user annotation box the label will have (x counter) where counter means how many times the same label was annotated from different fixed points.
      · In the User Annotation box, the user can select a vertex by typing the ID of it as it appears in the text editor, or archive it.
      · Archive a vertex will move its label to the bottom of the box with (-counter) and the (x) marker on the vertex will disappear.
  - Generate Story: Figure.3.12
    * When the user is exploring a fixed point, they can start the sparsenet process by pressing the keyboard key 's'.
    * When the first path appears the user can click the Generate Story button, where an automatic story generation based on the first path will start with a suggested title, every time the user click generate story button, the story will be shuffled and regenerated in a new way.

    - If the user show the next path in the sparsenet the generate story will add a new paragraph generated from the second path, etc.
    - Each path in the sparsenet is a paragraph of the story generated.
- Sparsenet Nodes Coloring:
    - When the user is exploring the fixed point sparsenet graph.
    - In the fifth setting tab the user can check the Color Nodes By Label checkbox, the nodes will be colored as instructed in the mapping that will appear, consecutive nodes can be highlighted on request.3.8
    - In the fifth setting tab the user can check the Color Nodes By Hotspots checkbox, choose from explore the type of hotspots to color, the user can download the hotspot and filter the graph by them. Figure.3.9
    - The hotspots slider filtration start from the first hotspot appears in the sparsenet and every other hotspot connected to it.
- Vertex Search and Neighbors Graph: Figure.3.13
    - The user can search for a vertex using the search box in the fifth tab, type the vertex Id and click "select by Id" button.
    - The selected vertex will be marked by (+) and the camera will zoom to.Figure.3.13
    - When the user hover on a vertex, the label will give an insight on where this vertex appears in other fixed points.
    - If the user discovered the vertex in all the fixed points using the "select by Id" search button, the vertex will be marked by a circle.
    - Now the user can press the keyboard key 'e' and over on the discovered vertex to show its neighbors from different fixed points.
- Fixed Point with More than One Connected Component: Figure.3.10
    - If the fixed point has more than one connected component in the third info tab there will be the connected components with number of vertices more than the average, sorted in descending order, labeled by the label of the vertex with the highest degree in it, with an icon show the shape of its sparsenet.
    - The sparsenet of each connected component can be explored by choosing it from that tab or select one of its vertices right click on the white space on the screen and choose "Draw Selected Sparsenet".

## 2.7 Assumptions and Dependencies

Assumed Factors:

- The user is using chrome web browser.

- The user is aware of the definition of the main algorithms and features are used in the system.

