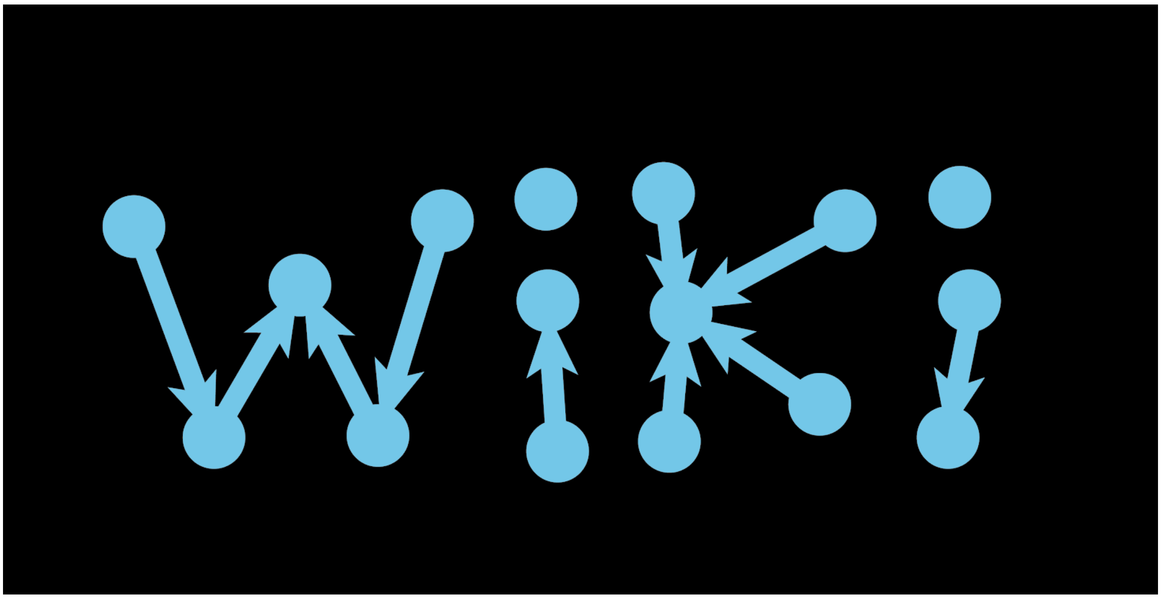
Wikipedia Analysis Using Keyword Based Graphs

Network Tour of Data Science Project Report



Team 24:

Marc Glettig - Matthias Minder - Yves Rychener - Charles Trotin

github.com/mattminder/wikilinks

18/01/2019

Executive summary

The Wikipedia network is a directed graph which contains Wikipedia articles as nodes and an edge between nodes A and B if the article A contains a link to node B. This project investigates whether it is possible to accurately recreate a subset of the Wikipedia network based on article text similarity measures. In particular, we generated a term frequency inverse document frequency (TF-IDF) matrix, based on which we inferred networks using three different methods, based on cosine similarity (COS), euclidean distance (DIST) and k-nearest neighbors (KNN). By comparison with the original Wikipedia network, the following was observed:

- Computing the page rank algorithm showed that 4 of the predicted top 10 most important articles are in the top 10 of the original network.
- The correct edge comparison showed a poor link prediction with only about 23% correct edges predicted.
- The network constructed based on KNN is the closest to the original one compared to the other we constructed based on average degrees and number of edges.

Finally we used the inferred KNN network to suggest new links between sites. Individual analysis of the proposed links show that they make sense. This algorithm could provide a valuable tool for Wikipedia development, where authors are suggested to add links to other sites.

Introduction & Problem definition

Wikipedia has grown to be one of the largest sources of information and is used by a very large community. One of its strong points is the fact that the articles contain the relevant information as well as links to the background information for more in-depth reading. It is therefore of interest that the links are of good quality, meaning that they link relevant articles. However, since sites are written collaboratively between many authors, it is thinkable that some links are redundant whereas others are missing.

