Exploration and application of complex graph properties

by

Zi Yuan Chen

Abstract

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Acknowledgements

I want to thank...

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

The application of mathematics has played a key role in the development of technological advancements and breakthroughs in science. Throughout history, mathematics has provided us with an increasing number of various sub-branches such as discrete mathematics, applied mathematics, cartesian geometry, algebra, calculus and many more. All of which can be applied to the real world, whether it is for construction, physics, or simple day-to-day activities.

A key branch under discrete mathematics is graph theory where models can be developed to represent relationships between different objects. Graphs contain a range of uses both in the mathematical world and the real world. They can be used to visually represent large sets of numerical data providing a way to deduce different properties and the correlations they have within the data. Examples include identifying clustering of certain areas, the connectivity between vertices or edges and many others. Graphs are also widely known as networks with some examples such as a friendship network, business networks and even food chain levels. Graph theory is the area that I will explore along with a method to visualise them through the help of using python. As well as analysing specific properties that may influence the visualisation of the graphs and thus the outcome of any key relationships between the vertices.

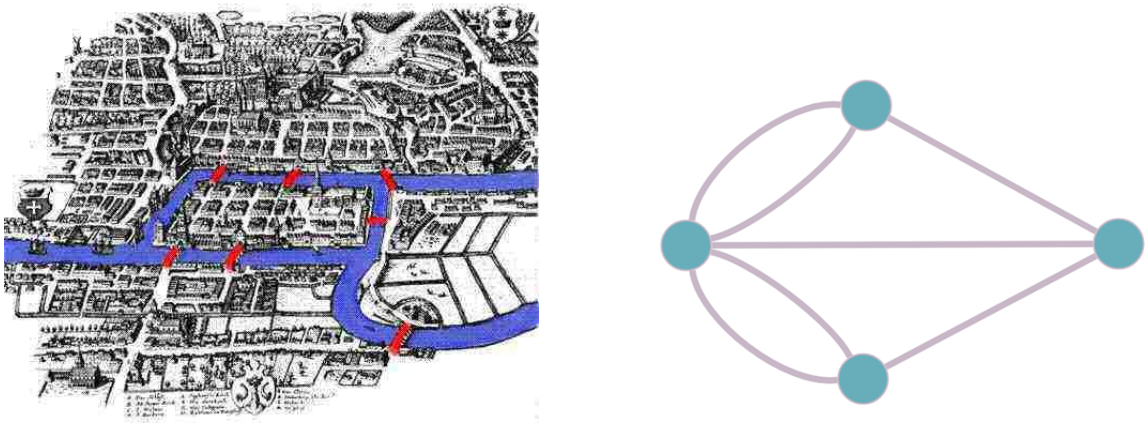
1.2 History

Initially, graph theory was introduced in 1735 as a form of a solution to the seven bridges of Königsberg problem which was solved by Leonard Euler [1]. This famous problem involved an island within Königsberg that had a river, Pregel, surrounding the island via a fork. There were seven bridges that crossed the river Pregel from the island to connect to other major landmasses of Königsberg, Prussia. The island had four bridges, two north, two south connecting the mainland and the island. Also

another bridge connecting to a neighbouring island and this neighbouring island itself had two bridges. Consequently, giving a total number of seven bridges as stated by the seven bridges of Königsberg problem, see Figure 1.1a sourced from MacTutor Archive [2].

Due to the location in which the island was situated, the problem was to determine whether a route exists that manoeuvres through all the bridges exactly once and must return to the starting location. Leonard Euler proceeded to analyse the problem by evaluating only the key areas, this was the land masses and the bridges. Other information such as the sizes of the island, bridge type or length were irrelevant. Consequently, the problem could be portrayed by utilising dots and lines [3] to give a simplistic view (see Figure 1.1b). Once developed, the dots are known as vertices which represent the key interests and the edges that are incident to these vertices represents the connections/relationships between them.

By removing the irrelevant information, Euler was able to simplify and visualise the problem. In doing so, Euler discovered the fact that for a solution to exist, each vertex must have an even number of edges incident to them (even degree). This is because you require one edge for entering and another for exiting. Otherwise not all edges will be included in the final path that is based on the conditions of the bridge problem. All vertices in the Königsberg problem have odd degree, meaning that a *Eulerian path* (a path that traverses all edges exactly once) does not exist for this problem. Since a Eulerian path does not exist then a *Eulerian circuit* (a Eulerian path that returns to the starting vertex) cannot exist either. Therefore, Euler proved that there were no solutions to the seven bridges of Königsberg problem. This proof is regarded as the first proof in relation to graphs and led to the birth of graph theory.



(a) (b)

Figure 1.1: The (a) original seven bridges of Königsberg and its (b) graph representation that only focusses on the key details and disregards the irrelevant information. Achieved by using vertices and edges.

Eulerian paths and cycles were a major milestone in the study of graph theory. Based on their unique definitions, they have been a useful tool in graph theory and other scientific fields. An example is the study of DNA fragment assembly through the identification of Eulerian paths. Through these Eulerian paths, repeated DNA fragments can be identified and "glued" together to construct a *de Brujin* graph. A de Brujin graph is a directed graph that represents overlaps between sequences or symbols. In DNA sequence assembly, traversing a de Brujin graph with overlapping regions can give the full genome sequence. This is studied extensively seen in the paper by Pevzner, Tang and Waterman [4].

After Eulerian paths and cycles [5] were introduced, the next famous puzzle in relation to graph theory was invented in 1857 and was known as the Icosian Game [6] by William Rowan Hamilton. The objective of the puzzle was to find a cycle that visits all vertices exactly once and returns to the starting vertex. This type of cycle is later defined as a *Hamilton cycle* [7] along with the definition of a *Hamilton path* which does not have the requirement to return to the starting vertex.

In the mathematical world, Leonard Euler is known for *Euler's identity* in complex mathematics which states that for a real number x, $e^{ix} = \cos(x) + i\sin(x)$. This has been crucial in many subject areas such as physics and engineering. Additionally in 1850, Euler uncovered another formula to be known as *Euler's polyhedra formula* which states that $F + V - E = 2$ where F denotes the number of faces, V as the number of vertices and E as the number of edges for a graph model. As polyhedrons can be depicted as graphs, algebraic topology benefited from Euler's polyhedron formula where more complex surfaces could be studied such as the surface of a torus. Based upon this formula, the *Euler characteristic* was formalised to describe the topological characteristic of various complex surfaces with its formula as $F + V - E = 2 - 2g$ where g is denotes the number of "holes" the surface has (formally known as the *genus*).

Furthermore, graph theory has assisted in problems such as the four-colour map problem which was introduced in the 1850s. The problem asks if all the countries in the world can be coloured with only the use of four colours such that no two adjacent countries were coloured with the same colour. In which the solution was not found until 1972 by Kenneth Appel and Wolfgang Haken [8] through the assistance of a computer. An alternative example of a four colour problem is the colouring of the counties in the UK with only four colours where the solution for this is shown in Figure 1.2 sourced from Robin Wilson [9].

The Chinese postman problem is another such graph theory problem where you must find the shortest path that uses all the edges in the graph at least once. A similar alternative version is called the travelling salesman problem. In which you identify a shortest path that uses all edges exactly once in the graph and must end at the starting vertex. These such problems are used in Linear programming to find optimal solution in routing or pathing between locations. Variants of the CPP (Chinese postman problem) includes undirected Chinese postman problem (UCPP) and contains different restrictions depending on the subset of edges used (see the

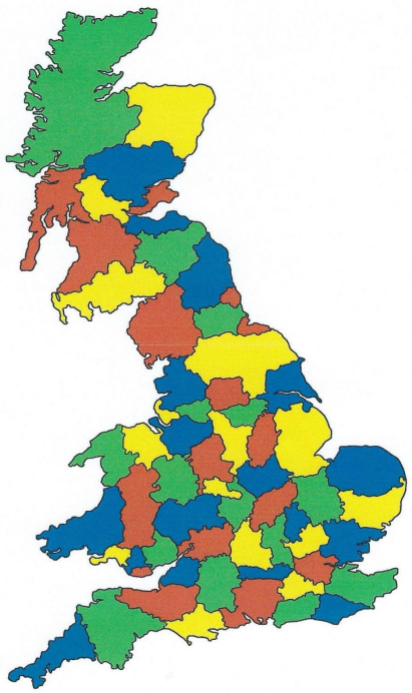


Figure 1.2: The solution for the four colour map problem with regards to the counties in the UK.

paper [10]). Also there exists the Chinese postman problem with load-dependent costs (CPP-LC [11]) that includes weights on the edges so that an optimal route can be generated. This can widely be used and applied to problems such as the best route in order to lower a vehicle's CO_2 emission. These problems can be applied to modern day scenarios which means that they are still valuable to companies now.

Therefore, studies within graph theory have been researched extensively since 1735 with many beneficial factors brought into the real world. This is possible due to the versatile nature that data structures in the real world can be represented as mathematical structures through graphs or networks. Examples of what they can represent ranges from simple relationships between people to the complex structure of the brain by studying the brain's anatomic structure and assigning vertices according to the sections of the brains and edges as the links between them. The links are typically representations of the neurons in the brain. Further details of brain mapping into graphs can be read by the paper [12]. Therefore, by using graph models, various patterns and correlations can then be derived to generate graph properties. These properties can be studied to develop useful information and possible improvements to the whole graph.

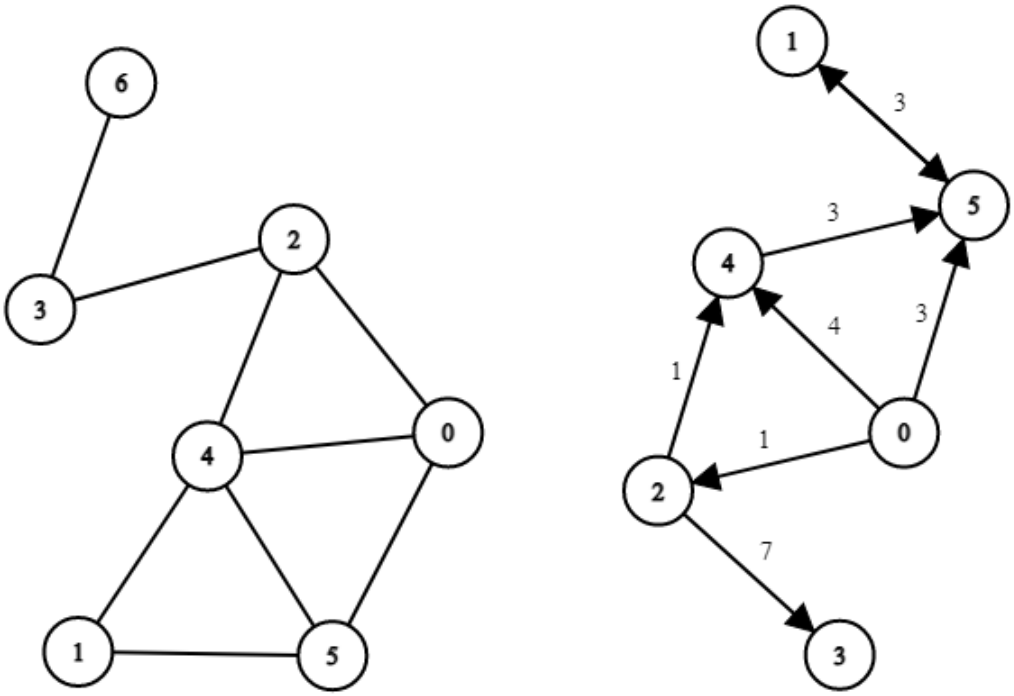
1.3 Basic notation and terminology

Graphs or networks are mathematical constructs that are formed by a collection of vertices and edges. Vertices V represents individual objects such as land masses, companies, houses, people, etc. Edges E represents the connections between the vertices such as their relationship, flow of water, supply chain, etc. Sets $V(G)$ or V and $E(G)$ or E forms the graph G and can be written as $G = (V, E)$ with E being a subset of $V \times V$. So an edge $e \in E$ can be written as v_1v_2 if e connects v_1 and v_2 where $v_1, v_2 \in V$. There exist variations among graphs as they can be directed or undirected, the edges may carry weights and they may contain self-loops. Figures 1.3a and 1.3b show simple graphs, one of which is undirected, and another is directed. A graph $H = (V', E')$ is a *subgraph* of $G = (V, E)$ if $V' \subset V$ and $E' \subset E$.

Order is the mathematical term that represents the number of the vertices in the vertex set $V(G)$. *Size* is number of edges or vertex pairs in the edge set $E(G)$. The *degree* of a vertex, denoted by $deg(v)$ is the number of edges that are connected (otherwise known as *incident*) to the vertex, discussed previously in Euler's solution to the seven bridges of Königsberg problem. Additionally, $\delta(G)$ and $\Delta(G)$ represents the *minimum* and *maximum degree* in G respectively. G is a *regular* graph if $\delta(G) = \Delta(G)$. Vertices v_1 and v_2 are *adjacent* if there exists an edge $e \in E$ that connects them. Vertex v_2 is also known as a *neighbour* to v_1 and is part of the set $N(v_1)$ which denotes the neighbours of v_1.

There are multiple ways to traverse a graph [13]. A *walk* $w = v_1v_2v_3v_4v_5...v_n$ is a sequence of vertices such that $E(w) = (v_1v_2, ..., v_{n-1}v_n)$ where vertices and edges can be revisited. They can be *open* or *closed* depending on whether the final vertex is equal to the starting vertex, open if equal and closed if not. A *trail* is an open walk where no edges are revisited but vertices may be revisited. When all the edges are traversed exactly once, it is known as a *Eulerian trail* (mentioned previously as a Eulerian path) and the graph is called *semi-eulerian* or *traversable*. Similarly for a trail that is closed (returns to the start), then it is known as a *circuit* and if all edges are used then it is called a *Eulerian circuit* (or Eulerian cycle) and the graph is defined to be *Eulerian*. A *path* is a trail but with no vertex repetitions and if the size of the path is equal to the size of the graph, then it is a *Hamilton path*. Finally, a *cycle* is a path that ends at the starting vertex and if all vertices are visited exactly once, then it is known as a *Hamilton cycle* meaning that the graph is *Hamiltonian*.

When considering network with flows [14], vertices can be known as nodes and may have capacities that limit the overall flow through the network shown in Figure 1.4. These networks are especially used when considering plumbing, water pipes and even evacuation routes in a building. Within a room there is a capacity that is represented by the node's capacity and the weights of the edges can demonstrate the rate of flow along with its maximal flow. In other words, when people are evacuating a building, the corridors have a limit to the amount of people that may



(a) An undirected graph with 7 vertices, 9 edges and an average vertex degree of 18/7.

(b) A directed graph with 7 vertices and 7 directed edges that contain weighted edges.

Figure 1.3: The two simple types of graphs within graph theory that are the basic building blocks for more complex structures and theorems.

pass through. Networks can be used to model social network processes through the study of small corporate groups to generate a communications network. This network representation will have a flow of sentiments based on social network theory [15] that is constrained in three ways. Firstly, by any existing direct relationships within the group, that will be denoted by vertices. Secondly, the frequency of communication of the relationships defined in the first point. Lastly, the breadth of the existing relationships in the network. Thus, by using graphs and networks, social behaviours within groups or companies can be studied giving ways to more phycological information represented by numerical data.

Additionally, graphs can be represented by the use of matrices to enable the use of matrix calculations on the datasets. These matrices are known as *adjacency matrices* [16] and this matrix contains the number of edges incident to each vertex. The connections of the vertices are based on the location of this value as the rows and columns represent the vertices. A *weighted adjacency matrix* will instead hold the weights of each edge in the matrix. An $n \times n$ adjacency matrix $A = (a_{ij})$ for $i, j = 1, ..., n$ is defined by $a_{ij} = 1$ if there exists an edge from vertex i to vertex j. A matrix is always symmetric when considering an undirected simple graph as an

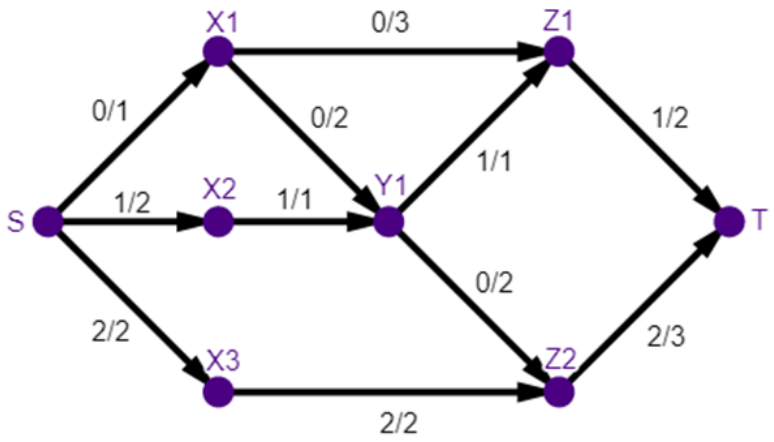
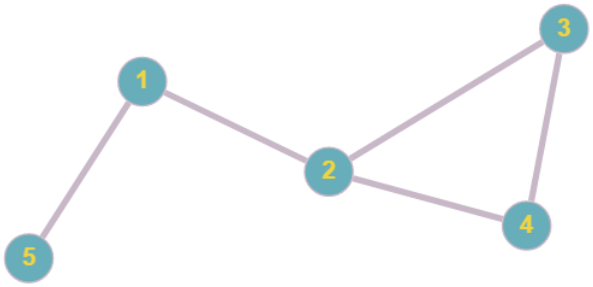


Figure 1.4: A simple network flow with a source node and a sink node, the current flow is 3 and the edge capacities are shown.

edge will contribute to both sides of the matrix. Examples of an adjacency matrix along with its graph representation is shown in Figure 1.5.



$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Figure 1.5: A simple graph along with its adjacency matrix.

1.4 Linguistic and Lexical Analysis

Both the linguistic and lexical field can use the implementation of graphs to further generate relations between words or characters based on a given text [17]. Vertices would represent each word or character and the directed edges would be the connections in relation to their text. A simple word graph example is seen in Figure 1.6 created from the poem "Hey Diddle Diddle" by Walter Crane [18]. Instead of directed graphs, trees can be used as an alternative to display the same text. However for clarity, we only look at directed graph representation.

Correlations can be identified through the study of word graphs and they can exhibit unique characteristics such as small-world characteristic. The small-world

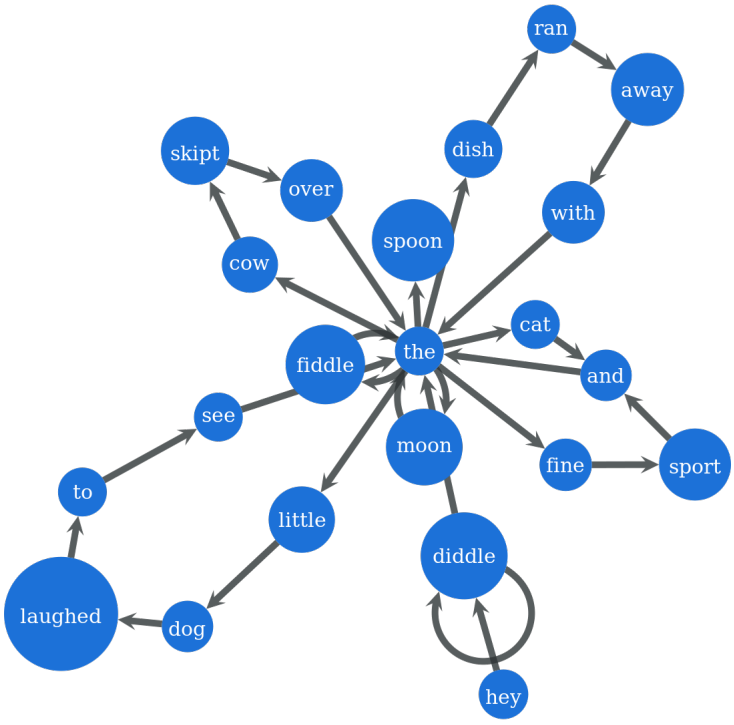


Figure 1.6: Directed word graph for the poem "Hey Diddle Diddle" by Walter Crane.

characteristic is based on the combination of high clustering and short path lengths where it is observed that a high degree of local clustering also has efficient global connectivity (short path lengths to other vertices). Clustering will be explored in detail in the next chapter. This small-world characteristic is often described as the "six degrees of separation" and more can be read in the paper by Watts and Strogatz [19].

In any natural language, words are essential building blocks to formulate expressions and meaning when conversing or writing. They can become complex because words may have various meanings and tenses such as the past, present, and future tense in the English language. Linguistic analysis is the study of natural language texts such as English, its grammar and the structure. Lexical analysis is similar but instead of studying words, it is the study of tokens or characters. For example, lexical analysis can be on the variants of the same word given their tense [20]. Whereas linguistic analysis determines the importance of words in a text and their relative positioning in its structure. Additionally, their syntactic and semantic information [21]. Essentially Lexical analysis is a branch of linguistic analysis. The goal of my research is the study of languages with various graph representations, to study properties of relationships between words in a given text. Therefore we are mainly focussed on linguistic analysis rather than lexical analysis.

The focus for future chapters will be on directed graphs and the data that can be extracted from them. Chapter 2 will describe and outline specific complex graph properties that will be applied to datasets later. This means that by analysing the graphs, numerical values can be generated to represent certain factors of the graph. These factors can then be applied to the graph to rearrange its shape so that further correlation can be identified between all the vertices and edges. We will explore linguistic analysis and study datasets generated based on different languages. But to achieve this, we will first discuss the graph properties that could assist in identifying correlations before the language datasets.

Chapter 2

Graph Properties

There exist many different graph properties that can be applied depending on the specific graph type. Hence, we will focus on properties that can be applied to connected graphs to provide a collection of values targeted for the same layout structure and giving a wider selection of these such properties. Our aim is to apply these properties onto linguistic graphs (also known as word graphs) to enable the analysis of languages. To accomplish this, each property must be able to calculate their values in a connected graph that can be directed. So, a detailed study of each property will be given and the graphs that they can be applied to.

