Imperial College London

Network Analysis of Wikipedia Biography Pages of Physicists

Author: Chao Fan
Project Partner: Kaiyu Hu
May 2022

CID: 01726262

Supervisor: Dr. Tim S. Evans

Assessor: Dr. Dave Clements

Word Count: 5643

Declaration of work undertaken:

This is a one-term project which is comprised of two major goals: developing a clustering algorithm using natural language toolkits, and network construction and evaluation of centrality measures. The former aspect is carried out by Chao Fan (the author of this report) and the latter is implemented by Kaiyu Hu.

Contents

Abstract

In this project, the Wikipedia pages of 1019 physicists are used to construct a network between the physicists by defining the edges with the hyperlinks in the main body texts which redirect one physicist's page to another. The physicists are put into 5 clusters using Kmeans algorithm, which uses the tf-idf matrix built from the main body texts of the biographies. 5 distinct centrality measures are evaluated and combined to produce an average mark, which is used as an indication of the importance of the physicists. Albert Einstein is identified as the most important physicist of all time, followed by Fermi, Heisenberg and others. The physicists are classified into 5 clusters, with the 3 major clusters identified by their research field and 2 minor ones identified by their awards and nationalities. A power law relationship is found between the lengths of the web pages and their average marks, and the biases of the English Wikipedia biographies are discussed.

1 Introduction

The subject of physics is a vast idea that reaches out to numerous branches, each have the most brilliant minds in the history of human civilization contributing to the building of the understanding of the nature of our world. Many of their stories and findings are archived digitally in free websites such as Wikipedia so that we as successors can learn about their beautiful lives and historical importance. In this project we focus on answering these questions:

1. How can we construct a network between physicists using their Wikipedia biography pages?
2. How do we evaluate uncertainties in network centrality measures?
3. Can we classify physicists into different clusters based on keywords appeared in their biographies?
4. Who is the most important physicist from each cluster and overall?
5. Can we draw any stories from the result about physicists themselves and Wikipedia web pages?

The project is naturally split into two directions: document clustering and networks. The first part of this project involves the construction of a network using the hyperlinks in the pages and various centrality measures are evaluated for the network. The second part of this project follows from the first part and uses natural language processing tools to separate the texts from the biographies and use them as keywords for clustering, so that we can compare the results between each cluster and draw stories from them.

2 Data Extraction and Methods

The data used for this project contains the html files for the English language Wikipedia biography pages of 1019 most well-known physicists from Wikipedia's list of physicist page[1]. This page is constantly being edited and we worked on the local file downloaded in Jan. 2022 to preserve consistency. The files are loaded into python using a Beautifulsoup html reader. The texts and hyperlinks can be extracted using different methods from the Beautifulsoup object.

2.1 Data Cleaning

The main problem faced is determining which part of the html file we consider to contain useful information. Here an example of a typical Wikipedia web page of a physicist is shown. Most of the side bar contents are either summary or repeats of the main body texts highlighted in red so it is assumed that most of the information about the person is well described in the texts and the hyperlinks redirecting to other physicists are usually found in the main boy texts as well. Moreover, the side bar contents sometimes list the names of the physicists working on the same field along with the links, which disrupts the degree of the network. The same can be concluded from the reference section down the bottom of the page. Usually the reference section does not contain useful information and creates meaningless keywords to clustering algorithm such as 'ISBN' and 'doi'. Any physicist who had distinct achievement(s) e.g. won a Nobel Prize, took part in the Manhattan Project or

was the president of the American Physical Society once and others is also listed among all other people at the end of the web page in a separated section, which disrupts the degree of the network hugely. The argument is that just because two physicists had both won a Nobel Prize or took the role of the president of the American Physical Society does not mean that they will have connection at all, given that there maybe a huge time gap between the two, or they worked on very distinctive aspects. Moreover, even if they had connections it should appear in the main boy texts already. Therefore we decide to remove all side bar contents and sections including and after references so that we are left with only the main body texts.

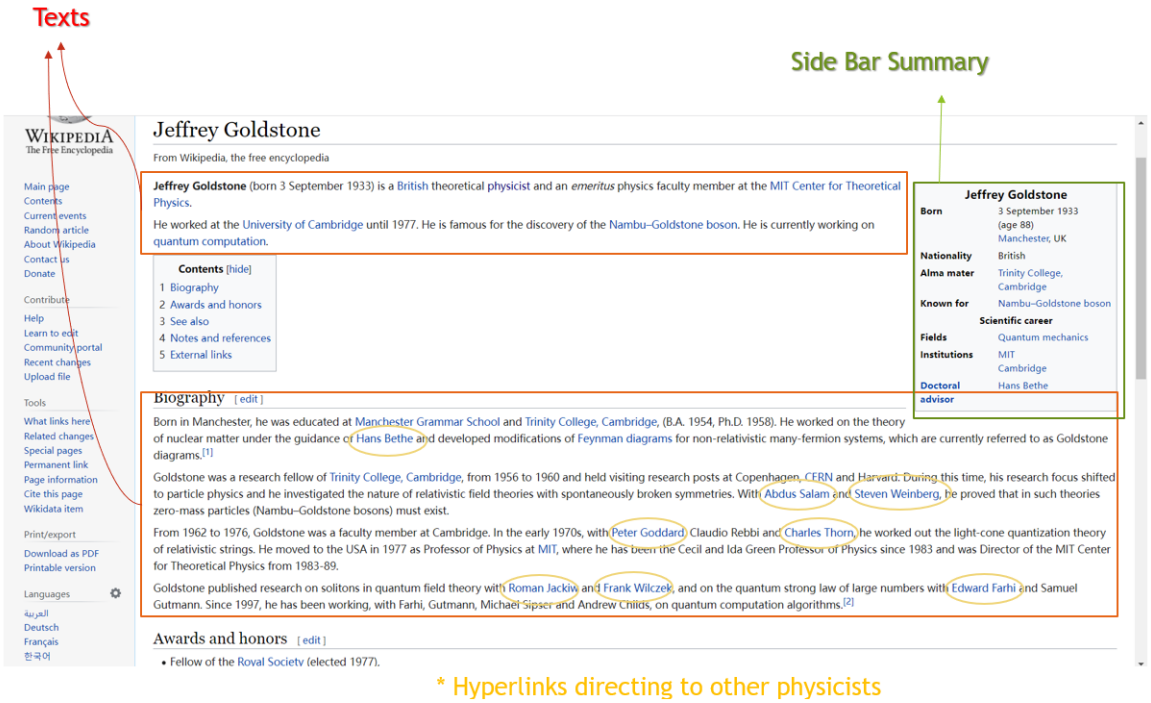


Figure 1: An example of the Wikipedia web page of a physicist.