Within the scope of this project we will focus on the latter, missing links between sites. The aim will be to construct a network based only on the texts of the articles. This will be done using different network construction methods, resulting in both directed and undirected networks. We will then investigate the properties of the constructed networks and compare them to the original network. Finally, we will try to identify missing links in the original network, and propose new links between sites.

Since the task would be too computationally intensive to be done for the entire Wikipedia network, we will focus on a subset of 4’549 articles of the English Wikipedia site, based on the SNAP dataset[1].

1. Graph construction based on text mining

a. Article retrieving using Wikipedia’s API

Using the python Wikipedia library[2] (which wraps the Wikipedia API), we retrieve the article texts. Some nodes of the network refer to disambiguation sites and our extraction fails. Since

[1] "SNAP: Web data: Wikispeedia navigation paths." https://snap.stanford.edu/data/wikispeedia.html. Date de consultation : 18 janv.. 2019.

[2] "wikipedia · PyPI." https://pypi.org/project/wikipedia/. Date de consultation : 18 janv.. 2019.

we do not expect the disambiguation sites to contain useful text features (they just explain the different meanings), we decided to not include those sites in our analysis.

b. Text mining for article similarities

The article contents are translated into so-called TF-IDF values, which reflect how important a given word is in an article within a collection of articles. This approach was chosen since we argue that articles in which the same words are given a high importance score will tend to be similar. The TF-IDF values were obtained as follows:
Step 1: Text cleaning. A pre-cleaning was done in the article retrieving in order to keep only the text. As we focus on keywords we first removed every stop words ('that', 'because', 'what'...). Then we stemmed our text to remove suffixes. Note that with Wikipedia articles we work here with an "easy" text without misspelling and mistakes.
Step 2: Bag of words representation. As we want to focus on keywords we will for our entire corpus a collection of individual words using spaces as out separation.
Step 3: Term Frequency calculation. For each document we then compute the frequency (number of occurrences) of each word in our bag of words.
Step 4: Inverse Document Frequency. In order to have accurate similarities we want to put more weight on keywords that appear in few articles. We measured the rarity of a word w using the inverse document frequency (IDF):

$$IDF(w) = 1 + log(\frac{Total\ number\ of\ documents}{Number\ of\ documents\ containing\ w})$$

Step 5: Combining TF IDF. For a given document and a given word we have TF*IDF.
These steps were implemented using the scikit-learn package in python.

c. Graph construction based on articles similarities

Based on the TF-IDF matrix we obtained a graph using three different methods. In order for the constructed network to be similar, we always constructed the networks such that it will contain the same number of edges between the sites.
In the first approach, the TF-IDF matrix was turned into a similarity matrix simply by multiplication by its transpose. Since the rows are normalized, this process yields the cosine similarity measure, as it is equivalent to calculating the cosine of the angle between two row vectors corresponding to articles. The edges with the largest scores were then retained in order to obtain a network.
As a second way to construct the network, we used the features given by TF-IDF to build a k-nearest-neighbor graph. This algorithm creates an edge between an article and its 26 closest neighbors in terms of euclidean distance. The result is a directed graph where every node has the same outgoing degree equal to k. kNN was performed using the kneighbors_graph function of scikit learn[3].
The third way for network construction is also based on Euclidean distance in the feature space of the TF-IDF matrix. In this algorithm, we computed the pairwise distances between articles using the pw_dist function of scikit learn. We then ordered the distances and retained the smallest distances as edges in our network. The result is an undirected graph with no constraints on the degrees of individual nodes.

[3] "Scikit-learn." https://scikit-learn.org/. Date de consultation : 18 janv.. 2019.

