# Evaluating Layout and Clustering Algorithms for Visualizing Named Entity Graph

K. Ibrahim[1], B. Ranaivo-Malançon[1], T. Lim[1] and Y.-N. Cheah[2]
[1]*Department of Information System, Faculty of Computer Science and Information Technology, Universiti Malaysia Sarawak, 94300 Kota Samarahan, Sarawak, Malaysia.*
[2]*School of Computer Sciences, Universiti Sains Malaysia, 11800 USM, Penang, Malaysia.*
*khairunnisaibr@gmail.com*

***Abstract*—Myriad of layout and clustering algorithms exist to generate visual graphs of named entities. Consequently, it is hard for researchers to select the appropriate algorithms that fulfill their needs. This paper intends to assist the researchers by presenting the performance evaluation of the combination of graph layout algorithm followed by a clustering algorithm. The layout algorithms are OpenORD and Hu's algorithms, and the clustering algorithms are Chinese Whispers and Givan-Newman algorithms. The evaluation is carried out on bio-named entities that are linked by some annotated relations. The results of the experimentations highlight the strengths and weaknesses of the four combinations regarding running time, loss of relations (or edges), edge crossing, and cluttered display.**

***Index Terms*—Bio-Named Entities; Graph Clustering Algorithm; Graph Layout Algorithms; Network Visualization.**

## I. Introduction

Due to the continual advancement in digitization technology, more and more printed documents are accessible to researchers as well as the public. However, the access is often limited to a simple search. Recently, the use of graph networks to visualize and analyze these digitized documents has gained a wide attention. As one can read in the Visualizing Historical Networks website by the Harvard University in 2011, "network visualization presents a way of conceptualizing relationships and transmission of ideas in historical communities" [1]. A graph is a powerful representation that can support many natural language processing (NLP) tasks such as machine translation, information retrieval, information extraction, word sense disambiguation, document summarization, key-word extraction, topic identification, co-reference resolution, syntactic parsing, part of speech tagging, etc. [2].

A crucial point when creating a graph is to display a good drawing of the layout for its viewer. For example, too many edge crossings may confuse the viewer. Sometimes, a node in a graph is better to be placed close to other nodes related to it. These decisions are determined by graph layout algorithms. Clustering is a process of arranging similar objects to form a single cluster. Many NLP applications have applied this process to discover automatically similar documents, similar words, similar named entities, etc. The definition of the term "named entity" (NE) remains unclear and for Marrero et al. [3], the definition depends on the purpose and domain of application. The most common NEs are Person, Location, Organization, and Date. However, in the domain of bioscience, NEs can be the names of disease and treatment.

The purpose of this paper is to evaluate the performance of layout and clustering graph algorithms when visualizing the relations between NEs. The evaluated algorithms are OpenORD [4] and the efficient force-directed graph drawing algorithm by Yifan Hu [5], which will be called as Hu's algorithm for layout algorithms, and Chinese Whispers [6] and Girvan-Newman [7] for clustering algorithms. The results of the evaluation highlight the strengths and weaknesses of the combination of a layout algorithm followed by a clustering algorithm.

The remainder of this paper consists of the following sections. Section II presents related work on entities relations and visualization of documents in the form of graph. Section III describes the proposed evaluation method in detail. Section IV presents the results of the evaluation, which is discussed in Section V. Section VI gives a summary of the work with future research directions.

## II. Related Work

Many documents have been digitally archived worldwide, enabling computer analysis to be done. The trend is now to visualize these documents in many different forms such as network graphs, map as can be seen in [8], word cloud (e.g., IBM Watson News Explorer), vertical tag cloud [8], timeline (e.g., WikiSAGA [9]), treemap (e.g., the Yale project Photogrammar), dashboard (e.g., the Yale project Photogrammar). The common objective of document visualization systems is to get access visually and interactively into the document content to enable the exploration, analysis, search, and browsing.

### *A. Visualizing NE Relations*

Grobelnik and Mladenic [10] created Contexter to visualize the summaries of news articles by creating a graph of NEs. Two NEs are connected if they occur in at least one shared document. The news articles are pre-processed and transformed into two alternative representations: NEs and bag-of-words. The latest is used to extract keywords. The identification of NEs is simply based on word capitalization and phrase similarity identification ("frequent and significant consecutive sequences of several words" [10]) to unify the different surface forms of the same NE such as 'Bill Clinton', 'President Clinton', and 'Clinton'. The graphical user interface of Contexter allows a user to visualize and browse the graph of NEs as well as the contexts of a NE (a set of keywords collocated with the selected NE, a set of other NEs collocated with the selected NE, and a set of keywords collocated with the simultaneous appearance of the selected and most frequent other NEs). Grobelnik and Mladenic [10] illustrated their approach by processing 11,000 article summaries of the length 200-400 words from the ACM Technology News service. However, they did not provide any evaluation of Contexter.

Osaki et al. [11] visualized Hyohanki, a personal diary written from 1132 until 1171 by a Japanese aristocrat. Their objective was to visualize through a network graph the relations between historical persons cited in the diary. They used the co-occurrence of an entity Person and an entity Location within a paragraph to establish a relation between the two entities. Each person is represented by a vector "having the number of co-occurrences with a place name as a component" [11]. The graphs were generated based on the clusters of historical persons by computing the cosine similarity and a modified K-means algorithm.

Itoh and Akaishi [12] proposed "an interactive visualization system to extract networks of historical figures from historical data and to show time-varying changes in their relationships". The authors used Dai-Nihon Shiryo historical database. It contains Japanese historical documents (from ninth to seventieth centuries) arranged chronologically. Each record in the database represents an event consisting of the names of historical figures, their titles, a list of location names, a list of keywords, and the text corresponding to the event. The relation between two persons p1 and p2 in a particular year is computed by dividing the number of records that contain p1 and p2 by the number of records containing p1 only. In the generated graph, the size and color of a node represent the importance of a person. The size of a node is defined by "the summation of other people's person dependency on person" and "the color of the node is defined by the ratio of the number of in-links to out-links of the node" [12]. The strength of a relationship between two persons is emphasized by the length of an edge. A short edge indicates a strong relationship.