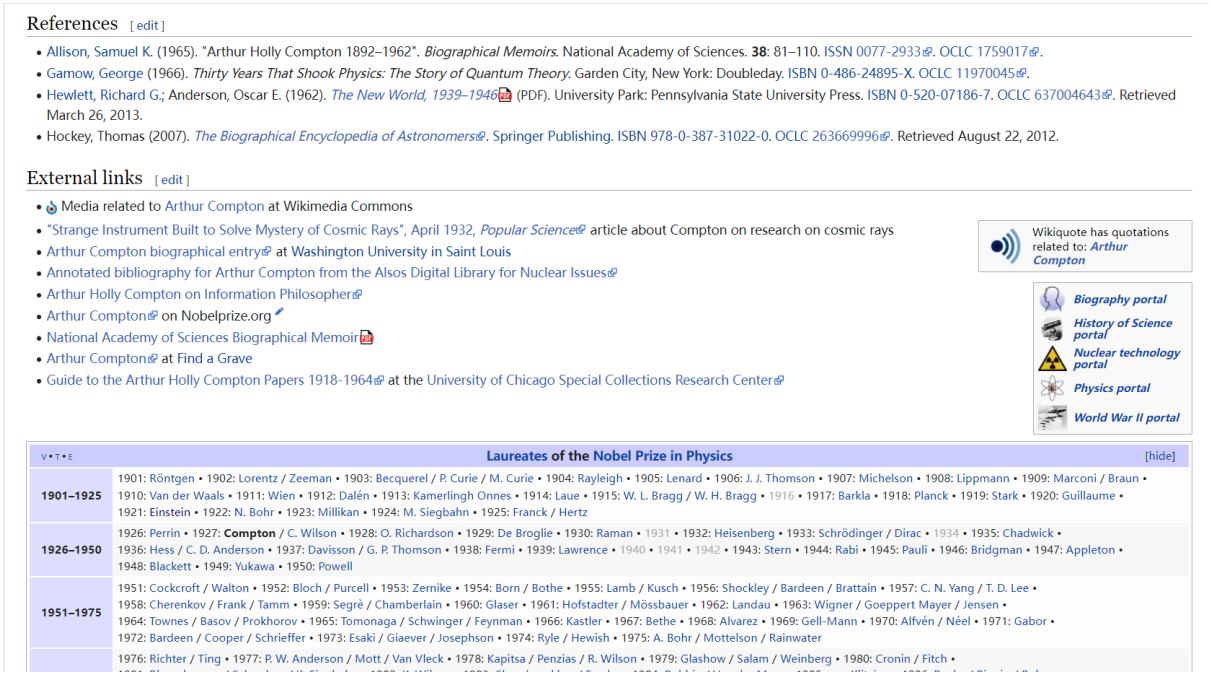


Figure 2: An example of the end of Wikipedia web page of a physicist. Here Arthur Compton's page is used. Compton was a laureate of the Nobel Physics Prize so all Nobel Physics winners appeared after the references section. The same applies to other pages.

2.2 Building the Network

A network is a set of objects, that we call nodes or vertices labelled $\mathcal{V}$, which are connected by properties we call edges. In this project the physicists are considered to be the nodes and the edges are the hyperlinks from one physicist's web page to another. Following from section 2.1, we have already extracted all hyperlinks in all physicists' pages, but not all of them are linked to another physicist in our list. Here is an example of how the hyperlinks extracted from the web pages look like:

.....
'/wiki/George_I_of_Great_Britain'
'/wiki/Isaac_Newton'
'/wiki/Royal_Society'
.....

Table 1: A section of the hyperlinks extracted from the web page of Willem Gravesande.

From the above example we see the last piece of the URLs contains the name of the redirected page, therefore it can be used to identify the edges we want by reading the string after the last / sign and scan through the name list to see if it matches with any name of the physicists. If there is a match, an edge is connected between the owner of the web page and the redirected person of the URL. Iterating this process through all physicists in the list generates an adjacency list of the physicists and all other physicists that they are related to through the URLs in their web page, which we define as the vertices and edges for our network.

Using we have constructed an undirected network with its accompanying adjacency list. Although the directions of the hyperlinks can be acquired, we have neglected them as it is often hard to interpret the meaning of the direction without context, which would require us to locate the link in the biography and use natural language processing tools beyond the level of this project.

2.3 Centrality Measures

5 different centrality measures are evaluated for our network: degree, closeness, betweenness, eigenvector and PageRank.

2.3.1 Degree

The degree of a vertex, labelled as k, is the number of edges connected to it.

$$k_i = \Sigma_j A_{ij}$$

where A_{ij} is the element of the adjacency matrix of this vertex. In a simple graph,

$$A_{ij} = \begin{cases} 1 & \text{if vertex i connects to vertex j} \\ 0 & \text{if vertex i does not connect to vertex j} \end{cases}$$

The assumption is that the more connection a physicist have with other people in the list, the more important and popular the physicist is.

2.3.2 Closeness

The closeness of a vertex, labelled as c, is the inverse of the average of the shortest paths from all other vertices in the network to it.

$$c_i = \frac{n-1}{\Sigma_{j \in \mathcal{V}_i} d_{ij}}$$

where n is the total number of vertices in the network and d_{ij} is the distance or shortest path between vertex i and j. In an undirected graph the distance is defined as the number of vertices it passes to connect from vertex i to vertex j plus 1. i.e. if two vertices are directly connected their distance is 1; if one extra vertex is needed to connect vertex i and vertex j their distance is 2.

The assumption is that if the physicist has a small value for the closeness, it means that the person is in a central position in the network and has small distances to other people. Hence the physicist is considered to be more important.

2.3.3 Betweenness

The betweenness of a vertex, labelled b, is defined as the fraction between the number of shortest paths passing through this vertex and all shortest paths in the network.

$$b_i = \sum_{s,t \in \mathcal{V}} \frac{\sigma(s,t|\mathcal{V}_i)}{\sigma(s,t)}$$

where $\sigma(s,t)$ is the number of shortest paths between vertices s and t and $\sigma(s,t|\mathcal{V}_i)$ is the number of those paths passing through $\mathcal{V}_i$.

The assumption is that the more shortest paths passes through a physicist, the more vital this physicist is.

2.3.4 Eigenvector

The eigenvector centrality of vertex $\mathcal{V}_i$ in our network is defined as the i^{th} element of the Perron vector $\vec{x}$ in the equation:

$$A\vec{x} = \lambda\vec{x}$$

where A is the adjacency matrix of the network with eigenvalue λ. The Perron-Forbenius theorem states that for the largest eigenvalue λ, there is a unique corresponding eigenvector solution $\vec{x}$, which has all of its entries greater than 0. This vector is referred to as the Perron vector and is used to calculate the eigenvector centrality. A starting centrality value is set to all vertices in the network. Then the values are broadcast to their neighbours, weighted by the edge weight. The neighbours sum up all the values received as its new centrality value. Therefore we expect links to important vertices to have a big effect on this new value, while weaker vertices play a smaller part. By iterating the broadcast process for a large number of times, the Perron vector term dominates since it has the largest eigenvalue, so we can say that the centrality values are proportional to the elements in the Perron vector. Hence the elements of the Perron vectors are used as one of the centrality measures.

The interpretation is that the more other important physicists a person knows, the more likely this person is also important too.

2.3.5 PageRank

Instead of a broadcast process, one could consider using a diffusion scheme, where the centrality value of a vertex is 'diffused' or equally divided and passed to its neighbours. This process can be described by the transfer matrix T, where T_{ji} represents the fraction of the value from source vertex i to neighbour j.

The PageRank centrality measure is defined as the long-time limit of the centrality measure through iterations of the following process:

$$R_i = \lim_{t\to\infty} w(t)_i, \qquad w(t+1)_j = (1-\alpha)h_i + \alpha \sum_i T_{ji} w(t)_i$$

where w is the centrality value of each vertex, and α is the damping parameter, which means that the probability of a random walker making a 'hyperjump' to other vertices instead of its neighbours is $1-\alpha$, and the probability of jumping to vertex i is h_i. Here it is assumed that hyperjumps to all vertices are equally likely, so $h_i = 1/N$.

The interpretation of the PageRank value is the number of random walkers at the vertex, so a larger value means more popularity of the physicist.