2. Comparison of Constructed Networks with Original

After having constructed different networks, we will now compare the constructed networks to the original.

a. Comparison using Pagerank algorithm[4]

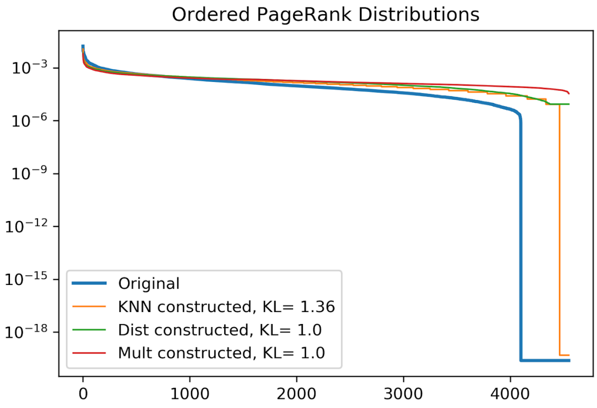
Since our original network is a hyperlink network, it makes sense to calculate the pagerank probabilities.

The pagerank algorithm looks at the network as if it were a Markov chain. At each node and timestep, the agent navigating the chain will move to its neighboring nodes with equal probability. When the agent reaches a sink node (a node with outdegree 0), the agent moves to a random node with equal probability. Also, sometimes the agent will move to a random node, simulating a new landing on Wikipedia pages. The pagerank algorithm calculates the leading eigenvector of the transition matrix of the chain described above to estimate the stationary distribution of the agent on the Markov chain. This can be used to order the nodes by importance. Moreover, we will also use it also to calculate the KL-divergence between the stationary distributions of the different networks to measure their similarity with respect to a random surfer. We also plot the (ordered) Pagerank probabilities, in order to estimate the difference between the stationary distributions of the random surfer.

Table 1. Most Important Articles according to PageRank

Original	**KNN**	**Distance**	**Multiplication**
United States	United States	United States	United States
United Kingdom	Modern history	Modern history	Modern_history
Scientific classification	United Kingdom	Bird	Bird
Europe	England	United Kingdom	United Kingdom
England	20th Century	New York City	New York City

Figure 1: Ordered PageRank probabilities and KL-Divergence



Looking at the important articles and pagerank probabilities (see Table 1 and Figure 1 above), we see that our results fit quite well in terms of stationary distribution. The KL-distribution suggests that KNN is slightly inferior to the others.

[4] "The PageRank Citation Ranking: Bringing Order to the Web. - Stanford ...." 28 déc.. 2008, http://ilpubs.stanford.edu/422/. Date de consultation : 18 janv.. 2019.

b. Comparison of Degree Distribution

Since the networks were reconstructed to have the same amount of edges (up to rounding errors) as the original, it is also possible to compare their degree distribution.

Figure 2: In-Degree Distribution

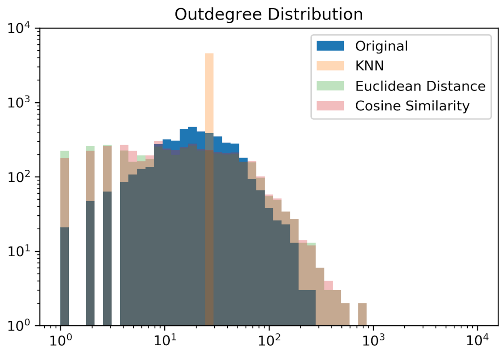
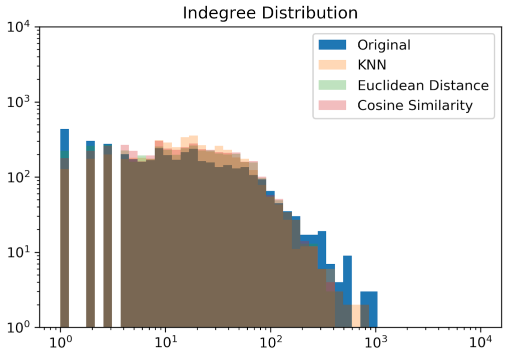


Figure 3: Out-Degree Distrubtion



The degree distributions shown in Figures 2 and 3 suggest that the distance based and cosine similarity networks are very similar. Due to its nature, the KNN constructed graph performs poorly in out-degree distribution, since every node has a fixed amount of nodes (K=26) it points to. This drawback could be avoided using a dynamic K, which could for example be estimated using a regression model from the text length.