2.1 Trophic Levels and Coherence

We begin with the graph property of *trophic coherence* [22]. In terms of ecology, this property determines the stability of the graph, i.e., the ability that an ecosystem maintains its structure and functions. For example, a food chain [23] can be represented in levels, higher levels are labelled as predators and lower levels as the prey. Trophic coherence measures whether this structure has clear partitions and uses *trophic levels* to determine their positions within this food chain. Trophic levels are calculated for individual vertices within the graph. The idea of trophic levels are taken from ecology and can be applied to graphs to generate a height-based format for the vertices of a graph. Thus, for a connected directed graph $G = (V, E)$, we have V which is the set of all vertices in G and E which is the set of all the edges within G. The graph can be represented with an adjacency matrix A where a_{ij} denotes the elements within the matrix. The standard trophic level definition on vertices uses the in degree and the out degree of the vertex v_i given by Equation 2.1.

$$k_i^{\text{in}} = \sum_j a_{ij}, \qquad k_i^{\text{out}} = \sum_j a_{ji} \tag{2.1}$$

The standard trophic level formula for vertices $v_i \in V$ is defined as:

$$t_i = 1 + \frac{1}{k_i^{\text{in}}} \sum_j a_{ij} t_j \tag{2.2}$$

By ecological convention, $t_i = 1$ if the vertex v_i is *basal*. A vertex is known to be basal if it has no edges directed to it, i.e., $k_i^{\text{in}} = 0$. The trophic level equation can also simply be written in matrix form by Equation 2.3 with $\mathbf{z}$ defined as $z_i = \max(k_i^{\text{in}}, 1)$ and $\Delta = \text{diag}(\mathbf{z}) - A$.

$$\Delta \mathbf{t} = \mathbf{z} \tag{2.3}$$

All vertices will receive a trophic level if and only if the Laplacian matrix Δ is invertible as the row sum of elements in Δ equals 0 for non basal vertices. If there are no basal vertices in the graph then Δ will be equal to the zero matrix and thus be singular, i.e., no inverse. This is discussed by Johnson [24] in the investigation of stability and such dynamical features within graphs. So, for standard trophic levels equation to be applicable for the entire graph, there must exist at least one basal vertex meaning that this is a limitation to the definition of trophic levels. In the case of linguistic analysis, a basal vertex exists because sentences will always have a word that is the start, i.e., a vertex without any in edges. However, in a larger dataset, the starts of sentences may be used in other areas meaning that they are no longer basal. Consequently, we study the improved equation for trophic levels and the possibility of weight inclusion.

2.1.1 Inclusion of weights and equation improvement

In a weighted graph $G = (V, E)$, edges carry a weight between a vertex v_i to a vertex v_j where $i, j = 1, 2, \ldots, |V|$. Weighted matrix W are used as a representation of the entire graph along with incorporating the direction of each edge and self-loops if they exist. The elements of the weighted matrix are w_{ij}. If the graph is not weighted then the edge is valued as 1 if the edge exists and 0 otherwise, i.e., the adjacency matrix of G. The total weight (also known as the *strength*) for each vertex is defined by the weights into a vertex v_i and the weights out of vertex v_i, which is shown by s_{v_i} in Equation 2.4. This is essentially the same as the in and out degrees of Equation 2.1 mentioned previously but instead of 1s and 0s, the weighted values are considered. The imbalance of the vertex v_i is defined by the weights in of a vertex minus the weights out of the vertex shown by i_{v_i} in Equation 2.5.

$$s_{v_i} = \sum (w_{v_i v_j}) + \sum (w_{v_j v_i}) \tag{2.4}$$

$$i_{v_i} = \sum (w_{v_i v_j}) - \sum (w_{v_j v_i}) \tag{2.5}$$

Vectors $\mathbf{s}$ and $\mathbf{i}$ hold all the values of the strength and imbalance for all vertices respectively. We let $\mathbf{h}$ be a vector, then the graph Laplacian operator in matrix form is defined by Equation 2.6.

$$\Delta = \text{diag}(u) - W - W^T \tag{2.6}$$

Therefore to get the trophic levels for each vertex with consideration for the additional weights, we solve the system of equations for vector $\mathbf{h}$ as shown by Equation 2.7.

$$\Delta \mathbf{h} = \mathbf{v} \tag{2.7}$$

The values within this vector h correspond to the trophic levels for the relative vertices. The values are used to illustrate the various trophic levels within a graph to give a hierarchical format visualisation. Trophic levels are not unique solutions because an arbitrary constant can be added to each component of the graph if there are multiple components to generate new levels that would be correct. The benefit to this is that Equation 2.7 can use an arbitrary constant c for a vertex in $\mathbf{h}$ to influence the value of other vertices in a component. If there are multiple components to the graph, then a vertex in each component is needed. A unique solution can be found this way in which the trophic level values can be shifted so that a better graphical display can be generated. For example, having the lowest trophic level to be 0.

Trophic levels can also be used to equate the overall *trophic incoherence* of the graph rather than just looking at the graph on a vertex level. By using the trophic levels from $\mathbf{h}$, the equation for the trophic incoherence is defined by MacKay, Johnson, Sansom [24] to be:

$$F_0 = \frac{\sum_{v_i v_j} w_{v_i v_j} (h_{v_j} - h_{v_i} - 1)^2}{\sum_{v_i v_j} w_{v_i v_j}} \tag{2.8}$$

The possible shifting of the trophic levels does not affect the trophic incoherence meaning that the equation is independent. The incoherence is strictly ranged from 0 to 1. If $F_0 = 0$ then the graph is *maximally coherent* as it would mean that all levels in the graph have a precise equal spacing which means that the graph is perfectly separated into levels. Whereas if $F_0 = 1$ then the graph is *maximally incoherent* and levels are harder to decipher. As F_0 measure the incoherence, by taking $1 - F_0$, this would instead measure the coherence of the graph as they are each other's converses.

In conclusion, trophic levels can be applied to weighted directed graph and any subcategories to achieve a hierarchical view of the graph. This eases the visualisation of many datasets and is used to decipher valuable information that may be of use. Through the combination of other graph properties which are described in this chapter, various combinations of these properties will yield different visualisations. This will prove beneficial in finding key vertices and correlations when analysing various languages. Through using trophic levels in languages, the graph can be formatted to demonstrate the sentence structure visually. Additionally, as shifting their trophic values does not affect the overall coherence, values can be modified so

0 indicates the start of the sentence and larger values indicate positions further up the sentence.

2.2 Clustering Coefficient

The *clustering* of a graph, also known as *transitivity* [25], is a property of a graph that measured the density of triangles within the graph. Triangles are where 3 vertices are connected. Clustering is used to quantify the graph's connectivity strength as it determines the fraction of triangles over the possible triangles that could be formed within the graph. Another perspective is that the coefficient quantifies the probability of a vertex a having an edge to vertex c if ab, $bc \in E(G)$. Thus, the *clustering coefficient* determines how complete the graph is with a value of 1 meaning it is complete and 0 if not. There are two popular introductions of clustering coefficient, the *local clustering*, and the *global clustering*. The global clustering coefficient essentially measures the completeness of the graph by measuring the number of existing triangles divided by the number of possible triangles. The local clustering coefficient measures the clustering coefficient for each vertex rather than the whole graph. The measurement is taken by the number of triangles that has a connection to this vertex over the number of triples centred on this vertex. In other words, the local value demonstrates how close the neighbours of this vertex are to being a complete graph (a *clique*).

2.2.1 Clustering for simple graphs

For simple connected graphs that are unweighted and undirected, we determine the coefficients of the clustering. The global clustering coefficient is defined by equation 2.9 where $\sum T$ denotes the number of triangles (closed triplets) and $\sum \tau$ denotes the number of connected triplets in the graph.

$$C = \frac{3(\text{Number of all triangles})}{\text{Number of all connected triples}} = \frac{\sum T}{\sum \tau} \tag{2.9}$$

An alternative equation which was demonstrated by M.E.J. Newman [26] through the studies of complex networks in terms of social networks. This is where the clustering coefficient determines the likelihood that a friend of your friend is also your friend. So, the alternative equation is written in the form of equation 2.10 where $\sum P_2$ denotes the number of paths with length two within the graph.

$$C = \frac{6(\text{Number of total triangles})}{\text{Number of paths with length 2}} = \frac{\sum T}{\sum P_2} \tag{2.10}$$

By considering the vertices of the graph, the local clustering coefficient can be defined to give such a coefficient to each vertex $v \in V(G)$ and is achieved by the equation 2.11 where i is the index of the vertex. This definition is from paper [26]

and proposed by Watts and Strogats [27] where they analysed small world networks in relation to various real-world systems by the use of clustering coefficients and random graphs to formulate certain similarities. Note that if the degree of a vertex is 1 then the coefficient can be determined as 0, otherwise the equation will lead to 0/0.

$$C_i = \frac{\text{Number of triangles connected to vertex } i}{\text{Number of triples centred on vertex } i} \tag{2.11}$$

Another representation of the global clustering coefficient is to take the averages of all the local coefficients [28]. When the vertices have a degree of 0 or 1 then $C_i = 0$ so global clustering coefficient can also be defined by equation 2.12.

$$C = \frac{1}{n} \sum_i C_i \tag{2.12}$$

Later the clustering coefficients will be used as assistance to model graphs generated from a language dataset. However, to provide more accurate coefficients, variations of the clustering formulas will be defined based on the inclusion of weights and/or directions. In terms of our goal of linguistic analysis, directions are of key importance since the word graphs will be directed. Therefore, the definitions of clustering coefficients must be developed further, starting with a overview of the inclusion of weighted edges.

2.2.2 Clustering for weighted graphs

Now by considering graphs as before but weighted instead, the equations undergo changes. For the instance of weighted graphs, there are multiple different definitions of clustering coefficients, each with slight variation in coefficients. This section will summarise a couple of the different definitions for weighted graphs and further detail can be analysed from Tanguy and Anna Levina on weighted directed clustering [29]. In this paper, four different definitions are reviewed which are the Barrat definition, Onnela's definition, Zhang & Horvath and their own continuous definition for weighted graphs. Zhang & Horvath [30] have used their definition of weighted clustering coefficients to analyse gene co-expression networks to review their functionality. Additionally, by soft or hard thresholding, it enables them to determine relationships between the clustering coefficient and gene networks within biology.

Alternatively, a simple idea to associate the clustering coefficient with regards to the edge weights is to define a value w that represents the value of the triplet. w can be the summation of the triplet, the mean of the triplet or another suitable method depending on the purpose. We take w to be the summation of edges of the triplets. Then Equation 2.13 calculates the weighted clustering coefficient [31] where T denotes the triangles in the graph and τ, the triples.

$$C = \frac{\text{Total of closed } w}{\text{Total of } w} = \frac{\sum_T w}{\sum_\tau w} \tag{2.13}$$

2.2.3 Clustering for directed graphs

We see that weights can be added trivially through Equation 2.13. So now considering the addition of directions as well as the possibility of weighted edges. Directions cause further complexities in the coefficients of clustering due to the various number of different *motifs*. Motifs are various patterns in graphs that are reoccurring and can be used here to describe the nature of the triangles. For instance, there are 16 possible motifs for directed graphs of 3 vertices shown in Figure 2.1. However, if we consider only connected triangles, they can be organised into 4 types of motif groups known as *Cycles*, *Middleman*, *Fan-in* and *Fan-out*. Demonstrated by Figure 2.2 which are used in the study of higher order motifs and synaptic integration by Bojanek, Zhu and Maclean [32]. An interesting result used in this paper is the isomorphisms between the middleman, fan-in, and fan-out motifs.

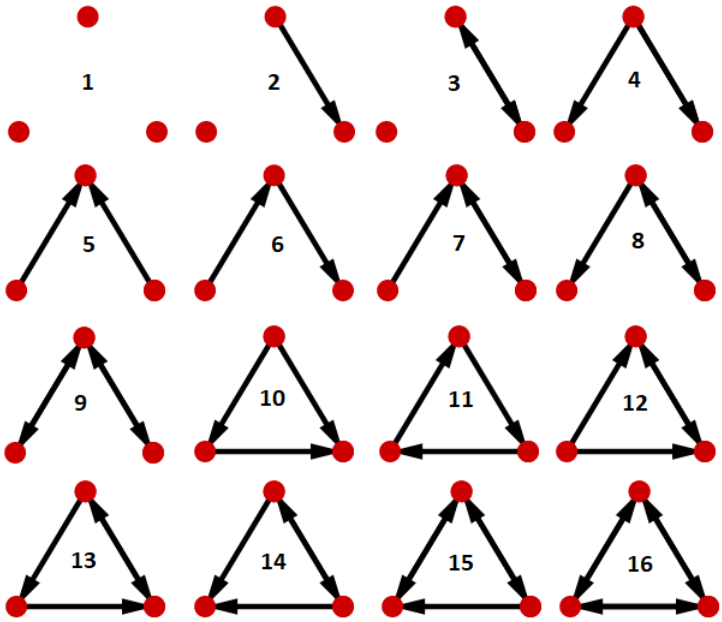


Figure 2.1: All 16 motifs of possible connections between 3 nodes and their directions shown on the edges. Ranges from 0 directed edges to the maximum of 6 directed edges.

Consideration of the directional edges yields better accuracy in the coefficient values. One of the versions mentioned in the paper [29] was Fagio's where he introduces the clustering coefficient to binary directed networks which are equivalent to simple directed connected graphs. Firstly, the equation for the directed version without the consideration of weights is defined by the ratio of all directed triangles centred on a vertex $i \in V(G)$ over the number of all possible triangles that could be formed with vertex i. Which are called t_i and T_i respectively. Before the directed equation for clustering, prior properties of the graph are necessary so that

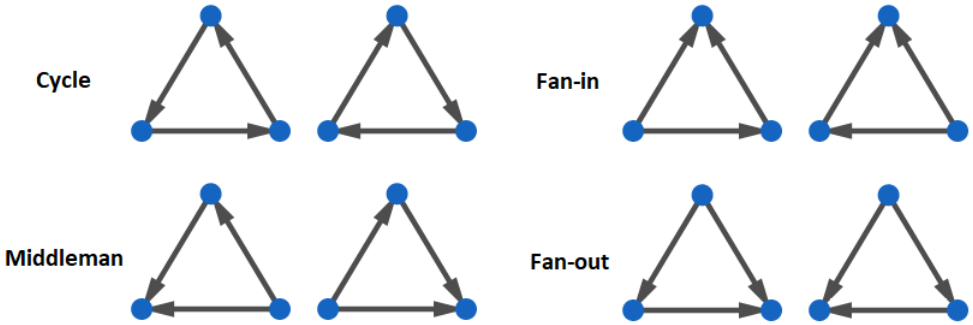


Figure 2.2: The 4 named categories of motifs in which all directed connected triangles fall under. Each pair of motifs is listed as the corresponding types which are cycles, middleman, fan-in, and fan-out.

the equation can be easily formulated. Thus, consider a graph $G = (V, E)$ with its matrix representation as the adjacency matrix A along with V_1 as the column vector, dimension n of the graph, of only 1s. A_i is the i-th row of the adjacency matrix. The in-degrees and out-degrees of a graph are the total number of edges going in or out of a vertex $i \in V(G)$ respectively, the total degree is the sum of the in and out degrees shown by Equations 2.14.

$$\begin{aligned} \text{in}(d_i) &= \sum_{i \neq j} a_{ji} = (A^T)_i V_1 \\ \text{out}(d_i) &= \sum_{i \neq j} a_{ij} = (A^T)_i V_1 \\ \text{tot}(d_i) &= \text{in}(d_i) + \text{out}(d_i) = \sum_{i \neq j} a_{ji} + \sum_{i \neq j} a_{ij} = (A^T + A)_i V_1 \end{aligned} \tag{2.14}$$

When the edge is directed both ways, this degree is the summation of the products of all the edges of vertex v that are bidirectional. Formally can be shown as equation 2.15 with A_{ii} as the i-th element of the diagonal for the matrix product of A.

$$\text{bi}(d_i) = \sum_{i \neq j} a_{ij} a_{ji} = (A^2)_{ii} \tag{2.15}$$

Equation 2.16 is demonstrated by Fagio [33] that measures the local clustering coefficient for each vertex with directed edges. Amended based on the consideration of the 8 triangles that this vertex could form shown previously in Figure 2.2.

So, this equation can be demonstrated with vertex i and pairs of neighbours j and k. Essentially showing that Equation 2.16 calculates the triangles formed by v over all possible triangles with the deduction of $2\text{bi}(d_i)$. The deduction is necessary.

Otherwise, if vertex i and vertex j have edges directed to each other, this causes a count of two additional triangles due to the nature of bidirectional edges.

$$\begin{aligned} C_i &= \frac{\frac{1}{2}\sum_j \sum_k (a_{ij} + a_{ji})(a_{ik} + a_{ki})(a_{jk} + a_{kj})}{\text{tot}(d_i)(\text{tot}(d_i) - 1) - 2\text{bi}(d_i)} \\ &= \frac{(A + A^T)^3_{ii}}{2(\text{tot}(d_i)(\text{tot}(d_i) - 1) - 2\text{bi}(d_i))} = \frac{t_i}{T_i} \end{aligned} \tag{2.16}$$

Weights of the edges can be implemented into Fagio's equation of local clustering coefficients by simply using a weighted adjacency matrix W instead of the adjacency matrix A for graph G. Mentioned before, the weights of the triangles can be considered differently. However, in Fagio's generalisation of the local clustering coefficient equation, the mean weights of the triangles are utilised. This is shown by taking a cube root of all elements of the weighted matrix W which can be denoted as $W^{[1/3]}$. Therefore by subbing $W^{[\frac{1}{3}]}$ in the place of A in Equation 2.16, the weighted directed version can be achieved, formally shown as Equation 2.17.

$$C_i = \frac{(W^{[\frac{1}{3}]} + (W^{[\frac{1}{3}]})^T)^3_{ii}}{2(\text{tot}(d_i)(\text{tot}(d_i) - 1) - 2\text{bi}(d_i))} = \frac{t_i}{T_i} \tag{2.17}$$

However, with this generalisation of the formula, it considers any triangle formed by a vertex i as equal values. This means that the directions in the triangles are meaningless when utilising this equation. For directed graphs, their directions are what gives the graph its flow of information and different directions will lead to different interpretations of the graph. Consequently Equation 2.17 must be improved to properly include the directions of edges and to consider each motif separately. In Fagio's case, treat them by considering the 4 types of categories the motifs can fall under mentioned before in Figure 2.2.

The 4 motifs were cycles, middleman, fan-in, and fan-out. By measuring each specific category, we can get more accurate coefficients for the relative pattern. The definition of the number of all possible triangles was defined in equations 2.16 and 2.17 as T_i. Similarly with the directed triangles that are formed by a vertex i as t_i. These can be both decomposed into the 4 types of motifs which can then be used to create the local clustering coefficient for each specific motif. Note that the sum of the local clustering coefficient for each specific motif will equal the local clustering coefficients for all triangles. The equations are decomposed as follows:

$$\begin{aligned} T_i &= \text{tot}(d_i)(\text{tot}(d_i) - 1) - 2\text{bi}(d_i) \\ &= \text{in}(d_i)\text{out}(d_i) - \text{bi}(d_i) + \text{in}(d_i)\text{out}(d_i) - \text{bi}(d_i) \\ &+ \text{in}(d_i)(\text{in}(d_i) - 1) + \text{out}(d_i)(\text{out}(d_i) - 1) \\ &= \text{cyc}(T_1) + \text{mid}(T_2) + \text{fan-in}(T_3) + \text{fan-out}(T_4) \end{aligned} \tag{2.18}$$

$$
\begin{aligned}
t_i &= (W^{[\frac{1}{3}]} + W^T)_{ii} \\
&= (W^{[\frac{1}{3}]})^3_{ii} + (W^{[\frac{1}{3}]}W^{[\frac{1}{3}]^T}W^{[\frac{1}{3}]})_{ii} + (W^{[\frac{1}{3}]^T}W^{[\frac{1}{3}]^2})_{ii} + (W^{[\frac{1}{3}]^2}W^{[\frac{1}{3}]^T})_{ii} \\
&= \text{cyc}(t_1) + \text{mid}(t_2) + \text{fan-in}(t_3) + \text{fan-out}(t_4)
\end{aligned} \tag{2.19}
$$

Both equations can be split according to their different motifs through logical and algebraic reasoning. For T_i, the maximum number of directed triangles for cycles, middleman, fan-ins, and fan-outs that can be formed equates to the total number of triangles for a vertex i. For example, when calculating cycles, the maximum number of directed cycles that can be formed with vertex i is the number of edges into the vertex multiplied by the number of edges out, hence giving $\text{in}(d_i)\text{out}(d_i)$ in the equation. Although, if a neighbour of vertex i has an edge to and from the another vertex then this would account to an additional triangle counted. So the subtraction of the bidirectional edges, $\text{bi}(d_i)$, remains. Notice that middleman and cycles are only differentiated by the direction of the pairs of neighbours connected to vertex i which forms the triangle. This is the reason why $\text{cyc}(T_1) = \text{mid}(T_2)$. Then for the calculations of fan-in motif, it is the multiplication of the in degrees of i and in degrees of $i-1$ (as an edge is already considered) which gives the maximum fan-in motif styled triangles. Similarly for the fan-out motif.

Then for the actual triangles formed, they can be broken down by algebra. Equation 2.20 breaks down the calculations for actual cycles based on the motif of 3 vertices. The others follow a similar process. Hence by algebraic manipulation, t_i can be separated into the 4 motifs shown before.

$$
\begin{aligned}
\text{cyc}(t_i) &= \frac{1}{2}\sum_j \sum_h w_{ij}w_{jh}w_{hi} + w_{ih}w_{hj}w_{ji} \\
&= \frac{1}{2}((W^{[\frac{1}{3}]})_{(i)}W^{[\frac{1}{3}]}(W^{[\frac{1}{3}]})^{(i)} + (W^{[\frac{1}{3}]})^T_{(i)}(W^{[\frac{1}{3}]})^T(W^{[\frac{1}{3}]})(W^{[\frac{1}{3}]})^{(i)} \\
&= (W^{[\frac{1}{3}]})_{(i)}W^{[\frac{1}{3}]}(W^{[\frac{1}{3}]})^{(i)} = (W^{[\frac{1}{3}]})^3_{ii}
\end{aligned} \tag{2.20}
$$

Finally, the local clustering coefficient can be calculated according to the specific motifs of cycles, middleman, fan-in, and fan out. Which can be formally defined as:

$$
x(C_i) = \frac{x(t_i)}{x(T_i)} \tag{2.21}
$$

where $x = \{\text{cyc, mid, fan-in, fan-out}\}$.

By demonstrating one specific formulation of the local clustering coefficient applied to weighted directed graphs, we notice that depending on variations of properties in relation to the triangles, different values may occur for the same graph. Such properties include the various different motifs of the triangles and the consideration

of the edge weights. This is reason why there exists multiple different versions of clustering calculations as each has their benefit depending on what the graph represents. The different versions of clustering formula are thoroughly examined in the paper [29]. Additionally, all versions can also be analysed against each other to determine which has the best performance depending on the ideal requirements of the task [34].

For the goal of linguistic analysis, we will initially take the calculations of the local clustering coefficients for directed graphs. So that we can determine if vertices form a relationship with specific areas of the text or with individual words. Using local clustering coefficient will help determine which words are important in its local cluster. Additionally local clustering can help visualise the various clusters in a word graph and identifies the graph's completeness. Note that the possibility of patterns may also occur when analysing the word graph since the clustering coefficients utilises triangles in a graph.

2.3 Centrality

Positioning within a graph is vital in displaying information in a simple manner. We want the positions to be useful for extracting important information. Key benefit of useful positioning is that the positions may be used to identify key areas within a graph or network such as the links between companies and which one has the most influence. This can also be applied to brain networks, the spreading patterns of disease and is useful to characterise specific areas to give a new interpretation of the graph. To assist in accomplishing a better positional system for graphs, we look at centrality values. Centrality values focus on the vertex location amongst its neighbours or connections within the whole graph. Assigning numerical values to them depending on their importance or usefulness. One of the major centrality values is the betweenness centrality of vertices in a graph which will be described in further detail in the next section.