2.4 Noise Model

The nature of this network makes it easy to see the robustness of our result: we expect the top names who have the highest importance in the network to be very familiar to any physics student and at least some of them can be recognised by ordinary people who do not have to have studied science. However, there are certainly errors and disadvantages in our network. The URLs in the biographies along can sometimes bring false connections between physicists or missing some connections as they heavily depend on the editor of the page and different pages are likely to be written by different editors with different preferences and knowledge. Both of these issues raise errors to the edges of the network. A noise model is set to model the re-editing process of the Wikipedia page to test the robustness of the network.

5% of the existing edges are removed randomly to simulate the process of deleting poor connections between physicists, and 5% of new edges are added to the network to simulate process of adding a new link to the biographies. The centrality measures for the new network with added 'noises' are calculated and this process is iterated for 100 times. The standard deviation of the centrality measures are used as the errors in our data.

2.5 Average Mark

To combine all of the above centrality measures, each different measure must firstly be normalised as they all scale differently. Here we used the following scheme for each measure:

$$C_i^{rescaled} = \frac{C_i}{C_{max}}$$

where C_i corresponds to the centrality measure of physicist i and C_{max} is the maximum value of this measure.

Having done so, the five measures are combined by averaging together to produce an average mark for all physicists in the network. Errors obtained from the noise model is also considered during this calculation.

$$Average\ Mark = \sum_j C_j$$

The results of the top 30 physicists with the highest average marks from the network are listed below:

Ranking	Name	Average Mark	Degree	Closeness	Betweenness	Eigenvector	PageRank
1	Albert Einstein	100.0 ± 0.28	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 1.42	100.0 ± 0.0
2	Niels Bohr	77.85 ± 1.47	80.83 ± 1.56	93.47 ± 0.87	39.89 ± 3.57	99.28 ± 2.74	75.78 ± 1.77
3	Enrico Fermi	75.16 ± 1.88	72.5 ± 1.8	94.09 ± 0.85	51.6 ± 3.83	87.15 ± 3.38	70.46 ± 1.87
4	Werner Heisenberg	70.64 ± 1.27	73.33 ± 1.22	92.43 ± 0.79	28.51 ± 3.52	93.54 ± 2.86	65.39 ± 1.36
5	Max Born	58.13 ± 1.2	55.83 ± 1.19	89.19 ± 0.79	16.7 ± 2.46	77.81 ± 2.89	51.1 ± 1.25
6	Hans Bethe	58.12 ± 0.81	53.33 ± 0.64	90.65 ± 0.73	27.49 ± 2.79	67.71 ± 2.29	51.43 ± 0.97
7	Arnold Sommerfeld	56.85 ± 1.41	50.83 ± 1.51	86.7 ± 0.83	32.28 ± 2.24	56.96 ± 2.62	57.45 ± 1.86
8	Paul Dirac	56.11 ± 1.19	46.67 ± 1.05	93.66 ± 0.87	31.36 ± 2.57	64.12 ± 2.68	44.76 ± 1.17
9	Isaac Newton	54.7 ± 1.3	50.0 ± 1.24	89.15 ± 0.79	51.42 ± 3.14	28.01 ± 1.66	54.93 ± 1.51
10	Lise Meitner	52.99 ± 1.12	47.5 ± 1.02	89.11 ± 0.8	17.97 ± 2.1	68.12 ± 2.75	42.28 ± 1.08
11	Richard Feynman	52.89 ± 1.18	45.83 ± 1.08	88.72 ± 0.84	32.15 ± 2.79	48.25 ± 2.32	49.51 ± 1.23
12	Wolfgang Pauli	51.85 ± 1.15	43.33 ± 1.05	89.06 ± 0.76	18.56 ± 2.1	65.96 ± 2.64	42.36 ± 1.27
13	Edward Teller	50.72 ± 1.26	43.33 ± 1.17	89.37 ± 0.98	20.49 ± 2.16	57.34 ± 2.27	43.06 ± 1.28
14	Ernest Rutherford	50.42 ± 1.27	46.67 ± 1.27	88.76 ± 1.0	26.24 ± 2.28	44.29 ± 2.22	46.15 ± 1.35
15	Max Planck	48.29 ± 0.96	42.5 ± 0.94	87.66 ± 0.76	13.64 ± 2.1	58.64 ± 2.64	39.02 ± 1.05
16	J. Robert Oppenheimer	48.25 ± 0.82	41.67 ± 0.67	88.72 ± 0.8	17.52 ± 2.32	53.65 ± 1.99	39.67 ± 0.99
17	Erwin Schrödinger	46.91 ± 0.92	39.17 ± 1.0	89.02 ± 0.71	12.69 ± 1.62	57.46 ± 2.35	36.21 ± 1.02
18	James Clerk Maxwell	46.6 ± 1.18	43.33 ± 1.46	85.53 ± 0.79	28.92 ± 2.0	30.57 ± 1.73	44.64 ± 1.48
19	Isidor Isaac Rabi	45.31 ± 0.83	38.33 ± 0.82	86.54 ± 0.81	11.37 ± 1.66	54.96 ± 2.22	35.33 ± 0.91
20	Rudolf Peierls	44.36 ± 0.8	36.67 ± 0.81	84.89 ± 0.67	11.62 ± 1.77	53.32 ± 2.15	35.31 ± 1.05
21	Eugene Wigner	44.12 ± 0.78	36.67 ± 0.68	86.62 ± 0.91	9.68 ± 1.59	53.67 ± 2.35	33.96 ± 0.74
22	John von Neumann	40.57 ± 0.76	33.33 ± 0.72	83.92 ± 0.73	12.39 ± 1.62	38.57 ± 1.92	34.64 ± 0.94
23	Otto Hahn	39.57 ± 0.6	33.33 ± 0.53	82.12 ± 0.7	4.65 ± 1.84	48.31 ± 2.05	29.42 ± 0.79
24	J. J. Thomson	39.48 ± 0.85	34.17 ± 0.77	83.69 ± 0.91	16.25 ± 1.96	26.65 ± 1.8	36.65 ± 1.02
25	Stephen Hawking	38.97 ± 0.98	31.67 ± 0.99	80.07 ± 0.75	33.06 ± 2.59	9.34 ± 0.65	40.73 ± 1.34
26	Lev Landau	38.53 ± 1.25	25.83 ± 1.24	85.61 ± 0.92	20.78 ± 2.03	30.86 ± 2.15	29.57 ± 1.36
27	James Chadwick	37.64 ± 0.8	29.17 ± 0.94	84.23 ± 0.71	4.76 ± 0.95	43.61 ± 2.26	26.44 ± 0.89
28	George Gamow	37.57 ± 0.83	28.33 ± 0.9	86.82 ± 0.79	11.22 ± 1.16	33.38 ± 1.85	28.09 ± 0.87
29	Subrahmanyan Chandrasekhar	35.1 ± 0.64	24.17 ± 0.54	84.0 ± 0.78	13.96 ± 1.28	25.57 ± 1.55	27.81 ± 0.87
30	Arthur Compton	34.52 ± 0.82	25.83 ± 0.92	81.39 ± 0.8	6.39 ± 1.34	34.2 ± 2.1	24.82 ± 1.11

Table 2: The centrality measures along with the errors evaluated from the noise model.

2.6 Clustering

The main body texts are used to cluster the physicists by means of natural language processing techniques. The main computational package used in this project is NLTK[2], which refers to natural language tool-kit, and scikit-learn[3], which is a standard machine learning package. The main idea is to use a bag-of-words approach where the texts are split into individual words, which we call tokens, and observe the frequencies of occurrences of each token then compare between documents to put them into different clusters according to different high-frequency keywords using Kmeans clustering from sk-learn.