# 3 External Interface Requirements

## 3.1 User Interfaces

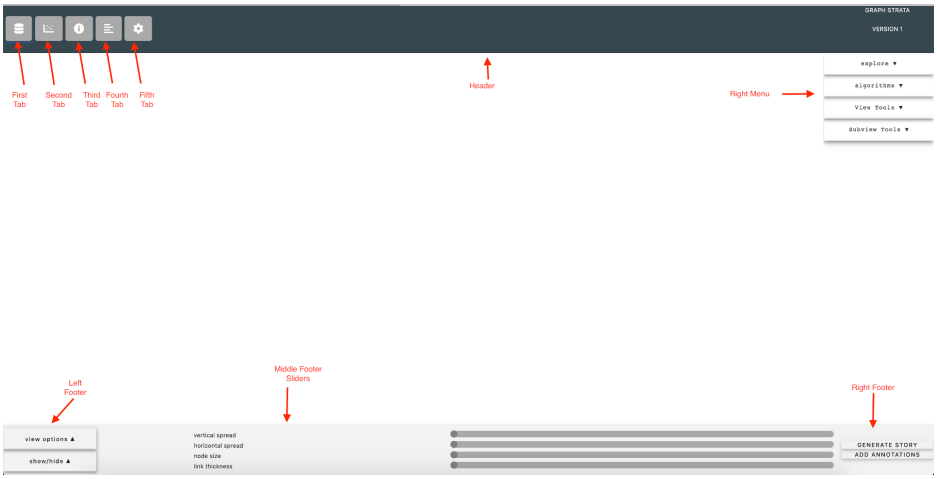


Figure 3.1: Graph System Main Screen

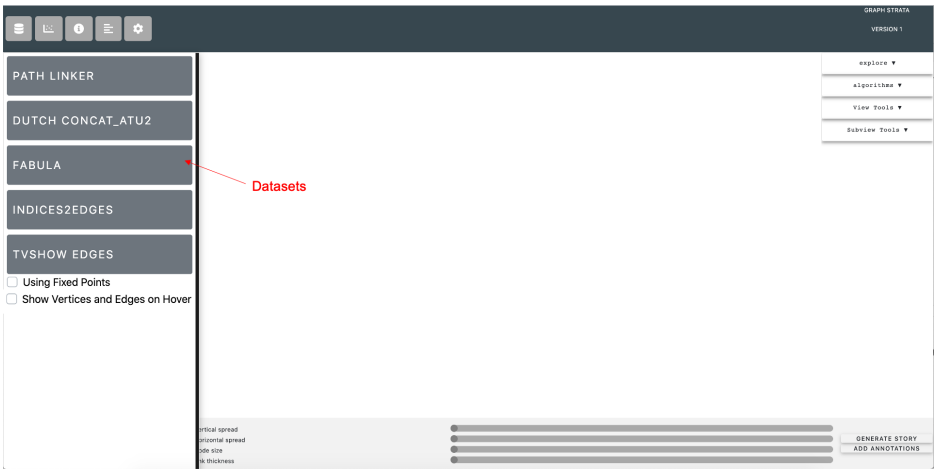


Figure 3.2: Main Screen with Datasets Tab [6]