Due to the availability of many open-source softwares like Gephi [13] for network graph generation and analysis, a great number of graph projects have emerged in recent years. These projects are usually exhibited on websites. For instance, the website Visualizing Historical Networks [1] provides network graphs that featured "the way people in the past interacted with each other and their surroundings". Another website using Gephi is the visualization of the history of philosophy created by Simon Rapper [14], where data about philosophers in Wikipedia are connected in a graph with an "influenced by" relation. The author used the information stored in DBpedia from an infobox on a Wikipedia page. A website created by Chris Harrison [15] exhibits the visualization of various data using his own visual tools. One of them is a network graph of NEs from the King James Bible. The focus is on Person and Location. The nodes in the graph are people and places and the edges were defined based on the co-occurrence of pairs of NEs mentioned in the same verse. A clustering algorithm was used to layout the nodes so that related NEs are placed close to each other. Labels are scaled according to the number of connections they have.

### *B. Evaluation of Graph Layout Algorithms*

Layout algorithm is used to draw graphs by laying out the components of the graph such as nodes and edges based on various mechanisms. Layout algorithms are usually assessed based on a set of aesthetic rules such as even distribution of nodes and edges, similar length of edges, and minimum number of edge crossings [16]. Huang et al. [17] once published their work where practices of quality evaluation in graph drawings are reviewed. They stated that in order to describe a graph as a graph of highest quality, all sorts of quantitative and qualitative aspects should be considered. The quantitative aspects discussed include minimising the number of edge crossings, maximising crossing angle resolution, maximising node angular resolution and uniformizing the edge lengths. The aesthetic of crossings is often used to judge the layout quality out of convenience. However, using crossing alone as the criteria is proven to be not necessarily equal to layout quality as a graph with one extra crossing is not always less readable than a graph with one less crossing. From qualitative aspects, graph drawing of highest quality is proposed to give the smallest amount of time and effort by the users to answer questions about the graph, while giving the highest average accuracy of the users' answers and highest efficiency in its drawings.

In 2016, another work published [18] indicated that graph aesthetics should not be considered based on one aspect only. Aesthetics include number of crossings, size of crossing angles, edge lengths and angular resolution. This paper concluded that graph visualization would be more effective if compromises are made between multiple aesthetics. The readability of the drawings made by different algorithms are compared based on the response time taken by a group of viewers to complete several given exploration tasks on the graph, their response accuracy, mental effort used to answer the questions and the visualization reading efficiency.

Jacomy et al. [19] compared ForceAtlas2 layout algorithm to two other layout algorithms that are Hu's algorithm and Fruchterman Reingold algorithm. They compare the speed time of the algorithms in generating graphs until the significant form of the graphs are met. The iteration performance, which is the number of iterations the layout algorithms need to reach the ideal quality of the graph are also compared. They also compared the short and long term quality of the algorithms which are the quality reached just after the most efficient phase of the layout and the quality measured specifically at the 750th iteration of the algorithm respectively. It was found that ForceAtlas algorithm shows better iteration performance and a better short term quality than Fruchterman Reingold and Hu's algorithm. It also showed better long-term quality in all but one dataset tested.

Hu et al. [5] created a graph algorithm and then compared it with Walshaw's algorithm in terms of central processing unit (CPU) time taken for each algorithm to run. Walshaw's algorithm is a heuristic method that is used to draw graph by using a multilevel framework and a force-directed placement algorithm. The evaluation is further done by printing the 2D layouts of the graphs created by each algorithm. Each graph was then compared in terms of the cluttering of nodes in it, and whether the layouts of the graphs are visually more or less acceptable compared to each other. He concluded that Hu's algorithm is efficient and of high quality for large graphs and are competitive to Walshaw's algorithm.

Hachul et al. [20] in their work on comparing fast algorithms for generating large graphs used running time, scalability, edge crossings and uniformity of edge length in their evaluation. The work focused on straight-line drawings of general large graphs that have been invented based on force-directed or algebraic approaches. The dataset used ranged from real-world and artificial large graphs.

### *C. Evaluation of Graph Clustering Algorithms*

Graph clustering algorithm is an algorithm used to group nodes that have similarity among each other in a graph. Clustering can be done by computing the value of edge betweennes, the number of edges each node has, and sometimes according to the k-means value, depending on the method used by the algorithm. The quality of a clustering algorithm can be measured using various criteria including modularity, conductance, silhouette index, relative density and cluster path lengths [21].

## III. PROPOSED METHOD

The proposed method goes through three main steps as depicted in Figure 1.

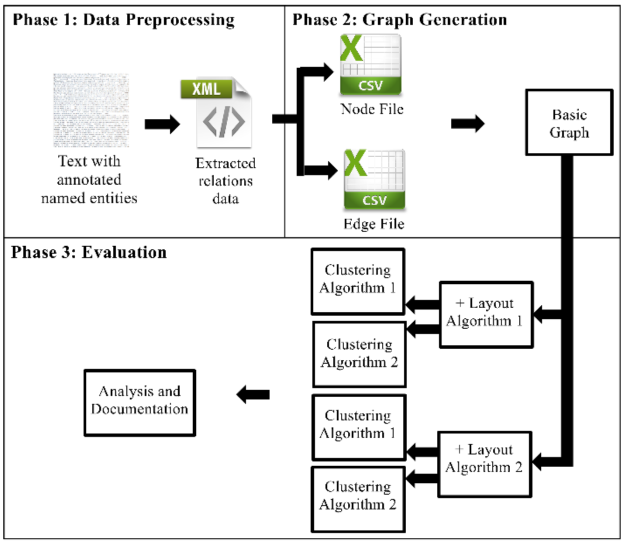
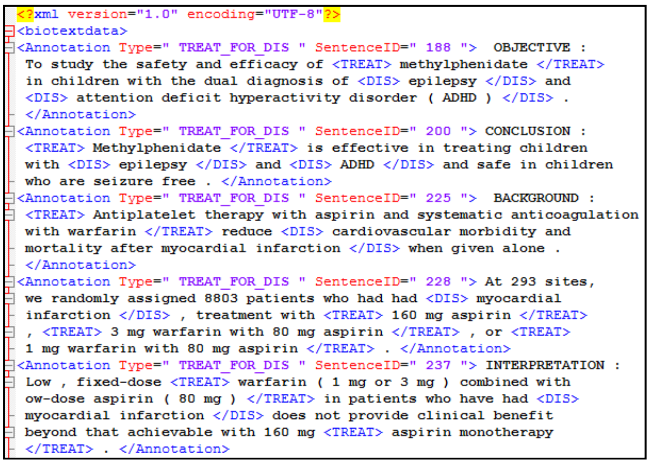



Figure 1: Overview of proposed method

In the first phase, which is the data preprocessing phase, the labelled text is converted into XML file for the extraction of NEs and relations. Once extracted, the new XML is transformed into CSV file, one of the file formats accepted by Gephi [13], a visualization tool for the interactive exploration and visualization of networks. Two CSV files are required to create the graph, namely the node file and the edge file. During the graph generation phase (phase 2), a basic graph is generated without any specific algorithm. This graph just displays the relations between nodes as specified in the node and edge files. In phase 3, which is the evaluation phase, two graph layout algorithms are applied to the initial graph. Then, each of the graphs generated from the two layout algorithms are clustered by two clustering algorithms. Further evaluation on the effect of the clustering algorithms on the graph with different layout is made. The analysis is documented and conclusions are then made.