2.6.1 Tokenization of Texts

The texts are first split into individual tokens, with their document index saved. It is worth noting that in the English language, their are many words that do not carry actual meaning, e.g. am, is, are, for, to, by.... Such words should not be considered as a valid token for the analysis so they are left out by storing them to a 'stopword' list that will be passed on to the tokenization and clustering algorithm. A standard English language stopword list is generated using NLTK package, however some extra words must also be appended to the list after observing the documents. Words such as physics, physicists, equation, formula etc. naturally appears in the majority of the biographies of physicists therefore they do not act as good tokens either and should also be excluded from the analysis.
Another pre-requisite is to consider the effect of English grammar, such as tenses and plurals. Without doing so, the same word in its singular form and plural form will be considered as two distinct tokens but they essentially carry the same meaning. The standard routine in NLP is to reduce the words into their stems by stripping away the suffixes of words, a classic algorithm to do so is the Porter Stemmer[4]. The Snowball Stemmer in NLTK follows from the Porter Stemmer algorithm and is initially tried. The method created two problems: Firstly, in the field of physics there are many academic or scientific words appearing frequently, which share the same stem as some other ordinary words, e.g. 'university' and 'universe' will both be reduced to 'univers', while the first words refers to a location and the second word links to astronomical study; 'electron' and 'electronic' will both be reduced to 'electron', while they carry very different meanings and may refer to completely different fields of physics study! An atomic physicist may have worked on particle collisions involving electrons and protons for a long time but has very little knowledge in the field of electrical and electronic engineering, and vice versa. The most heavily affected case is the mis-interpretation of words 'relate', 'relative' and 'relativity', which are all reduced to the stem 'relat'. The first word is a often-used verb, the second word is an adjective and the last word is a very specific physics topic. The algorithm messes the three and produced wrong clusters by relating all of them together. Secondly, to extract the keywords the stems must be re-converted back to their original form, the initial attempt was to save the original words and stemmed words into a data frame and use index searching to recover the word. However as previously mentioned, one stem can point to many different words in the data frame and it is impossible to determine which original word has the highest contribution to the frequency of this stem.
Having realised the drawback of the stemmer, the alternative method tried is lemmatization of texts. This method is similar to stemming but, unlike stemming, it considers the part of speech of the word and leave the word in its dictionary form. The lemmatizer used in this project is the WordNet lemmatizer[5] in NLTK, which uses its built-in morphy function to look up for the word in the dictionary.

	Output	Remove Tenses	Remove Plurals	Differentiate the Part of Speech
Snowball/Porter Stemmer	Stems	Yes	Yes	No
WordNet Lemmatizer	Words	Yes	Yes	Yes

Table 3: Comparison between the functionalities of stemming and lemmatization.

The significant advantage of this method is that not only it differentiates between the part of speech of the word and is able to separate academic words from ordinary words, but also it outputs complete words for direct display without having to map for original words like stemming so the aforementioned problems of the stemmer algorithm is automatically solved.
By comparison, the WordNet lemmatizer function is eventually chosen to transform the texts into tokens, which will be used in the next section.

2.6.2 Term Frequency–Inverse Document Frequency Matrix

The term frequency-inverse document frequency (tf-idf) is a measure of how important a word or a token is in a collection of documents which we call a corpus. It is proportional to the number of occurrences of a token in a specific document and is offset by the number of documents that contain this token.
The term-frequency (tf) is defined as follows:

$$tf(t,d) = \frac{f_{t,d}}{max\{\Sigma_{t',d} : t' \in d\}}$$

where $f_{t,d}$ is the number of occurrences of a term t in document d, and the denominator is the total number of terms in d.
The inverse document frequency (idf) is defined as follows:

$$idf(t,D) = log(\frac{N}{|\{d \in D : t \in d\}|})$$

where N is the total number of documents in a corpus and the denominator is the number of documents where the term t appears in.
Having known the above definitions, the tf-idf is defined as the product of the two:

$$tfidf(t,d,D) = tf(t,d) \cdot idf(t,D)$$

The logic behind this measure is that the more frequent the appearance of a term in a document, the better it can be used as a feature of the document, given that the term does not also frequently appear in other documents within the corpus. If it does have a shared high frequency in the majority of the documents, it is more likely that the term is a common word with no special meaning, so its importance goes down.
This project uses sklearn.feature_extraction.text.TfidfVectorizer[3] to convert the tokens generated from the last section to a sparse matrix, the tf-idf matrix, where the columns are document indices, the rows are term indices and so the entries represent the frequency of a term inside a document. The column vector is referred to as the document vector as it shows the frequencies of each word inside a document, so we can relate it to term frequency. The row vector is referred to as the 'word vector' as it illustrates the term frequency of a word in different documents, so we can relate it to inverse document frequency.

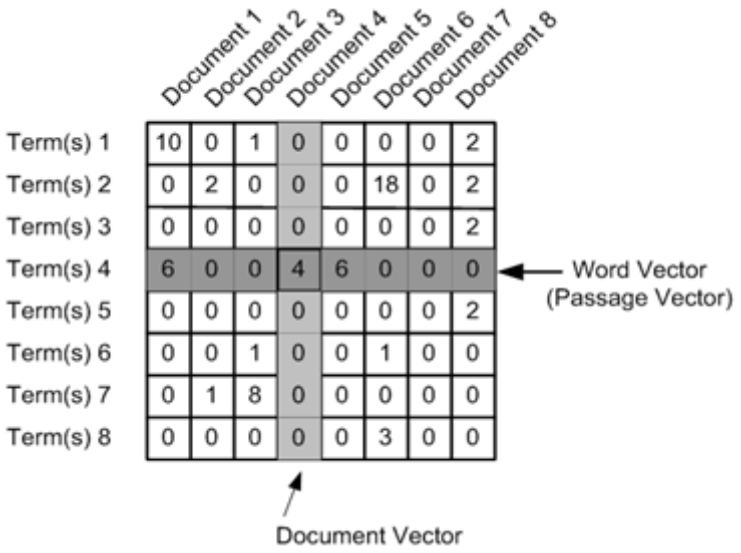


Figure 3: A visual representation of the tf-idf matrix[6].

2.6.3 Kmeans Clustering

Having calculated the tf-idf scores for all tokens, they are now ready to be fed in to the clustering algorithm. The unsupervised Kmeans algorithm from sk-learn is chosen to cluster the tokens. It assumes the following:

1. Define a 'cluster centroid' as the arithmetic mean of all the data points in the cluster.

2. Each point is closer to its own cluster centroid than other centroids.

To do this an initial number of clusters is put in and each point is assigned to a cluster label. The cluster means are calculated and used as the new cluster centroids. Then the data points are re-allocated based on these centroids and this process is iterated until the within-cluster-sum-of-squares (WCSS) value converges. The WCSS is calculated using the built-in algorithm:

$$W_k = \Sigma_{i=0}^{k} min(|x_i - u_j|^2)$$

where x_i are data points and u_j are cluster centroids.
The top keywords for each cluster can be extracted using the cluster_centres() method from sklearn.Kmeans class.