However, there seems to be an asymmetry between the in- and out-degree distribution in the original network, as seen by the different shapes of the blue curve in Figure 2 compared to Figure 3. This asymmetry can only be captured by a directed network, since undirected networks have as many outgoing as incoming edges. However, due to the out-degree constraints of KNN, our only directed approach cannot capture this asymmetry either. It would therefore be of interest to use more sophisticated directed network inference techniques, such as with a dynamic K, in order to better recreate the out-degree distribution.

c. Direct Edge Comparison

Another straightforward way to compare the networks is the compute the percentage of identical edges between the reconstructed and the original edges. The percentages are 23.92% for KNN, 23.60% for the distance based and 22.84% for the cosine similarity based network construction methods.

Those results suggest that the methodology is not accurate for network recreation. This is most likely because Wikipedia authors include cross references for the reader to have a large view point on a subject. However, the difference between the actual and created networks could be interesting and will be used for link suggestion in the following section.

3. Link Suggestion

Finally, we used the constructed networks to suggest new links between Wikipedia articles. Since all networks behave similarly when compared to the original network, we used KNN. This approach was chosen because it has the advantage that every node is assured to have the same amount of outgoing edges. In this way we want to assure that all articles will get link suggestions, and not only the ones that have a central position in the TF-IDF feature space. In order to assure that we suggest the most important links, we computed the

pairwise distances between articles. Then, we identified the edges with the lowest distances that were present in the KNN network, but not in the original. We retained a total 13’647 edges in order to suggest three edges per article on average.
Unfortunately, there is no good performance measure to assess the quality of the predicted edges. We therefore manually assessed the quality of random subsamples of the suggested links. Such a subsample can be seen in Table 2.

Table 2: Five Suggested Links Selected at Random

Number	From	To
1	Avacha Volcano	Galeras
2	Byzantine Empire	6th century
3	A Tale of a Tub	Augustan literature
4	Post-glacial rebound	Sea level rise
5	Lake Chad	Lake Superior

We observe that the proposed links are all between related topics, a desirable property. Especially the links with number two to four seem to link a more specific topic to a more general category, which is interesting to the reader of the article. On the other hand number one and five are less interesting, since they just link two volcanos (number one) and two lakes (number five) with each other. This is likely because they have similar characteristics, but since they are located in different geographical contexts, a link between them is less interesting.
This shows that link suggestions will still have to be assessed by a human before being integrated. In practice Wikipedia could propose authors to integrate certain links when adding to articles. One could further imagine a more sophisticated supervised learning implementation, where a model tries to learn the quality of an edge based on more features such as the article categories or multiple similarity measures.

Conclusion and criticisms

We saw that the constructed networks were similar in terms of degree distribution to the original network. Direct comparison of the created links however showed that only about 23% of the edges were the same of the different methods with the original. Asymmetry inward and outward degree distributions of the original network however indicate that directed networks will better recreate the original network properties. Application of more sophisticated network inference methods could thus be of interest for improving the network reconstruction.
Moreover, the predictive performance could be improved by using different text representations than TF-IDF. For instance, a well-known drawback of the TF-IDF method is due to the fact that we study each word individually hence we lose important groups of words such as “Game of Thrones” which make sense when considered as one block.
Finally, text similarity is not the main reason to include hyperlinks. Authors will include links that they find relevant for their reader and they do not really consider similarity. This is in our opinion the main reason for the poor network recreation. In order to improve performance, one will have to incorporate other article metadata than its text.