2.3.1 Betweenness centrality

Betweenness centrality measures the centrality of a graph by using the shortest paths for pairs of vertices. It identifies vertices who are most influential within the graph, i.e., vertices that are required in the paths of other vertices. The introduction of this idea was through the view of a communications network. Where a vertex (or point) of the communications network is deemed to be central if it lies on the shortest path between another pair of vertices. Alex Bavelas [35] formularised this idea of centrality where he suggests that a person in a group is in the central position if that person lies on the shortest path between other connecting pairs. With the implication that this person then holds the power or responsibility for the others. Due to the fact that when exchanging information, the others must go through that

person. A simplistic way of viewing the betweenness value for a vertex is that the larger the value, the greater number of shortest path connections it has to other vertices. In other words, if the value is large for a vertex, then the travel time from this vertex to other vertices is much shorter.

The betweenness centrality will be discussed based on Freeman's interpretation of betweenness centrality measure [36]. For a simple unconnected and undirected graph $G = (V, E)$, consider all the unordered pairs of vertices $v_i, v_j \in V$ with $i \neq j$. This pair must either be disconnected or has at least one path connecting them with its path length based on the number of edges contained within. The path or paths connecting v_i to v_j with the shortest length is known to be *geodesic*. If the path is larger than one edge (the vertices were adjacent) then vertices in this geodesic are central. Depending on whether the vertex is the only central vertex between the pair of vertices determines if it has complete or partial control of their link. The intuition of control in betweenness was expressed by Shimbel [37] where work sites are considered as vertices. Meaning that the vertices with control will be the connecting sites between other sites. Which in terms means that the connecting sites hold responsibility to the other sites and must relay information and resources to them. Figure 2.3 expresses the idea of central vertices and the geodesics between them. For the path between vertices v_1 and v_3, there are 2 shortest paths of length 3 which means they have 2 geodesics. Suggesting that there could be up to 2 vertices which share the power of information transfer between vertices 1 and 3. Another example is that vertices v_3 and v_6 only have one geodesic through v_4 and v_0 so either v_4 or v_0 can have complete control of power.

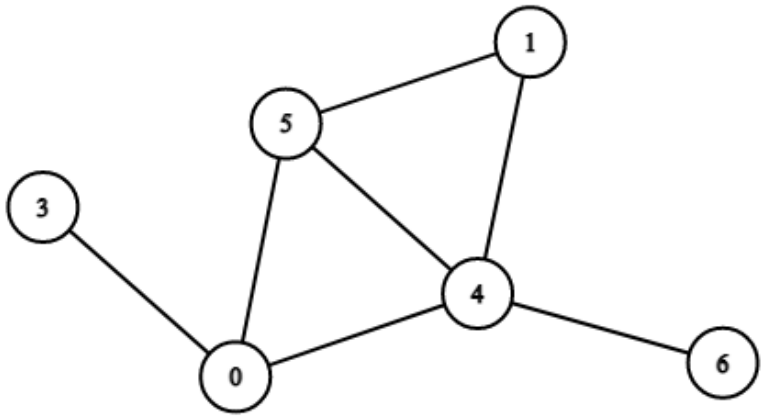


Figure 2.3: A simple graph of 6 vertices and 7 edges. Vertices v_1 and v_3 have two geodesics meaning that the central vertices are v_4 and v_5 who share partial control but v_0 has full control and is most central between v_1 and v_3. Note that v_3 and v_6 cannot be central as they only have one edge each and they only have one geodesic because if v_5 or v_1 is included then it is no longer the shortest path between v_3 and v_6.

If there is more than one geodesic, then they are considered to have equal probability in deciding which one to be used. This is simply given by $1/g_{ij}$ where g_{ij} is the number of geodesics between vertices v_i and v_j. The formalisation of partial betweenness $b_{ij}(v_k)$ is defined using the idea of geodesics, so for a vertex v_k in G and a vertex pair v_i and v_j, the partial betweenness for v_k can be calculated based on the vertex pair and is given by Equation 2.22.

$$b_{ij}(v_k) = \frac{1}{g_{ij}}(g_{ij}(v_k)) = \frac{g_{ij}(v_k)}{g_{ij}}(i \neq j \neq k) \tag{2.22}$$

where $g_{ij}(v_k)$ is the number of geodesics connecting vertices v_i and v_j that include v_k in its path. This essentially translates to the probability that the geodesic chosen for v_i and v_j contains v_k. Additionally, notice that $b_{ij}(v_k) = 0$ if there does not exist a path between the vertex pair, v_i and v_j using v_k. This equation can be extended to calculate the betweenness centrality of each vertex. So, for a graph of size n, the betweenness centrality value for a vertex $v_k \in V$ is be defined by Equation 2.23.

$$C_B(v_k) = \sum_{i}^{n} \sum_{j}^{n} b_{ij}(v_k) \tag{2.23}$$

where $i < j$. This defines the betweenness centrality for v_k and its value increases depending on the number of geodesics that v_k is a part of. The maximum value [38] was proved by Freeman using a star graph with the central vertex as v_k as all vertices are reachable if they go through the central vertex. Furthermore, this means there are $n(n-1)/2$ paths between all the unordered pairs of the star graph S. With $n-1$ edges connected to the central vertex v_k. Thus, the betweenness centrality for v_k in S is

$$C_B(v_k) = \frac{n(n-1)}{2} - (n-1) = \frac{n^2 - 3n + 2}{2} \tag{2.24}$$

And if any new edge is added that does not increase the branches of the star, then a new geodesic would form without v_k causing the betweenness of v_k to fall. Therefore Equation 2.26 expresses the betweenness centrality of any vertex in a graph G by its representation through a ratio with the maximal value as shown using the star graph S.

Directed edges are required in linguistic analysis so we study the use of betweenness centrality and its equations for directed graphs as well.

$$C'_B(v_k) = \frac{2C_B(v_k)}{n^2 - 3n + 2} \tag{2.25}$$

2.3.2 Generalisation to directed graphs

The key idea of betweenness centrality is to evaluate the graph and produce values based on the shortest paths of all the possible pairs of the graph. The higher the

betweenness value, the more likely the vertex is on the shortest paths of any two vertices. As this concept investigates the vertices on shortest paths, weighted edges would not benefit this graph evaluation as weights could cause a pair of vertices to be seen as further apart than they are within the graph. Also, for experimentations done later with word graphs, the calculations for betweenness will benefit on edges without weights. As the weights may influence the paths of edges between words when the words have a strict order within a sentence. Nonetheless a brief overview of weight inclusion is given.

In the experimentations of social networks [39], the centrality values are used on undirected graphs however an idea to generalise the betweenness to weighted graphs is to take the weights as indication of the distance of the vertices. Meaning that the geodesics of any pair will be defined on the smallest total value of paths between them rather than the shortest path length as shown on Figure 2.4.

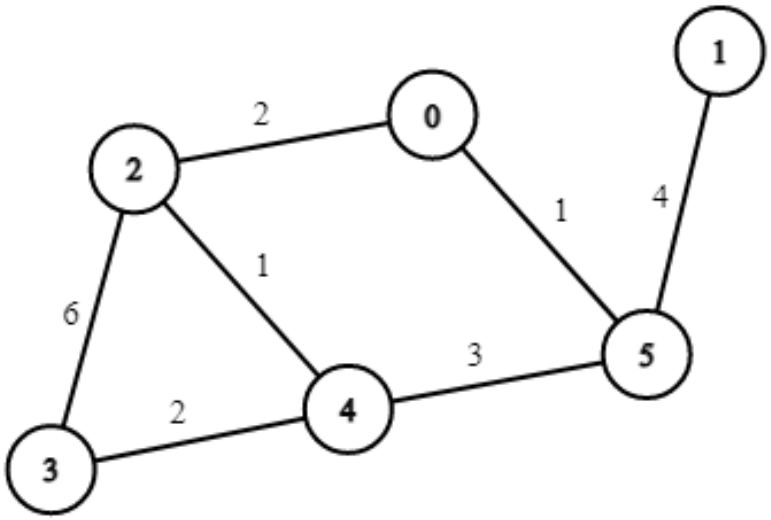


Figure 2.4: Shows a simple weighted graph. The path between vertex v_0 and v_3 has 1 geodesic but when considering the weights, this geodesic has a weight of 6. The smallest weighted geodesic is using the path $v_0v_2v_4v_3$ with a weight of 5 instead of the original geodesic.

Since weighted edges do not benefit the word graphs when considering centrality, we discuss the generalisation in this section only to directed graphs. If the graph has weights, then they are not implemented to ensure betweenness values will not be influenced and remain congruent throughout experimentation. The geodesic proportions of paths from v_i and v_j was defined earlier in Equation 2.22. Consequently, this equation can be utilised to define pair-dependency of vertices v_i to v_k where the vertex v_i must depend on vertex v_k in order to get to other vertices such as v_j on its geodesics. In other words, v_k acts like a gatekeeper to v_i. Therefore, for a graph with n vertices, the pair dependency is defined to be

$$d^*_{ik} = \sum_{j=1}^{n} b_{ij}(v_k) \tag{2.26}$$

where $i \neq j \neq k$. Matrices can be used to store the pair-dependency values to provide an ease of use and a better representation for all values. The results can be arranged into the matrix D defined as $D = (d^*_{ik})$. The elements of the matrix measures how much the vertex x (corresponding to the row number) depends on vertex y (corresponding to the column number) to connect to other vertices in the graph. Additionally, the betweenness centrality can also be calculated based on this matrix D through the summation of the columns in which the sum will give the betweenness centrality for the column number that represents the vertex. Otherwise shown as

$$\sum_{i=1}^{n} d^*_{ik} = 2C_B(v_k) \tag{2.27}$$

Which means that the betweenness centrality of v_k is double of the pair dependency column sum [40]. This is because for an undirected graph, the upper and lower diagonal matrix are equal due to the symmetry of graph G. The generalisation for directed graph can then be shown to be

$$C_B(v_k) = \sum_{i=1}^{n} d^*_{ik} \tag{2.28}$$

Therefore, we are able to use the directed version of betweenness centrality to identify centralised vertices of importance. This will be applied to word graphs generated by a dataset of under a specified language. We continue to discuss other centrality values that may be of interest.

2.3.3 Other centrality values

Centrality is a larger area within graph theory that is continuously being studied. Other than betweenness centrality, there are other centralities with similar attributes such as the *closeness centrality*, *eigenvalue centrality*, *Katz centrality* [41] and the *Hyperlink-Induced Topic Search (HITS) centrality*. All of which calculate values for the vertices or edges given their locations amongst their neighbours within the graph by various different methods. Interestingly the closeness centrality can be seen as the duality of betweenness centrality as they can be obtained from row and column summations of the dependency relation defined in the paper by Brandes, Borgatti and Freeman [42]. Betweenness studies the vertices that act as bridges between other geodesics whereas the closeness centrality measures the average distance of the shortest paths between any pairs of vertices. A vertex with high closeness value means that the distance to any other vertex is short on average.

Closeness centrality [43] can simply be calculated as the inverse of total shortest distance from a vertex i to all of vertices, demonstrated by equation 2.29.

$$C_C(v_i) = \frac{1}{\sum_j^n d(v_i, v_j)} \tag{2.29}$$

where $i \neq j$. This is the calculation is for simply connected graphs however can be easily expanded to directed and weighted graphs. Accomplished by modifying the calculation to use the measure of distances for the graph in question. I.e., take the total weights of the path length rather than the path length for weighted graphs and to consider only correct paths (travelling along an edge in the permitted direction) when investigating directed graphs. Of course, we take the version for directed graphs to be used later in duality of the betweenness centrality. This duality of the centrality values may lead to unique vertex identifications.

Therefore, by taking these properties, the graph can be rearranged in accordance to their centrality values. This is accomplished by having their property values as their position vector and is done so similarly in the paper by Juan, Alvarez, Villasante and Ruiz-Frau [44]. In which they relabelled graphs with their normalised centrality values. This ranges from the betweenness centrality to eigenvector centrality. Thus, a similar idea can be applied along with the other properties explained and studied in this chapter.

2.4 Webpages

The World-Wide Web contains webpages and hyperlinks of tremendous scale. It can also be considered as one of the largest graphs in the modern world because the webpages and hyperlinks can be see as the vertices and edges in a directed graph [45]. The World-Wide Web is traversed through the assistance of a search engine such as Google's search engine. To help navigate through the vast quantity of pages, Brin and Page [46] developed an algorithm known as the *Page Rank* algorithm. The Page rank gives a quantified meaning to the importance of the webpages and their links to other webpages. Additionally, the page rank is known as a centrality value which was discussed briefly in the last section along with the Hyperlink-Induced Topic Search (HITS) which was also created with a similar goal to page rank. Both values are based upon digraphs (directed graphs) and were generated to help with the navigation of the world wide web. The algorithms provided users with the highest quality webpages that were also most relevant to their search criteria. The idea of finding the most relevant page can be translated to finding the most relevant word within a sentence. Consequently, we can transfer the values generated for page rank or HITS onto our word graphs later on. In doing so, words of relevance can be found.

In addition to the centrality property, which we use later, we identify a couple

web page algorithms and deduce the main one that will be used during linguistic analysis. These will either be the HITS algorithm or Page Rank algorithm.

2.4.1 Hyperlink-Induced Topic Search

HITS algorithm calculates the ranks of *authorities* and *hubs* in relation to their in-links and out-links [47]. In other words, the edges pointing into vertices and the edges pointing out of the vertices in the graph/network. Authorities and Hubs are assigned to webpages (vertices) depending on their number of in-links and out-links. For the HITS algorithm, the webpages that have lots of in-links pointing to it are denoted as authorities. The webpages that have lots of out-links pointing to other webpages are denoted as the hubs. These can be identified by the HITS algorithm as the calculations are an iterative process. The iterative process enforces the authorities and hubs by bringing the authorities to the surface of the graph. Leading to the isolation of the authorities and hubs from the other webpages. This is achieved by generating the hub and authority values through mutual reinforcement. Based on vertices $i \in V$ from a graph of webpages $G = (V, E)$, the hub value h_i and authority value a_i are first set to a value of 1. This is so that they are indistinguishable and will later be relabelled if necessary. Then by Kleinberg's HITS algorithm, they are updated iteratively through the formulas:

$$a_i^{(k)} = \sum_{j:j\to i \ \& \ j\neq i} h_j^{(k-1)}, \qquad h_i^{(k)} = \sum_{j:i\to j \ \& \ j\neq i} a_j^{(k)} \tag{2.30}$$

where j is denoted as the links from and to the webpages of i and j. The k depicts the k^{th} iteration of the algorithm, so the authority values depend on the previous iteration of hub values and the hubs are calculated based on the current k^{th} authority values. Which gives the mutual reinforcement of both values. As the graph G can be represented by an adjacency matrix $A = [a_{ij}]$, the formulas can be expressed through matrices [48] and vectors instead as shown by Equation 2.31.

$$\mathbf{a}^{(k)} = \mathbf{A}^T\mathbf{h}^{(k-1)}, \qquad \mathbf{h}^{(k)} = \mathbf{A}^T\mathbf{a}^{(k)} \tag{2.31}$$

where the hub and authority values are adapted into vectors $\mathbf{a}^{(k)}$ and $\mathbf{h}^{(k)}$. The vectors are expanded and shown as

$$\mathbf{a}^{(k)} = \begin{bmatrix} a_1^{(k)} \\ a_2^{(k)} \\ \vdots \\ a_n^{(k)} \end{bmatrix}, \qquad \mathbf{h}^{(k)} = \begin{bmatrix} h_1^{(k)} \\ h_2^{(k)} \\ \vdots \\ h_n^{(k)} \end{bmatrix} \tag{2.32}$$

Accomplishing the goals of generating the values which depict the hubs and authorities more clearly within the graph. Although this algorithm is currently

defined only for directed weightless graphs, we extend the algorithm to include a weighted edge version.

As the number of iterations increases, the values generated may increase. This means that at some point, the values will become too large to be used for any calculations. In order to mitigate this problem, the values can be normalised which prevents the issue of tending to infinity. After k iterations, the normalised formulas are demonstrated by Equations 2.34 with the inclusion of weights. The general outline for the HITS algorithm is demonstrated by Agosti and Pretto [49] as the following:

$\mathbf{a}^{(0)} := \mathbf{u}$, $\mathbf{h}^{(0)} := \mathbf{u}$;
for $k := 1$ to K **do**
    $\mathbf{a}^{(k)} = \mathbf{A}^T\mathbf{h}^{(k-1)}$;
    $\mathbf{h}^{(k)} = \mathbf{A}^T\mathbf{a}^{(k)}$;
    normalise $\mathbf{a}^{(k)}$ such that $\left\|\mathbf{a}^{(k)}\right\| = 1$;
    normalise $\mathbf{h}^{(k)}$ such that $\left\|\mathbf{h}^{(k)}\right\| = 1$;
end for
$\mathbf{a} := \mathbf{a}^{(K)}$, $\mathbf{h} := \mathbf{h}^{(K)}$;

where K denotes the maximum number of iterations and $\mathbf{u}$ as the vector for the first iteration of hub and authority values, also known as the base case. The base case for $\mathbf{u}$ will just be the vector of 1s known as $\mathbf{1}$ or $\mathbf{e}$ in linear algebra. Then the normalised values after k iterations is given as follows:

$$\mathbf{a}^{(k)} = (\mathbf{A}^T\mathbf{A})^{k-1}\mathbf{A}^T\mathbf{u}, \qquad \mathbf{h}^{(k)} = (\mathbf{A}\mathbf{A}^T)^k\mathbf{u} \tag{2.33}$$

To incorporate weights of the edges into the algorithm, the weighted matrix W for the graph G can be used in place of the adjacency matrix. Simply by replacing the A in the formulas, the weighted version can be generated as the formulas:

$$\mathbf{a}^{(k)} = (\mathbf{W}^T\mathbf{W})^{k-1}\mathbf{W}^T\mathbf{u}, \qquad \mathbf{h}^{(k)} = (\mathbf{W}\mathbf{W}^T)^k\mathbf{u} \tag{2.34}$$

Concluding the study on the HITS algorithm, we now proceed with the Page Rank algorithm which is more commonly known and used.

2.4.2 Page Rank

A widely known algorithm that contributes to internet navigation within Google is the page rank. The page ranks are needed because many different search queries can be entered onto the search engine. These queries produce lots of results that contain the same or similar words to what was searched. Consequently, a method to help organise and prioritise the results is necessary. The page rank is used to rank the webpages according to the number of backlinks a page may have and the number

citations that reference a particular page. An example of webpages and links in the form of a graph is shown in Figure 2.5 with their page ranks displayed. The Page Rank algorithm can also be compared to the HITS algorithm to see the benefits of either. This is explored in the paper by Devi, Gupta and Dixit [50].

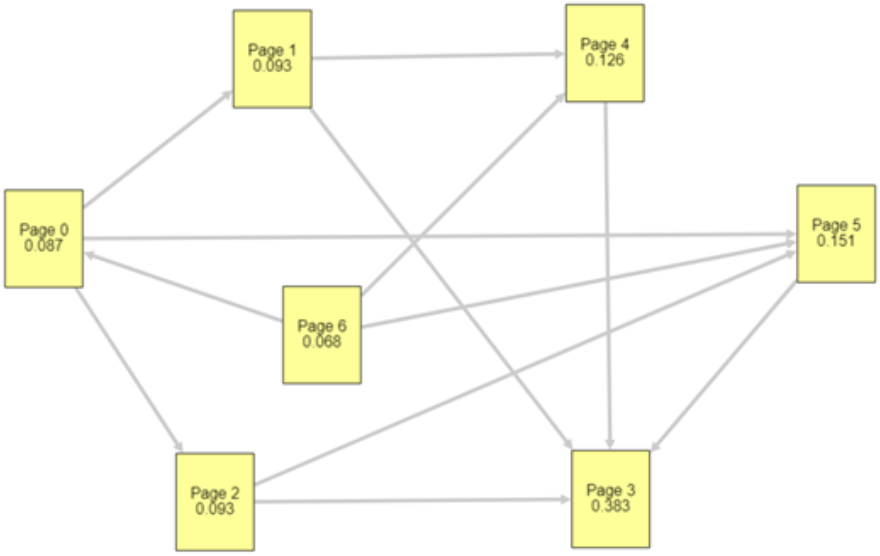


Figure 2.5: A graph representation of webpages and their in and out-links. Algorithms such as Page Rank and HITS use this graph format to help demonstrate their calculations.

For the graph $G = (V, E)$ the webpages can be seen as vertices $v \in V$ with their links as the directed edges between pages, the same way to how they were represented in the HITS formulas. We can let F_v be the set of webpages that v points to, i.e., the forward links. Similarly, the backwards links as B_v which is the set of webpages that points to v. The simplified version of Page Rank [51] can then be defined with $N_v = |F_v|$ in Equation 2.35.

$$PR(v) = c \sum_{w \in B_v} \frac{PR(w)}{N_w} \tag{2.35}$$

where c is a variable used for normalisation purposes so can be modified accordingly. The page rank is calculated by the even distribution of the page rank for webpage v among webpages that v points to. The page ranks that links to other webpages from v are then used to calculate their page rank. Hence giving an iterative approach to calculating their ranks whilst the algorithm travels along the links of the webpages. On the other hand, if a webpage does not have out-links and only in-links from other webpages then their rank is never distributed to others causing its value to accumulate. This situation is known as a *rank sink*.

A remedy to having a rank sink is to use a a damping factor to illustrate the probability that the user follows the links on the webpage. This takes into the consideration that users could skip pages or go directly to another webpage that were not linked through the URL. Hence $(1-d)$ is considered as the distribution of the page rank from webpages that were not directly linked to it, i.e., no direct edges between them. Thus ,for the page v with $b_i \in B_v$, the page rank [46] is defined through Equation 2.36.

$$PR(v) = (1-d) + d(PR(b_1)/F(b_1) + \ldots + PR(b_n)/F(b_n)) \tag{2.36}$$

where $F(v)$ was retrieved from the set F_v for the webpage v. Page rank is applied to each page and is repeated on further pages until the equation converges. The damping factor adds randomness into the network of webpages so that the rank sink does not occur and ensures the convergence is not reached too quickly. Usually, the damping factor is taken to be 0.85.

By using summation, the Page Rank formula can be simply reduced to

$$PR(v) = (1-d) + d \sum_{w \in B_v} \frac{PR(w)}{N_w} \tag{2.37}$$

The algorithm for calculating page rank can be modified depending on the use and aims. Such modifications include adjusting the vertex and edge values, modifying the damping factor or introducing a new variable into the algorithm. These such changes are known as *personalisation* and an example of personalisation is the *Weighted Page Rank* [52]. Where larger values are assigned to more popular or important webpages rather than the even distribution that occurred beforehand. Webpages that have out-links will instead receive a value proportional to the page's popularity. Their popularity is based off the number of in-links and out-links they hold. These popularity values of the in-links and out-links are represented by $W^{in}_{v,w}$ and $W^{out}_{v,w}$ respectively. $W^{in}_{v,w}$ is calculated by the in-links of webpage v, shown to be I_v, and the in-links of all the webpages that the webpage w references. This is represented as a set and we call this set of pages $R(w)$ with I_p where $p \in R(w)$. Furthermore with these changes, $W^{in}_{v,w}$ and $W^{out}_{v,w}$ can then be formularised into Equations 2.38 and 2.39 respectively.

$$W^{in}_{v,w} = \frac{I_v}{\sum_{p \in R(w)}} I_p \tag{2.38}$$

$$W^{out}_{v,w} = \frac{O_v}{\sum_{p \in R(w)}} O_p \tag{2.39}$$

where O_v, O_p is defined the same as the in-links previously but with the use of the out-links as a replacement. Therefore, the Page Rank formula can be modified to include the webpage's importance giving us the Equation 2.40.

$$PR(v) = (1-d) + d \sum_{w \in B_v} PR(w) W^{in}_{v,w} W^{out}_{v,w} \tag{2.40}$$

So, this formula is focussed more upon the webpages that are visited more frequently by users and ensures they end up with a higher rank.