2.6.4 Optimising for the Number of Clusters Used

Before generating the outcome, one crucial question remained is choosing the most appropriate number of clusters to be used. Due to both the nature of this project and the flaw in Kmeans clustering algorithm, in principle the number of clusters should not be large to be indicative and easy to interpret. The range of choices of the number of clusters to be optimised,k, is then set to be between 2 and 10. Three different optimisation methods are tried: the Gap statistics, the Elbow method and the Silhouette scores.
The initial method tried is the Gap statistics. It measures the difference of the log of the WCSS score of the clusters based on our data and the expectation value of the log of the WCSS score of the clusters with the same k but based on a reference data set[7]:

$$GAP_n(k) = E_n^*\{logW_k^*\} - logW_k$$

where the * denotes the reference data set. It can be generated using the NumPy random_sample function. Here we use 3 random samples for each k and calculate their average to get the expectation of the reference data.

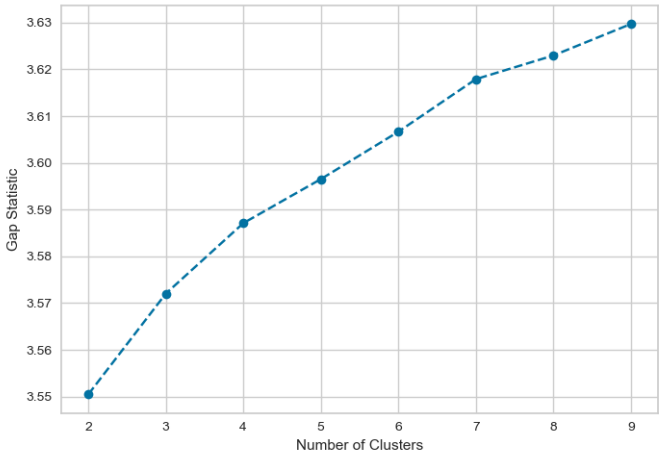


Figure 4: The gap statistics of the tf-idf matrix for different number of clusters.

As seen in the figure, the gap statistics diverge in the range (2,9) and even beyond. We tried running the algorithm in the range (2,30) and it is still divergent so this method is not indicative.

The silhouette score is another metric used to calculate the goodness of clustering, ranging between -1 and 1. It is calculated using the formula:

$$Silhouette\ Score = \frac{b - a}{max(a, b)}$$

where a is the average distance between each point within a cluster, and b is the average distance between a point and the nearest cluster it is not a part of. This metric can be evaluated in the code using sklearn.metrics.silhouette_score.

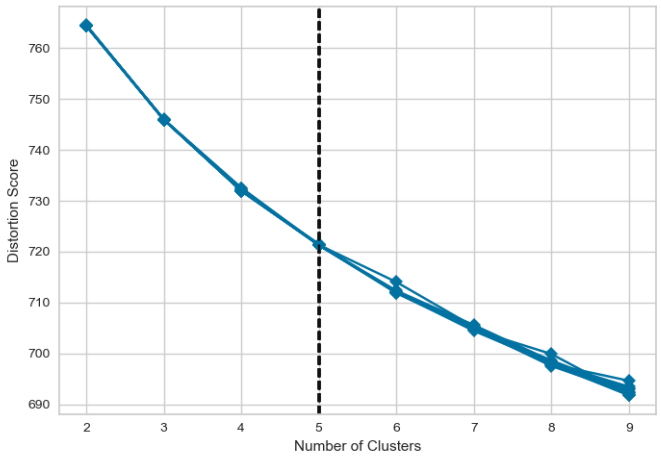


Figure 5: The output of 10 iterations of the Elbow Method.

However, the outcome of this method is highly unstable as Kmeans is a machine learning algorithm that has built-in randomness each time it runs, therefore the silhouette scores for different k vary significantly, even after taking averages of multiple loops so this method is also discarded. Example outputs of this method can be found in appendix B.

The last method we tried is the Elbow method. It determines the optimal choice for k by plotting out the distortion score against k, and locating the elbow, where the curve visibly bends, so that adding another cluster does not give much better modelling of the data. This solves the divergence problem of the gap statistics, allowing us to choose an appropriate k in the desired range. The outcomes of all 10 iterations of the Elbow method for the tfidf-matrix are stable and an elbow at $k = 5$ can be clearly identified. Hence, it is finally determined that the optimal number of clusters, k, to be used in this project will be 5.

2.6.5 Identifying Centroid Keywords

The centroid keywords are found by storing their corresponding indices to a dataframe before putting into the model, and matching the indices from the centroid terms from the model with the dataframe to see which words contributed to the features of a cluster.

The output from the clustering algorithm is shown below:

Cluster	Centroid Keywords	Include...
0	['become', 'german', 'name', 'article', 'study', 'institute']	Wilhelm Wien, Gustav Kirchoff, Paul Drude, Otto Stern
1	['quantum', 'theory', 'mathematics', 'mechanics', 'field', 'particle']	Wolfgang Pauli, Paul Dirac, Erwin Schrodinger, Paul Ehrenfest
2	['american', 'nobel', 'academy', 'medal', 'society', 'national']	Marie Curie, Carl David Anderson, John Bardeen, Tsung-Dao Lee
3	['nuclear', 'atomic', 'laboratory', 'project', 'war', 'german']	Enrico Fermi, Werner Heisenberg, Robert Oppenheimer
4	['experiment', 'theory', 'first', 'royal', 'year', 'father']	Albert Einstein, Isaac Newton, James Maxwell, Max Planck

Table 4: Example output from the clustering algorithm with number of clusters = 5.

3 Results and Discussions

We can now combine the results from the previous parts together and produce a clustered networks for the dataset with all centrality measures values and errors calculated. The following is a visualisation of the network using software GePhi:

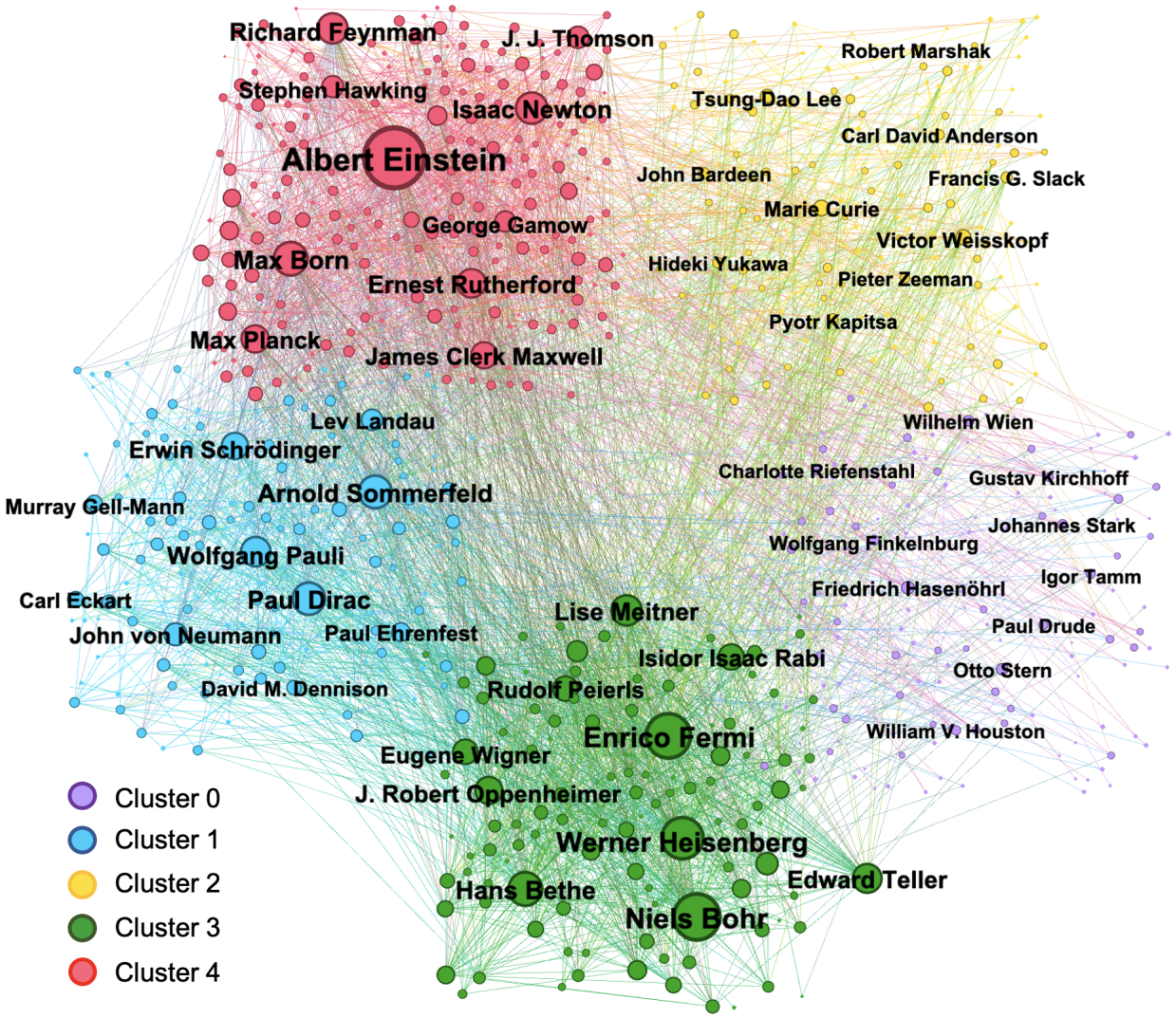


Figure 6: A visual representation of the network constructed between physicists. The size of the vertex is proportional to the average mark assigned to it. Different colours indicate the cluster index. Only the names of the top 10 physicists from each cluster are shown for clarity.

A spreadsheet recording all centrality measures, cluster index and ranks for all physicists can be found in the GitHub repository. A table showing the top physicists and their data can also be found in the appendix.

3.1 Comparing Clusters

It can be seen by the relative sizes of vertices that the clusters have a clear differentiation in overall importance. Cluster 3 & 4 have a larger number of important physicists, where cluster 0 & 2 appears to have much less popularity. This trend can be more easily spotted by a directed acyclic graph. Although the network we have constructed is not a directed graph, we can define a vertex to be in a higher hierarchy than another vertex if all of its centrality measures have larger values than the other one, and create a direction by pointing from the vertices with higher hierarchies to lower ones. In this way the different 'layers' of the network in terms of importance ranking can be shown clearly.

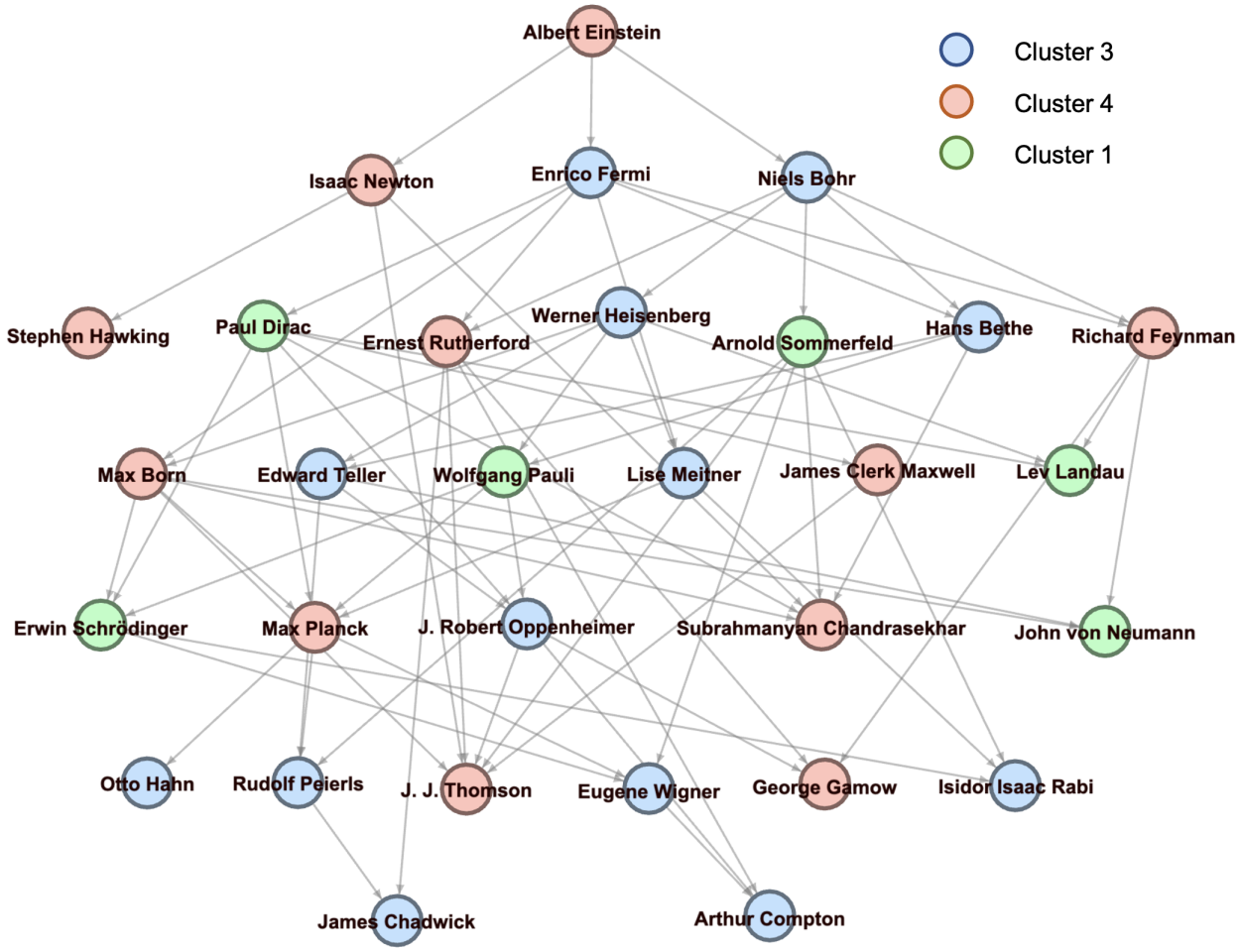


Figure 7: The directed acyclic graph of the top 30 physicists based on their centrality measures.