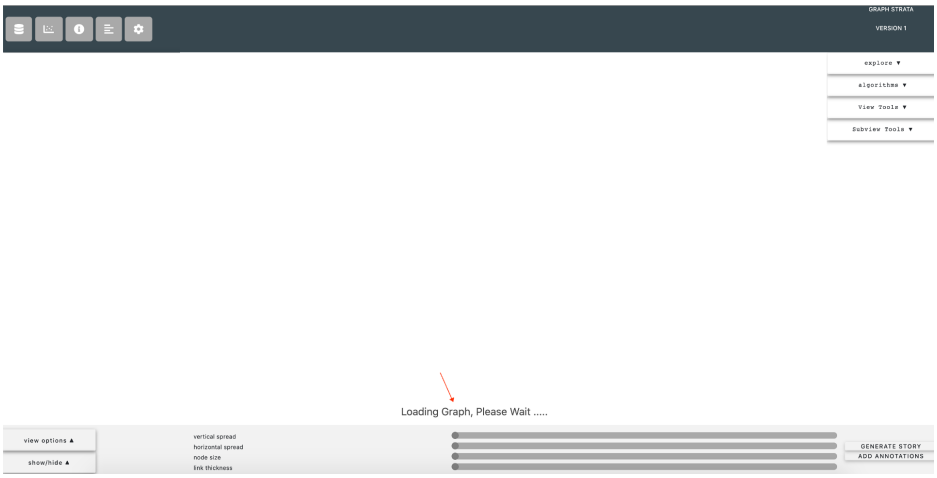


Figure 3.3: Main Screen Loading

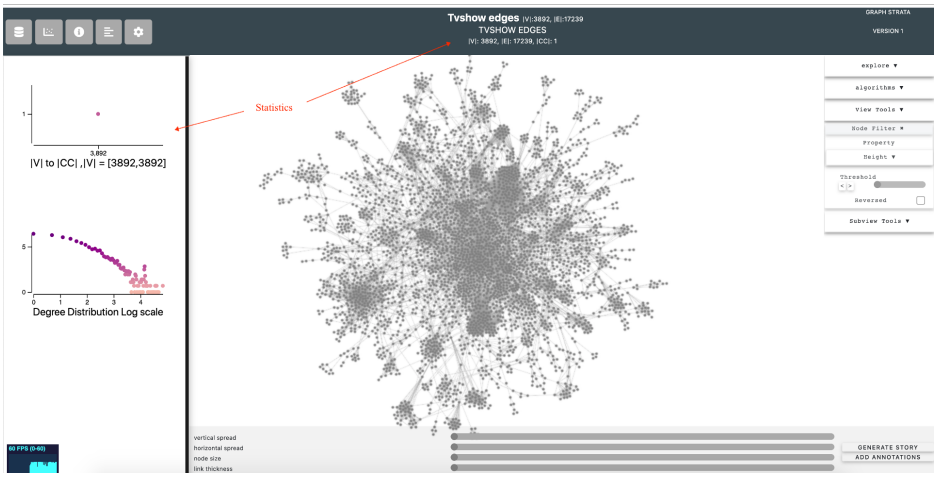


Figure 3.4: Statistics Header and Tab

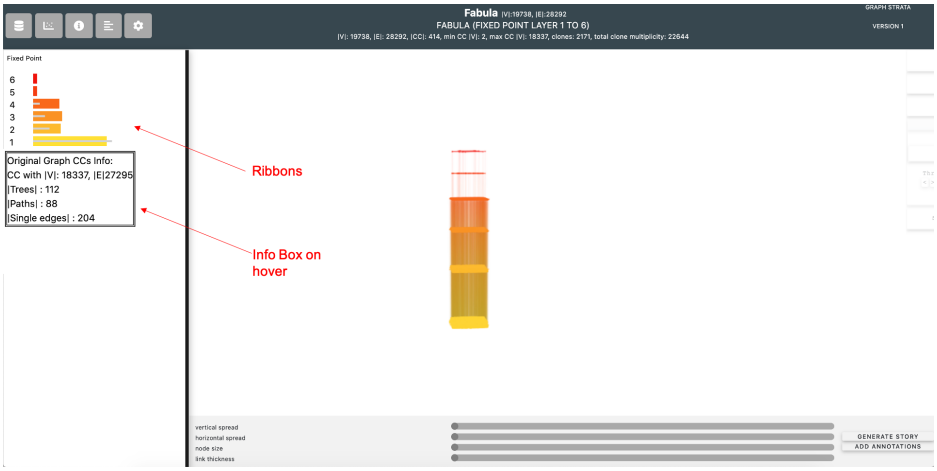


Figure 3.5: Fixed Points Exploring

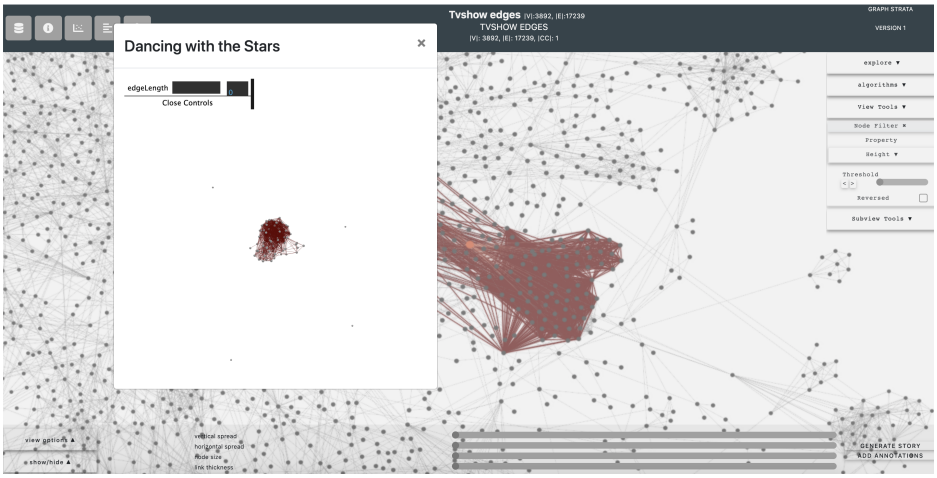


Figure 3.6: Egonet Exploring

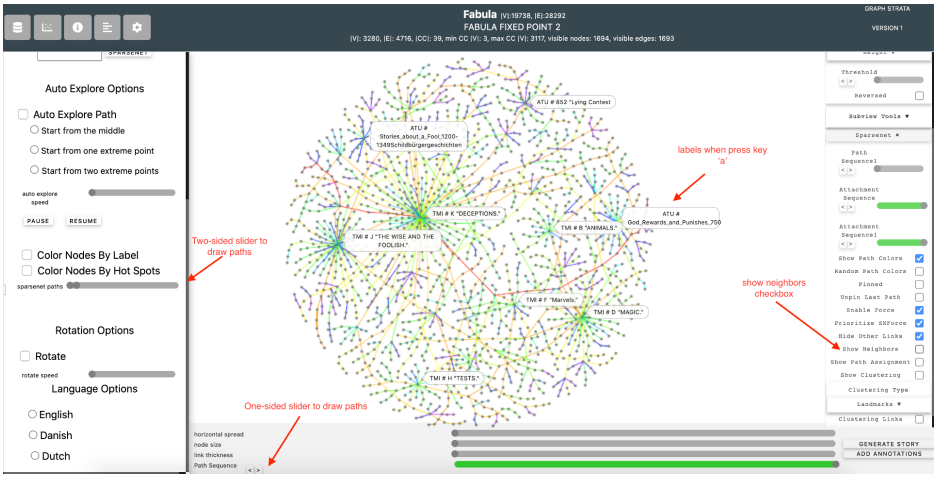


Figure 3.7: Sparsenet Exploring

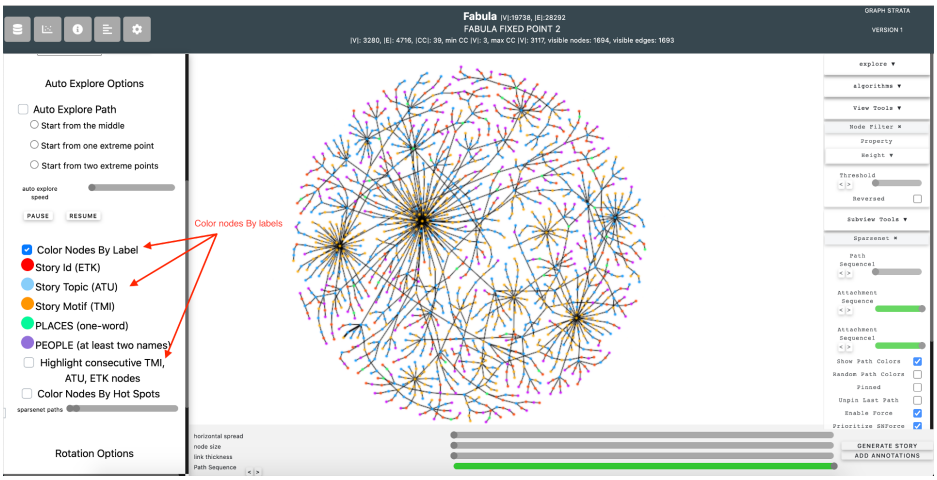


Figure 3.8: Sparsenet Color Nodes By Labels

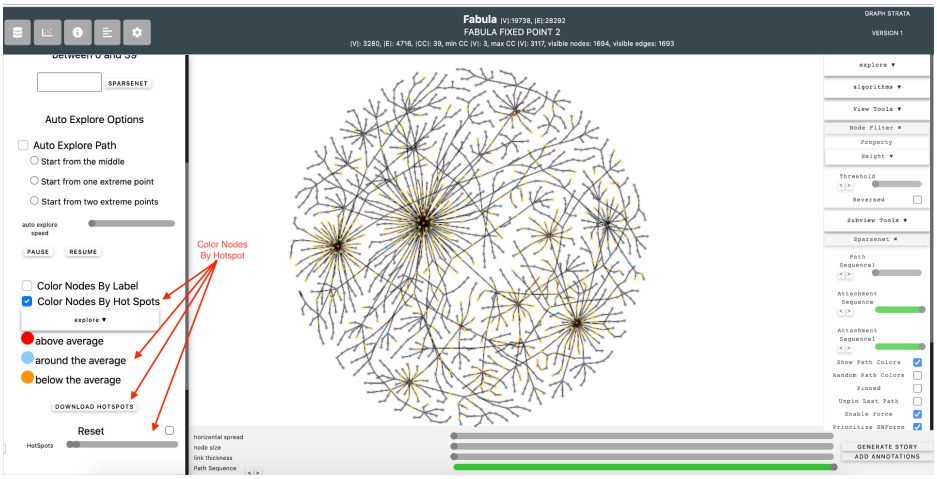


Figure 3.9: Sparsenet Color Nodes By Hotspots

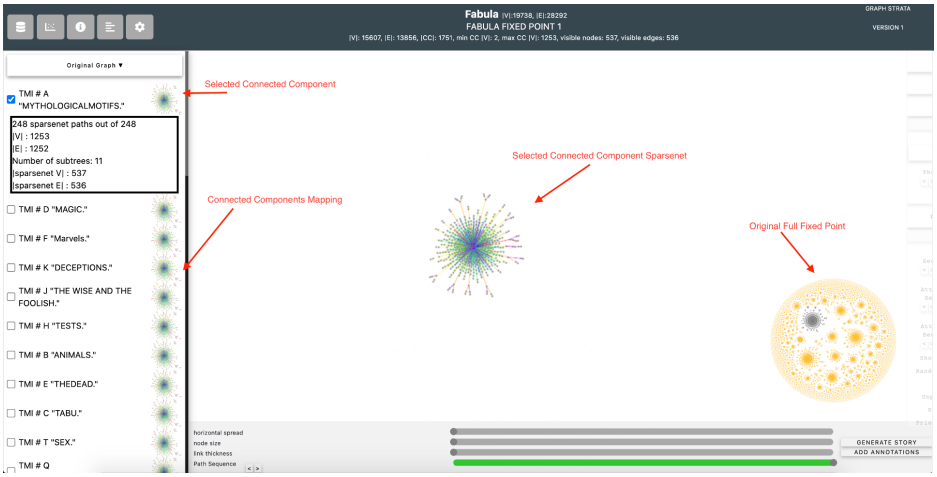


Figure 3.10: Fixed Point with more than one Connected Component

