Deep into the DBPL citation network

Nour Ghalia Abassi
nour.abassi
@epfl.ch

Khalil Mohamed Cherif
mohamedkhalil.cherif
@epfl.ch

Aymen Gannouni
aymen.gannouni
@epfl.ch

Abstract

The DBLP comupter science bibliography is a handy collection of scientific publications, mainly in the field of computer science. It started as a small project back in 1993 to test html files, but it grew so fast to gather over 3 million journal articles, conference papers, and other publications on computer science. DBLP is not only about publications, but also the authors and the whole community of researchers. This work aims at detecting the communities in the network of authors and finding the most influential publications in DBLP.

1 Introduction

In a data-driven era, researches have been done to detect the present communities in a social network and analyze the interactions between these. The DBLP computer science bibliography is a rich source of metadata on publications about computer science. In this work, the social network of authors will be analyzed to find the present communities in DBLP. Moreover, the network of citations will be studied to reveal more insights on the most influential publications.
The main goal of our project, is to address the following research questions:

- What are the communities of the DBLP social network of authors?
- Is it possible to cluster the citation network based on term similarity?
- What are the most influential publications in DBLP?
- How is the evolution of citations over the last decades?

In order to tackle these questions, this paper is structured as in the following.
The subsequent section 2 tells about the previous works that have carried out in the context of analyzing the DBLP computer science bibliography. In section 3 the used dataset is presented along with the preprocessing tasks that have been done to ensure a higher quality data as an input for the data analysis process. The section 4 constitute the part where all the algorithms used for the data analysis are illustrated. Besides, this section highlights the results for each algorithm. At last, section 5 handles the conclusion of this work with a brief discussion of the future work.

2 Related Work

During the past years, network analysis is gaining more and more interest thanks to the wide range of applications where it can be applied. In particular, social network analysis is arousing even more interest with all the social media networks that evolved in the last decade. Considering the DBLP use-case, a social network depicts the authors of publications as nodes and the co-authorships as edges between these nodes. Finding communities in such networks is one of the prominent tasks in social network analysis. In this context, some works have been done as in [ZCG07] and [BD10].
The DBLP bibilopgraphy is hosted at the university of Trier in Germany and Michael Ley is considered to be one of the major contributors, who greatly helped to maintain the DBLP database. He also has some publications that describes the structure of DBLP system and gives advice on how to extract and handle the DBLP data. His publications [Ley02] and [Ley09] were very helpful in the course of the project.

3 Data Collection

3.1 The DBLP dataset

For the sake of the project, a dataset from Aminer, namely DBLP-Citation-network V10, was used. It is in a JSON format and contains 3,079,007 papers published until 27^{th} of October 2017 and 25,166,994 citation relationships with a total size of nearly 4Gb. The next table shows the general structure of the dataset.

Field	Type	Description
`abstract`	string	paper abstract
`authors`	list of strings	paper authors
`id`	string	paper id
`references`	list of strings	references ids
`title`	string	paper title
`venue`	string	paper venue
`year`	int	published year

Table 1: Font guide.

3.2 Data Preprocessing

As any real-world data, the dataset contained many inconsistencies such as missing values and duplicates. Totally 530475 abstracts were missing in the used dataset. Another challenge was that the authors could not be uniquely identified due to the name duplicates such as 'Wei Wang' who refers according to the DBLP website to over 131 different persons. Therefore, a cleaning strategy was crucial to cleanse the data from its dirt. Common data cleaning practices consist of filling in the missing values and removing duplicates. In our case, these tasks were very challenging and time-demanding, but this is more of a classic for data science projects, where data preprocessing is worth to plan for in advance.

Given the data quality that plays a major role in delivering more accurate results, the DBLP API was used at an early stage to fill in the missing values and correct the names of the authors with the help of the unique enumerated names provided by the API such as Wei Wang0001, Wei Wang002 etc. for different authors having the name 'Wei Wang'. For the missing abstracts, the API helped to get the source link of a paper that was scraped in order to extract the abstract. A total of 300000 from 500000 abstracts were collected during this process which involved the web scraping of different publishers.