From Fig 7 we see that with no doubt Albert Einstein has the highest hierarchy so he is considered to be the most important physicist of all time, and all other physicists are shown. What is interesting is that of the top 30 physicists, no one is from cluster 0 or 2, which is similar to the observation made before. The statistical differences between clusters are plotted out below:

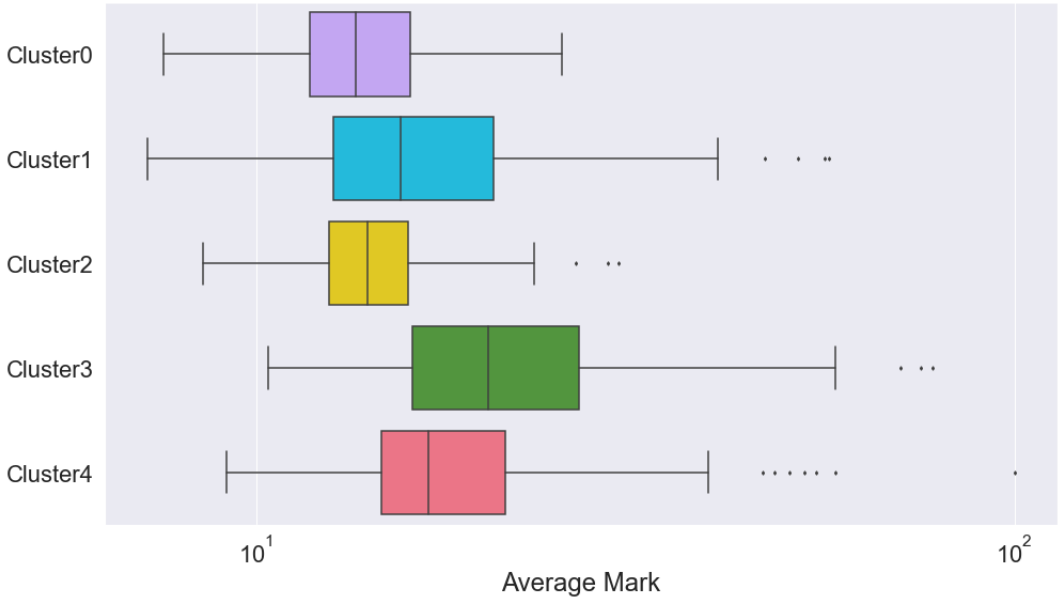


Figure 8: The boxplots of the distributions of average marks of physicists from each cluster. The horizontal axis has been set to log scale to make easier visual comparisons.

This statistical pattern must be understood by how the clusters are assigned. To do that we need to look into the details of the cluster centroid keywords from Tab. 4 or Fig. 9 below:

Figure 9: Word Clouds for the 5 clusters of physicists. The size of the text is proportional to its feature importance.

The group of physicists with the highest average mark is cluster 3, which mainly involves physicists who have worked during the Manhattan project on the invention of nuclear bombs against Nazi Germany and Japan in World War II. Oppenheimer is unsurprisingly selected as a feature keyword as he was the lead of the Manhattan Project, and also one of the top 10 most important physicists in Cluster 3. Other names appeared in Cluster 3 such as Werner Heisenberg, Niels Bohr and Eugene Wigner all had major contributions in quantum and particle physics in the 20th century, so we can say that quantum and particle physics form a large aspect of the modern physical research.

Similarly, Cluster 1 also has a strong emphasis on quantum and particle physics. Names appeared in this cluster includes Erwin Schrodinger, Wolfgang Pauli and Paul Dirac who all contributed hugely to the foundation of quantum understanding of the physics between particles and systems. This cluster has a stronger focus on theory rather than experiments and projects.

Cluster 4 is another strong group with many familiar names, most notably Albert Einstein, who is the most important physicist in the network and is likely to be the most popular choice when any person is asked this question. From the boxplot he is a major outlier to the group and his name is also used as one of the clustering feature. This cluster is harder to describe as the physicists within span quite largely in both fields of researches, location and time. Some of the features such as 'Royal Society' and 'Cambridge' do indicate a considerable proportion of British physicists in this group, including Isaac Newton and Stephen Hawking but geography is not a dominant feature. The word 'father' appeared as a major feature in this group, which may be interpreted that the physicists are mostly pioneers in some fields and claimed some major discoveries and is then named as 'the father' of something, which does not heavily overlap with quantum and particle physics are all grouped together. One exception of this interpretation is the presence of Max Planck in this group, who is usually considered as the father of quantum physics, so by the nature of his work he should suit in cluster 1 more but it would also make sense to put him in cluster 1.

Unlike above, cluster 0 has a strong geographical feature, with the word 'German' as the most important clustering feature. Indeed a big portion of the physicists in the cluster are German and this can be used to show the bias in the English Wikipedia Archive. The majority of the physicists recorded comes from an English speaking country (mainly the USA or the UK) so the inverse document frequency of a different nationality will be high. In addition, a German physicist had likely received higher education in a Germany university and/or worked in a Germany institute, given that Germany is a country with a strong background and resources on

physics research.

Cluster 2 is similar to cluster 0 as it has the word 'American' as a major feature, although less important than 'Nobel'. Clearly most physicists in the group had won an Nobel prize and are American or joined the American Physical Society since its name is also one of the features shown. Unlike cluster 0, there are physicists who were born and raised in a different culture, but are still grouped to cluster 2. This is because most of them came to the US for higher education later in life, as their original countries did not have so much resources on science like the West had. Beyond that, there is not much information about their fields of study.

Summarising all discoveries, 2 out of the 3 more important clusters have a clear common field of study, while the 2 weaker clusters depend much more on geographical features such as nationality. A hypothesis is proposed based on the findings:

- The more important a physicist is, the more stories there are in the biography to describe the person's achievement so keywords from the person's work dominates in the clustering process;
- Conversely if the physicist has less stories to tell in the biography, basic descriptions of the person such as nationality/country of birth/place of work/place of education will dominate.

To examine this hypothesis, we will need to see if there is a correlation between the lengths of web pages and the average marks.

3.2 Page Length and Importance

The lengths of web pages are found by reading the number of line breaks '/n' from the main body texts extracted earlier. The correlation plot between page length and the average mark is shown below:

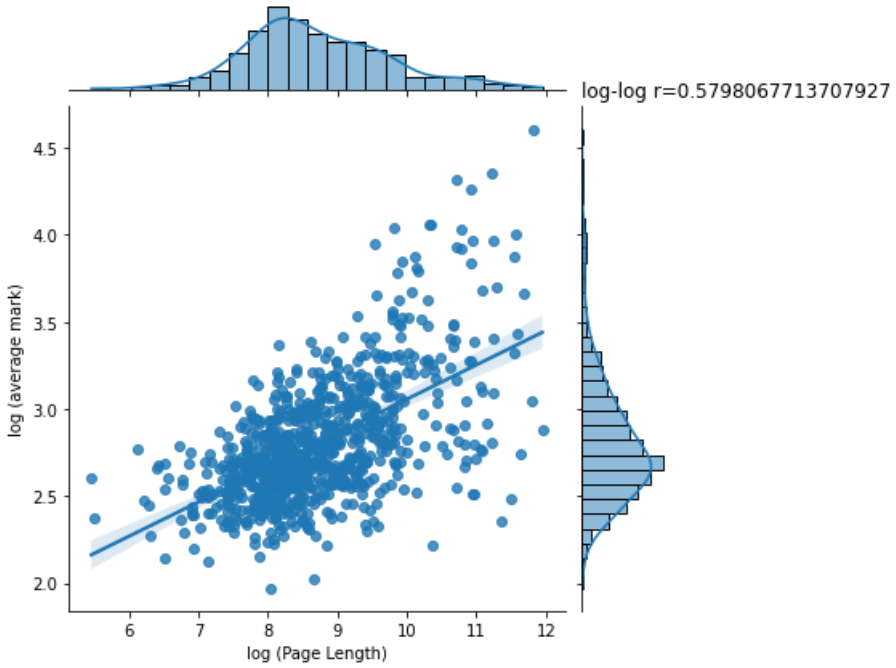
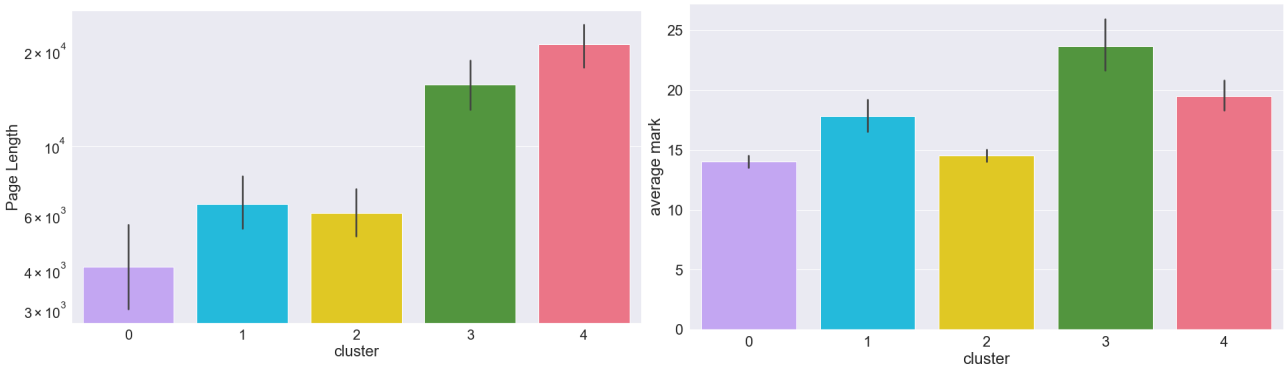