The dataset used for the experiments was used in the BioText Project of the University of California, Berkeley [22]. The source texts for the annotation are from MEDLINE 2001 and were then classified according to their semantic relations [23]. The sentences in the dataset were labelled manually by one single annotator. The labels correspond to the relations between entities Disease and Treatment. Overall, the annotator has determined seven relations: Cure, Only Disease, Only Treatment, Prevent, Side Effect, Vague, and Does Not Cure. For the work presented in this paper, only five of these relations were considered: Cure, Prevent, Side Effect, Vague, and Does Not Cure. The NEs involved in these relations are DIS, DIS_NO, DIS_PREV, DIS_SIDE_EFF, DIS_VAG, TREAT, TREAT_NO, TREAT_PREV, TREAT_SIDE_EFF, and TREAT_VAG. The Only Disease relation and Only Treatment relation are discarded from the experiments because they do not show significant relations instead of just a list of diseases or treatments mentioned in the same sentence. In total, the dataset for the experiments contains 964 sentences, and thus 964 relations, and 1656 NEs. A sample of the annotated NEs and relations is shown in Figure 2.



```
<?xml version="1.0" encoding="UTF-8"?>
<biotextdata>
<Annotation Type=" TREAT_FOR_DIS " SentenceID=" 188 ">  OBJECTIVE :
 To study the safety and efficacy of <TREAT> methylphenidate </TREAT>
 in children with the dual diagnosis of <DIS> epilepsy </DIS> and
 <DIS> attention deficit hyperactivity disorder ( ADHD ) </DIS> .
 </Annotation>
<Annotation Type=" TREAT_FOR_DIS " SentenceID=" 200 "> CONCLUSION :
 <TREAT> Methylphenidate </TREAT> is effective in treating children
 with <DIS> epilepsy </DIS> and <DIS> ADHD </DIS> and safe in children
 who are seizure free . </Annotation>
<Annotation Type=" TREAT_FOR_DIS " SentenceID=" 225 ">  BACKGROUND :
 <TREAT> Antiplatelet therapy with aspirin and systematic anticoagulation
 with warfarin </TREAT> reduce <DIS> cardiovascular morbidity and
 mortality after myocardial infarction </DIS> when given alone .
 </Annotation>
<Annotation Type=" TREAT_FOR_DIS " SentenceID=" 228 "> At 293 sites,
 we randomly assigned 8803 patients who had had <DIS> myocardial
 infarction </DIS> , treatment with <TREAT> 160 mg aspirin </TREAT>
 , <TREAT> 3 mg warfarin with 80 mg aspirin </TREAT> , or <TREAT>
 1 mg warfarin with 80 mg aspirin </TREAT> . </Annotation>
<Annotation Type=" TREAT_FOR_DIS " SentenceID=" 237 "> INTERPRETATION :
 Low , fixed-dose <TREAT> warfarin ( 1 mg or 3 mg ) combined with
 ow-dose aspirin ( 80 mg ) </TREAT> in patients who have had <DIS>
 myocardial infarction </DIS> does not provide clinical benefit
 beyond that achievable with 160 mg <TREAT> aspirin monotherapy
 </TREAT> . </Annotation>
```

Figure 2: Sample bio-NEs relations data

## IV. RESULTS

### *A. Results of Graph Layout Algorithms*

There are many graph layout algorithms but for this study, two of them were evaluated, namely OpenORD and Hu's algorithms.

OpenORD is a force-directed layout algorithm with the number of iterations controlled by a simulated annealing with five stage cooling schedule (liquid, expansion, cool-down, crunch, and simmer). It has been designed by its authors to overcome the problems of force-directed layout algorithms that do not scale well to large graphs and do not work well on real data [4]. As stated by Kobourov, graphs generated by force-directed layout algorithms "tend to be aesthetically pleasing, exhibit symmetries, and tend to produce crossing-free layouts for planar graphs" [24]. The algorithm optimizes the following function:

$$\min_{\mathbf{x}_1,\ldots,\mathbf{x}_n} \sum_i \left( \sum_j \left( w_{ij} d(\mathbf{x}_i,\mathbf{x}_j)^2 \right) + D_{\mathbf{x}_i} \right) \qquad (1)$$

where: $x_i$ = positions of nodes
$w_{ij}$ = edge weights
$D_{xi}$ = density of edges near $x_i$

This algorithm is originated from Fruchterman Reingold algorithm and aims to better distinguish clusters in graphs. OpenORD algorithm is expected to display clearly nodes that are closely connected to each other by putting them closer and exhibit communities in the graph. To avoid the cost of the density term $D_{xi}$, the authors of the algorithm used a grid-based method that can reduce the computation complexity from *O(n²)* to *O(n)*. In Gephi implementation of the algorithm, the setup used is the default for version 8.1. To prevent white spaces in the layout of a graph, users can make use of the edge-cutting heuristic available in OpenORD. Edges are cut if they are long and have large weight. The value 0 indicates no cutting of edges and 1 indicates aggressive cutting of edges. In the default set up of the algorithm, 0.8 which is closed to 1 is used. This high value is used so that clustering is encouraged to occur in the resulting graph without cutting every edge that present in it.

The overall graph generated by OpenORD algorithm with the subset of the bio-named entities dataset is shown in Figure 3 and an insight of the same graph is shown in Figure 4.