However, for the purpose of general directed weighted graphs, the edge weights are lost in the formula as the algorithm updates the rank of every vertex in each iteration meaning that the weights can be disregarded and replaced with the page ranks instead. To ensure this does not happen, the weights of all the edges must be included in the rank's calculation. This is accomplished by summing up the weight values of the in and out edges and incorporating it into the formula as achieved by Equation 2.41.

$$PR(v) = (1-d) + d \sum_{w \in B_v} \frac{PR(w) w_{(w \to v)}}{N_w} \tag{2.41}$$

where $N_w = \sum_y A_{w,y} w_{(w \to y)}$ is redefined as the sum of weights of the out-linked edges in relation to vertex w.

Whilst HITS and Page Ranks are designed for the search engine, the network of webpages essentially is a large directed graph that can contain weights hence why the Page Rank can be used on any derivatives of a weighted directed graph. Thus, these are additional graphical properties that can be used to help analyse the structure and linkage of a graph.

In conclusion, we take the page rank formulas into the experimentations of linguistic analysis. This is because we do not have the requirements of hubs and authorities as shown in the HITS algorithm, so we use the page ranks for directed graphs in future. In addition to the page ranks, we will incorporate the use of local clustering coefficient, trophic levels and the betweenness centrality so that various graph visualisations can be produced. Before applying these to linguistic data, we begin with initial experimentations on a smaller dataset shown in the next chapter. This is to ensure that the visualisations work and to implement any modifications that may be required. Also gives us further insight as to what various values represent in a different dataset.

Chapter 3

Application of Graph Properties

Now we apply the properties discussed in the last chapter onto connected graphs to demonstrate their use. These properties are used to outline and modify the graph such that an alternative layout may be given. A different layout means a new visualisation that can reveal correlations between the vertices or edges. By using Python, I have coded a program to display a graph either generated from an adjacency matrix, a weighted matrix or a dataset that is pre-existing. To help accomplish this, I have used Tiago P. Peixoto's Graph Tool library for python [53] which contains useful documentation and functions to achieve the graph generations. Along with Peixoto's library, I have used other mathematical libraries for complex arithmetic. The general idea is to compare various graph properties by modifying their positions according to the values of their graph properties. For simplicity and the goal of being comparable, we choose the y-axis of the graph to be based upon the trophic levels and the x-axis to vary between the different properties discussed in the last chapter.

3.1 Early Experimentations

Out of the many pre-existing datasets from the graph tool library, I chose to experiment on a smaller dataset that demonstrates the relationships between karate clubs in a city. This is so that I can test and generate a visualisation of this dataset before beginning the linguistic analysis. This dataset involves 34 karate clubs and the initial graph can be seen by Figure 3.1. This is also to test that the visualisations are viable for a simple undirected graph prior to directed graphs. However, the trophic levels are not used ideally in this scenario as they focus on directed graphs and their hierarchical structure. Consequently, they will be analysed thoroughly in linguistic analysis later and will not be analysed for this karate dataset.

The current positioning of the graph is determined based on the idea that the vertices do not overlap with each other, and their connections are easily visible. So, the positioning of the elements in the dataset has no real benefits other than

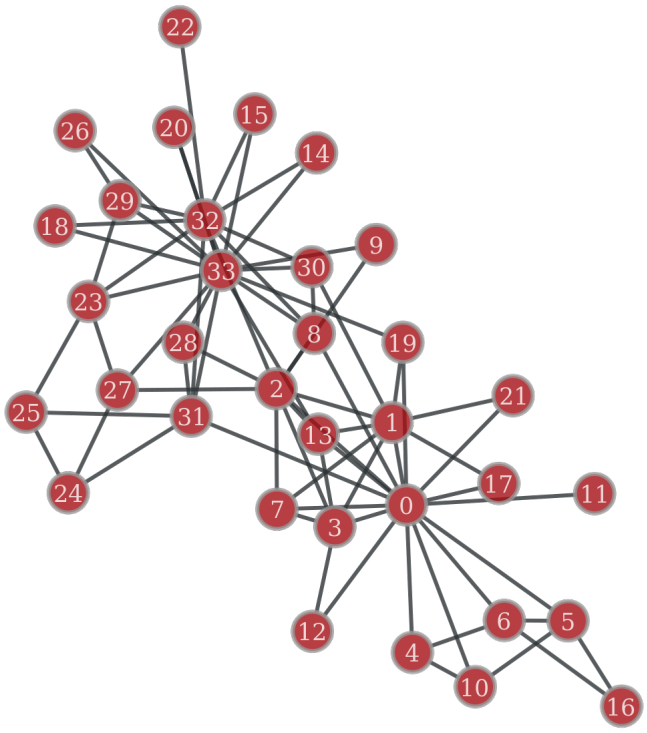
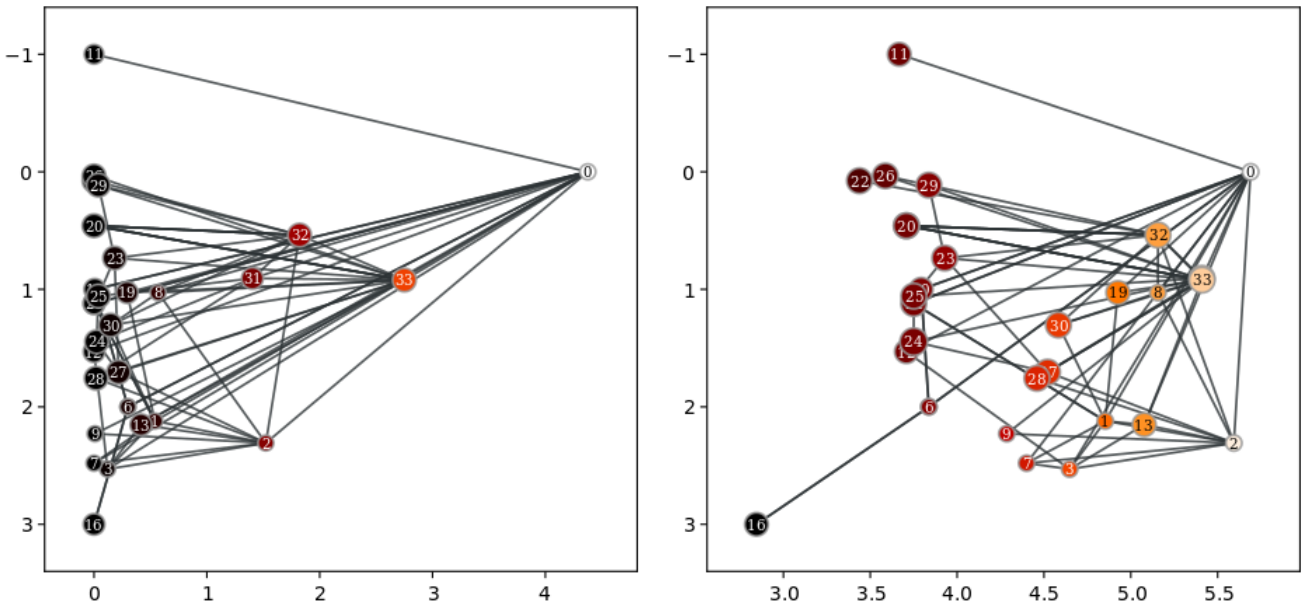


Figure 3.1: The initial graph based on the karate club dataset pre-existing within the library that was generated through the python program. Contains 34 clubs and their connections to one another with no important meaning with the positions of the vertices.

having good visibility. Any correlations or vital information cannot be derived from the initial graph. The only information that can be derived by initial examination are the certain outliers who only have one edge, vertices 22 and 11. Thus, we now include various graph properties discussed in the last chapter to ensure that more can be visualised. This can be seen in Figures 3.2 and 3.3 by using the trophic levels values for each vertex as their y-value and another property value for their x-value.

For karate clubs, the trophic levels do not represent a clear hierarchical format due to the undirected edges as there is no distinction between "upstream" or "downstream" of information. Even though they may not be accurate, we use them temporarily to distinguish the vertices as they can implement some structure into the karate dataset. Further datasets involving languages will be directional so that trophic levels can be used optimally.

Figure 3.2a shows the karate graph with the x-axis representing the betweenness value. The betweenness values are scaled by a factor of 10 to give a clearer visualisation with betweenness value increasing on the right of the x-axis. We can see that vertex 0 is the furthest vertex on the x-axis meaning it has a high betweenness



(a) (b)

Figure 3.2: Graph generated by karate dataset, the y-axis represents the trophic levels, and the x-axis represents the (a) betweenness centrality values and (b) closeness centrality values for all the vertices. Additionally, added colour changes between the x-axis to give a clearer visualisation of the separations.

value. This means that vertex 0 is involved in the highest number of shortest paths between all the vertices of the karate graph. On the other hand, vertices such as 11, 7, 16, etc., are clubs who are on the outskirts with no convenient connections to other clubs. These vertices are shown to be on the left side of the graph. Therefore, the clubs with larger betweenness are the clubs who are more centralised in a city and have more meaningful links to others.

We compare this to another centrality value, the closeness value, which has also been scaled by a factor of 10. Figure 3.2b represents this in place of the betweenness value. Through comparison of both, vertices are positioned similarly to betweenness. This is because betweenness values considers all vertices within the graph whereas closeness considers all neighbours of a specific vertex. In other words, betweenness measures the control a vertex has over the flow of information through the entire graph, whereas closeness measures the control over the flow of information with vertices in proximity (i.e., neighbours). Consequently, the vertices on the right of both the betweenness and closeness graph would be the largest clubs. Additionally, clubs such as 19 who have a larger closeness compared to betweenness means that it is important to the clubs in proximity of itself. In other words, the club is the largest within its local area.

Figure 3.3 is shown with local clustering coefficient as the x-axis instead. Notice that the clubs with less connections to the major clubs (vertices of high degree) are

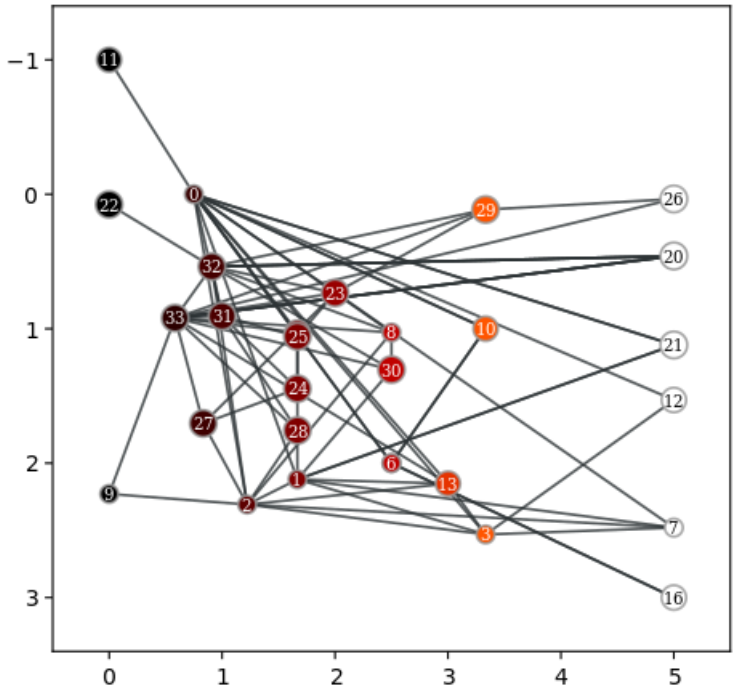


Figure 3.3: Graph generated similarly to the betweenness graph for the karate dataset, but the vertices are plotted with local clustering coefficient as the x-axis and trophic levels as the y-axis.

visually seen to the left in both graphs. These vertices would have a low clustering coefficient as well as a low betweenness. On Figure 3.3, the vertices with the best connections to major clubs are located further to the right of the graph which are the karate clubs 26, 20, 12, etc. However, vertices with high betweenness such as vertex 33 has a smaller local clustering. Due to their club having connections to other smaller clubs, decreasing the overall value of its own connections. This means that clubs on the right have quicker access or communication with larger clubs and are the closest to them.

This concludes the early experimentations on a smaller undirected dataset. We see that the graph properties are successful in identifying correlations and certain vertices that are important. Therefore, instead of using simple datasets, I will generate the datasets based upon different languages as well as their sentence structure. To understand this further, words in languages must be given a rank to judge their importance. This can be demonstrated through *Zipf's Law* discussed in the next section.

3.2 Zipf's Law

Zipf's law analyses the natural languages and the frequency of words that appear in them. Alternatively, Zipf's Law [54] is generally seen as the frequencies of specific events are inversely proportional to their rank that is determined through this law. The law was proposed by George Kinbgsley Zipf when researching the various frequencies of words within the English language. The law states that the r^{th} most frequent word in the language has a frequency of $f(r)$ which has a relation with the inverse of r. Denoting r as the *frequency rank* for the word and $f(r)$ as the frequency of the word in the corpus examined (The *corpus* means the collection of written text).

$$f(r) \propto \frac{1}{r^{\alpha}} \tag{3.1}$$

This is the scale for $\alpha \approx 1$ and means that the most frequent word in the examined text which is $r = 1$ has its frequency of appearance to 1, the next most frequent word which is $r = 2$ has a frequency appearance of $\frac{1}{2^{\alpha}}$ and so on. Zipf's law can be drawn on a graph to show a relation and when $log(f)$ is drawn against $log(r)$, the graph generates a curve that closely resembles a straight line with a slope of -1. This is known as Zipf's curve and later in the 1960s, the curve's nature was reinforced by the law being correct for smaller *corpora* (the plural of corpus) [55]. However, the curve varies depending on the corpora as expected and the higher ranking words deviated more from the straight line. Therefore, Mandelbrot derived a generalisation for Zipf's law to adjust to the frequency distributions within the different languages. Mandelbrot proposed to adjust the rank by a constant β, demonstrated by

$$f(r) \propto \frac{1}{(r+\beta)^{\alpha}} \tag{3.2}$$

Generalisation of Zipf's law can then be applied to various corpora of languages so that a frequency distribution can be viewed for the corpus. An example of this can be seen in Figure 3.4 sourced from [56].

Words in a corpus have a systematic relationship between their rank in their occurrence table. Meaning that they are the words most used such as "the", "or", "of". These words account for most of the word occurrences in the English language. Other words such as "xylophone" and "accordion" have the least occurrence in English. A larger corpus is studied by Bentz, Kiela, Hill and Buttery [57] where they study Zipf's law for Old English and Modern English. They study the frequency and ranks of each word whilst comparing them between the old and new English. In doing so, for old English, the words "and" is ranked first with a frequency of 1731 whereas in modern English, "the" is ranked first with a frequency of 1775 and "and" is second instead with a frequency of 1024. By looking at more words and the comparisons between them, old English has a larger number of distinct words

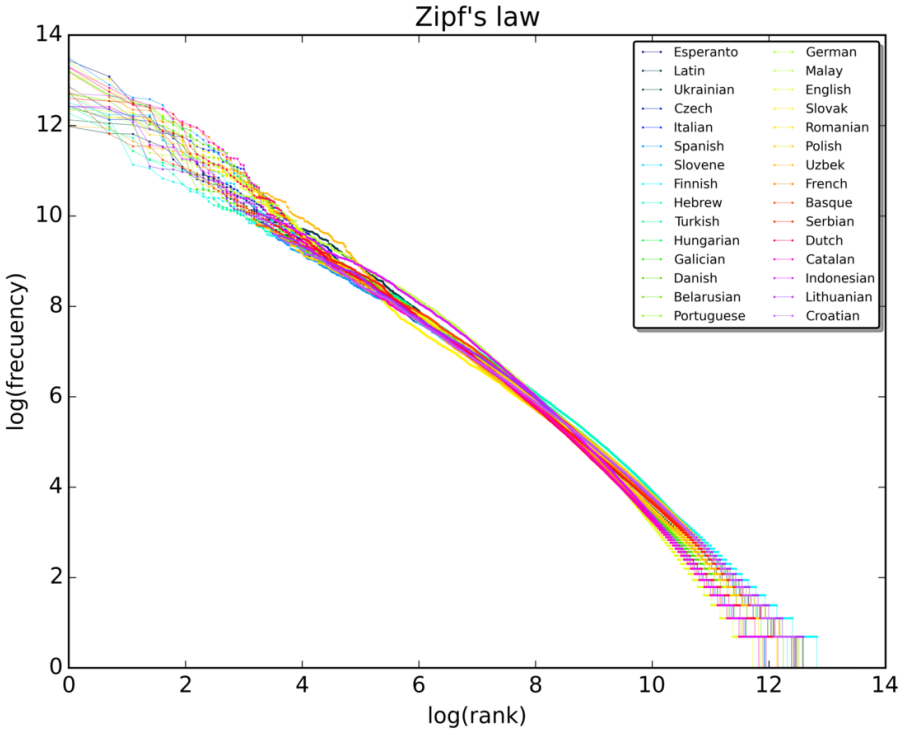


Figure 3.4: The plot of Zipf's law containing 30 different language corpora generated from the first 10 million words in each language from Wikipedias.

whilst modern English has less. However, modern English has a higher frequency for its first 100 words.

In conclusion, Zipf's Law is a useful tool because languages tend to follow Zipf's curve in terms of their frequencies and rank. Consequently, by following an existing corpus of language, the data can be extrapolated and used in other corpora to determine similarities between the known and the unknown texts. As well as to use it to determine the types of words that are deemed to be most common in a language.

Chapter 4

Analysing Languages

Learning from prior graph generations of each complex property in the last chapter, adjustments have been made for improvement in analysis. This includes normalisation of the values calculated into a range of $0-10$ instead of a scale factor. In doing so, ensures that the graphs have the same size and axis to provide easier visual comparison. Additionally, I have included page rank in the properties to analyse. These properties were trophic levels, betweenness centrality, closeness centrality and local clustering coefficients. Finally the linguistic datasets I will use will be generated through my program by an input of text. Afterwards will be converted into graphs where the words represent the vertices and the edges are directed to the next word in the order of the text. A factor that affected the dataset was the punctuation so to achieve congruent data, the punctuation are stripped unless they symbolise the end of a sentence such as full stops, question marks, exclamation marks, etc. Therefore, the graphs we experiment on are directed graphs generated from different language families.

4.1 Linguistics

Modern languages are descendants of ancestral languages through evolution of linguistics. Throughout the different ages of the world, language have been a key part in communication between societies. They are developed and taught to newer generations to reach the stages in the current world. The history of languages can be viewed as a family tree where modern languages are nearer the bottom. Within this tree, there are groups of languages that will share a common ancestor. These are defined to be the *language family* of the languages branching off it.

Estimations of around 500 language families exist and Campbell [58] has reported that there are exactly 406 independent language families including dead languages and *language isolates* (where the language does not fit into any language family). According to the Ethnologue [59], the research centre for language intelligence, there are 142 different living language families. Of the living families, 6 are considered

to be the major families. These families are known as the Indo-European, Afro-Asioatic, Niger-Congo, Austronesian, Sino-Tibetan and Trans-New Guinea.

The aim of my research is to study modern languages that fall under the Indo-European language family and the Sino-Tibetan language family. *Proto-languages* are alternative names for major language families since they are the parent language to many other languages [60]. English, German and Dutch are Germanic languages under the Indo-European language family. Russian and Polish are Balto-Slavic languages under the Indo-European family. French and Spanish are Latin languages under the same language family. Chinese falls under a different language family, the Sino-Tibetan family. Furthermore, I will also look at Japanese which is part of the Japonic family but would have been considered a language isolate if the Ryūkyūan languages were not distinct from Japanese [61]. Therefore all the languages mentioned will be used in translating the chosen text extract into their relative datasets.

4.2 Text Corpus

The best way in comparing the results to other languages is to have datasets based on the same text extract. Thus, the text extract chosen should be simple and well know. In my case, I have chosen to use the popular story in many languages known as "Sleeping Beauty". To ensure that the same version is used, the Grimm Brothers version is utilised where the original was in German and extracted from the book of children stories "Kinder- und Hausmärchen" [62]. So using translations of this story, word graphs will be generated and experimented on. Additionally, instead of using the entirety of the story, the first two paragraphs are used so that the graphs are not overwhelmingly dense whilst maintaining any key attributes in the various languages. A partial extract of the first two paragraphs is given as follows (the full two paragraphs is shown in Appendix A.1):

> "In times past there lived a king and queen, who said to each other every day of their lives, "Would that we had a child!" and yet they had none. ... There were thirteen of them in his kingdom, but as he had only provided twelve golden plates for them to eat from, one of them had to be left out."

In conclusion the 9 datasets are created based on this text extract for graph property calculations. In the next section, the study of Indo-European languages will be undertaken, beginning with English.

4.3 Indo-European Language Family

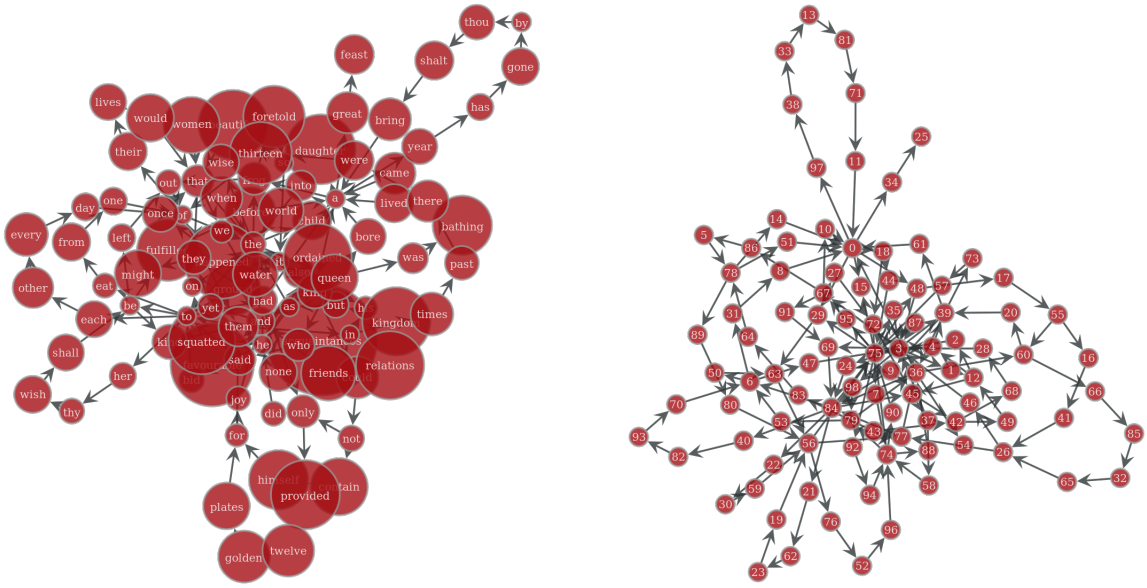
Five languages of the Indo-European language family were analysed but detailed analysis of English, German and French will be given in this section. Throughout

the analysis, words are referred to as vertices and vice versa. The two paragraph extract of "Sleeping Beauty" will simply be referred to as the story corpus of its relative language. Edges of the generated graph will represents the connections to the subsequent word in the story corpus.