Figure 10: The correlation plot between Page Length and the average mark in log-log space of base 10.

With 1019 data points and a correlation coefficient of 0.580, the p-value for a correlation test is to the order of 10^{-74}, which is negligibly small, so we can conclude that there is indeed a power law relationship between page length and the average mark, i.e. $Average\ Mark \propto Page\ Length^{\alpha}$, where α is a constant.

In terms of the individual clusters, the correlation also varies in a similar way to how mean average marks among different clusters vary:



(a) (b)

Figure 11: Barplots of the means of (a). Pages Length; (b). Average Mark of the 5 clusters. Errorbars correspond to ±1 standard deviation. The y-axes for both plots are in log scale of base 10.

cluster	Correlation coefficient
0	0.2096
1	0.5922
2	0.3247
3	0.7261
4	0.4985

Table 5: The correlation coefficient between page length and average mark in log-log space for individual clusters, rounded to 4 d.p..

The correlation coefficients for the 3 'big' clusters are significantly bigger than the 2 others as shown in Tab. 5. Cluster 0 has the smallest correlation coefficient of all, this can be seen from Fig. 11 that cluster 2 and 0 have similar mean average marks but cluster 0 has much shorter biographies than cluster 2. This further explains why the centroid keywords for cluster 0 focus on geographical factors but does not have much information about areas of research, since in a limited amount of words, place of birth and study will have the highest term frequencies. It can be seen that the English language Wikipedia pages have its bias: the editors for English Wikipedia are more likely to know physicists who come from a English speaking country, or produced most of their works in English Also it is easier for editors to look up for resources describing the person in English, as they may lack the ability to read records about the person in another language, or they are not aware of a foreign source.

4 Conclusions

From the directed acyclic graph, the most obvious conclusion is that Albert Einstein is the most important physicist of all time with no parallel, on the basis of the Wikipedia data. The network has been tested to be robust from the noise model, as we see the percentage error for the centrality measures are small. However we have also seen drawbacks of using the Wikipedia pages, such as biases towards English-speaking physicists. The method of establishing edges using hyperlinks worked well but it does not give us a direction, as it is impossible to tell the true direction of the edges by looking at the contexts before and after, which would require natural language processing techniques far beyond the scope of this project.

The clustering algorithm produced sensible results as well, but it is harder to test and interpret its errors. The Kmeans algorithm is suitable for this type of large dataset with a flat geometry, so the main problem is the dataset again. Although attempts like using lemmatization instead of more standardised stemming were tried to reduce overlapping words, many words in academics carries different meaning to what they are normally used in ordinary life, such as 'state' or 'even'. This can be solved by using n-grams in tokenization.

Cross-validation could be another way to improve the robustness of our result. This can be done by either applying the same approach to another dataset, or just gathering results directly through surveys or other independent researches and observing the differences.

5 Acknowledgements

The author of this report would like to thank his project partner Kaiyu who devoted himself fully into this project and produced convincing results in the building of the network whilst being a good communicator and thinker who helped generate solutions to many problems during the project. Also the author would like to thank the supervisor of this project Dr. Tim S. Evans for his continuing support throughout the project and many resources and ideas he shared turned out to be useful when further exploring the directions of this project. Lastly the author would like to specially mention his parents and girlfriend who has given him many support in studying and living abroad and motivated him to carry on learning and working.

References

[1] Wikipedia, "List of physicists." `https://en.wikipedia.org/wiki/List_of_physicists`, 2022. Last accessed 09 March 2022.

[2] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009.

[3] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[4] M. F. Porter, "An algorithm for suffix stripping," *Program*, 1980.

[5] Princeton University, "About WordNet," 2010. WordNet. Princeton University. Last accessed 20 March 2022.

[6] B. Rose, "Document clustering with python." `http://brandonrose.org/clustering`, 2019. Last accessed 15 March 2022.

[7] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 2, pp. 411–423, 2001.

Bibliography

[8] B. Chen, Z. Lin, and T. S. Evans, "Analysis of the wikipedia network of mathematicians," *CoRR*, vol. abs/1902.07622, 2019.

[9] J. VanderPlas, *Python data science handbook : essential tools for working with data.* Sebastopol, CA: O'Reilly Media, Inc, 2016.

[10] H. Jabeen, "Stemming and lemmatization in python." `https://www.datacamp.com/community/tutorials/stemming-lemmatization-python`, 2018. Last accessed 10 March 2022.

[11] I. D. Baruah, "Cheat sheet for implementing 7 methods for selecting the optimal number of clusters in Python." `https://towardsdatascience.com/cheat-sheet-to-implementing-7-methods-for-selecting-optimal-number-of-clusters-in-python-898241e1d6ad`, 2020. Last accessed 18 March 2022.

[12] M. B. Ross Barnowski, Riccardo Bucco, "NetworkX: Network analysis in Python." `https://networkx.org`, 2022. Last accessed 21 April 2022.

Appendix A: Github Repository

The github repository containing all written code, initial data, outputs graphs and spreadsheets of this project can be accessed by the following link: `https://github.com/Kaiyu-cpu/BSc-Project-Complexity-of-Physicists`

Appendix B: Silhouette Score Outputs

Here are some example outputs from repeating the silhouette method for optimising the number of clusters used. The method is highly unstable hence discarded from consideration.

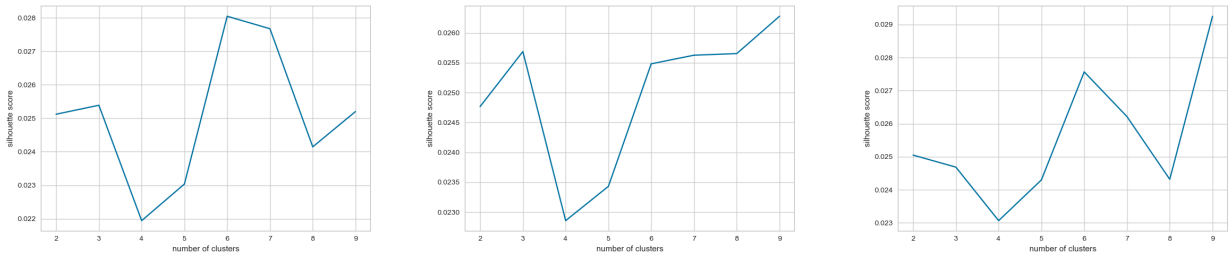


Figure 12: Example of different attempts of the silhouette score method.