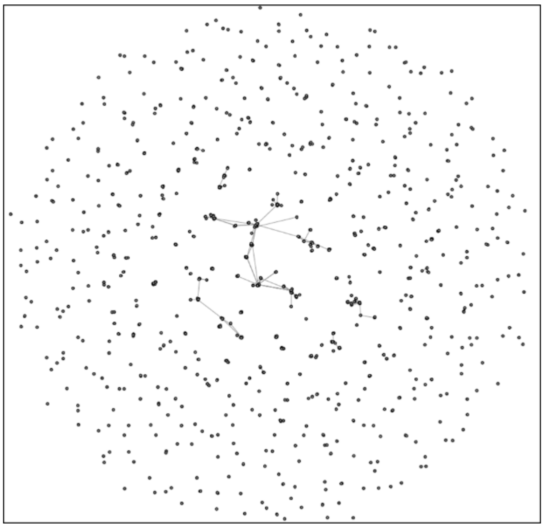


Figure 3: Overall graph generated by OpenORD algorithm

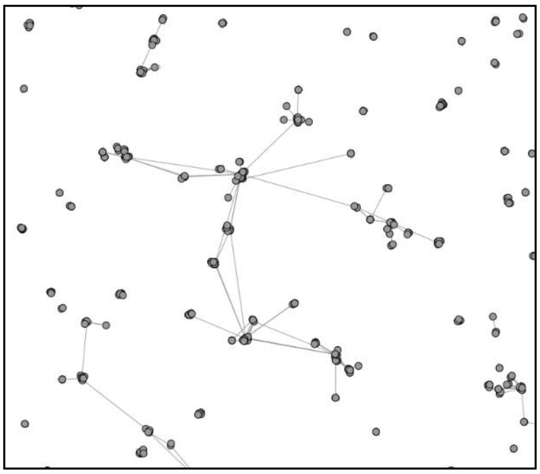


Figure 4: Insight of the graph generated by OpenORD algorithm

Hu's algorithm [5] is a force-directed layout algorithm that uses various combinations of techniques intended to overcome several problems faced by other algorithms. It uses multilevel approach to find global optimal layouts and to reduce complexity. It also manipulates the octree technique to approximate short and long range forces efficiently. As mentioned in the Gephi Tutorial Layouts, "the repulsive forces on one node from a cluster of distant nodes are approximated by a Barnes-Hut calculation, which treats them as one super-node." Apart from that, it also uses other techniques such as hybrid coarsening scheme, adaptive step and octree depth control to work efficiently and adapts a general repulsive force model to overcome peripheral effect of Fruchterman Reingold spring electrical model.

In Gephi implementation of this algorithm, the setup used is also the default for version 8.1. Optimal distance indicates the natural length of the repulsion springs between nodes where higher values will place the nodes in the graph farther apart. The value of optimal distance used in this work is 100.00. A bigger and unnecessarily spacious graph will be produced if the value is increased more than that. Theta (ϴ) is used to approximate the Barnes-Hut calculation in the algorithm. The smaller the value of ϴ, the more accurate the approximation to the repulsive force, and the more computationally expensive the algorithm is. In this work, ϴ is set to 1.2. Hu in his work [5] found out that setting ϴ as 1.2 is a good compromise and uses this value throughout his work.

The overall graph generated by Hu's algorithm with the subset of the bio-named entities dataset is shown in Figure 5 and an insight of the same graph is shown in Figure 6.

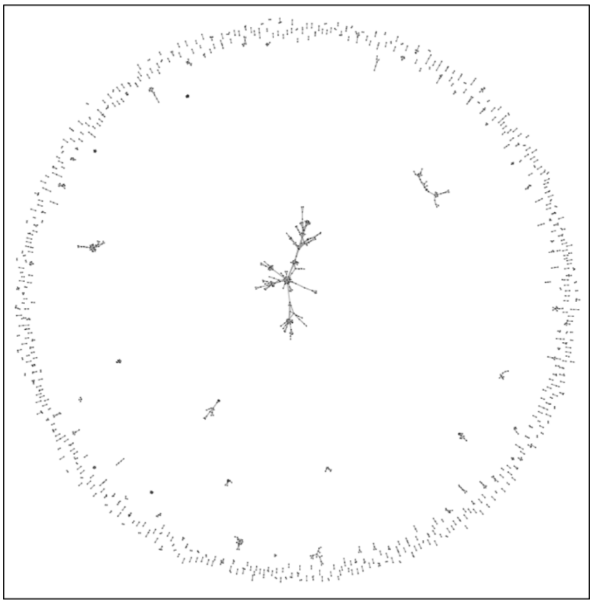


Figure 5: Overall graph generated by Hu's algorithm

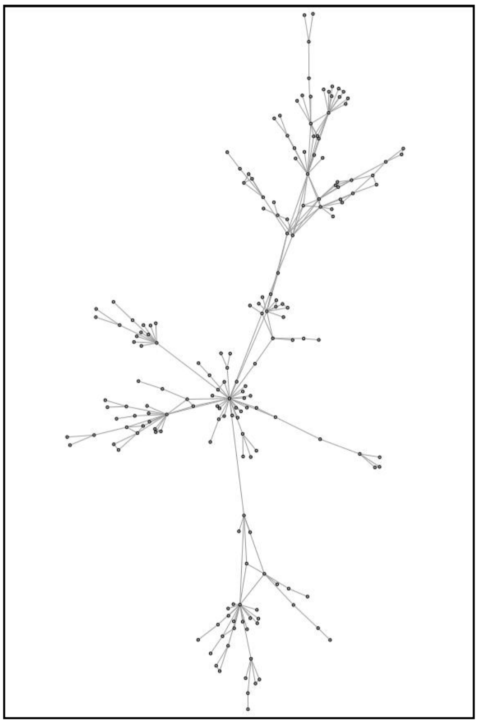


Figure 6: Insight of the graph generated by Hu's algorithm

A summary of the properties of OpenORD and Hu's layout algorithms is given in Table 1.

Table 1
Properties of the Two Graph Layout Algorithms

| | OpenORD (OO) | Hu's algorithm (YH) |
|---|---|---|
| Time complexity | $O(n^2)$, where $n$ is the number of nodes | $O(\|n\| \log(\|n\|) + \|E\|)$, where $n$ is the number of nodes and $E$ the number of edges in the graph |
| Scalability | From 100 to one million nodes | From 100 to 100,000 nodes |
| Stop criterion | Automatic | Automatic |

The two layout algorithms were evaluated based on four criteria: time taken to generate a graph, edge crossing, and cluttered nodes. Table 2 shows the results of their evaluation, where OO refers to OpenORD algorithm and YH to Hu's algorithm.

