Analyzing Wikipedia Talk Network with GraphX

Allu Hanuma Reddy
Department of Computer Science and Engineering.
Indian Institute of Information Technology Dharwad.
Dharwad, Karnataka
20bcs008@iiitdwd.ac.in

Erigi Vaishnavi
Department of Computer Science and Engineering.
Indian Institute of Information Technology Dharwad.
Dharwad, Karnataka
20bcs044@iiitdwd.ac.in

Golla Anjaiah
Department of Computer Science and Engineering.
Indian Institute of Information Technology Dharwad.
Dharwad, Karnataka
20bcs048@iiitdwd.ac.in

Kotha Balaji
Department of Computer Science and Engineering.
Indian Institute of Information Technology Dharwad.
Dharwad, Karnataka
20bcs073@iiitdwd.ac.in

***Abstract*—This project focuses on using Apache Spark's GraphX module to identify communities in the Wikipedia Talk Network. To identify distinctive communities within this enormous social network dataset, we use the Connected Components algorithm. We address scalability issues by utilizing Spark's capabilities for distributed computing. For the purpose of meaningful community discovery, our work entails data preprocessing, graph construction, and algorithm execution. We investigate optimization techniques that guarantee both performance and accuracy. The project's results improve our comprehension of social dynamics in the Wikipedia Talk Network and show how useful GraphX and Spark are for performing graph-based analytics. These revelations have larger applications in anomaly detection, recommendation systems, and social network analysis. The interconnectedness of participants and their communities is revealed, demonstrating the power of distributed graph processing.**

***Index Terms*—Big data, Community detection, Wikipedia Talk Network, Large-scale network analysis.**

I. Introduction

Analysis of large-scale network topologies has become essential in the current big data era for comprehending complicated relationships and interactions across numerous fields. Graph-based analytics and community discovery techniques are of utmost importance in this environment. This research explores the field of graph processing by conducting a thorough community detection analysis on the Wikipedia Talk Network using the GraphX library and the Apache Spark framework. The Wikipedia Talk Network is a complex social network where users converse, exchange ideas, and add to the platform's shared knowledge base. This dataset gives opportunities and challenges for thorough analysis because of its size and dynamic nature.

The Connected Components approach, a fundamental graph analysis technique, will be used as our main tool to reveal the underlying community structures within the Wikipedia Talk Network. We overcome the inherent scalability limitations caused by the enormous scale of the dataset by utilizing Apache Spark's computational capabilities. By completing this task, we hope to offer insightful information about the social dynamics taking place within the Wikipedia Talk Network. Beyond this particular dataset, our research serves as a real-world illustration of the usefulness and promise of graph processing with Spark and GraphX. This research's findings can now be applied to a wider range of fields, such as social network analysis, recommendation systems, fraud detection, and more.

In summary, this project demonstrates the ability of Spark and GraphX to handle big, real-world network datasets while bridging the fields of graph analytics and social network research. It provides a thorough examination of the community structures on the Wikipedia Talk Network.

II. Dataset Description

Wikipedia, a free encyclopedia run by volunteers, has user talk pages for communication and article discussions. We created a network by extracting all user talk page modifications from the Wikipedia edit history dump on January 3, 2008. This network depicts Wikipedia user interactions, with directed edges from one user to another showing instances of talk page updates, promoting collaborative conversations and knowledge sharing on the platform. From the platform's conception until January 2008, the network covered all Wikipedia users and their discussions. Nodes represent individual Wikipedia users, and directed edges between nodes i and j indicate that user i has modified user j's talk page, allowing for smooth communication and collaboration across platform contributors.

- **Nodes:** Each node representing a distinct Wikipedia user.
- **Edge:** An edge from node i to node j indicates that user i has edited the talk page of user j at least once.

III. Work FLow

In this case, we are utilizing the connected components approach to analyze the Wikipedia Talk Network with GraphX.

- **Load the Graph:** To begin, we import the necessary libraries and load our network data from a dataset file.

This data represents the Wikipedia Talk Network, where nodes represent Wikipedia users, and edges indicate connections, such as discussions or interactions between them.

```
// Load the Wikipedia Talk
//Network graph from a file
val graph = GraphLoader.edgeListFile
(sc, "D:/7th Sem/Data Science System
/wikipedia$_$talk$_$network.txt")
```

- **Identify Connected Components:** We use the 'graph.connectedComponents()' function to apply the connected components algorithm to our network graph. This algorithm groups nodes that have connections to one another due to various interactions. Each group or element is assigned a special identifying number.

  ```
  // Find connected components and
  // assign component IDs to nodes
  val cc = graph.connectedComponents()
                .vertices
  ```

- **Calculate Component Sizes:** Now, we want to know how many nodes are in each connected component. We do this by mapping each node's component ID to the number 1 (indicating one node) and then summing up the nodes within each component.

  ```
  // Calculate the size of each
  // connected component
  val componentSizes = cc
    .map { case (_, componentId) =>
             (componentId, 1) }
    .reduceByKey(/_ + /_)
  ```