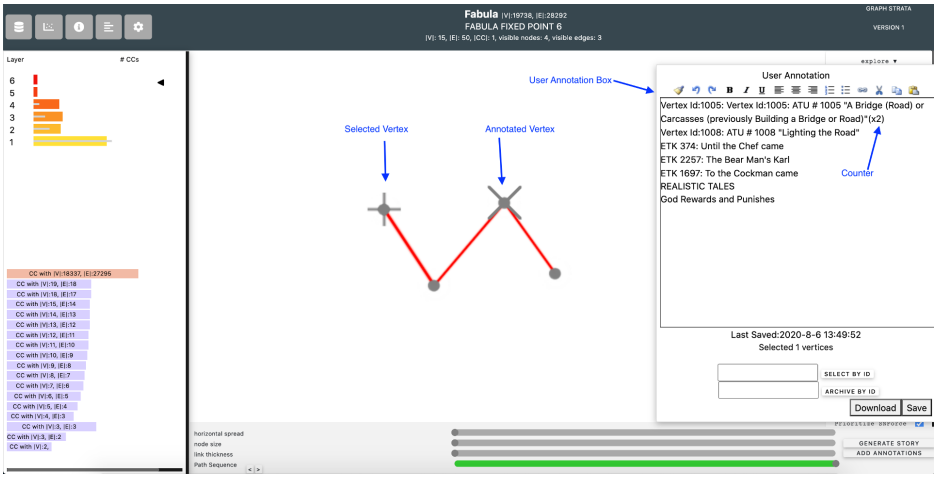


Figure 3.11: User Annotation

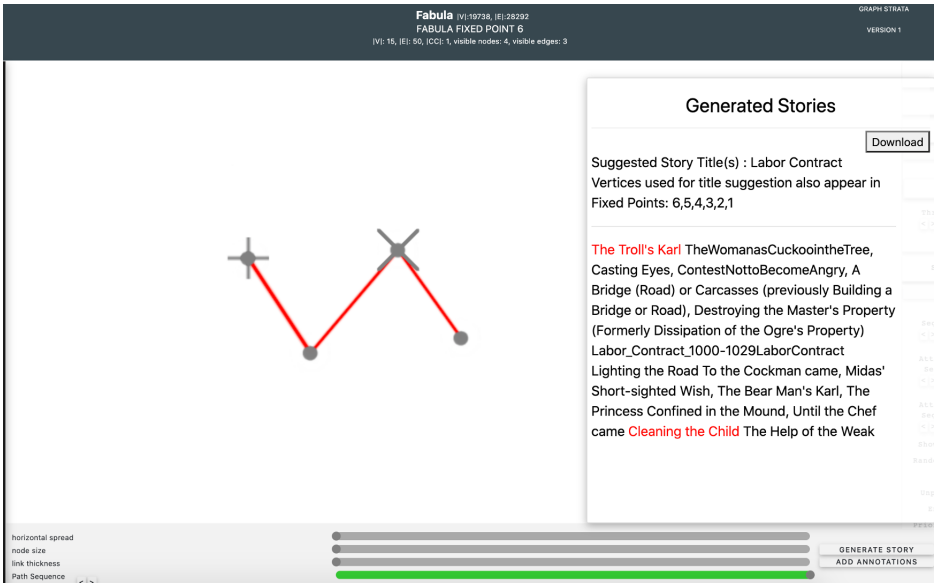


Figure 3.12: Story Auto Generation

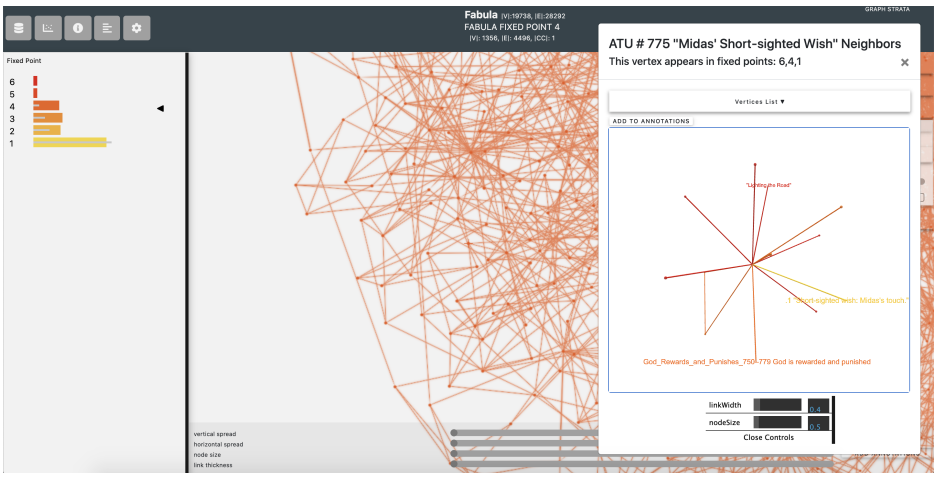


Figure 3.13: Neighbors Graph of a Hovered Discovered Vertex


Figure 3.14: Second Exploration Option - On hover

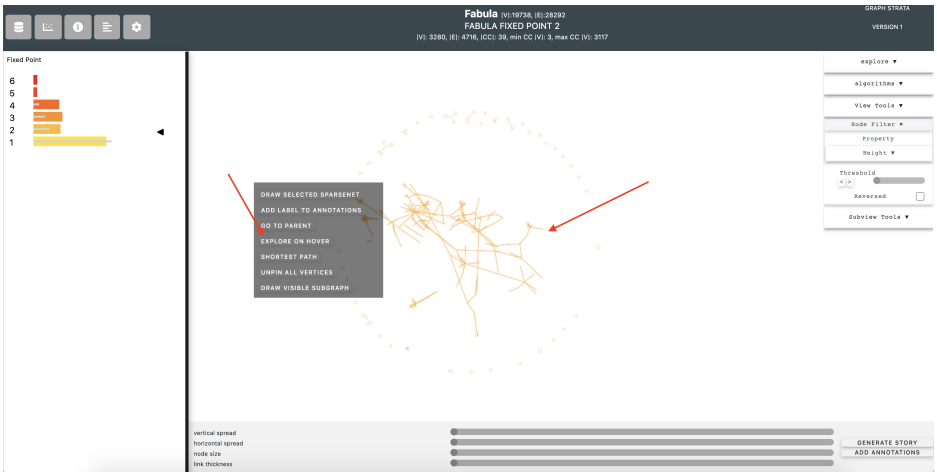


Figure 3.15: Second Exploration Option - Multiple Connected Components

## 3.2 Hardware Interfaces

- Windows, MacOS, and Linux.
- Chrome browser which supports WebGL, HTML  Javascript.

## 3.3 Software Interfaces

| Software Used | Description |
|---|---|