Table 2
Results of Graph Layout Algorithms

| Algorithm | Time | Aesthetic Criteria | |
|---|---|---|---|
| | | Edge Crossings | Cluttered Nodes |
| OO | 00:05.96 | Edges overlaps and crossed making them not visible at all | Many cluttered nodes |
| YH | 00:10.95 | Very few edge crossings | No cluttered nodes |

### B. Results of Graph Layout and Clustering Algorithms

Like graph layout algorithms, there are many graph clustering algorithms. The two graph clustering algorithms studied in this paper are Chinese Whispers and Girvan-Newman. These algorithms were given as inputs to the outputs of the previous two layout algorithms.

Chinese Whispers algorithm [6] is a clustering algorithm that has been designed by its author to generate large graphs – the case of datasets in NLP – with low time complexity. Chines Whispers run with few iterations. At the initial stage, all nodes are assigned to different clusters. Then, the algorithm randomly processes the nodes. A processed node is assigned to a neighborhood cluster with the maximum sum of edge weights. If more than one neighborhood cluster is possible, the algorithm selects randomly one cluster. For the bio-named entities dataset, the weight of an edge is represented by the number of relations that exists between two nodes. The graph obtained from Chinese Whispers algorithm is weighted, undirected, and "dense regions in the graph are grouped into one cluster while sparsely connected regions are separated" [6].

The mechanism of the algorithm in an undirected unweighted graph is as follows:

1. Initially, each node is assumed to represent one cluster.
2. All nodes are selected randomly one by one. Each node is moved to the cluster it is most linked with. If there is equal number of links, the node is randomly assigned to any of the cluster.
3. Repeat Step 2 until a predetermined number of iterations or until the process converged.

Girvan-Newman algorithm [7] is a clustering algorithm that was designed by its author to determine community structure (or cluster) in a graph using edge (and not node) betweenness centrality. The algorithm creates clusters by removing edges that have the highest betweenness centrality value in the graph. "The betweenness centrality of an edge corresponds to the number of shortest paths between nodes that go through that edge." [2]. The edge elimination process is performed iteratively until the edge with the highest betweenness centrality in the graph falls below a user-defined threshold. For this experimentation, the threshold was set to be 541 for the bio-named entities dataset, the minimum number of clusters that can be generated by the algorithm based on their edge betweenness value. The final clusters are expected not to overlap.

The mechanism of the algorithm is as follows:

1. Calculate betweenness value for all edges in the graph.
2. Remove the edge with the highest betweenness value.
3. Recalculate betweenness value for all edges affected by the removal.
4. Repeat Step 2 and Step 3 until no edges remain.

A summary of the properties of Chinese Whispers and Girvan-Newman clustering algorithms is given in Table 3.

Table 3
Properties of the Two Graph Clustering Algorithms

| | Chinese Whispers (CW) | Girvan-Newman (GN) |
|---|---|---|
| Time complexity | $O(n)$, where $n$ is number of nodes | $O(n^3)$, where $n$ is number of nodes |
| Scalability | Suitable for large graphs, but can be inconclusive on small graphs | Impractical for graph that is too large |
| Stop criterion | Need not to be set up | Need to be setup (number of clusters) |
| Change in cluster | Clusters does not change significantly after 40-50 iterations even if there is no converge in a network with approximately 10000 nodes | Number of clusters depends on number of stop criterion |

Figure 7 shows the result when applying Chinese Whispers and Girvan Newman algorithm respectively on the basic graph without any layout algorithm pre-applied to it. The difference in the color of the nodes indicates the different clusters each node belongs to. Nodes of same color belong to the same cluster. Both output display clusters that are very scattered and at the same time very hard to be differentiated from one another. From the result, it would be inappropriate if only clustering algorithm is applied to the graph.

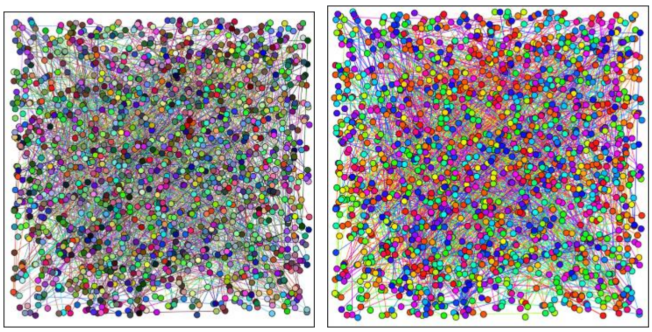
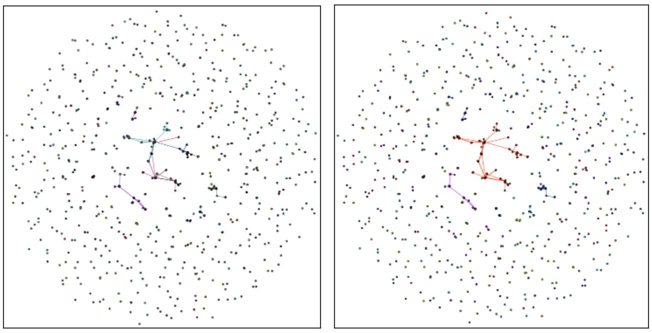
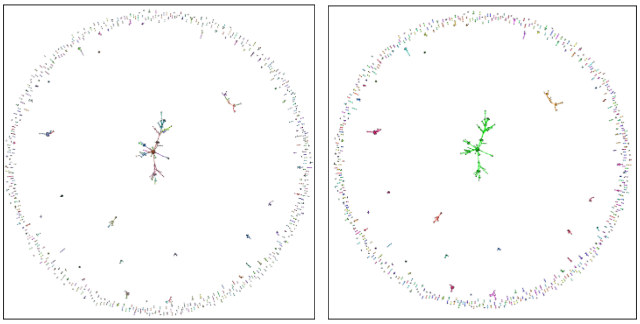


Figure 7: Clustering of CW (left) and GN(right) on the graph with no layout algorithm applied

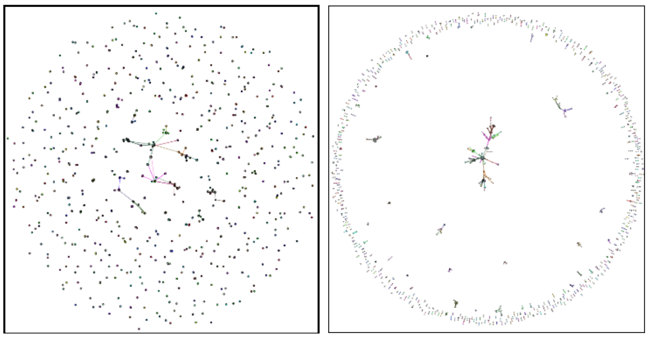
Therefore, in order to find a better visualization, we have run the combination of the two layout algorithms with the two clustering algorithms. The results of the four types of possible combinations are shown in Figure 8.