Before the analysis can be undertaken, a brief summary of the languages structure and grammar is given. This is to ensure the increased understanding of correlations and results achieved through this process.

4.3.1 English

English words can be organised into eight different *parts of speech*; Nouns, Pronouns, Adjectives, Adverbs, Verbs, Prepositions, Determiners and Conjunctions. Linguistic researchers focus on the use of these categories in different situations such as through speaking or in magazines [63]. We will study the appearances of these categories in our story corpus. To achieve this, the English story corpus is converted to a dataset so that a directed graph can be generated shown in Figure 4.1a. Additionally, in replacement of having each vertex labelled by the corresponding word, each vertex will be labelled with an integer shown in Figure 4.1b. The corresponding integer for each word will be shown on the table of values for the graph. The entire table can be seen in Appendix A.2 which holds 99 unique words.



(a) (b)

Figure 4.1: Initial graphs generated off the English story corpus. (a) shows the graph with vertex labelling of their corresponding words. (b) shows the same graph but with integer labels rather than word labels to provide better visibility.

As done in the Early Experimentations of the karate club dataset, we calculate the values of the various graph properties explained in Chapter 2. These are graph properties such as local clustering coefficient, betweenness centrality, closeness centrality, trophic levels and additionally, page rank. Values are organised into their corresponding columns and presented as a table using abbreviations of each graph property. For example, "TL" for trophic levels, "CC" for closeness centrality, etc. The ten most frequently used words are shown in Table 4.1 below. Where the count denotes the frequency of the words appearance in the English story corpus.

Vertex	Word	Count	TL	BC	CC	LC	PR
3	and	9	2.03	10.00	2.49	0.57	8.13
75	the	9	3.34	9.19	2.20	0.44	10.00
0	a	7	3.83	9.26	2.22	0.00	7.65
84	to	6	2.22	7.48	2.28	0.18	5.54
36	had	4	2.75	5.32	2.48	0.36	3.82
39	he	4	1.80	3.68	2.07	0.48	2.38
56	of	4	3.64	6.21	2.37	0.00	6.03
74	that	4	2.80	3.49	2.19	0.36	5.12
6	be	3	2.86	3.15	1.91	0.00	3.24
12	but	3	1.33	1.29	1.95	0.00	1.73

Table 4.1: The first 10 most common words of the dataset. Generated from the English version of "Sleeping Beauty" in a table format.

We begin by analysing words with the most recurrences (see Table 4.1). In the order of most frequent to least, the words are "and", "the", "a", "to", "had", "he", "of", "that", "be" and "but". Note that nouns, adverbs and adjectives do not appear in the most frequent words. These are the words that are deemed more vital in the creation of structure within a sentence of the story corpus. Any sentence in "Sleeping Beauty" will have a high chance of containing at least one of these words. Since this is only a portion of a dataset, we can compare these words to a larger dataset to see whether or not the importance of these words remain same. Word frequencies follow the Zipf curve for languages, as discussed in the previous section, so we can take another corpus to compare its frequency of words to ours. We choose to compare the story corpus to the British National Corpus (BNC) [64] which is a 100 million word collection that includes both written and spoken English language. The benefits in choosing BNC is that it contains older English so may provide clearer correlations to the story of "Sleeping Beauty" (the Brothers Grimm version began in the late 18th century). So for the BNC, the top ten words [65] in order of frequencies are "the", "of", "and", "a", "in", "to", "it", "is", "to" and "was". Comparing the most frequent words in both corpora, correlations are achieved such as the repetitions of words "the", "to", etc. Such similarities reinforce the fact that the English language has a structured form that requires the use of these such words, as demonstrated with a much smaller corpus compared to the the BNC.

Vertex	Word	Count	TL
25	feast	1	10.00
34	great	1	6.91
15	child	2	6.67
95	world	1	6.42
54	none	1	5.83
63	out	2	4.77
0	a	7	3.83
11	bring	1	3.83
13	by	1	3.83
33	gone	1	3.83

(a)

Vertex	Word	Count	TL
43	in	2	0.00
55	not	2	0.09
83	times	1	0.43
16	contain	1	0.50
64	past	1	0.86
41	himself	1	0.92
37	happened	2	1.06
42	his	2	1.29
78	there	3	1.29
49	kingdom	1	1.31

(b)

Table 4.2: Partial extracts of the table data ordered by their trophic levels. (a) top 10 words and (b) bottom 10 words ranked by their trophic levels based on the English story Corpus.

Now onto the analysis of Trophic levels and coherence (see Tables 4.2a and 4.2b). When applying trophic level calculations on directed word graphs, the levels represent the positioning of the word within its relative sentence. Similarly shown in the analysis of network data in empirically-derived directed networks [66]. For the "Sleeping Beauty" English word graph, the lower trophic levels denotes the vertices most likely to be sentence starters and the higher levels denote the sentence enders. This is supported by the data because the top six words with largest trophic levels values (ranging from $10.00 - 4.77$) are all sentence enders or accompanied a sentence ender (vertex 34). Along with the bottom five being words (trophic values ranging from $0.00 - 0.86$) nearer the start of sentences such as "there" and "times" in relation to the corpus. However trophic incoherence is calculated to be 0.91 which means that the levels in the graph can not be distinguished and are not clear. Since the trophic coherence $= 1 -$ trophic incoherence $= 0.09$ which is due to the fact of the vast difference of sentence lengths in the corpus. Which varies from the shortest sentence of five words and the longest of forty seven words. Consequently in relation to the extract used, a clear hierarchical layout may not be provided but there remains a good layout for sentence flow. Demonstrated by further graphs with the trophic levels as the y-axis ranging from 0 to 10 downwards to provide a normal cascade of words. Then the x-axis ranging from 0 to 10 from left to right which will be based on other graphical properties.

Extraction of the top ten vertices for each graph property to provide ease of comparison later on with the graph visualisations.

As visually demonstrated on Figures 4.2a and 4.2b, the centrality values for each word is plotted against their trophic levels. Both have been normalised to a range of 0-10 with Figure 4.2a showing the betweenness centrality on the x-axis and Figure 4.2b showing the closeness centrality. The axis are provided here to demonstrate

Vertex	Word	BC
3	and	10.00
0	a	9.26
75	the	9.19
84	to	7.48
56	of	6.21
48	king	5.79
36	had	5.32
77	them	5.03
39	he	3.68
74	that	3.49

(a)

Vertex	Word	CC
34	great	10.00
3	and	2.49
36	had	2.48
77	them	2.40
56	of	2.37
84	to	2.28
0	a	2.22
75	the	2.20
74	that	2.19
26	for	2.17

(b)

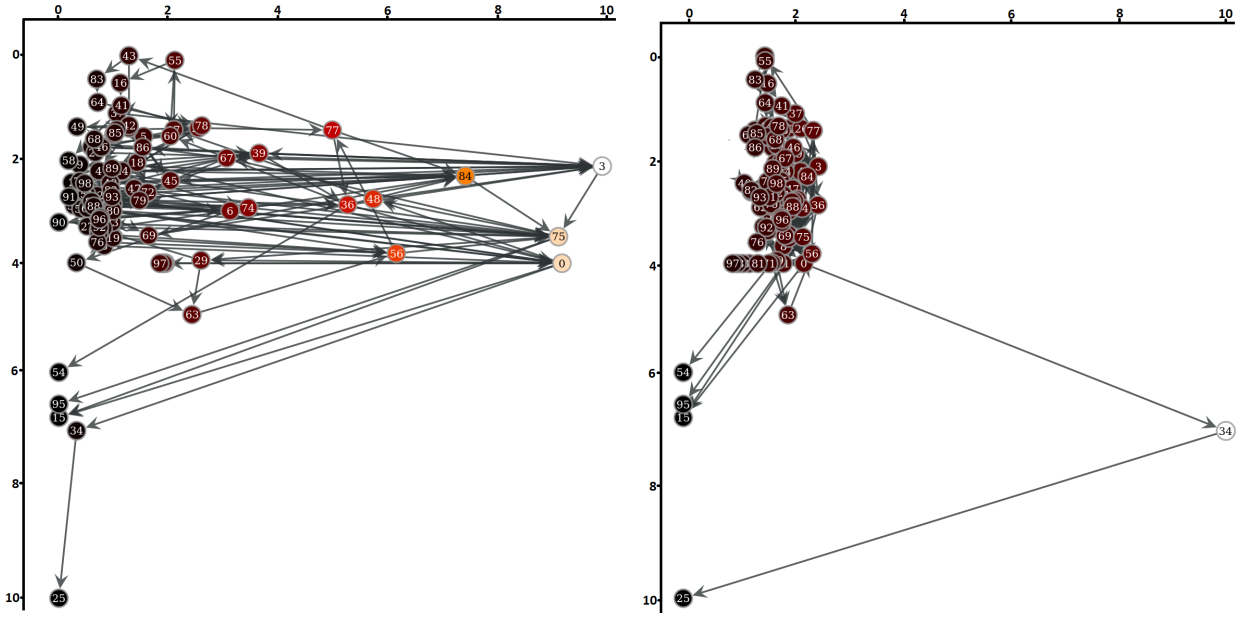
Vertex	Word	LC
35	ground	10.00
87	water	10.00
90	when	10.00
4	as	3.33
48	king	1.67
77	them	1.00
67	queen	1.00
3	and	0.57
39	he	0.48
75	the	0.44

(c)

Vertex	Word	PR
75	the	10.00
3	and	8.13
0	a	7.65
56	of	6.03
84	to	5.54
74	that	5.12
36	had	3.82
6	be	3.24
77	them	2.70
39	he	2.38

(d)

Table 4.3: Partial extracts of the English table data ordered by their (a) betweenness centrality values, (b) closeness centrality values, (c) local clustering coefficients and (d) page ranks.



(a) (b)

Figure 4.2: The x-axis positioning of vertices are altered based on their (a) betweenness centrality and (b) closeness centrality values. The y-axis uses the trophic levels. Since the axis is the same for any graph, due to normalised values, further graphs will not contain the axis for clarity.

the ranges however will not be included for future graphs because they follow the same axis layout.

Key vertices can be visually identified based on their betweenness values. These

vertices are vertex 3, 0 and 75 which are the words “and”, “a” and “the” respectively (shown by the Table 4.3a). These are conjunctions and determiners of the English language and they have the largest frequency of appearance in the story corpus discussed before. Demonstrating a strong link between the betweenness centrality values of vertices to the word frequencies within the text. Furthermore, these words are common in forming correct structure of a typical English sentence meaning that high betweenness associates the words as key bridges in a sentence.

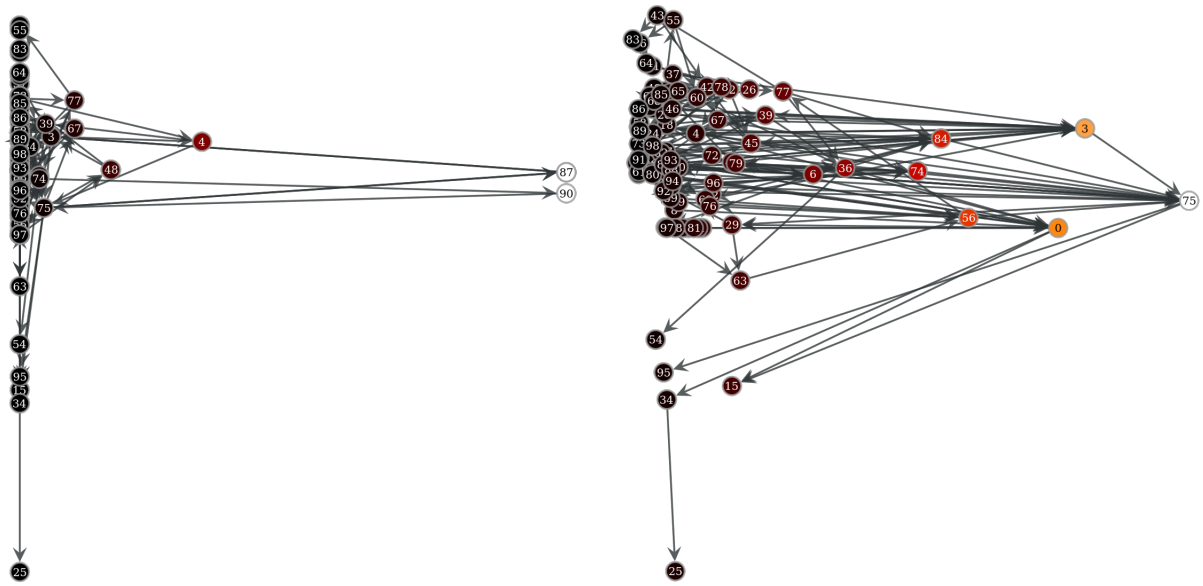
When considering closeness centrality, the graph shows that almost all vertices have a similar closeness value in comparison to their betweenness. This is because the closeness centrality analyses the importance of the words within their local clusters rather than the graph as a whole. Meaning that words with high closeness values are key connections in their relative clusters, in other words the sentences that they are part of. However vertex 34 (the word “great”) is an outlier and by further analysis, vertex 34 is the only connection between vertex 0 and 25. Also vertex 25 has the highest trophic level and vertex 0 has a low trophic level but a higher degree. Consequently, the closeness value for 34 is an extreme due to the fact that it is the only predecessor of vertex 25. Thus meaning that vertex 34 is the only local bridge in its sentence giving it an extreme closeness whereas other vertices contains multiple connections.

In conclusion, based on the story corpus, betweenness finds the words most commonly used as connectors in a sentence and closeness finds the words that are likely to isolate later vertices.

Finally, local clustering coefficients and page ranks of the vertices within the English word graph are presented similarly as before. Local clustering coefficients in Figure 4.3a and page ranks in Figure 4.3b along with their respective table extracts shown before in Tables 4.3c and 4.3d.

Immediately, the local clustering graph shows very few vertices who has a high local clustering, the main ones being vertices 87 and 90 (“water” and “when” respectively). However, there does not exist a clear relationship between these words other than neighbours of these words have have a high degree or importance in the graph. Important words such as “and” and “a”. On the other hand this is only for the English version of the story so other languages may lead to different results. The page rank of each vertex shows the importance of each word beyond their direct contact. Essentially it has elements of both closeness and betweenness centrality which the graph reinforces since it is visually similar to the betweenness English graph.

In conclusion, when analysing the English language, trophic levels have provided a naturally flow of data presentation from top to bottom in the various graphs generated. Where some sections of the graph are grammatically correct when following their flows. Betweenness centrality and page rank both identified the words of most importance with respects to their neighbours. Closeness centrality also identified words of key importance but at a local level. Finally local clustering coefficients do not provide sufficient benefits when visualising the dataset in the English language. Therefore the results generated based on the English version can be extrapolated



(a) (b)

Figure 4.3: Instead of centrality values like before, (a) local clustering coefficients and (b) page ranks are used for the x-axis. The y values remains the same representing the trophic levels.

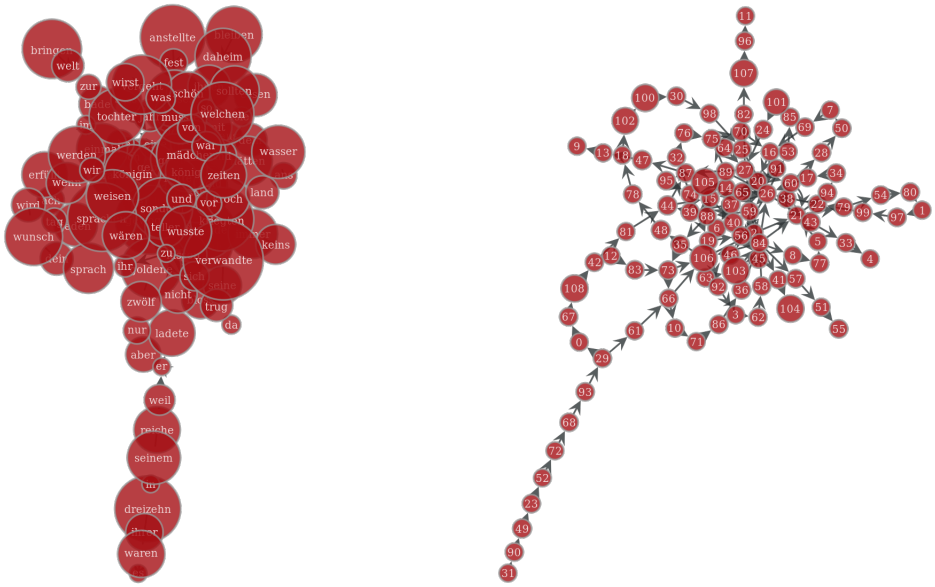
to represent the English language. Which is achieved through the identification of the words position in the structure of a sentence and its relative importance. This could also be used for languages that have a similar grammatical structure to English. Now we move onto the analysis of a different translation of the story corpus, German.

4.3.2 German

German is also a Germanic language under the Indo-European language family, same as English. However whilst modern English no longer uses the inflectional case system in grammar, the German language still does [67]. So as well as the parts of speech in English, German words can be divided into two groups, the words which are *inflectable* and *uninflectable*. If a word is inflectable then their form changes based on the context that the word is used in. These include the three genders for the words (masculine, feminine and neutral), the four cases (nominative, accusative, genitive and dative) and the number (singular or plural). Uninflectable words are known as *modal particles* and mainly used to highlight an emotion of the sentence in spoken language. Therefore the German language is considerably more varied compared to the English language. Meaning that, in theory, the dataset would contain more unique words overall. Which is proven to be true as the dataset based

on the story corpus for German contains 109 words whilst English had only 99.

Expectations of the graph properties for the German language is that there would be more unique words of higher importance based on the different genders for words and particles. Word graphs for the German version of the story corpus are generated and shown in Figure 4.4. Similarly as before, each number references the same word in its position.



(a) (b)

Figure 4.4: The German (a) word graph and (b) numbered equivalent of the word graph generated from the German translation of the "Sleeping Beauty" corpus.

Vertex	Word	Count	TL	BC	CC	LC	PR
26	ein	7	7.42	10.00	2.19	0.66	10.00
84	und	7	7.37	9.90	2.51	0.77	8.35
21	die	4	7.43	3.37	1.56	0.00	4.49
27	eine	3	7.61	2.93	1.73	0.00	4.17
58	königin	3	7.40	2.84	1.78	0.00	2.67
106	zu	3	6.86	4.38	1.65	4.00	2.77
15	das	2	7.32	2.42	1.94	0.00	2.75
16	dass	2	7.26	1.55	1.84	0.00	2.91
19	dem	2	7.53	1.69	1.54	0.00	2.59
20	der	2	7.02	0.76	1.95	0.00	1.72

Table 4.4: Top 10 words with the highest frequency in the German dataset including values of other graph properties.

Table 4.4 shows the most common ten words in the German dataset along with each graph property value. The translation to English for each word in the order of the table is “and”, “a” (Masculine), “the” (Feminine), “to”, “a” (Feminine), “Queen”, “King”, “not”, “himself/herself/itself” (dependent on the pronoun this refers to) and “the” (Neutral). As expected when comparing to the English dataset most words appear in both translations as the top ten, in particular the top three. Which we recall was “and”, “the” and “a” for the English dataset. However rather than a cumulative count of “the” in English, the German translation has multiple versions. This may counteract each others importance and bring they values lower. By retaining the inflectable changes in grammar, the average count is lower for the German translation. Additionally the range of frequencies has also decreased from 9-1 to 7-1. For example, “the” in English was split into “die”, “der” and “das” in German. By noting these key differences, the graphical properties will be analysed.

Vertex	Word	Count	TL
11	bringen	1	10.00
9	bleiben	1	9.50
96	welchen	1	9.40
4	anstellte	1	9.21
55	keins	1	9.16
13	daheim	1	8.90
107	zur	1	8.80
33	fest	1	8.62
51	immer	1	8.57
104	wusste	1	8.57

(a)

Vertex	Word	Count	TL
31	es	1	0.00
90	waren	1	0.60
49	ihrer	1	1.20
23	dreizehn	1	1.79
52	in	1	2.39
72	seinem	1	2.99
68	reiche	1	3.59
93	wasser	1	4.19
29	er	2	4.78
12	da	1	5.18

(b)

Table 4.5: Tables for (a) top 10 and (b) bottom 10 trophic levels of the German dataset along with other graph values.

Analysis of the trophic levels demonstrates that most words congregate at around 7.3. This can be more evidently seen in the graphical representations on Figures 4.5 and 4.6 later on. From the same graphs, there is a unique path from vertex 31 to 29 which is equivalent to the translation “Es waren ihrer dreizehn in seinem Reiche, weil er”. This path is grammatically correct and represents the beginning of a sentence belonging to the German story corpus. Furthermore this portion holds only unique words, hence has not been influenced by other vertices so demonstrates a clear hierarchical structure. Which is the reason the words used are of low trophic levels. Whereas most other sentences share words which causes the conglomeration nearer 7.3. Consequently the German language has more options for word choices which means the increased probability of unique sentences.

Trophic coherence has also benefited from the uniqueness of word choices since for the German graph, trophic coherence is calculated to be 0.29. This is larger than the English graph which was 0.09. Therefore the German graph has a clearer hierarchical

structure compared to the English graph with clearer levels as demonstrated in further graphs below when analysing other properties.

Vertex	Word	BC
26	ein	10.00
84	und	9.90
57	könig	5.37
106	zu	4.38
21	die	3.37
27	eine	2.93
58	königin	2.84
15	das	2.42
66	nicht	2.41
63	lassen	1.88

(a)

Vertex	Word	CC
97	welt	10.00
13	daheim	10.00
33	fest	10.00
51	immer	10.00
41	gewogen	10.00
107	zur	6.67
43	grosses	6.67
47	ihnen	6.67
59	kriegten	6.67
82	tochter	5.00

(b)

Vertex	Word	LC
57	könig	10.00
73	sich	10.00
66	nicht	10.00
106	zu	4.00
84	und	0.77
26	ein	0.66
97	welt	0.00
13	daheim	0.00
33	fest	0.00
51	immer	0.00

(c)

Vertex	Word	PR
26	ein	10.00
84	und	8.35
21	die	4.49
27	eine	4.17
16	dass	2.91
74	sie	2.81
44	hatte	2.81
106	zu	2.77
15	das	2.75
58	königin	2.67

(d)

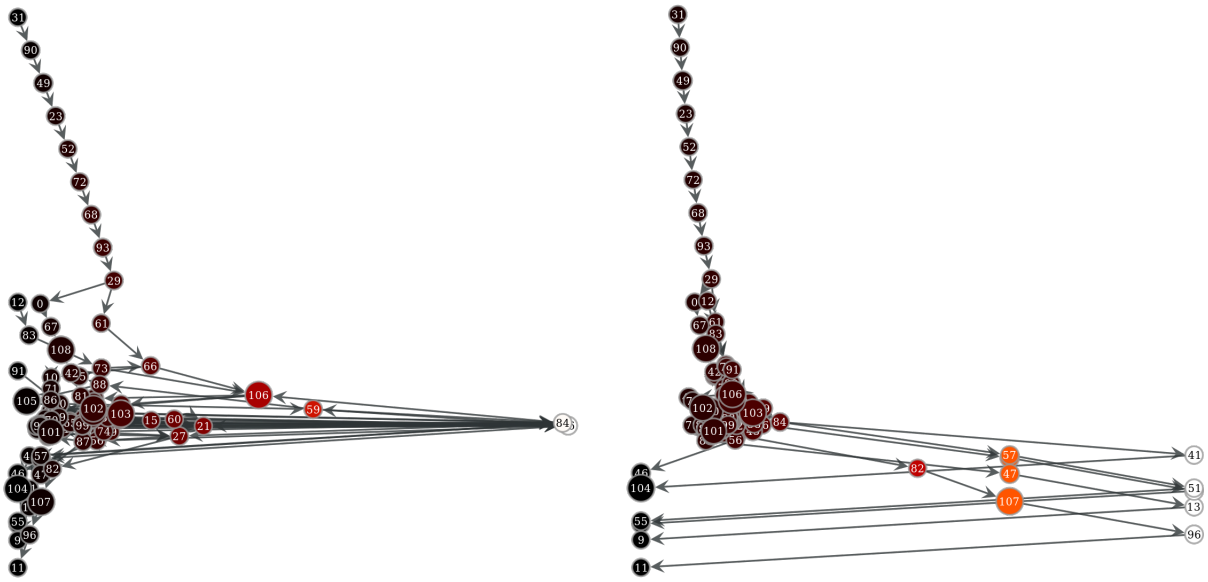
Table 4.6: Partial extracts of the German table data ordered by their (a) betweenness centrality values, (b) closeness centrality values, (c) local clustering coefficients and (d) page ranks.