However querying over the API was time-consuming due the API rate limit and the limited number of papers that could be queried in a certain period of time.

Fortunately, the DBLP website provided a way to download the whole metadata contained in the website as a XML file. Considering the relatively large size of the file, event-driven iterative parsing was used to get quicker results and avoid memory overload. Using the XML file, the names, titles and links were extracted in order to be matched with the initial dataset.

Scraping the websites of over 500000 links to get the abstracts was not a simple task and required a rethought way of execution. Since it would take a lot of time to analyze the responses of so many HTTP requests, multi-threading was used at a first stage along with asynchronous requests to optimize the time performance, but this was not a very stable solution. Hence only asynchronous requests were used and this has reduced the overall processing time in comparison to regular requests by averagely 5.

4 Data Analysis

4.1 Building the networks

An essential step to start the data analysis, is to build both the network of authors and the citation network. For the last, each paper id is added as a node to the graph then for each cited paper in the references, an edge between from the paper to the cited paper is built. Almost the same is done to the authors graph, in fact the authors are firstly added as nodes to the graph and then for each publication an edge between each author and the co-authors is built.

The next table shows some basic properties of both networks:

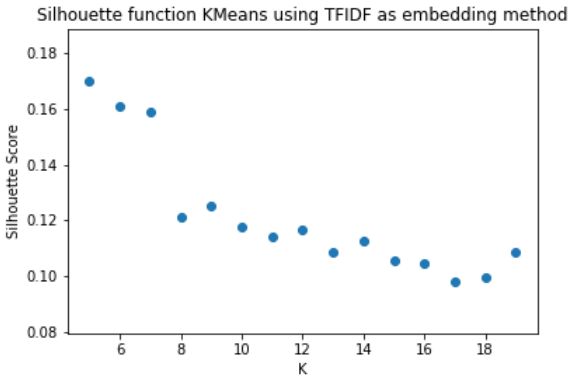
	Citations	Authors
Type	directed	undirected
#Nodes	3079007	1729816
#Edges	25166994	29448792

Table 2: Network characteristics

4.2 K-means clustering

Clustering is the task of grouping a set of objects into smaller subsets, also known as clusters. Each cluster is characterized by the similarity between

the objects within the subset, and dissimilarity to objects from outside the subset. One of the well-known algorithms, is K-means clustering, which is a partitional clustering algorithm with k as the number of clusters that to be given in advance.[HW79]
In this project, we collected the abstract, title and venue of each paper in a corpus and computed the corresponding term frequencyinverse document frequency(TFIDF) vector, which is a popular statistical measure for term-weighting.
Furthermore, the Global Vectors, also known as GloVe vectors where computed using the pre-trained word vectors from [reference].
K-means clustering was applied on the dataset using the aforementioned term representation models(TFIDF and GloVe) and different k values that range from 5 to 19. In order to determine the best suitable k value, the silhouette score of each clustering was calculated to measure the clustering quality of each setting. The following figure shows the silhouette scores for different k values using TDIDF as embedding model



Considering the plot figured above, the best value k to choose for k-means clustering is 8, since it has the lowest silhouette score value. On one hand, the 8-means clustering using the TFIDF measure yields 8 clusters with uniformly distributed points. When checking the document, they all seem to be related in each cluster, networking, speech recognition, data visualization... On the other hand, 8-means clustering using GloVe vectors returns clusters of papers written in different languages: One of the clusters contains mainly french documents with some in spanish, another one only german documents, so gloVe was able to encapsulate the semantics of the languages, but for the 6 other there is no apparent reason for them to be in the same cluster. So we decided reducing the number of K to see if we can find 4 clusters for example each for a certain language, but 3 of them were mainly in english and one in German. So to get a good separation of the languages we had to increase K, but this also led to clusters in which the element were unrelated based on their topics.