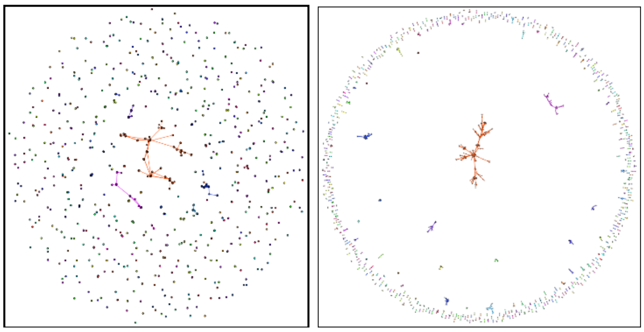
(a) Result of OO followed by CW(left) and OO followed by GN(right)



(b) Result of YH followed by CW(left) and YH followed by GN(right)



(c) Result of CW followed by OO(left) and CW followed by YH(right)



(d) Result of GN followed by OO(left) and GN followed by YH(right)

Figure 8: Results of different combinations between OO, YH, CW and GN

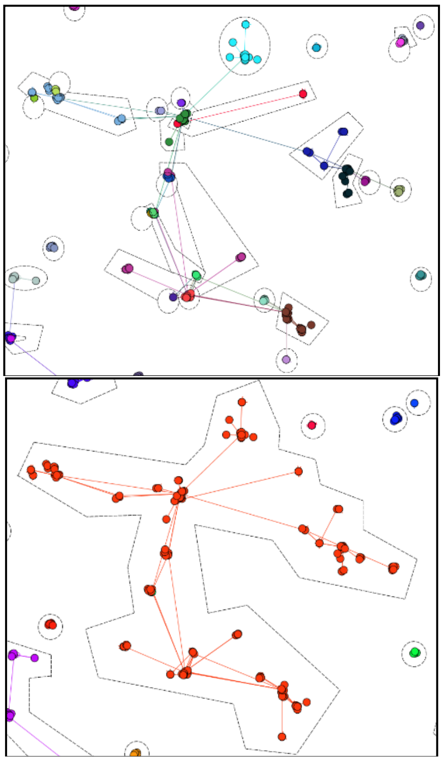
From the results of the combined algorithms, it is observed that the clustering made by Chinese Whispers algorithm changes every time applied on graphs. This is because Chinese Whispers algorithm assigned a node to a cluster if its edges are mostly connected to nodes from that cluster. When Chinese Whispers algorithm encounters a situation where one node is equally connected to two different nodes from two different clusters, the node is assigned to any of the two clusters randomly. Girvan Newmann, gives a more stable clustering result based on the threshold decided before running the algorithm.

Any sequence of algorithm application, which is whether applying layout algorithm first followed by clustering algorithm, or vice versa, does not affect the shape of the graph or the position of nodes in it. The only difference observed when the different arrangements of algorithms are applied is the difference in the clusters created by Chinese Whisper, as stated previously. The difference in the clustering is not caused by the change in the sequence, but caused by the method of the algorithm itself. Since the results are the same for any sequence, for the next discussion, we will only be explaining the result of applying clustering algorithms after applying layout algorithms sequence.

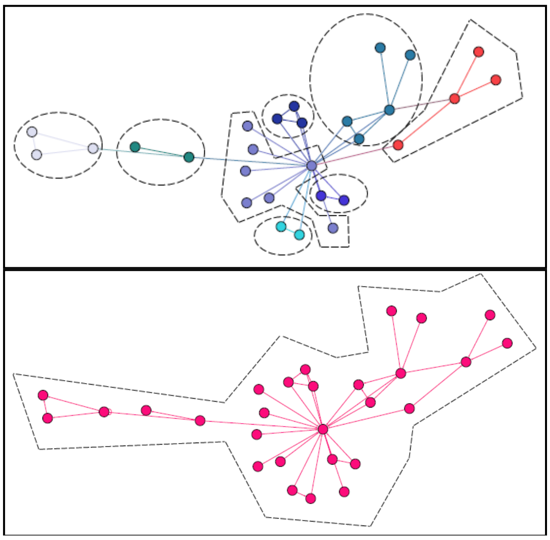
Table 4 shows the evaluation results of applying the two clustering algorithms to each layout graph. Figure 9 provides the clearer view on the differences in clustering made by each algorithm on each type of layout graph.

Table 4
Evaluation Results of Clustering Algorithms on Layout Graphs

| Layout Algorithm | Clustering Algorithm | Clustering time | Number of Clusters | Number of Nodes in Biggest Cluster | Number of Nodes in Smallest Cluster |
|---|---|---|---|---|---|
| OO | CW | 00:00.48 | 579 | 51 | 2 |
| | GN | 00:05.39 | 541 | 184 | 1 |
| *YH* | CW | 00:00.42 | 584 | 27 | 2 |
| | GN | 00:01.43 | 541 | 184 | 1 |



(a) Clustering of CW(top) and GN(bottom) on OO layout graph



(b) Clustering of CW(top) and GN(bottom) on YFH layout graph

Figure 9: Comparison of clustering made by CW and GN

## V. DISCUSSION

This section discusses the effects of running clustering algorithms on graphs generated by layout algorithms. To ease the reading, the combinations are abbreviated as follows: OO-CW for OpenORD and Chinese Whispers algorithms, OO-GN for OpenORD and Girvan-Newman algorithms, YH-CW for Yifan Hu's and Chinese Whispers algorithms, and YH-GN for Yifan Hu's and Girvan-Newman algorithms.

### *A. Running Time vs. Visual Effect*

If we combine the time taken by each combination of algorithms to run, then the combination of OO-CW gives the shortest running time (6.44 sec = 5.96 + 0.48). The combination YH-GN has slightly the longest running time (although not too distant) (12.38 sec = 10.95 + 1.43). The other combinations are nip and tuck: 11.35 sec for OO-GN and 11.37 sec for YH-CW. Even though YH doubles its running time compared to OO, the difference is only within seconds and is not really significant. Extra credit is given to YH as it has the advantage of generating a readable graph without cluttered nodes and with very few edge crossings. The graph generated by OO is hard to be read due to many overlapped edges and nodes. To summarize, if one needs a graph of large NE relations in a fast way and give ease presentation on the eye to its viewer, then YH is more preferred.

