Extracting Hub and Authority Scores for Wikipedia Pages using Apache Spark

KIARASH GOLZADEH, University of Waterloo, Canada

Having sight of the architecture and structure of links between entities in a large network is a fundamental requirement for performing deep analysis on it. The hub and authority scores are two useful metrics for assessing the contribution of each node in a graph. In this research, we used Apache Spark, which is a distributed computing framework, to calculate these scores for articles on Wikipedia using the HITS method, display the leading pages in each score, and determine their likely underlying relationships.

Additional Key Words and Phrases: Network Analysis, HITS Algorithm, Hub, Authority, Wikipedia, Apache Spark

ACM Reference Format:
Kiarash Golzadeh. 2022. Extracting Hub and Authority Scores for Wikipedia Pages using Apache Spark. 1, 1 (December 2022), 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Encyclopedias are generally considered a starting point for every attempt of searching for information. As a result of recent advances in computer and data systems, online encyclopedias have been emerged and used widely. Wikipedia is a free, online encyclopedia where everybody is permitted to voluntarily add and edit entries, without the need for official, strict review processes. This will lead to a wide range of articles on many topics, with high or low qualities.

Wikipedia articles mainly consist of hyperlinks to other related wiki pages. In other words, every word that points to a relevant concept from another article is an anchor to that page. This creates a massive graph network of articles, where every article is a node, and links between articles are directed edges.

To have a good intuition of the graph structure and find out the most important and valuable nodes, various metrics have been introduced and discussed by network analysis methods, such as PageRank and HITS. In this project, the HITS algorithm and its output metrics (hub and authority scores) are used. In a nutshell, as a result of this project, we want to find out the top pages on Wikipedia with the most hub and authority scores, and visualize them in clusters to gain a broad view of Wikipedia's structure. In section 2 the details of the HITS method are described. After that, the properties of our used dataset are stated in section 3. Then, the steps of coding the framework are described in section 4. The attributes of the executor nodes from the cluster come in section 5. Finally, the process of evaluating the metrics and results of the project are displayed in section 6.

2 HITS ALGORITHM

The HITS[1] algorithm, is a popular link analysis and ranking algorithm for documents, based on the network edges[2]. The main idea of HITS is that informative web pages mostly consist of two types of data: good information about the desired topic, or good links to other pages. This iterative algorithm produces two scores for each document, hub score and authority score. A hub is a node with a high hub score, and an authority is a node with a high authority score. These metrics reinforce each other in every iteration so that a good hub would point to good authorities, and a good authority would have links from good hubs. To achieve that, the hub score of a node is set to the sum of authority

[1] Hyperlink-Induced Topic Search

Author's address: Kiarash Golzadeh, University of Waterloo, Waterloo, Canada.

2022.

scores of its outside neighbors. Similarly, the authority score of a node is set to the sum of the hub scores of its inside neighbors. More formally, after each iteration, the authority score of node v a_v would be equal to

$$a_v = \sum_{(u,v)\in E} h_u \tag{1}$$

where h_u is the hub score of node u. Plus, h_v, the hub score of node v would be equal to

$$h_v = \sum_{(v,u)\in E} a_u \tag{2}$$

Finally, after updating the metrics for each node in an iteration, the scores are normalized so their square sums are equal to 1.

3 DATASET

Throughout this project, the Wikipedia pages were analyzed. A dump image of wiki pages and their links had become available by Wikimedia. Here, we used the page-to-page links from 2018; however, more recent data are available on the Wikimedia website[1].

In the aforementioned dataset which is a text file, an ID is assigned to each article, and in each line, there is a node ID and its adjacency list. The size of the dataset is large; it contains about 5 million nodes and 128 million edges, and running the HITS algorithm on this graph isn't feasible on a single computer. Hence, the Apache Spark framework would come to the rescue.

4 PROPOSED METHOD

To leverage horizontal scaling and distributed computing, a Spark job was developed in Scala language.

First of all, the graph adjacency list was parsed, and the incoming edges for every node were calculated using a map on the edges and a groupByKey, where the key is the endpoint. Then, each node id and its incoming and outgoing edges for each node were gathered using a full outer join and stored in a case class called Node. Furthermore, another case class named NodeAttribute was created to contain the hub and authority scores, related to each node id. As the HITS algorithm runs, instances of Node class aren't changed, unlike NodeAttribute instances.

After creating the graph structure, the iterations of updating hub and authority scores would start. To update the authority score, we know that the authority score of every node is the sum of the hub score of its outside neighbors. So in other words, the hub score of every node will take a part in the authority score of its inside neighbors. Thus, for each node v, the pair (n, h_v) is generated for each in-neighbor n. These pairs are reduced by key to form the new authority score of each node, and the result of this reduction is an RDD of (v, a_v) pairs. After normalizing the authority scores (by mapping each score to its square, summing all scores, calculating the square root, and mapping each value of the RDD to its division by the normalization factor), the NodeAttributes are updated in their authority field by first creating an RDD of pairs $(id, nodeAttribute)$ for each NodeAttribute instance, then performing a left outer join on the old attributes and the new authority values (and putting 0 for null entries), and finally mapping each NodeAttribute instance to a new copy of it, with having the authority field changed.