Entirety of the data presented in table form can be seen in Appendix **??**. Table 4.6 (a)-(d) shows an extract of the top 10 complex values in order of highest to lowest for each property.

Assessing the visual graphs (Figures 4.5a and 4.5b), we identify vertex 84 and 26 who have the highest betweenness values. A correlation to the frequencies of the words in the German Story Corpus can be seen through these vertices. Additionally, the same correlation was also true in the English version because the vertices in both versions referred to the words that are most commonly used as bridges or links in sentences. In the German case, the words are “ein” and “und”.

With closeness centrality, it is a measure of influence on nearby words, the graph in Figure 4.5b shows that the vertices 97, 13, 33, 51 and 41 have the largest closeness values. These all correspond to the second to last word of each sentence apart from the first sentence which has the word “Kind” as the second to last word. However “Kind” is also used elsewhere meaning that its closeness value was influenced through other connections. The vertices of high closeness are unique and essential as a bridge in its nearby words. Otherwise the neighbours of these vertices are likely to be isolated. Note that the orange vertices in Figure 4.5b are the unique words predecessors to the vertices of high closeness and the last word of every sentence always has a value of 0.

Therefore betweenness identifies the words of key importance that are used most commonly as connectors. Meanwhile closeness identifies the words most likely to isolate vertices of the graph when following the sentence flow, i.e., breaks up the rest of the sentence into unique words. Observe that the same pattern was seen in the English version but was set aside due to there being only one vertex of high closeness.



(a) (b)

Figure 4.5: Graphs with the x positioning based on (a) betweenness and (b) closeness centrality. The y is based on trophic levels.

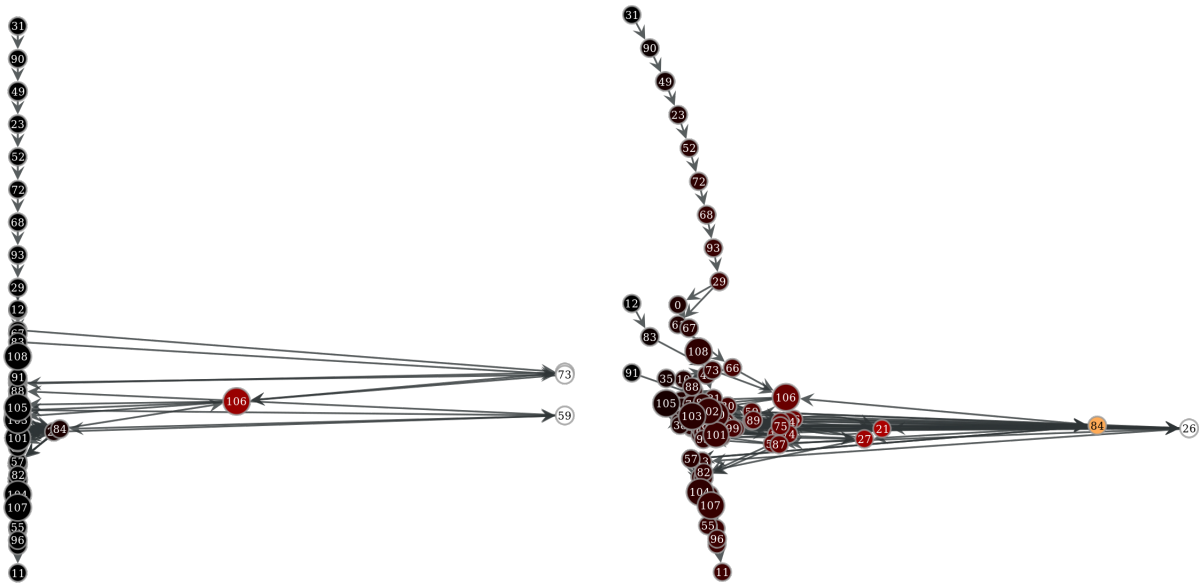
However the German version shows a correlation of local importance to unique words which means that the English version may contain the same correlation.

Nothing prominent can be derived by studying the local clustering coefficients (see Figure 4.6a) because almost all vertices hold a local clustering coefficient of 0 apart from six vertices. Three of which are the vertices 57, 73 and 66 which corresponds to the words “könig”, “sich” and “nicht” respectively. These are not unique words in the German story corpus and no other vertices demonstrate anything unique with its local coefficient. On the other hand, vertices with a high page rank (see Figure 4.6b) correlate to either conjunctions, vertex 84 (“und”), or words that accompany other words like pronouns or articles, vertices 26 (“ein”), 27 (“eine”), etc. Therefore high page rank relates to key words that are used frequently in German, which remains true when extended to the German language as a whole.

In conclusion, similar results were seen when analysing the graphical properties based on the German and English translations. With German having a clearer visualisation and stronger correlation compared to the English language due the the uniqueness of inflectional words.

4.3.3 French

English and German are Germanic languages under the Indo-European family. Instead of another Germanic language, we study a different branch under the Indo-



(a) (b)

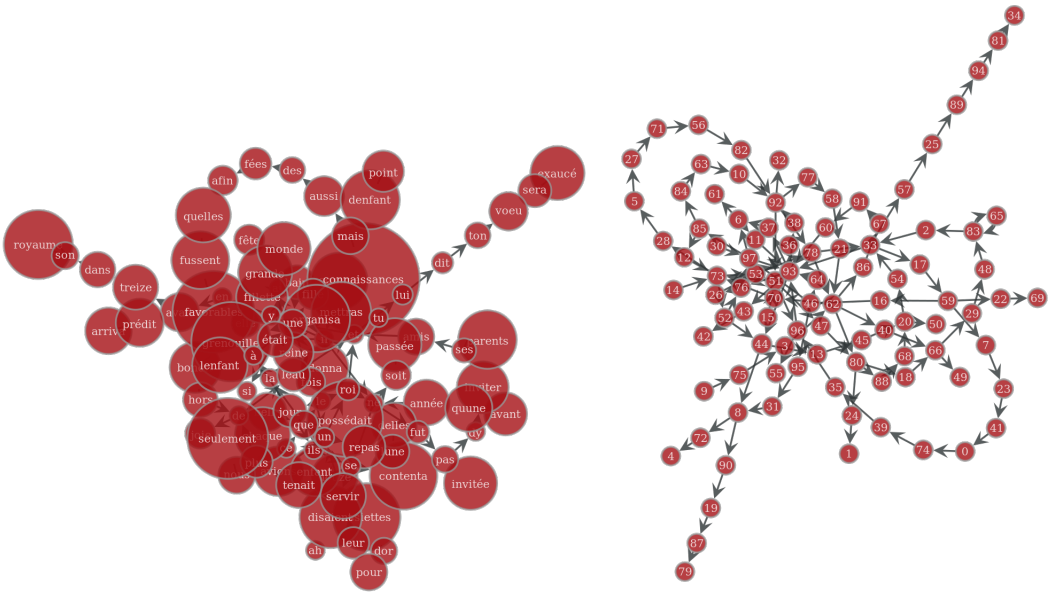
Figure 4.6: Displays the (a) local clustering and (b) page rank on the x-axis instead of the centrality values. The trophic levels for y remains the same.

European family. The language being French which lies under the Italic branch of the same family. Similarly to German, French contains the same parts of speech as English but also inflectional words. So words can be inflected by number (singular or plural), gender, person, case, aspect and mood. However whilst German has three genders (Masculine, Feminine and Neutral), French [68] only uses two, masculine and feminine.

The story corpus is translated into the French version so graphs can be generated off this. Where the initial word graphs are shown in Figure 4.7.

Words of highest frequency are shown in the extract Table 4.7 where the majority of words seen are inflectable and used as a way to give more information. Such as the particle of vertex 93 meaning "a/an" or vertex 46 meaning "he/it". So these would be used more frequently and it is generally true for the French language.

In relation to the story corpus, French has a high trophic coherence which is calculated to be 0.36 which is higher than both English and German. Even though it may not be close to 1, in comparison to previous languages, a clearer level structure can be identified for the French language. Top 10 trophic levels can be seen in Table 4.8a where the high values represent the words nearer to the end of a sentence. All top 10 are within the last four words of sentences in the story corpus. However the predecessors of the words have an impact on their trophic level. If the predecessors of a word is involved in other sentences at a earlier stage then the predecessors trophic value will be lowered. Subsequently lowering the trophic levels of the neighbourhood



(a) (b)

Figure 4.7: The French (a) word graph and (b) numbered equivalent of the word graph generated from the French translation of the "Sleeping Beauty" corpus.

Vertex	Word	Count	TL	BC	CC	LC	PR
93	une	6	3.83	10.00	1.81	0.91	10.00
46	il	5	2.33	1.86	1.86	0.00	1.26
62	ne	5	3.03	5.16	1.64	0.00	5.87
73	que	4	3.08	4.31	1.85	2.86	4.65
92	un	4	3.26	8.19	1.94	0.00	6.58
33	et	3	3.93	8.75	1.90	0.00	6.30
51	jour	3	3.16	3.40	1.82	4.00	3.73
52	la	3	3.17	1.19	1.54	0.00	1.50
59	mais	3	2.32	3.69	1.62	0.00	2.24
76	reine	3	3.31	1.92	1.68	10.00	2.43

Table 4.7: Top 10 words with the highest frequency in the French translation of the corpus. Shown in table format with other graphical properties.

of the word. This can be seen by vertex 49 who has a trophic level of 5.08 even though it is last in its sentence.

Tables 4.9 (a)-(d) shows an top 10 extract of Table **??** in the order of their corresponding graph property value.

Vertices with high betweenness will have many other vertices that must connect to it in order to reach other vertices within their sentence. As mentioned in previous language analysis, betweenness helps identify the key bridges in the sentences by

Vertex	Word	Count	TL
34	exaucé	1	10.00
79	royaume	1	9.25
81	sera	1	8.99
87	son	1	8.23
94	voeu	1	7.98
4	arriva	1	7.22
19	dans	1	7.22
89	ton	1	6.97
72	prédit	1	6.21
90	treize	1	6.21

(a)

Vertex	Word	Count	TL
9	avant	1	0.00
75	quune	1	1.01
30	elle	1	1.95
3	année	1	2.02
14	ce	1	2.07
15	chaque	1	2.15
59	mais	3	2.32
16	comme	1	2.33
46	il	5	2.33
85	si	2	2.46

(b)

Table 4.8: Trophic levels, (a) top 10 and (b) bottom 10 in table format including other values.

Vertex	Word	BC
93	une	10.00
33	et	8.75
92	un	8.19
78	roi	6.73
38	fois	5.54
62	ne	5.16
73	que	4.31
59	mais	3.69
17	connaissances	3.62
51	jour	3.40

(a)

Vertex	Word	CC
87	son	10.00
81	sera	10.00
72	prédit	10.00
22	denfant	10.00
24	disaient	10.00
43	grande	10.00
19	dans	6.67
94	voeu	6.67
90	treize	5.00
89	ton	5.00

(b)

Vertex	Word	LC
53	le	10.00
76	reine	10.00
51	jour	4.00
97	était	4.00
73	que	2.86
93	une	0.91
87	son	0.00
81	sera	0.00
72	prédit	0.00
22	denfant	0.00

(c)

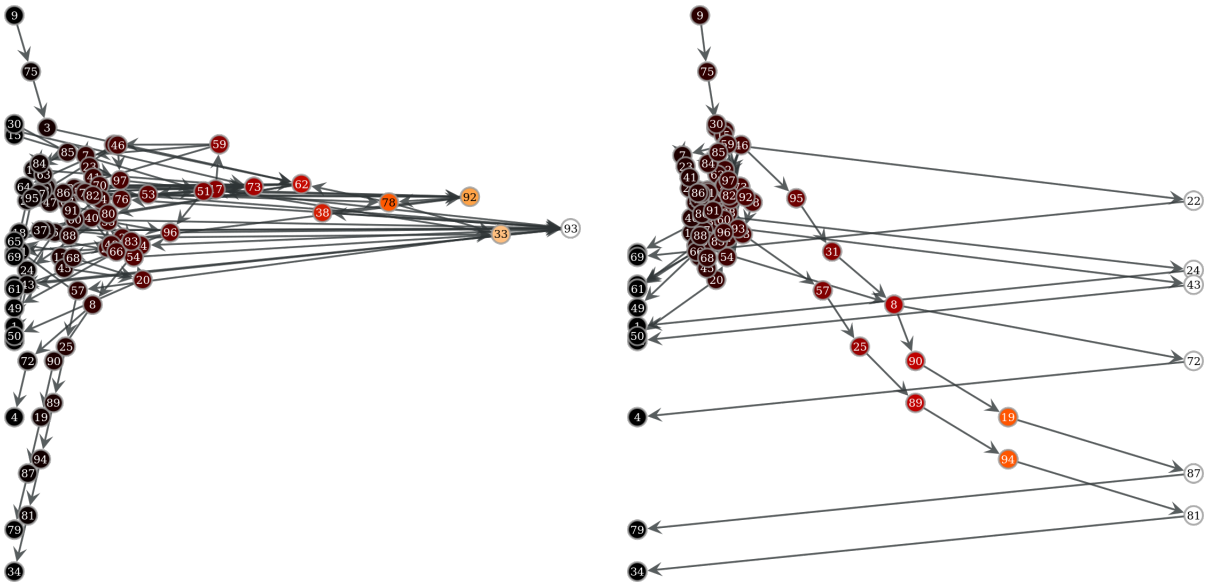
Vertex	Word	PR
93	une	10.00
92	un	6.58
33	et	6.30
62	ne	5.87
83	ses	4.73
73	que	4.65
96	à	4.05
20	de	3.78
51	jour	3.73
66	pas	3.19

(d)

Table 4.9: Partial extracts of the French table data ordered by their (a) betweenness centrality values, (b) closeness centrality values, (c) local clustering coefficients and (d) page ranks.

finding the vertices which control the majority of the flow of data. Furthermore the betweenness coincides with the frequency. On the other hand there are vertices such as vertex 78 (“roi”) and vertex 38 (“fois”) who have a betweenness value of 6.73 and 5.54 (see Table 4.9a)respectively but only a count of 2 and 1. By further analysis, the reason they have high betweenness, is that they are used with a predecessor and successor of higher betweenness value, i.e. “un roi et” and “une fois un”. Consequently causing its own betweenness value to be inflated because ”un”, ”et” and ”une” must pass through “roi” or “fois” to complete their word pathing. Therefore, if vertices have a low count but are connected to other vertices of high betweenness, then their values will relatively increased.

As discovered in the German analysis, closeness centrality identifies the vertices



(a) (b)

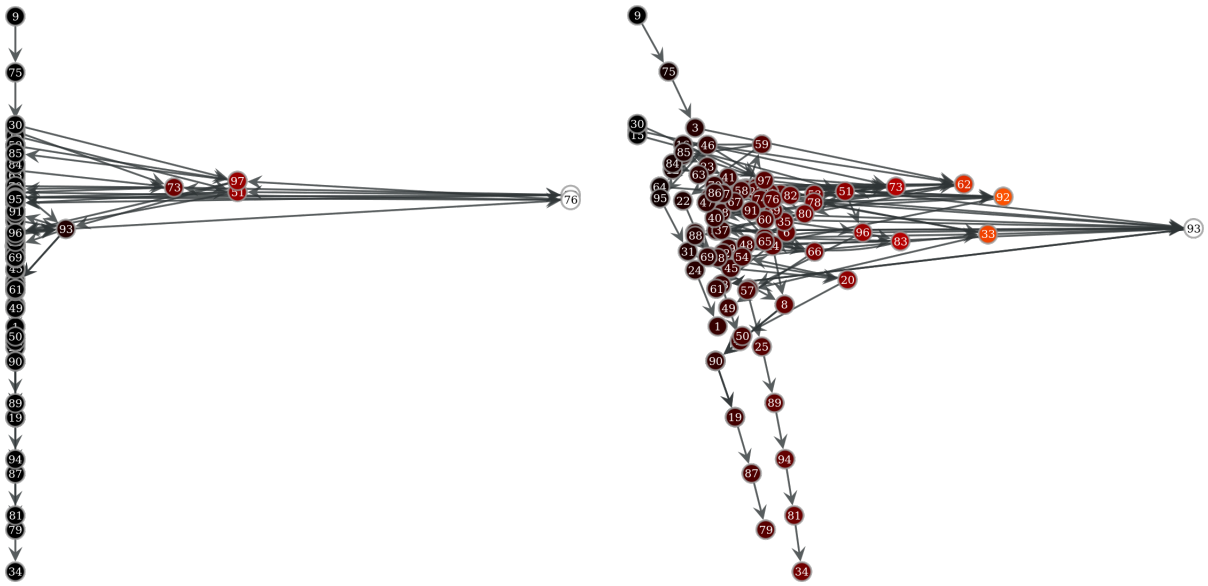
Figure 4.8: Graphs with (a) betweenness and (b) closeness centrality values displayed on the x-axis based on the French numbered word graph. The y-axis being the trophic levels.

most likely divide the graph from their directed path, i.e. the words most likely to isolate the remaining sentence. Otherwise vertices are involved in a diverse range of sentences causing a high degree where the formation of vertex conglomerates can be seen (see Figure 4.8b where a clustering of vertices is shown in the upper left quadrant). Studying the graph in Figure 4.8b, French words have more vertices with a variety of high closeness values. The graph shows clearly that vertices of high closeness would cause isolation for further vertices along its path. According to the visualisations, French is the most unique language out of the three languages seen so far as there is a higher number of clear divisions in the closeness values.

Finally the local clustering coefficients and page rank. Only two vertices have a local clustering coefficient of 10 which are vertex 53 “le” and vertex 76 “reine” (see Figure 4.9a and Table 4.9c). In the context of French, nothing particular can be derived. However when studying the graphs of local clustering in various languages, the vertices with a local clustering value are never an start or end to a branch.

Page ranks for the French story corpus measures the influence the vertices have on one another other than for their immediate neighbours. This is why vertex 46 “il” has a low page rank value of 1.26 as it is located at the start of the sentence so does not have enough unique vertices encompassing it to be influential.

In conclusion, French has similar results in comparison to English and German which makes sense as they are a part of the same language family. So to see a



(a) (b)

Figure 4.9: Graphs displaying the (a) local clustering and (b) page rank on the x-axis instead of the centrality values whilst keeping their y positions.

different perspective, the study of other language families are undertaken such as the Japonic language family.

4.4 Japonic

The Japonic language family is the protolanguage for Japanese and Ryūkyūan languages [69] so it is a relatively small language family compared to other language families. Translating the corpus into Japanese gives us a new dataset to use for graph property analysis which will be studied in comparison to the previous language family.

4.4.1 Japanese

To give a larger variety of languages, Japanese is studied as it uses vastly different grammar compared to the languages seen so far. Japanese has only five linguistic word classes which includes nouns, verbal nouns, nominal adjectives, verbs and adjectives. The order in which the words are structured is different to Indo-European languages since Japanese uses SOV (Subject-Object-Verb) compared to languages like German and English which uses SVO (Subject-Verb-Object). Furthermore Japanese does not rely heavily on grammatical number or gender and is

more focused on their system of honorifics which indicates the speaker, listener or person of reference. Therefore, even if Japanese contains words borrowed from other languages (these are referred to as *loanwords* [70]), their grammar is vastly different.

We follow the same process as before and input the Japanese Story corpus into my program to produce the basic graphs. For simplicity, we show the numbered version of the word graph (see Figure 4.10) with the words they correspond to in the full table of data (See Table **??**).

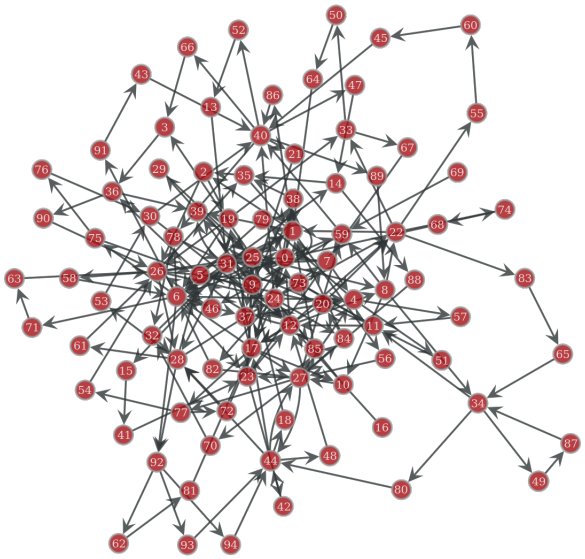


Figure 4.10: The Japanese word graph generated from the Japanese translation of the story corpus.

Vertex	Word	Count	TL	BC	CC	LC	PR
17	た	14	5.73	8.12	9.50	1.06	6.12
12	し	13	6.91	8.19	8.73	0.30	6.89
31	ま	13	7.90	10.00	9.31	0.77	8.66
1	い	11	6.73	7.46	8.51	0.99	7.72
6	が	11	5.86	9.97	8.68	0.59	7.98
25	な	11	7.15	9.40	9.56	0.66	10.00
24	と	9	4.80	8.35	10.00	0.77	5.64
27	の	9	4.74	6.24	8.19	0.33	5.71
26	に	7	5.98	6.74	9.56	0.77	7.81
0	あ	6	6.40	4.02	9.08	1.56	6.04

Table 4.10: Top 10 words with the highest frequency in the Japanese translation of the corpus. Shown in table format with other graphical properties.

Out of the previous languages, the larger frequency of a words appearance was

9. With Japanese, there are 6 words with 11 or more appearances (see Table 4.10), largest being vertex 17 with 14 appearances. Vertex 17 represents a form of conjugation which means that their use depends on the inflections of its associated words. Generally vertex 17 is most commonly used to signify the past tense. So rather than having a separate word for different tenses, Japanese uses specific accompanying words which leads to their frequency being significantly higher. Additional examples such as vertex 12 who has a large frequency and its associated word is used to empathically add more information (an empathic and). Vertex 6 is a particle that means "but" or indicates the sentence subject. Therefore, Japanese words have a heavy reliability on each other to define their overall meaning which is demonstrated by the many edges signifying these relationships.