4.3 Label Propagation Algorithm

The Label Propagation Algorithm(LPA) is a popular community detection algorithm with an almost linear time performance. It does not yield a unique result, but an aggregate of many results. [RAK07] The algorithm starts by assigning a unique community label to each node and then diffuse the labels through the whole graph. Membership to a certain community changes then based on the labels of the neighbours. This results in quick labelization of high-density connected nodes as a single community. When many dense groups start to evolve, community labelization is expanded outwards until all nodes are reached.
Applying LPA to both networks of citations and authors led to the following results:

	Citations	**Authors**
#Communities	502654	255797

Table 3: Label Propagation Algorithm results

4.4 Page-Rank

Page-Rank, or the secret behind Google success since the debuts to deliver the most relevant web links based on the user query. It is a link analysis algorithm that assigns each node a score based on its relevance in the network. The algorithm is mainly designed for directed graphs, where degree centrality measures such as indegree and outdegree have a strong impact on the algorithm outcome. Generally, the higher the outdregree of a node, the higher it can get ranked by the algorithm. Even higher a node can get ranked, when its neighbours have a high Page-Rank score.[Pag+99]
Considering the large size of the dataset, the Page-Rank algorithm was applied on the citation network using a Spark cluster. The results address one of the most interesting research questions: Which are the most influential papers in the DBLP database ? The answer is represented in the table on top of the next page of the paper.

Paper title	Year	Page-Rank
Finite automata and their decision problems	1959	2018.18
The reduction of two-way automata to one-way automata	1959	1820.10
The Design and Analysis of Computer Algorithms	1974	1396.06
The complexity of theorem-proving procedures	1971	1364.04
New Directions in Cryptography	1976	1238.15
A method for obtaining digital signatures and public-key cryptosystems	1978	1015.43
Genetic Algorithms in Search, Optimization and Machine Learning	1989	1010.77
Computability and Unsolvability.	1959	993.98
Reducibility Among Combinatorial Problems	2010	939.41
Picture Processing by Computer	1969	857.18

Table 4: The 10 most influential publications in DBLP

4.5 Strongly connected components

In directed graphs, a node is called strongly connected when each node can be reached from any other node. Strongly connected components are maximal subgraphs of nodes that are strongly connected between each others. Any node that breaks the property can not considered in the strongly connected partition. In fact, strong connected components of a graph can be computed in linear time.[Sha81] The strongly connected components of the citation network were computed using the cluster.

	Citations	Authors
#Communities	63572	36134

Table 5: Label Propagation Algorithm results

5 Conclusion

Analyzing the DBLP network was certainly a challenging task, that was tackled thanks to best practices of data science from data cleaning to data analysis. The DBLP shows over 250000 communities of authors, who contributed together in the technology growth of computer science. Through this work, some of the most influential publications were revealed to see how publications from the past are still very significant nowadays.
Future work to this project includes fine-tuning the data by adding more information to the papers and survey the graph analysis algorithms that can be applied to both networks.

References

[HW79] John A Hartigan and Manchek A Wong. “Algorithm AS 136: A k-means clustering algorithm”. In: *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 28.1 (1979), pp. 100–108.

[Sha81] Micha Sharir. “A strong-connectivity algorithm and its applications in data flow analysis”. In: *Computers & Mathematics with Applications* 7.1 (1981), pp. 67–72.

[Pag+99] Lawrence Page et al. *The PageRank citation ranking: Bringing order to the web.* Tech. rep. Stanford InfoLab, 1999.

[Ley02] Michael Ley. “The DBLP computer science bibliography: Evolution, research issues, perspectives”. In: *String processing and information retrieval*. Springer. 2002, pp. 481–486.

[RAK07] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. “Near linear time algorithm to detect community structures in large-scale networks”. In: *Physical review E* 76.3 (2007), p. 036106.

[ZCG07] Osmar R Zaiane, Jiyang Chen, and Randy Goebel. “DBconnect: mining research community on DBLP data”. In: *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*. ACM. 2007, pp. 74–81.

[Ley09] Michael Ley. “DBLP: some lessons learned”. In: *Proceedings of the VLDB Endowment* 2.2 (2009), pp. 1493–1500.

[BD10] Maria Biryukov and Cailing Dong. “Analysis of computer science communities based on DBLP”. In: *Research and advanced technology for digital libraries* (2010), pp. 228–235.