### *B. Cluster Attributes vs. Visual Effect*

GN, regardless of the layout algorithm used, does not vary in its clustering decision. The number of cluster depends on the threshold value that has been set when running the algorithm. For both layout algorithms, GN found 541 clusters with the largest cluster containing 184 nodes. This regularity makes GN attractive if one wants a stable type of clustering. It can be seen in Figure 9(a) and 9(b) (bottom figures) that if the nodes are not related at all, then GN separates them into different clusters. In this case, it is said that GN gives a global cluster of the graph. However, if the nodes are related, even by just one edge, the algorithm put them into the same cluster. The benefit from this behavior would be no relation is lost.

GN was created to find community structure. In the context of NE relations, communities are a group of NEs which describe about similar thing. For example, when disease NEs, a number of treatment NEs and a number of side effect NEs are grouped in the same cluster by GN, we say that they are describing about similar things. Figure 10 shows an example of the said situation. All NEs in the given figure are grouped together in one cluster. In the cluster, HIV infection is the disease where other NEs that are clustered together with HIV infection are either its prevention or its treatment.

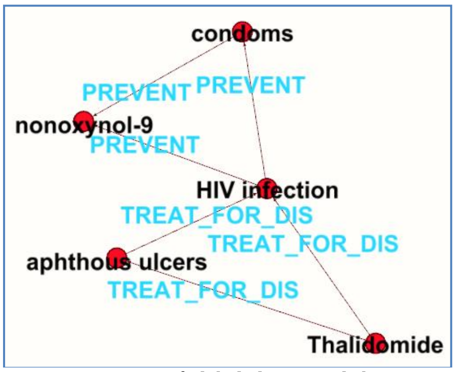



Figure 10: One of global cluster made by GN

There are differences in the number of clusters created by CW on each of the two layout graphs. This difference in cluster numbers are not caused by the choice of the layout algorithms since CW does its clustering process based on random decision. The number of cluster made will change again when CW is applied on the same graph for the second time.

In Figure 9(a) and 9(b) (left figures), the nodes are clustered to the group where it has most relations too. When a node has the same number of edges connected to two clusters, the node is randomly assigned to either one of the cluster. CW creates random clusters on the graph every time applied depending on the position of the node it started to run first.

The dataset containing NE relations is usually large. CW is however suitable for large graph and is not suitable for small graphs as the "results can be inconclusive due to its non-deterministic nature." [6]. However, in our case, the result is still non-deterministic because the size of the NEs used in this work is still considered small, which is 1656 only.

CW however, compared to GN, gives out a larger number of clusters for both OO and YH layout; 579 and 584 respectively. CW exhibits a small number of nodes in its largest cluster: 27 when it is combined with YH layout algorithm and 51 when it is combined with OO layout algorithm. As highlighted by its author, CW outputs a large number of clusters and the majority of them contain a small number of nodes [6]. This statement is then confirmed by this study.

As a summary, Table 5 and Table 6 show the summarized result of the 4 algorithms, according to their own category. The grey columns indicate the best situation.

Table 5
Evaluation of Graph Layout Algorithms

| Layout Algorithm | OO | YH |
|---|---|---|
| Time Performance | Fast | Intermediate |
| Edge Crossings | Many | None |
| Cluttered Nodes | Many | None |

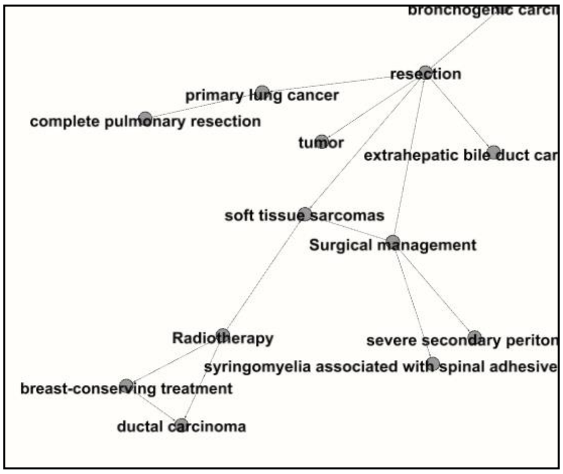
Table 6
Evaluation of Graph Clustering Algorithms

| Clustering Algorithm | CW | GN |
|---|---|---|
| Time Performance | Fast | Intermediate |
| Characteristic of Clustering Generated | Random cluster | Global cluster |
| Loss of Relations | ✓ | ✗ |
| Same Result When Applied to Same Graph with Different Layout | ✗ | ✓ |

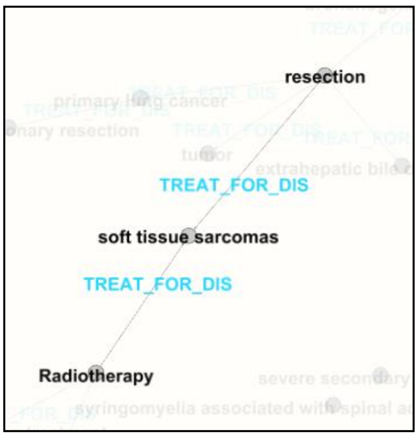
Based on the evaluation, it is found that YH layout algorithm provides the best three criteria out of four criteria taken into consideration. It is also found that GN clustering algorithm provides the best three criteria out of five criteria taken into consideration.

### *C. Edge Overlapping vs. Relations Display*

One advantage of displaying NEs relations in the form of graph is that hidden relations which are not easily captured by reading the text can be easily captured when looking at the nodes of the graph. However, the visibility of these hidden relations is very much affected by the overlapping of the edges and the nodes. This situation is hardly recognised when looking at the graph generated by OO algorithm as there are too many overlapped edges and cluttered nodes. For the graph generated by YH algorithm, the hidden relations are obvious and can be easily recognised. An example of the said situation is as depicted in Figure 11(a) and (b).