| Three.js and 3D force Graph | JavaScript library used to simulate and display an animated graph |
| Node.js | Used for the server side and data preprocessing |
| Javascript, HTML and Bootstrap | Used for the client side and user interface design |

## 3.4 Communications Interfaces

This project supports chrome web browsers.

# 4 Other Nonfunctional Requirements

## 4.1 Data and Performance Requirements

This system is interactive with datasets of edges in the range [$2^{14}$ - $2^{16}$). The data provided must be in the form of csv, the edges file contains source and target columns and the labels file contains new_id, name columns, if the dataset contains labels in multiple languages it should be added as in the labels file in the example below.
edges and labels file example :

| source | target |
|---|---|
| 924 | 924A |
| 924 | 924A |
| 924 | 924B |
| 924 | 924B |
| 924 | J1804 |

Edges file

| new_id | name_en | name_da |
|---|---|---|
| 924 | ATU # 924 "Discussion in Sign Language" | ATU#924 "Discussion in Sign Language" |
| 924A | ATU # 924A | ATU#924A |
| 924B | ATU # 924B | ATU#924B |
| J1804 | TMI # J1804 "Conversation by sign language mutually misunderstood." | TMI#J1804 "Conversation by sign language mutually misunderstood." |
| 196 | ETK 173: The Troll's Karl | ETK 173: Troldens Karl |

