Bachelor Thesis

Social Network Analysis on Research

Yusuf Erdem Nacar

Mehmet Emre Akbulut

Advisors:

Suzan Üsküdarlı

Hüseyin Birkan Yılmaz

TABLE OF CONTENTS

1. INTRODUCTION

1.1. Broad Impact

Today, many articles are published by many organizations and researchers. It has become very difficult to follow up-to-date research published in a certain academic field. In addition to all these, when we also consider the interrelationship of academic studies in different fields, we see a wide network. In our project, Social Network Analysis on Research, we aim to create a publicly available web application/tool that showcases these relationships and uses them in a specific domain. In this way, we think that everyone, from undergraduate students to university professors, will have the chance to examine and analyze academic knowledge in a particular field.

Seeing which academic writers have participated in such studies, examining the connections between citations, and seeing the clusters formed are supportive of academic studies.

Identifying rapidly developing sub-disciplines, seeing influential authors in a particular academic field, or analyzing the reflection of a current issue in academia are some of the possible outcomes within the academic environment. One of these current issues is naturally COVID-19. A time-dependent graph and analysis can be created on COVID-19 using the tool. In the last few years, the publicly presented COVID-19 graph can explain which articles, authors, organizations, or journals in academia have been effective on this subject, in conjunction with their relations with each other.

1.2. Ethical Considerations

We have tried to pay attention to ethical concerns in the academic environment in our work. Metadata and detailed data of academic papers are momentous for our study. It is one of the situations in that we do not want to present a paper by separating

it from its authors in a presented interface or graph, or not to show all contributors. To this end, we decided to include the authors in our graph whenever an article is added to the graph

In this direction, protecting the rights of researchers, research institutes and journals is one of the ethical considerations we paid attention to at first.

In addition, validation of the data presented and used is also critical. Since it will be used in some metric analysis in any data graph, the margin of error mustn't be higher than a certain rate. We made the data source selection according to this consideration. For example, we decided not to use the IEEE API because it only provided data about the papers they published, and did not include outside data.

The integrity of the research conducted within the academic environment is also a part of our ethical considerations. In this direction, it is essential to use up-to-date data sources.

Last but not least, we have made sure that the evaluation and analysis of our graphs do not involve any subjective interpretations that would undermine a researcher or their work.

2. PROJECT DEFINITION AND PLANNING

2.1. Project Definition

Social Network Analysis on Research is a project that aims to uncover hidden relations and data that is present in the research graphs that include research articles and researchers using social network analysis techniques and graph measures. A tool that will help the analysis will also be built to achieve this goal. In total, the project includes a tool that will enable the users to create research graphs and analyze them as well as analyze results that we wish to study.

2.2. Project Planning

This project aims to produce certain results according to Milestones, which are expected to be conducted in 2 semesters.

2.2.1. Project Time and Resource Estimation

Basically, this project aims to produce certain results according to Milestones, which are expected to be conducted in 2 semesters.

Table 2.1.

Name	**Start**	**Finish**
Design	01.10.2022	
Requirements Specification	18.10.2022	
Sequence Diagrams	08.11.2022	22.11.2022
Deciding on Graph Structure	08.11.2022	
Setting up Graph Database	06.12.2022	13.12.2022
Create Graph from Data stored	06.12.2022	13.12.2022
Frontend Application Development	22.02.2023	06.06.2023
Design Revision	06.03.2023	20.03.2023
Literature Survey	20.03.2023	09.04.2023
Ontology Research	20.03.2023	09.04.2023
Graph Metric Calculations	20.03.2023	09.04.2023
Deciding on Research Domain	09.04.2023	17.04.2023
Collecting Dataset	09.04.2023	17.04.2023
Analyzing Graph with respect to graph topology and metrics	17.04.2023	08.05.2023
Result and Final Analyzes	08.05.2023	08.06.2023

2.2.2. Success Criteria

As stated in the project definition, we aimed to create a web application that helps people to create graphs of articles and authors. The metric analysis will be conducted by using this data structure.

Our first success condition for this semester is to go over the design of the application and redesign and reimplement the parts that should be improved or corrected to achieve a more efficient, scalable, and versatile codebase.

The second success criterion is to revise the application so that it represents the

research networks more correctly. This includes redesigning the parts that misrepresent the data and the network structure at hand.

The third success criterion is to bring the project to a point where the users can realize their use cases, such as finding researchers for collaboration.

2.2.3. Risk Analysis

2.2.3.1. External Data Source. The extendable structure of the codebase has been updated to better represent the layered and separated services that we utilize. This allows us to include other data sources easier and more robustly than before if the one at hand fails.

2.2.3.2. Performance. The research graph is now persistent meaning that the data added to it accumulates over time. This removes the overhead caused by several operations

- Removing data that can be utilized later on
- Making duplicate requests to the data source because of the deleted data
- Building the graph over time instead of big chunks

In addition to these changes, several memory optimizations are implemented to support a higher number of users at the same time such as removing graph projections after they are used for analysis.