- **Sort Components by Size:** To understand which connected components are the largest, we sort them in descending order based on their sizes.

  ```
  // Sort connected components by size
  // in descending order
  val sortedComponents = componentSizes
    .sortBy { case (_, size) => size }
    .collect()
    .reverse
  ```

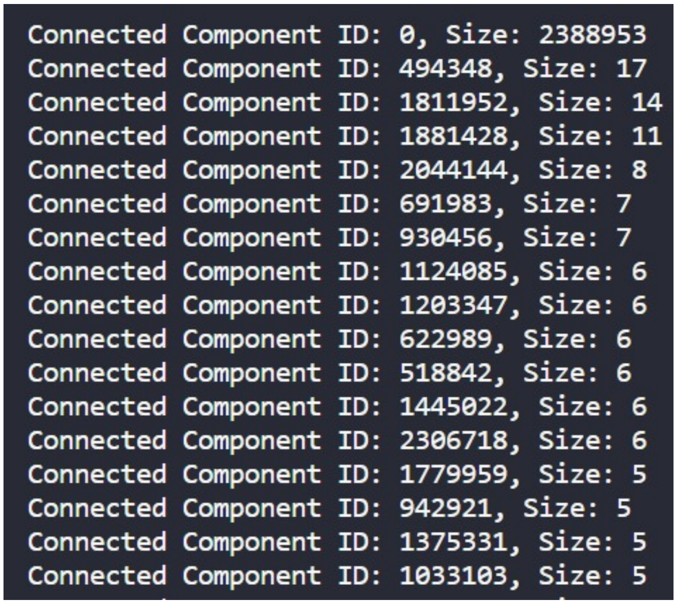
- **Print the Top 50 Components:** Lastly, we print out information about the top 50 connected components, which are the largest ones. This information includes the component ID and the number of nodes (size) in each component.

  ```
  // Print information about the
  //top 50 connected components
  val top50Components = sortedComponents
                              .take(50)
  top50Components.foreach {
           case (componentId, size) =>
    println(s"Connected Component
        ID: $componentId, Size: $size")
  }
  ```

In a nutshell, this code helps us explore the Wikipedia Talk Network's structure by finding groups of nodes that are tightly connected, determining the size of each group, and highlighting the top 50 largest connected components. This analysis can provide insights into the network's organization, revealing significant clusters or communities of pages or users.

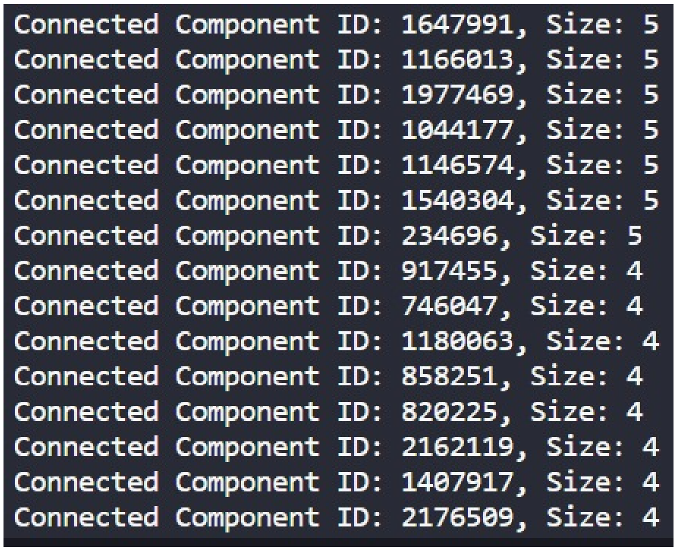
IV. Results & Discussion

The dataset represents a communication network of Wikipedia users, where each node represents a user, and directed edges indicate communication interactions (user A edited the talk page of user B). Connected components analysis was performed to identify clusters of users who are interconnected through discussions on their user talk pages. The largest connected component (Component ID: 0) is significantly large, containing approximately 99.8% of all users. It represents a highly interconnected community within Wikipedia. Smaller connected components (e.g., Component ID: 494348, 1811952) represent various discussion groups or communities within Wikipedia. The analysis provides insights into how Wikipedia users form communication clusters and communities, revealing patterns of interaction and collaboration. It sheds light on the dynamics of knowledge exchange and collaboration within the Wikipedia platform, highlighting both the larger Wikipedia community and smaller, specialized discussion groups.