(a) A subgraph of Hu's layout graph




(b) Highlighted relations of soft tissue sarcomas

Figure 11: One of global cluster made by GN

Figure 11(a) shows a subgraph of the graph generated by YH algorithm. Figure 11(b) then highlighted two relations displayed in the graph where soft tissue sarcomas are related to radiotherapy and resection in a "treatment_for_disease" relation. These edges indicated that soft tissue sarcoma can be treated by both radiotherapy and resection. These relations are examples of hidden relations that cannot be captured immediately when human read through the text as the two relations are stated in two different sentences of different parts of the text. A thorough reading of the text would be required to come up with such a summary of the text. Figure 12 shows the two different sentences that stated the two different relations.

```
1. <TREAT> Surgical management </TREAT> of <DIS> soft tissue sarcomas </DIS> :
   principles of <TREAT> resection </TREAT> and reconstructive plastic
   procedures ||TREAT_FOR_DIS

2. <TREAT> Radiotherapy </TREAT> of <DIS> soft tissue sarcomas </DIS> ||TREAT_FOR_DIS
```

Figure 12: Original text of the data

## VI. CONCLUSION

As a conclusion, from the overall visual observation on the graphs generated by the combination of two algorithms, graph with the layout of Hu's algorithm are able to display the clusters created in the graph more clearly compared to the other layout, as it displays the nodes and edges in a less clustered and overlapping way. The choice of the algorithms actually depends on what the viewer wants to view. If the viewer wants to view a global group of nodes in the graph, then a graph generated by the combination in sequence of Hu's algorithm and Girvan-Newmann clustering algorithm is the appropriate one.

The number of NEs used in this research is hoped to be increased in the future so that the evaluation results would be more applicable to the big real world data. More graph algorithms will be compared to ease the decision of choosing graph algorithms to visualize NE relations in the future.

## ACKNOWLEDGMENT

The authors thank the Ministry of Higher Education (MoHE), Malaysia, for the financial support of this research under the grant RACE/b(5)/1097/2013(05).

## REFERENCES

[1] "Visualizing Historical Networks," (Retrieved from http://www.fas.harvard.edu/~histecon/visualizing/).
[2] R. Mihalcea, and D. Radev, *Graph-Based Natural Language Processing and Information Retrieval*. Cambridge, UK: Cambridge University Press, 2011.
[3] M. Marrero, J. Urbano, S. Sánchez-Cuadrado, J. Morato, and J. M. Gómez-Berbís, "Named entity recognition: fallacies, challenges and opportunities. Computer Standards & Interfaces," 2013, pp. 482-489.
[4] S. Martin, W. M. Brown, R. Klavans, and K. Boyack, "OpenOrd: An Open-Source Toolbox for Large Graph Layout," in *Proceedings SPIE7868, Visualization and Data Analysis*, USA, 2011.
[5] Y. F. Hu, "Efficient and high quality force-directed graph drawing," *The Mathematica Journal*, vol. 53, 2005, pp. 1689-1699.
[6] C. Biemann, "Chinese Whispers: An efficient graph-clustering algorithm and its application to natural language processing problems," in *Proc. of TextGraphs: The Second Workshop on Graph-Based Methods for Natural Language Processing*, New York City, 2006, pp. 73-80.
[7] M. Girvan and M. E. J. Newman. "Community structure in social and biological networks," in *Proceedings of the National Academy of Sciences*, 2002, pp. 7821–7826.

[8] U. Hinrichs, B. Alex, J. Clifford, A. Watson, A. Quigley, E. Klein, and C. M. Coates, "Trading Consequences: A Case Study of Combining Text Mining and Visualization to Facilitate Document Exploration," *Digital Scholarship in the Humanities*, vol. 30, 2015, pp. 50-75.

[9] D. Y. Tan, B. Ranaivo-Malançon, and N. Kulathuramaiyer, "Wiki SaGa: An Interactive Timeline to Visualize Historical Documents," in *Information Science and Applications*, 2015, pp. 705-712.

[10] M. Grobelnik and D. Mladenic, "Visualization of news articles," *Informatica*, vol. 28, 2004, pp. 375-380.

[11] T. Osaki, S. Itsubo, F. Kimura, T. Tezuka, and A. Maeda, "Visualization of Relationships among Historical Persons Using Locational Information," in *Web and Wireless Geographical Information Systems*, 2011, pp. 230-239.

[12] M. Itoh and M. Akaishi, "Visualization for Changes in Relationships between Historical Figures in Chronicles," in *Proceedings of the International Conference on Information Visualisation,*USA, 2012, pp. 283-290.

[13] M. Bastian, S. Heymann and M. Jacomy, "Gephi : An Open Source Software for Explorating and Manipulating Networks," in *Proceedings of International AAAI Conference on Web and Social Media*, USA, 2009, pp. 361–362.

[14] B. S. Raper, "Graphing the history of philosophy," (Retrieved from http://www.coppelia.io/2012/06/graphing-the-history-of-philosophy).

[15] C. Harrison, "Visualization Projects," (Retrieved from http://www.chrisharrison.net/index.php/Visualizations/Welcome).

[16] G. Michailidis, "Data Visualization Through Their Graph Representations," in *Handbook of Data Visualization*, Springer Berlin Heidelberg, 2008, pp. 103-120.

[17] W. Huang, P. Eades, S. H. Hong and C. C. Lin, "Improving multiple aesthetics produces better graph drawings," *Journal of Visual Languages and Computing*, vol. 24, no. 4, 2013, pp. 262-272.

[18] W. Huang, M. L. Huang and C. C. Lin, "Evaluating overall quality of graph visualizations based on aesthetics aggregation," *Information Sciences*, vol. 330, 2016, pp. 444-454.

[19] M. Jacomy, S. Heymann, T. Venturini and M. Bastian, "ForceAtlas2 , A Graph Layout Algorithm for Handy Network Visualization," 2014.

[20] S. Hachul and M. Jünger, "An experimental comparison of fast algorithms for drawing general large graphs," *Graph Drawing*, 2006, pp. 235–250.

[21] F. Zaidi, D. Archambault and G. Melançon, "Evaluating the Quality of Clustering Algorithms Using Cluster Path Lengths," in *ICDM 2010: Advances in Data Mining. Applications and Theoretical Aspects*, Germany, 2010, pp. 42-56.

[22] "The BioText Project," (Retrieved from http://biotext.berkeley.edu/data.html)

[23] B. Rosario and M. A. Hearst, "Classifying Semantic Relations in Bioscience Text," in *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004)*, Spain, 2004.

[24] S. G. Kobourov, "Force-Directed Drawing Algorithms," in *Handbook of Graph Drawing and Visualization*, CRC Press, 2013, pp. 383-408.