The process of updating the hub scores is similar to the mentioned procedure.

5 RUNNING THE JOB

The Spark job was run on the Datasci cluster from the University of Waterloo using 10 executors, each with 4 gigabytes of RAM and 2 cores, and with 4 gigabytes of RAM for the driver program. The running duration of the HITS algorithm on Wikipedia after 20 iterations was about 1 hour. However, during the development, some memory errors occurred which were solved by increasing the memory.

6 EVALUATION AND RESULTS

To check the correctness of generated results, a simple Python code was written for a trivial implementation of the HITS algorithm. Then, the Spark job and this checker code were run on a smaller dataset and their results were investigated to be the same.

After calculating the hub and authority scores, a spark job was run to find the top 100 hubs and authorities. Their related page titles were found and added to the result by joining the titles file by page id. After that, the results were analyzed. The top hub pages were mainly about cities in United States. Such as:

- Westwood, California
- Saginaw, Michigan
- Gardena, California
- March Air Reserve Base
- Gardiner, Maine
- ...

Which shows that these pages contain links to articles for things that related to the cities, their states, and their country (USA). These pages were "hubs" to conduct viewers visit the most important things about their cities.

On the other hand, the situation was different for authorities, and they were from various categories. To have a better visualization, we intended to find an embedding for each page, group them into clusters, and visualize them on a scatter plot. A small script was written in Python to fetch and store the summaries of the candidate authorities from Wikipedia. Then, a BERT model (bert-base-uncased) was applied to generate an embedding of the summaries as a vector with 768 dimensions. The embeddings were clustered using KMeans algorithm, where the optimal number of clusters were found to be 6. Then, using the TSNE method, the features were reduced to 2 to create a 2D plot from the page titles, where the colors show different clusters. The plot could be viewed in figure 1. We can figure out from the graph that many authorities are countries and US states. This shows that the hubs (which were the cities) probably navigate us to the events and facts about a larger place. Moreover, other authorities are cheifly about different races, the US Census, articles about population welfare (poverty line, per capita income, ...), and group events (Football, Baseball, World Wars, Senate).

7 CONCLUSION

Knowing the characteristics of the entities and their relations in a large graph would lead the researchers to realize the underlying interconnection. Encyclopedias such as Wikipedia are a great resource of separate units, providing paths that connect each subject to others. Two helpful metrics in evaluating the contribution of each node in a graph are the hub and authority scores. In this project, we computed these scores using the HITS algorithm with the help of the Apache Spark framework, visualized the top authorities, and figured out the probable relations between them.

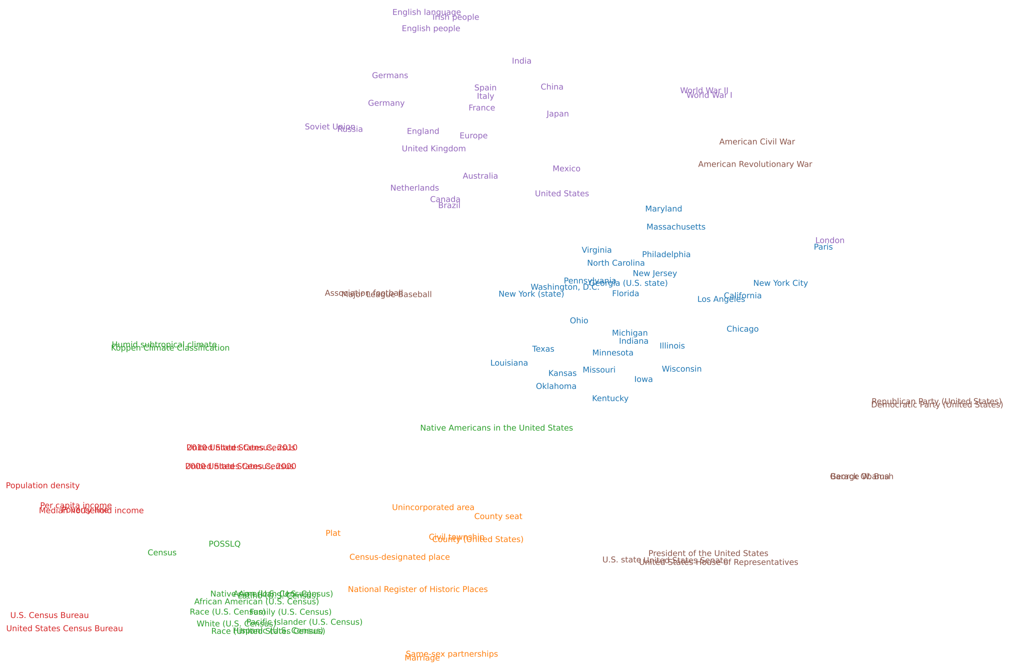

Fig. 1. Plot of the top 100 authority pages on Wikipedia

REFERENCES

[1] Wikimedia database backup dumps. https://dumps.wikimedia.org. Accessed: 2022-12-14.

[2] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, sep 1999.