Vertex	Word	Count	TL
3	え	2	10.00
39	れ	5	8.47
66	抑	1	8.24
35	よ	3	8.19
11	さ	5	8.19
38	る	3	8.00
36	ら	3	7.93
31	ま	13	7.90
29	ば	2	7.81
51	后	2	7.68

(a)

Vertex	Word	Count	TL
69	昔	1	0.00
16	そ	1	0.23
72	様	2	2.73
82	蛙	2	2.91
75	水	2	3.38
92	1	3	3.57
46	供	3	3.94
62	年	1	4.29
93	2	1	4.47
94	3	1	4.47

(b)

Table 4.11: Tables showing graph values ordered by (a) top 10 trophic levels and (b) bottom 10 trophic levels.

Due to the graph having an increased amount of bidirectional edges such as the edge between vertex 25 and 5. As well as an increased amount of self loops, the trophic incoherence is very high in comparison to previous languages. Thus, the trophic coherence for the Japanese extract is only 0.04. Caused by the ambiguity of hierarchical structure with bidirectional edges and self loops. An example of a self loop is vertex 0 which is part of the word "everyday". The subsequent repetition of a word that creates a different meaning is known as *reduplication*.

Unfortunately without a full understanding of the Japanese language, direct analysis of each word is difficult. This is because of the multitude of meanings each word has depending on the context. Therefore, the trophic levels can only provide a generic flow of text from lowest trophic levels (see Table 4.11b) as sentence starters to highest levels (see Table **??**) as sentence enders.

As with previous languages, Tables 4.12 (a)-(d) are ordered extracts from the main table in Appendix **??**.

The graphs in the centrality Figure 4.11 shows that the words are more evenly distributed unlike the Indo-European languages. Which correlates to the importance

Vertex	Word	BC
31	ま	10.00
6	が	9.97
25	な	9.40
24	と	8.35
12	し	8.19
40	を	8.12
17	た	8.12
1	い	7.46
22	て	6.92
26	に	6.74

(a)

Vertex	Word	CC
24	と	10.00
25	な	9.56
26	に	9.56
17	た	9.50
31	ま	9.31
0	あ	9.08
12	し	8.73
6	が	8.68
1	い	8.51
28	は	8.35

(b)

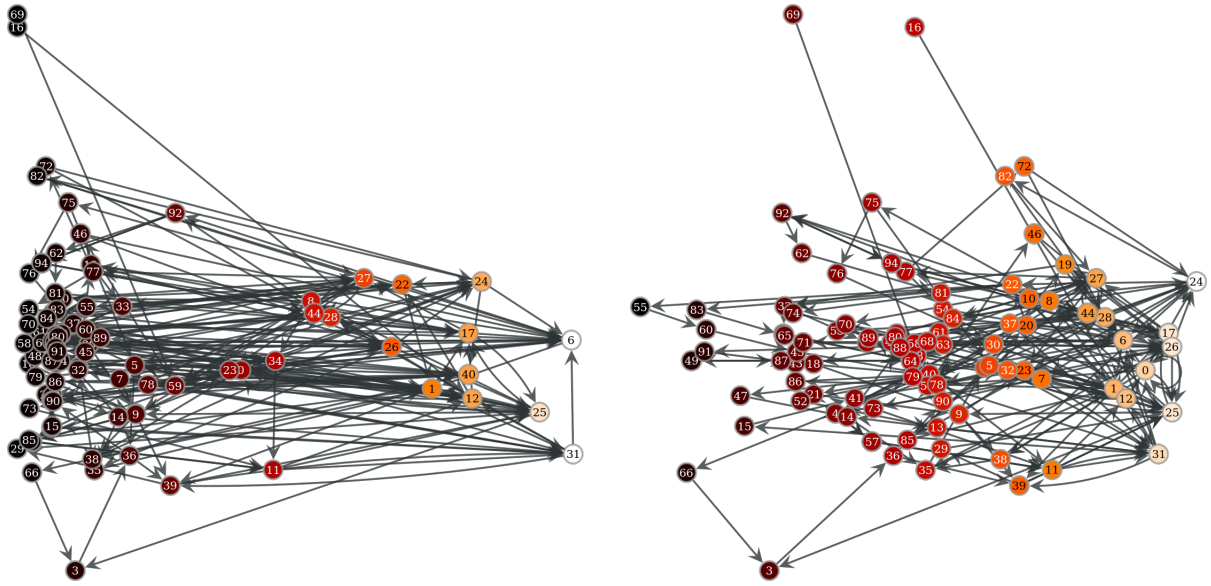
Vertex	Word	LC
29	ば	10.00
61	家	10.00
87	達	10.00
49	友	10.00
2	う	5.00
37	り	4.00
72	様	3.33
32	み	3.33
82	蛙	3.33
78	産	3.33

(c)

Vertex	Word	PR
25	な	10.00
31	ま	8.66
6	が	7.98
26	に	7.81
1	い	7.72
12	し	6.89
40	を	6.66
17	た	6.12
0	あ	6.04
27	の	5.71

(d)

Table 4.12: Partial extracts of the Japanese table data ordered by their (a) betweenness centrality values, (b) closeness centrality values, (c) local clustering coefficients and (d) page ranks.



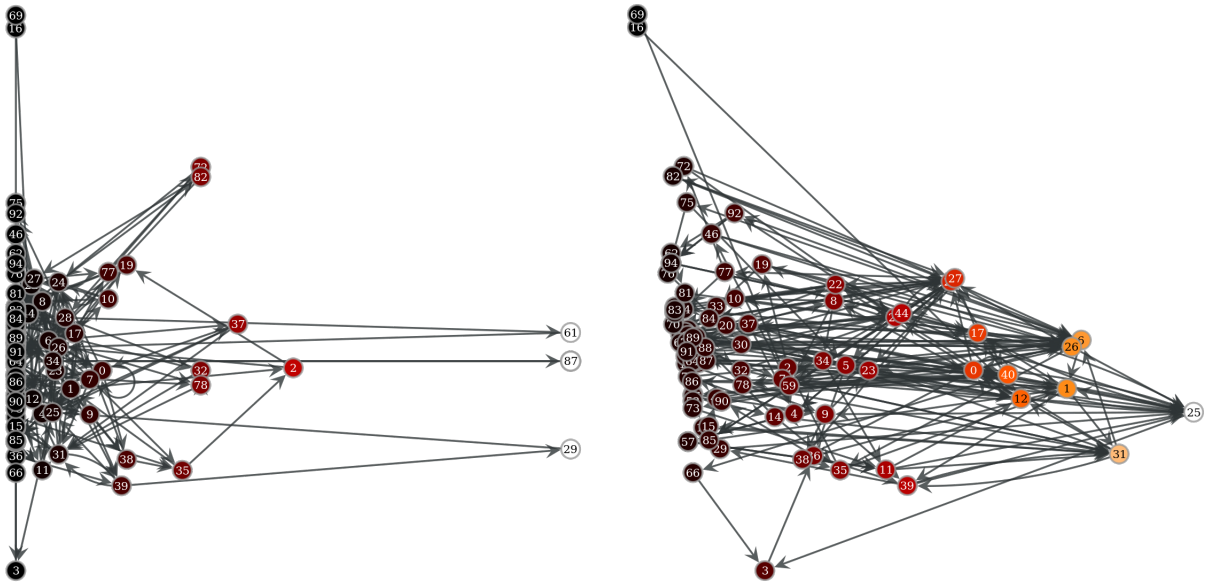
(a) (b)

Figure 4.11: Positions of the Japanese numbered graph but with trophic levels on the y-axis and (a) betweenness or (b) closeness on the x-axis.

of the Japanese characters nature of being reliant on one another for their meaning and demonstrates the complexities of the language visually. The basic correlations from other languages still hold here. Although further detail cannot be determined by the visualisation other than importance.

Japanese's closeness centrality values also has a larger average compared to the Indo-European languages. Instead of looking at the vertices in relation to the whole graph, closeness focusses on the closer clusters. Hence provides a similar shape betweenness but shifted right to emphasise the zooming in on the vertices. There are no vertices that stand out so no correlations can be easily derived.

Therefore only a general correlation can be demonstrated through the centrality graphs. Clearer results can be gathered with Japanese fluency.



(a) Displays the local clustering coefficient against the trophic levels.

(b) Displays the page rank against the trophic levels.

Figure 4.12: The x-axis showing (a) local clustering and (b) page rank instead of the centrality values. The y values remain.

Local clustering coefficients (see Figure 4.12a) measures how close the vertex is part of a unique directed triangle, i.e. a directed 3 cycle. The vertices 29, 87 and 61 are vertices who are involved in such triangles where they each have a unique in and out edge to other vertices. Lower values means they are less likely to be involved in unique triangles. Thus local clustering can identify certain triples in the Japanese texts that would have unique meanings.

Page rank identifies the vertices who have an influence to other vertices nearby but not immediate neighbours like what closeness does. However nothing further can be correlated based on this graph (see Figure 4.12b).

In conclusion the way the Japanese story corpus is split into the database (character by character), only general correlations can be determined based upon the graphs and their properties. Stronger links as to what exact words have importance cannot be obtained here without the full understanding of the language. On the

other hand, we see that in general the graphs are different to the ones based on the Indo-European languages which demonstrates the languages differences. To see this further we analyse another language and its language family.

4.5 Sino-Tibetan

Sino-Tibetan is the final language family in which we analyse and contains over 400 languages. Two groups of languages that are successors of this family are the Tibeto-Burman language group and Chinese language group. Focus on the Chinese language will be taken to translate the story corpus into. Mandarin Chinese is chosen here since it is the 2nd most spoken language after English.

4.5.1 Chinese

The main branches of Chinese are Mandarin or Cantonese with Mandarin considered to be the standard and is the official language of China. Mandarin is also the language in which the corpus was translated into but will refer to this as Chinese for simplicity. Chinese has a SVO (subject-verb-object) sentence structure like with English. However the language is made up of syllables [71] rather than words. Syllables come with 3 parts, an initial consonant, the tone, and a final. These parts determines the meaning to each syllable and instead of writing their unique characters, they can be written using Pinyin. Pinyin is the standard system for romanised spelling and a useful tool in learning Chinese and their pronunciations. A simple word like "hello" is "nǐ hǎo" in Pinyin where "n" and "h" are initials, "i" and "ao" are finals and the accents on the letters "i" and "a" are the tone for the syllables. The tone matches the pitch of the syllable and leads to different meanings for the each word. Although each syllable in Chinese has its own definitions, they can be combined to form a different word. E.g. In Pinyin, "dì" means earth but "dì tú" means map.

Similarly to Japanese, Chinese has very few inflectional characters and utilises other particles to accompany the syllables so various verbal aspects can be expressed. Another similarity is the use of reduplication where a syllable is repeated to give a different meaning. Parts of speech for Chinese is split into two major categories, the *function* words and the *content* words. The function words contains nouns, pronouns, verbs, auxiliary verbs, adjectives, number words, measure words, interjections and onomatopoeias. Function words contain conjunctions, prepositions and particles. Furthermore, each subgroup mentioned can have further branches dependent on the context that they are used in. For example, nouns can be proper nouns, location nouns, place nouns and time nouns in the Chinese language.

Therefore Chinese contains a larger amount of unique syllables compared to the words in the English language. Also syllables have a reliability factor and depends on the context of use (like when studying Japanese). Keeping this in mind, we generate

and analyse the graphs beginning with seeing the basic word graph generated for the Chinese translation of the story corpus, seen in Figure **??** (shown with integer labelling to reference each word of the entire table, Table ?? in the Appendix).

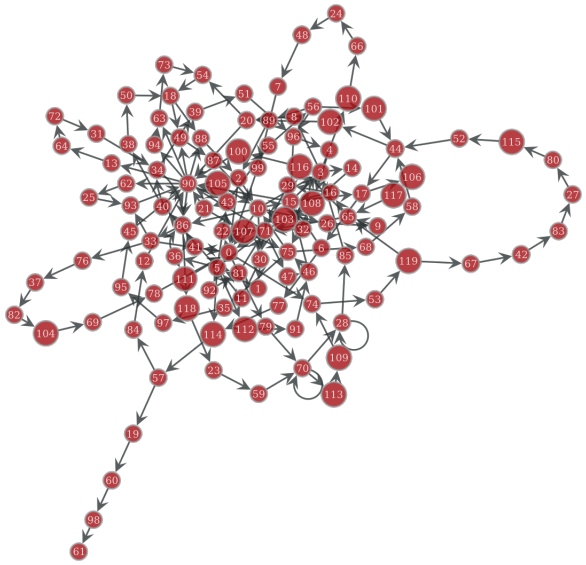


Figure 4.13: The Chinese word graph generated from the Chinese translation of the story corpus.

Vertex	Word	Count	TL	BC	CC	LC	PR
90	的	11	2.65	10.00	1.96	0.17	10.00
0	一	9	2.26	9.87	2.09	0.38	8.54
5	个	8	1.74	4.99	1.96	0.44	4.55
10	了	7	1.81	6.15	2.42	0.36	7.85
15	他	7	2.02	5.15	2.19	0.56	4.61
71	有	7	1.86	6.15	2.37	0.73	6.18
43	女	6	1.75	2.44	1.59	1.67	1.79
86	王	6	2.35	5.64	1.75	0.00	6.03
108	邀	4	2.68	3.02	1.76	0.67	3.90
2	下	3	2.21	3.60	2.13	1.67	3.67

Table 4.13: Top 10 words with the highest frequency in the Chinese translation of the corpus. Shown in table format with other graphical properties.

The syllables with the highest count (see Table 4.13) is vertex 90 which is “de” in Pinyin. This is a particle so has a grammatical meaning with different functions to denote possession. Also can be accompanied by other parts of speech. Other syllables with high counts include vertex 71, “yǒu”, vertex 5, “gè” and vertex 0, “yī”. These vertices all have various meanings depending on their use, “yǒu” can

mean “have”, “is” and “are”, “gè” can mean “this”, “that”, the party of a specific size or a classifier for people or objects. Vertex 0, “yī”, can mean “a” or is a part of a number like “1” in the English language. Therefore the syllables with a high count represents the words of varied meanings which are important in the sentence structure or identifiers for words with different genders.

Vertex	Word	Count	TL
61	恼	1	10.00
98	苦	1	8.19
60	心	1	6.39
19	伤	1	4.58
54	就	2	4.17
73	望	1	4.05
63	愿	2	3.93
49	实	2	3.57
18	会	3	3.56
88	生	2	3.41

(a)

Vertex	Word	Count	TL
105	过	1	0.00
56	师	3	0.52
119	鱼	2	0.90
16	以	2	1.05
67	把	1	1.11
55	巫	3	1.14
42	头	1	1.33
33	后	3	1.33
89	留	1	1.37
6	为	2	1.44

(b)

Table 4.14: Tables to show the (a) top 10 trophic level and (b) the bottom 10 along with other relative data.

An increased amount of reduplications and bidirectional edges appear in the Chinese graph. Like before with the Japanese data, trophic levels do not demonstrate a clear visual hierarchy. Meaning that the trophic coherence is low and is calculated to be 0.09 based on the Chinese translation of the story corpus. Therefore average positions of each syllable within a sentence can still be represented but with minor flexibility. The trophic levels go from low to high in relation to the sentence flow with Table 4.14a showing the top 10 syllables nearer the start of a sentence and Table 4.14b showing the top 10 syllables for the end. Grammatically correct can be seen by unique downward paths in the graph visualisations such as the path from vertex 19 to vertex 61 (see Figure 4.14 or 4.15).

The top 10 syllables for each complex graph property is seen here where the full table is in Appendix **??**.

Betweenness centrality in the Chinese translation Story corpus (see figure 4.14a) identified the syllables vertex 90 and vertex 10 (both are particles) to be of high importance. Re-emphasizing the fact that they are the syllables that act most like bridges and needed to derive further meaning. Hence betweenness relate closely to the counts of the syllables discussed earlier. The same correlation is achieved in previously discussed languages so we can conclude that this correlation of importance holds for the majority of languages.

As mentioned in the German analysis, closeness measures the importance of words directly around them. In other words, the vertices vital in the paths structure (controlling the flow of data). Therefore, demonstrated by Figure 4.14, vertex 98

Vertex	Word	BC
90	的	10.00
0	一	9.87
71	有	6.15
10	了	6.15
86	王	5.64
15	他	5.15
5	个	4.99
65	所	3.61
2	下	3.60
33	后	3.56

(a)

Vertex	Word	CC
98	苦	10.00
60	心	6.67
19	伤	5.00
10	了	2.42
71	有	2.37
15	他	2.19
65	所	2.13
2	下	2.13
75	来	2.10
0	一	2.09

(b)

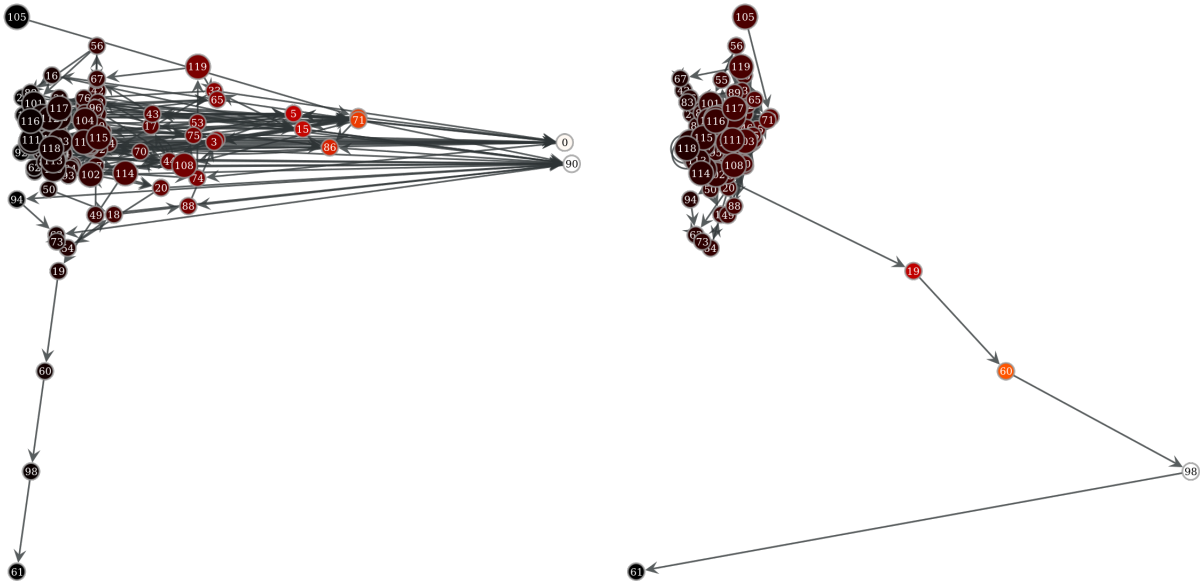
Vertex	Word	LC
22	共	10.00
94	祝	10.00
32	只	3.33
81	没	3.33
6	为	3.33
21	儿	3.33
63	愿	3.33
2	下	1.67
17	们	1.67
43	女	1.67

(c)

Vertex	Word	PR
90	的	10.00
0	一	8.54
10	了	7.85
71	有	6.18
86	王	6.03
18	会	4.62
15	他	4.61
5	个	4.55
70	时	4.29
65	所	4.15

(d)

Table 4.15: Partial extracts of the Chinese table data ordered by their (a) betweenness centrality values, (b) closeness centrality values, (c) local clustering coefficients and (d) page ranks.

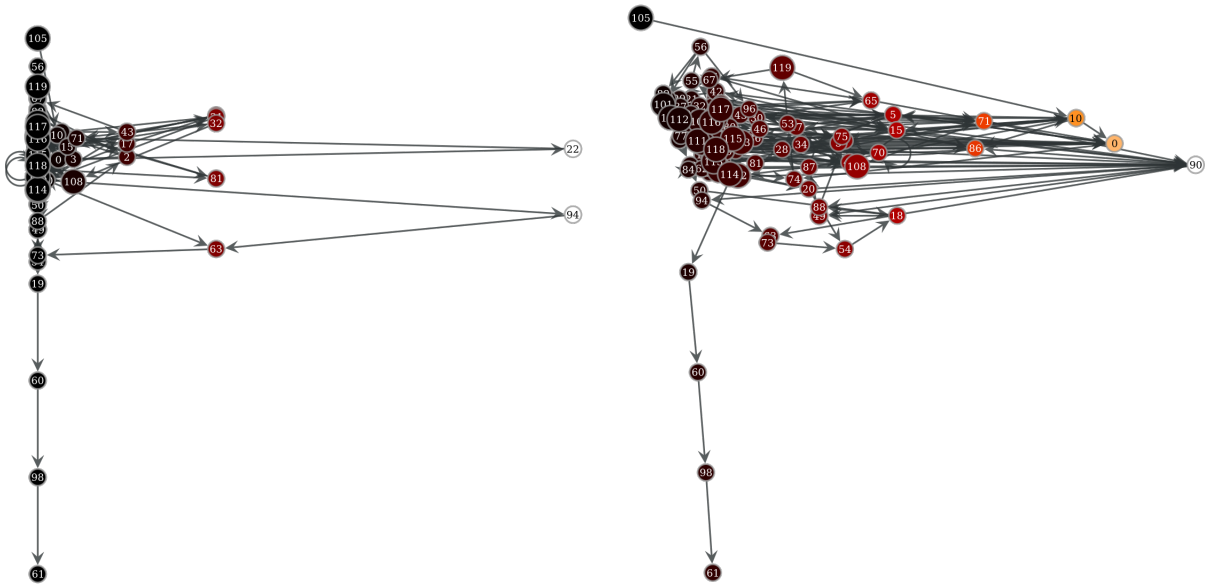


(a) (b)

Figure 4.14: Graphs showing (a) betweenness centrality and (b) closeness centrality values displayed on the x-axis based on the Chinese numbered word graph. The y-axis for their trophic levels.

has the highest closeness as without this vertex, vertex 61 is isolated. Subsequently the predecessors of vertex 98 have higher closeness. However there are no other vertices with high closeness since other syllables hold connections to various unique

vertices. Thus inferring that these syllables of high degree have multiple meanings rather than the unique meaning of vertex 98s section of the path.



(a) (b)

Figure 4.15: Displays the (a) local clustering and (b) page rank on the x-axis instead of the centrality values. The y values are unaffected.

By taking what we found in the Japanese graphs, we now know that the local clustering coefficient identifies vertices involved in unique directed 3-cycles. For example, vertex 22 is uniquely part of the directed 3-cycle, vertex $0 \to 22 \to 71 \to 0$ where vertex 22 is seen as the pinnacle of the cycle because vertices 22 and 71 hold other edge connections within the graph seen in Figure 4.15a. The same applies to vertex 94 as it is involved in its own directed 3-cycle. So local clustering does not identify the vertices of high importance but rather certain triples within the corpus.

The page rank (see Figure 4.15b) identifies the vertices with influence to vertices other than their immediate neighbours. Evidently vertices with high betweenness will also have a high page rank as they are more commonly used in the corpus. Similar results have been seen with other languages for page ranks so evidently, page rank and betweenness values are close in the context of linguistic analysis.