Labels file

The edges file should be placed in data2 folder and the labels file should be placed in temp2 folder. If there is any other classification files it should be added in the temp2 folder, for example in the figure below it shows a classification file, any vertices with an id between 1 and 99 will be classified under wild animals, etc.
For new classifications, the code should be edited accordingly.

| label | start | end |
|---|---|---|
| Wild Animals | 1 | 99 |
| Other Wild Animals | 70 | 99 |
| Wild Animals and Domestic Animals | 100 | 149 |
| Wild Animals and Humans | 150 | 199 |
| Domestic Animals | 200 | 219 |
| Other Animals and Objects | 220 | 299 |

Figure 4.1: Classification file

### 4.1.1 ER Diagram

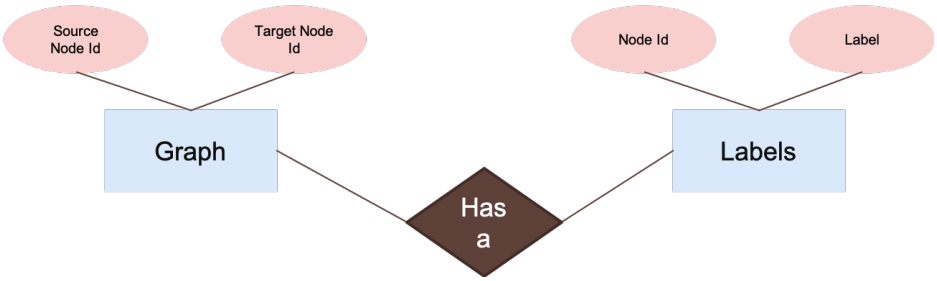


Figure 4.2: ER Diagram

## 4.2 Software Quality Attributes

- AVAILABILITY: The system should be available for graph drawing based on the user request.
- CORRECTNESS: The system should provide the correct graph with correct data upon request.
- MAINTAINABILITY: The administrators should maintain new datasets and sparsenet compilation.
- USABILITY: The system response time should satisfy the users.

# 5 Other Requirements

## 5.1 Appendix A: Demos

Full System Tutorial: https://drive.google.com/file/d/1sa-Tyt-7nNq8eL1k5t9oYoKzHlUiS9Lj/view?usp=sharing

Story Generation: https://drive.google.com/file/d/1u96qnwM35oudRv8M1kGiH-QOzvykgNZ-/view?usp=sharing

Story Generation: https://drive.google.com/file/d/1u96qnwM35oudRv8M1kGiH-QOzvykgNZ-/view?usp=sharing

Color Nodes By Labels: https://drive.google.com/file/d/1SXDlfu2bcBvL5eN2pMnKug8-DWE3JmU5/view?usp=sharing

Color Nodes By Hotspots: https://drive.google.com/file/d/1ONmjrSvNV4fASG9IHiihMpPKw65bggw0/view?usp=sharing

User Annotation : https://drive.google.com/file/d/1tTcEfssl0XeQOt3f50VpOQ9bZd-wT1bC/view?usp=sharing