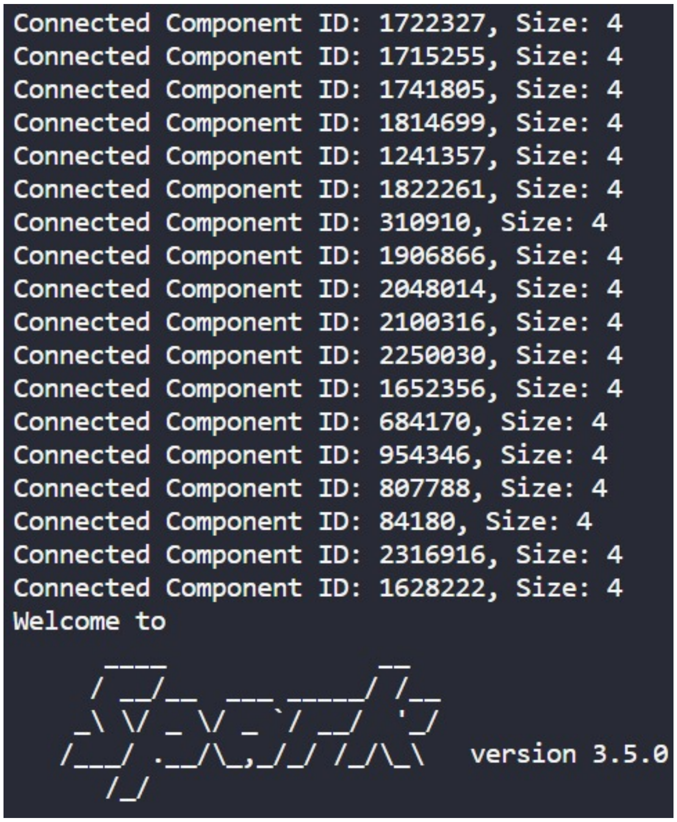
```
Connected Component ID: 0, Size: 2388953
Connected Component ID: 494348, Size: 17
Connected Component ID: 1811952, Size: 14
Connected Component ID: 1881428, Size: 11
Connected Component ID: 2044144, Size: 8
Connected Component ID: 691983, Size: 7
Connected Component ID: 930456, Size: 7
Connected Component ID: 1124085, Size: 6
Connected Component ID: 1203347, Size: 6
Connected Component ID: 622989, Size: 6
Connected Component ID: 518842, Size: 6
Connected Component ID: 1445022, Size: 6
Connected Component ID: 2306718, Size: 6
Connected Component ID: 1779959, Size: 5
Connected Component ID: 942921, Size: 5
Connected Component ID: 1375331, Size: 5
Connected Component ID: 1033103, Size: 5
```

Fig. 1. *Component ID with corresponding size.*



```
Connected Component ID: 1647991, Size: 5
Connected Component ID: 1166013, Size: 5
Connected Component ID: 1977469, Size: 5
Connected Component ID: 1044177, Size: 5
Connected Component ID: 1146574, Size: 5
Connected Component ID: 1540304, Size: 5
Connected Component ID: 234696, Size: 5
Connected Component ID: 917455, Size: 4
Connected Component ID: 746047, Size: 4
Connected Component ID: 1180063, Size: 4
Connected Component ID: 858251, Size: 4
Connected Component ID: 820225, Size: 4
Connected Component ID: 2162119, Size: 4
Connected Component ID: 1407917, Size: 4
Connected Component ID: 2176509, Size: 4
```

Fig. 2. *Component ID with corresponding size.*



```
Connected Component ID: 1722327, Size: 4
Connected Component ID: 1715255, Size: 4
Connected Component ID: 1741805, Size: 4
Connected Component ID: 1814699, Size: 4
Connected Component ID: 1241357, Size: 4
Connected Component ID: 1822261, Size: 4
Connected Component ID: 310910, Size: 4
Connected Component ID: 1906866, Size: 4
Connected Component ID: 2048014, Size: 4
Connected Component ID: 2100316, Size: 4
Connected Component ID: 2250030, Size: 4
Connected Component ID: 1652356, Size: 4
Connected Component ID: 684170, Size: 4
Connected Component ID: 954346, Size: 4
Connected Component ID: 807788, Size: 4
Connected Component ID: 84180, Size: 4
Connected Component ID: 2316916, Size: 4
Connected Component ID: 1628222, Size: 4
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 3.5.0
      /_/
```

Fig. 3. *Component ID with corresponding size.*

V. Conclusion

In brief, this project effectively employed Apache Spark's GraphX module to uncover distinct communities within the vast Wikipedia Talk Network. By implementing the Connected Components algorithm, we overcame scalability challenges and gained valuable insights into the network's social dynamics. The optimization techniques explored ensure both performance and accuracy, enhancing the utility of Spark and GraphX for graph-based analytics. These findings have broader applications in areas like anomaly detection, recommendation systems, and social network analysis. This project underscores the power of graph processing in uncovering complex network structures.

References

[1] A. Spark. (2023) Apache spark graphx programming guide. [Online]. Available: https://spark.apache.org/docs/latest/graphx-programming-guide.html

[2] S. N. A. P. (SNAP). (2023) Snap - stanford network analysis project. [Online]. Available: https://snap.stanford.edu/index.html

[3] A. Spark. (Accessed 2023) Apache Spark GraphX API. [Online]. Available: https://spark.apache.org/docs/latest/api/scala/org/apache/spark/graphx/lib/ConnectedComponents$.html

[1] [2] [3]