Chapter 5

Conclusion

Based upon the five of the nine total languages shown in the previous chapter. Results achieved in the previous chapter demonstrated the similarities of languages within the same family, particularly French and German. Their uniqueness could also be seen when comparing the graphs of each language as they each varied. Especially with languages of a different family as the Indo-European were all clustered together more but the Japonic and Sino-Tibetan families were much more evenly distributed.

Therefore the study of complex graphical properties such as trophic coherence, various centrality values and clustering have proved beneficially in distinguishing graphs from one another as well as a tool to represent possible information based upon the graph.

We summarise each graph property and the correlations that were derived from them in the next section.

5.1 Final Correlations

Based on each graph property that was studied, we take their correlations within the story corpus used and summarised as follows.

5.1.1 Local Clustering Coefficient

Local clustering with the lexical graphs corresponded to words that would most likely be involved in triples (a 3-cycle). However because of the common reuse of words, there were usually no triples that represented anything unique to the language. A way to improve the usefulness of local clustering in lexical analysis would be to use use specific motifs that could prove to generate a better correlation. Currently a generic 3-cycle is used for the local clustering calculations so using specific motifs may generate interesting results..

5.1.2 Betweenness Centrality

Betweenness centrality was chosen as it identifies the vertices with most control of informational flow. In the context of language analysis, betweenness values corresponded to the most commonly used words in each language which is further proven by the Zipf curve as mentioned in the English analysis. This was further verified when compared with the frequency of the words/vertices for every language dataset.

5.1.3 Closeness Centrality

Closeness centrality determines the vertices that on average have the shortest connections to all the other vertices within the graph. For lexical analysis, closeness centrality correlated to the vertices whom are most likely to isolate the rest of its sentence. This may be because the rest of the sentence uses more unique words which means that the vertex of high closeness is the closest to these words as well as the previous words whom did not have a high closeness. Therefore the reason these such vertices had a high closeness and its correlation in lexical terms.

5.1.4 Page Rank

Page ranks finds the vertices who are seen as important by finding the backlinks that refer to the vertex and other vertices which reference these vertices. This is similar to betweenness as both identified the vertices that held the most importance within the text so the ranks were similar to the betweenness but has more reference to the predecessors of the vertices. This causes the slight different of the betweenness graph and the page rank graph.

5.1.5 Trophic Level

Finally tropic levels are useful in determining structure and flow in a graph. Within lexical analysis, the flow of the sentences were represented by the ranges of trophic levels which were normalised to 0-10 like with the other values. Consequently were used as the y-positioning downwards from 0-10 so that the sentence flow was represented correctly. Trophic levels for each vertex represented the average position that they belonged to within a sentence with 0 at the start and 10 at the end. Languages with better sentence structure had more evenly distributed levels due to the fact that their sentence length was not as varied to other languages.

Additionally, through trophic levels visualisation, each graph property in the previous chapter had key subsegments of the y-range which held the relative graph properties high value. For example, in the betweenness graphs, the vertices of high betweenness value were centralised which means that they appeared most commonly in the middle of sentences. This is grammatically true because words of importance either appeared as a bridge in a sentence like "and" or are used commonly throughout the sentence. Thus leading to an average closer to the centre. A similar trend

was seen with local clustering where the vertices/words of high local clustering were lower in the y-range.

In conclusion, each graph property have been researched for their use and benefits within a graph. The calculations for each complex graphical property was also given along with the possible variations which depended on the graph types. I.e. the weights, whether the graph is directed, self loops, etc. After detailed descriptions of various properties were given, specific ones were transferred and used on text corpora based on various different languages and their corresponding language family. This was possible through the use of programming in Python code to generate a dataset from a text input (the story corpus for each language). The dataset can then be used to formalise a graph so that the calculations for each complex graph property can be calculated from. Furthermore, the values of each property are normalised and applied onto the positions of each vertex to generate a new graph that provides a visual representation for the dataset. Each new graph represents a different complex graph property which achieves the aim of my research. Therefore the newly generated visualisations of the graph have proven to be useful in analysis and to highlight key components of interest by showing them as extremes of clusters within the new graph. Results found can then be expanded onto a larger dataset to predict the key areas of the larger text such as the words that would be of interest. This also means that when given another graph of the same language but with unknown vertices, the graph generated based on the corpus can be used to predict and identify what the unknown vertices could be.

5.2 Further works and Applications

Briefly mentioned in the last part, the visualisations can be used as a prediction tool within the same language as similar vertices that represent each word would correspond to a similar location in this visualisation. As well as this expansion, the visualisation can be used to analyse new or unknown languages through the comparison of graphs since languages of the same family produce graphs that have resemblances. Once the best language family is chosen, predictions of what components of the graph represents can be given from the facts of other languages within the same language family. Improvements for each specific language representations can be remedied since languages such as Japanese and Chinese does not use words as their sentence structure is more complex than most other languages.

Therefore the current works on lexical analysis and complex graph properties can be taken further by introducing unknown languages, expanding on the graph properties mentioned such as using specific motifs for the local clustering or the application onto other subject areas rather than lexical analysis.

Other areas that could be analysed includes neuroscience. The brain [72] can be regarded as a complex network of neurons where each vertex represents units

or specific regions and the edges represents the links or connections between them. Analysis of neurological diseases such as schizophrenia, dementia and Alzheimer's can be used to generated relative graph visualisations. These visualisations can be studied further to find common links and correlations within each neurological disease so that action could be taken if early signs of similar correlations are found within a healthy brain. A similar idea was studied based on detecting changes in the brain which has Alzheimer's using graph theory [73]. In conclusion graph theory is crucial in representing various types of information in the real world from friendship groups to the complex structures of the brain. Many subject areas can be represented as a graph which means that they can undergo similar graphical analysis and visualisations which experimented on in Chapters 3 and 4 to deduce correlations and key components of interest in the relative graph.

Bibliography

[1] James Powell and Matthew Hopkins. *A Librarian's Guide to Graphs, Data and the Semantic Web*. Chandos Information Professional Series. Chandos Publishing, 2015.

[2] School of Mathematics and Statistics, University of St Andrews, Scotland. Königsberg bridges, 2000. [Online; accessed 2 January, 2023].

[3] Kane O Pryor and Jamie Sleigh. The seven bridges of königsberg. *The Journal of the American Society of Anesthesiologists*, 114(4):739–740, 2011.

[4] Pavel A Pevzner, Haixu Tang, and Michael S Waterman. An eulerian path approach to dna fragment assembly. *Proceedings of the national academy of sciences*, 98(17):9748–9753, 2001.

[5] Loh Bo Huai Victor. Eulerian path and circuit, 2010.

[6] Stephen C. Carlson. graph theory. Dec 2022.

[7] M Sohel Rahman and Mohammad Kaykobad. On hamiltonian cycles and hamiltonian paths. *Information Processing Letters*, 94(1):37–41, 2005.

[8] Koji Ohnishi. Towards a brief proof of the four-color theorem without using a computer: theorems to be used for proving the four-color theorem. *Artificial Life and Robotics*, 14(4):551–556, Dec 2009.

[9] Robin Wilson. Four colours suffice, 2017. [Online; accessed 19 January, 2023].

[10] Stefan Irnich. Undirected postman problems with zigzagging option: A cutting-plane approach. *Computers & Operations Research*, 35(12):3998–4010, December 2008.

[11] Ángel Corberán. The chinese postman problem with load-dependent costs. *Transportation Science*, 52(2):370–386, March 2018.

[12] Michael Hart, R.J.F. Ypma, Rafael Romero-García, Stephen Price, and John Suckling. Graph theory analysis of complex brain networks: New concepts in brain mapping applied to neurosurgery. *J. Neurosurg.*, 124:1665–1678, 06 2016.

[13] Md Saidur Rahman et al. *Basic graph theory*, volume 9. Springer, 2017.

[14] Lester R Ford Jr. Network flow theory. Technical report, Rand Corp Santa Monica Ca, 1956.

[15] Wayne W. Zachary. Modeling social network processes using constrained flow representations. *Social Networks*, 6(3):259–292, 1984.

[16] U. Knauer. *Algebraic graph theory : morphisms, monoids, and matrices / by Ulrich Knauer.* De Gruyter studies in mathematics ; 41. 2011.

[17] Michael S Vitevitch. What can graph theory tell us about word learning and lexical retrieval? 2008.

[18] Walter Crane. *The baby's opera.* anboco, 2016.

[19] Duncan J Watts and Steven H Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393(6684):440–442, 1998.

[20] Andrew R Hippisley. Lexical analysis. 2010.

[21] Andrew Broekman and Linda Marshall. Linguistic inspired graph analysis. *arXiv preprint arXiv:2105.06216*, 2021.

[22] Samuel Johnson, Virginia Domínguez-García, Luca Donetti, and Miguel A Munoz. Trophic coherence determines food-web stability. *Proceedings of the National Academy of Sciences*, 111(50):17923–17928, 2014.

[23] Miguel Saigo, Florencia Lucila Zilli, MR Marchese, and D Demonte. Trophic level, food chain length and omnivory in the paraná river: a food web model approach in a floodplain river system. *Ecological Research*, 30(5):843–852, 2015.

[24] Samuel Johnson. Digraphs are different: Why directionality matters in complex systems. *Journal of Physics: Complexity*, 1(1):015003, 2020.

[25] Thomas Schank and Dorothea Wagner. Approximating clustering coefficient and transitivity. *Journal of Graph Algorithms and Applications*, 9(2):265–275, 2005.

[26] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, jan 2003.

[27] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, Jun 1998.

[28] Liudmila Ostroumova Prokhorenkova and Egor Samosvat. Global clustering coefficient in scale-free networks, 2014.

[29] Tanguy Fardet and Anna Levina. Weighted directed clustering: Interpretations and requirements for heterogeneous, inferred, and measured networks. *Phys. Rev. Res.*, 3:043124, Nov 2021.

[30] Bin Zhang and Steve Horvath. A general framework for weighted gene co-expression network analysis. *Statistical Applications in Genetics and Molecular Biology*, 4(1), 2005.

[31] Tore Opsahl and Pietro Panzarasa. Clustering in weighted networks. *Social networks*, 31(2):155–163, 2009.

[32] Kyle Bojanek, Yuqing Zhu, and Jason Maclean. Cyclic transitions between higher order motifs underlie sustained asynchronous spiking in sparse recurrent networks. *PLoS computational biology*, 16:e1007409, 09 2020.

[33] Giorgio Fagiolo. Clustering in complex directed networks. *Physical Review E*, 76(2), aug 2007.

[34] G.P. Clemente and R. Grassi. Directed clustering in weighted networks: A new perspective. *Chaos, Solitons & Fractals*, 107:26–38, 2018.

[35] Alex Bavelas. A mathematical model for group structures. *Human organization*, 7(3):16–30, 1948.

[36] Linton C Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977.

[37] Alfonso Shimbel. Structural parameters of communication networks. *The bulletin of mathematical biophysics*, 15:501–507, 1953.

[38] Linton C Freeman et al. Centrality in social networks: Conceptual clarification. *Social network: critical concepts in sociology. Londres: Routledge*, 1:238–263, 2002.

[39] Linton C Freeman, Douglas Roeder, and Robert R Mulholland. Centrality in social networks: Ii. experimental results. *Social networks*, 2(2):119–141, 1979.

[40] Douglas R White and Stephen P Borgatti. Betweenness centrality measures for directed graphs. *Social networks*, 16(4):335–346, 1994.

[41] Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.

[42] Ulrik Brandes, Stephen P Borgatti, and Linton C Freeman. Maintaining the duality of closeness and betweenness centrality. *Social networks*, 44:153–159, 2016.

[43] Ulrik Brandes and Christian Pich. Centrality estimation in large networks. *International Journal of Bifurcation and Chaos*, 17(07):2303–2318, 2007.

[44] Silvia de Juan, Andres Ospina-Alvarez, Sebastián Villasante, and Ana Ruiz-Frau. A graph theory approach to assess nature's contribution to people at a global scale. *Scientific Reports*, 11(1):9118, 2021.

[45] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, Dandapani Sivakumar, Andrew Tompkins, and Eli Upfal. The web as a graph. In *Proceedings of the nineteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 1–10, 2000.

[46] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998.

[47] Amy N Langville and Carl D Meyer. A survey of eigenvector methods for web information retrieval. *SIAM review*, 47(1):135–161, 2005.

[48] Alexander Chatzigeorgiou, Nikolaos Tsantalis, and George Stephanides. Application of graph theory to oo software engineering. In *Proceedings of the 2006 international workshop on Workshop on interdisciplinary software engineering research*, pages 29–36, 2006.

[49] Maristella Agosti and Luca Pretto. A theoretical study of a generalized version of kleinberg's hits algorithm. *Information Retrieval*, 8(2):219–243, 2005.

[50] Pooja Devi, Ashlesha Gupta, and Ashutosh Dixit. Comparative study of hits and pagerank link based ranking algorithms. *International Journal of Advanced Research in Computer and Communication Engineering*, 3(2):5749–5754, 2014.

[51] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[52] Wenpu Xing and Ali Ghorbani. Weighted pagerank algorithm. pages 305–314, 2004.

[53] Tiago P. Peixoto. The graph-tool python library. *figshare*, 2014.

[54] William L. Hosch. Zipf's law. 2009.

[55] Elvira I Sicilia-Garcia, Ji Ming, Francis J Smith, et al. Extension of zipf's law to words and phrases. In *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002.

[56] Sergio Jimenez. Zipf 30wiki en labels, 2015.

[57] Christian Bentz, Douwe Kiela, Feli Hill, and Paula Buttery. Zipf's law and the grammar of languages: A quantitative study of old and modern english parallel texts. *Corpus Linguistics and Linguistic Theory*, 10(2):175–211, 2014.

[58] Lyle Campbell. How many language families are there in the world? *Anuario del Seminario de Filología Vasca" Julio de Urquijo"*, 52(1/2):133–152, 2018.

[59] David M. Eberhard, Gary F. Simons, and Charles D. Fennig. Ethnologue: Languages of the world.

[60] Bruce M Rowe and Diane P Levine. *A concise introduction to linguistics*. Taylor & Francis, 2022.

[61] Lyle Campbell. Language isolates and their history, or, what's weird, anyway? In *Annual Meeting of the Berkeley Linguistics Society*, volume 36, pages 16–31, 2010.

[62] J. Grimm and W. Grimm. *Kinder und hausmärchen: gesammelt durch die Brüder Grimm*. Number v. 1-2 in Kinder und hausmärchen: gesammelt durch die Brüder Grimm. Dieterich, 1857.

[63] Sukany Khaisaeng and NK Dennis. A study of part of speech used in online student weekly magazine. *International Journal of Journal of Research Granthaalayah*, 5(4):43–50, 2017.

[64] BNC Consortium et al. British national corpus. *Oxford Text Archive Core Collection*, 2007.

[65] Geoffrey Leech, Paul Rayson, et al. *Word frequencies in written and spoken English: Based on the British National Corpus*. Routledge, 2014.

[66] Samuel Johnson and Nick S Jones. Looplessness in networks is linked to trophic coherence. *Proceedings of the National Academy of Sciences*, 114(22):5618–5623, 2017.

[67] Martin Durrell. *Hammer's German grammar and usage*. Routledge, 2011.

[68] Roger Hawkins and Richard Towell. *French grammar and usage*. Routledge, 2015.

[69] Alexander Vovin. Origins of the japanese language. In *Oxford Research Encyclopedia of Linguistics*. 2017.

[70] Akira Miura. The influence of english on japanese grammar. *The Journal of the Association of Teachers of Japanese*, 14(1):3–30, 1979.

[71] Claudia Ross and Jing-heng Sheng Ma. *Modern Mandarin Chinese grammar: A practical guide*. Routledge, 2017.

[72] Fabrizio de Vico Fallani, Jonas Richiardi, Mario Chavez, and Sophie Achard. Graph analysis of functional brain networks: practical issues in translational neuroscience. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 369(1653):20130521, 2014.

[73] Joana B Pereira. Detecting early changes in Alzheimer's disease with graph theory. *Brain Communications*, 2(2), 08 2020. fcaa129.

Appendix A

Langauages

A.1 Text Corpus

In times past there lived a king and queen, who said to each other every day of their lives, "Would that we had a child!" and yet they had none. But it happened once that when the queen was bathing, there came a frog out of the water, and he squatted on the ground, and said to her: "Thy wish shall be fulfilled; before a year has gone by, thou shalt bring a daughter into the world."

And as the frog foretold, so it happened; and the queen bore a daughter so beautiful that the king could not contain himself for joy, and he ordained a great feast. Not only did he bid to it his relations, friends, and acquaintances, but also the wise women, that they might be kind and favourable to the child. There were thirteen of them in his kingdom, but as he had only provided twelve golden plates for them to eat from, one of them had to be left out.

A.2 English Language Table

Entire table of graph property values for the english word graph that was generated from the first two paragraphs of the story "Sleeping Beauty".

Vertex	Words	Counts	TC	BC	CC	LC	PR
0	a	7	1.98	9.26	2.22	0.00	7.65
1	acquaintances	1	2.53	0.60	1.64	0.00	0.36
2	also	1	3.20	0.24	1.82	0.00	0.16
3	and	9	3.00	10.00	2.49	0.57	8.13
4	as	2	2.95	0.73	2.16	3.33	1.13
5	bathing	1	1.24	1.56	1.51	0.00	0.33
6	be	3	3.78	3.15	1.91	0.00	3.24
7	beautiful	1	3.29	0.77	1.81	0.00	0.42
8	before	1	2.58	0.83	1.84	0.00	0.74
9	bid	1	3.11	0.38	1.88	0.00	0.11
10	bore	1	2.43	0.73	1.83	0.00	0.10
11	bring	1	1.98	1.96	1.84	0.00	1.24
12	but	3	2.05	1.29	1.95	0.00	1.73
13	by	1	1.98	1.91	1.22	0.00	1.00
14	came	1	1.20	0.51	1.84	0.00	0.12
15	child	2	7.17	0.00	0.00	0.00	1.78
16	contain	1	0.78	1.13	1.55	0.00	0.12
17	could	1	1.33	2.12	1.32	0.00	0.57
18	daughter	2	1.19	1.44	1.70	0.00	0.61
19	day	1	4.91	1.00	1.94	0.00	0.82
20	did	1	2.16	0.73	1.72	0.00	0.30
21	each	1	4.08	0.94	1.25	0.00	0.31
22	eat	1	4.15	0.70	1.42	0.00	0.31
23	every	1	4.63	0.98	1.64	0.00	0.68
24	favourable	1	3.40	1.16	1.87	0.00	0.36
25	feast	1	10.00	0.00	0.00	0.00	0.76
26	for	2	2.35	2.57	2.17	0.00	2.09
27	foretold	1	3.39	0.52	1.48	0.00	0.57
28	friends	1	2.71	0.67	2.02	0.00	0.57
29	frog	2	3.93	2.62	1.72	0.00	1.78
30	from	1	4.49	0.72	1.64	0.00	0.51
31	fulfilled	1	3.18	0.82	1.57	0.00	0.59

32	golden	1	2.17	1.05	1.55	0.00	0.67
33	gone	1	1.98	1.89	1.09	0.00	0.89
34	great	1	5.99	0.32	10.00	0.00	0.61
35	ground	1	3.67	0.60	2.00	10.00	0.55
36	had	4	3.44	5.32	2.48	0.36	3.82
37	happened	2	2.36	1.07	2.07	0.00	0.72
38	has	1	1.98	1.87	0.99	0.00	0.76
39	he	4	2.42	3.68	2.07	0.48	2.38
40	her	1	3.79	0.94	1.12	0.00	0.31
41	himself	1	1.57	1.15	1.81	0.00	0.35
42	his	2	2.13	1.29	1.52	0.00	1.33
43	in	2	0.49	1.29	1.50	0.00	0.44
44	into	1	2.77	0.36	1.82	0.00	0.07
45	it	3	3.52	2.05	1.79	0.00	2.12
46	joy	1	2.67	0.78	2.02	0.00	0.70
47	kind	1	3.39	1.38	2.01	0.00	0.59
48	king	2	2.66	5.79	2.10	1.67	1.78
49	kingdom	1	2.09	0.33	1.64	0.00	0.38
50	left	1	4.90	0.32	1.63	0.00	0.59
51	lived	1	1.20	0.51	1.84	0.00	0.12
52	lives	1	4.46	0.73	1.56	0.00	1.42
53	might	1	3.65	0.28	1.61	0.00	0.60
54	none	1	7.45	0.00	0.00	0.00	0.41
55	not	2	0.00	2.13	1.51	0.00	0.73
56	of	4	5.19	6.21	2.37	0.00	6.03
57	on	1	3.70	0.41	1.82	0.00	0.33
58	once	1	3.05	0.18	1.81	0.00	0.12
59	one	1	4.84	0.74	1.94	0.00	0.68
60	only	2	1.90	2.05	1.55	0.00	1.15
61	ordained	1	2.20	0.53	1.83	0.00	0.11
62	other	1	4.35	0.96	1.42	0.00	0.51
63	out	2	6.01	2.45	1.93	0.00	1.93
64	past	1	0.44	0.71	1.51	0.00	0.24

65	plates	1	2.26	1.06	1.81	0.00	0.81
66	provided	1	1.99	1.01	1.20	0.00	0.30
67	queen	3	2.89	3.09	1.88	1.00	1.53
68	relations	1	2.42	0.65	1.70	0.00	0.38
69	said	2	5.62	1.65	1.87	0.00	1.30
70	shall	1	3.79	1.00	1.62	0.00	0.82
71	shalt	1	1.98	1.94	1.57	0.00	1.18
72	so	2	2.85	1.63	1.72	0.00	1.42
73	squatted	1	3.06	0.39	1.55	0.00	0.11
74	that	4	3.73	3.49	2.19	0.36	5.12
75	the	9	4.34	9.19	2.20	0.44	10.00
76	their	1	4.83	0.71	1.36	0.00	1.38
77	them	3	2.90	5.03	2.40	1.00	2.70
78	there	3	0.41	2.63	1.76	0.00	1.59
79	they	2	3.52	1.48	2.05	0.00	1.85
80	thirteen	1	3.60	1.00	1.94	0.00	0.35
81	thou	1	1.98	1.93	1.37	0.00	1.10
82	thy	1	3.79	0.96	1.25	0.00	0.51
83	times	1	0.47	0.70	1.32	0.00	0.00
84	to	6	3.80	7.48	2.28	0.18	5.54
85	twelve	1	2.08	1.03	1.35	0.00	0.50
86	was	1	2.06	1.54	1.32	0.00	0.10
87	water	1	3.67	0.60	2.00	10.00	0.55
88	we	1	3.59	0.64	2.01	0.00	0.69
89	were	1	2.01	0.98	1.65	0.00	0.12
90	when	1	4.03	0.00	1.82	10.00	0.69
91	who	1	4.26	0.19	1.59	0.00	0.10
92	wise	1	4.14	0.75	1.54	0.00	0.55
93	wish	1	3.79	0.98	1.41	0.00	0.68
94	women	1	3.93	0.76	1.80	0.00	0.72
95	world	1	8.35	0.00	0.00	0.00	0.55
96	would	1	4.10	0.75	1.82	0.00	1.45
97	year	1	1.98	1.86	0.91	0.00	0.61
98	yet	1	3.26	0.48	1.72	0.00	0.36