## 5.2 Appendix B: Analysis Models

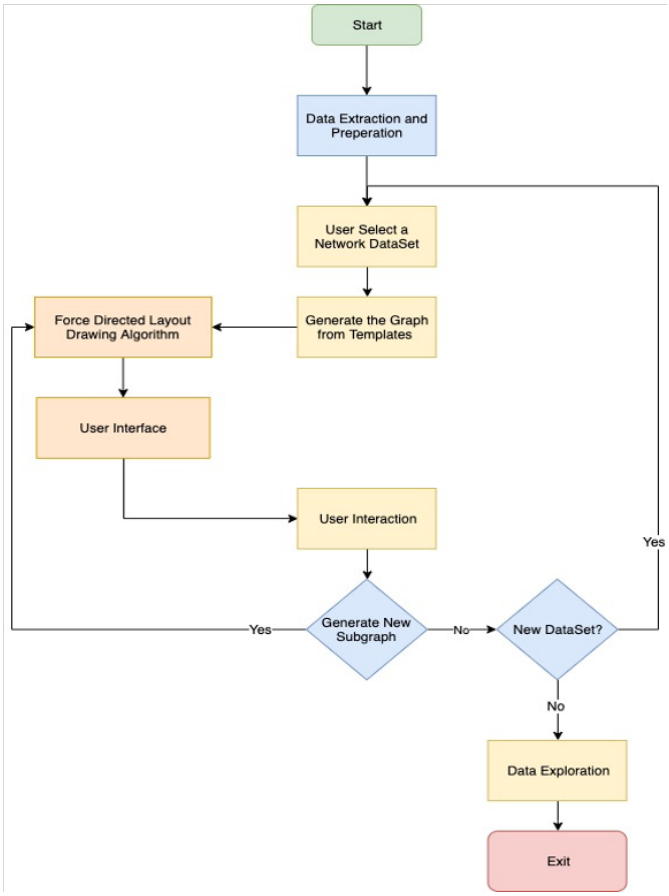


Figure 5.1: Partial Flow Chart Diagram

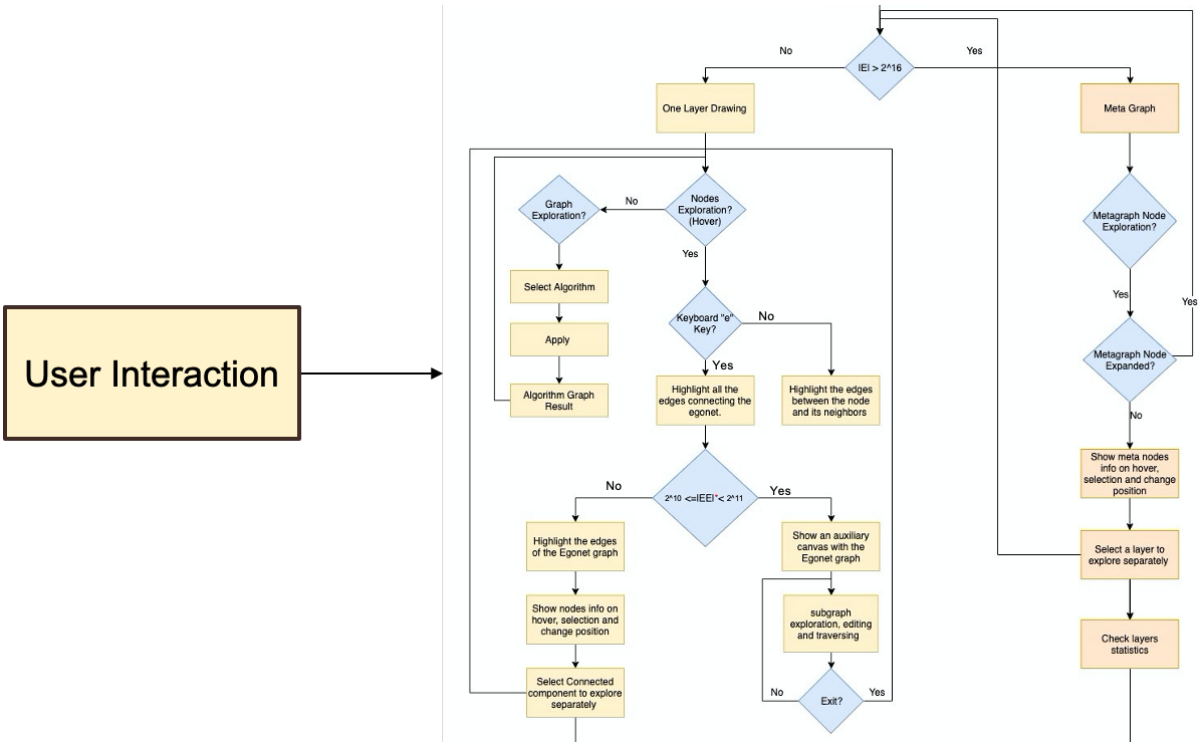


Figure 5.2: Partial Flow Chart Diagram

## 5.3 Appendix C: Fabula Dataset and Other Refrences

Fabula dataset is talking about The Danish stories, it is being collected by UCLA ,It has five main classifications :
1- ETK : Story Id
2- TMI : Story Motif
3- ATU : Story Topics
4- Places: One Word.
5- People: Two Words.
Each of ATU and TMI has further classification that is being used in the system and can be found in the following website :
https://sites.ualberta.ca/~urban/Projects/English/Motif_Index.htm
The documentation template credits goes to Jean-Philippe Eisenbarth was used with the help of the following template: https://github.com/jpeisenbarth/SRS-Tex