2.2.3.3. Scalability. The memory and CPU usage of the application has been analyzed and several steps have been taken toward eliminating wasteful usage of these resources such as cleaning the memory used for analysis after the analysis is done, optimizing CPU usage parameters such as thread pool sizes for the machine that the application is deployed to in order to reduce the risk of failure due to the hardware.

2.2.4. Team Work

In this project, which we carried out as two people, we successfully recorded our progress and weekly efforts. This helps us a lot in giving feedback to each other. We share the action items we have decided with our advisors equally and combine the results with the meetings we hold among ourselves.

3. RELATED WORK

Social Network Analysis is a method used to derive meaningful information on many datasets showing graph structure. In this direction, we conducted a literature survey to understand how it is used in different fields. "Social Network Analysis: Methods and Applications" [2] has been very influential in shaping our basic ideas about Social Network Analysis. "Models and Methods in Social Network" [1], which is also its sequel helped us to understand what skill sets we need to progress within our project.

We continued to direct our research in line with cluster analysis and graph theory. Especially, we have focused on the meaning of graph metrics in our research graph.

Additionally, we have benefited from works such as "On the Correlation between Research Performance and Social Network Analysis Measures Applied to Research Collaboration Networks" [6] and "Polarized public opinion responding to corporate social advocacy: Social network analysis of boycotters and advocators" [5] helped us get better insight into the meaning and usage of our metrics as well as helped us think about questions that we may want to do research on.

Additionally, works on the topological structure of social networks such as "Graph pattern matching revised for social network analysis" [3] helped us understand the technical methodology used in doing research on social network analysis.

As a result, we have observed that social network analysis is a method that is widely used in different domains, however, analyzing authorship and citation graph in terms of their domains can help to uncover the dynamics of research graphs and also academia.

Also, there are several online services that provide similar graph building, how-

ever, none of them aim to let the users analyze the graph they built. These services are mainly focused on providing the researchers with a more convenient way of finding articles that are related to an article that they are interested in. The most well-known tools of this nature are

- Connected Papers `https://www.connectedpapers.com/`
- Research Rabbit `https://www.researchrabbit.ai/`
- Litmaps `https://www.litmaps.com/`

Although these tools provide their users with a research graph of articles, they do not provide the measures of the graphs they built since their focus on building these graphs is to enhance the literature exploration of their users.

4. METHODOLOGY

4.0.1. What Data to Collect

Our main goal is to collect reliable and complete data related to articles, authors, and journals. Many academic platforms can serve this information such as Elsevier, arXiv etc. However, the relationship between them is also one of the main aspects we need. Semantic Scholar [4] is built with the data of plenty of Academic platforms.

Article Fundamental Fields

- paperId
- DOI
- title
- year
- referenceCount
- citationCount
- authors
- abstract
- cites
- references
- fieldsOfStudy

Author Fundamental Fields

- authorId
- Name
- alliases
- affiliations
- paperCount

- hIndex

4.0.2. How to Collect Data

Semantic Scholar Academic Graph API (S2AG) was the system that both met our demands in the best way and had the largest data source. We extract articles, authors, citations, and authorships by creating data pipelines with necessary validations. Articles that come up with a keyword search are presented in paginated form. We define it as 'catalog-base'.
Later, with some actions that can be applied to this base, its content can also change. Adding articles cited by my articles, adding articles that my articles cite, and adding papers of my authors are some of these global actions. The set of papers created with these actions is named 'catalog-extension
The Union of these sets composes of a 'catalog' and it is the main source to build a graph.

4.0.3. How to Analyze Data

The Graph Data Structure created will be stored in Graph Database (Neo4j) and graph metrics will be applied with CYPHER queries.
Available Metrics:

- Betweenness Centrality
- PageRank
- ArticleRank
- Eigenvector Centrality
- Degree Centrality
- Indegree Centrality
- Outdegree centrality
- Closeness Centrality
- Harmonic Centrality

and their changes with respect to time.

5. REQUIREMENTS SPECIFICATION

5.1. Glossary

- Catalog Base: Catalog is a user-selected set of authors, papers, and journal metadata. Users create catalogs corresponding to specific research interests, upon which various functionalities (such as network analysis) can be performed.
- Catalog Extension: Catalog extension is a set of authors, papers, and journals that are not included in a Catalog Base but can be derived from them via options.
- Catalog: The total of a catalog base and its catalog extension.
- Search Engine: A view where a User can search authors and papers by a keyword.
- Paper Node: A node that keeps the metadata of a paper.
- Author Node: A node that keeps the metadata of an author.
- Citation Edge: A directed edge from a paper to a paper that it cited.
- Authorship Edge: A directed edge from an author node to a paper node that describes a paper they wrote.
- Build Engine: A process/system which builds a graph from a catalog and its catalog extension.
- Inbound Citation Count: The number of papers that cited a paper.
- Outbound Citation Count: The number of papers that a paper cited.
- Author Citation Count: The total number of citations that an author's papers have.
- Graph: A graph is a data structure where the nodes are either paper or author nodes, and the edges are either citation or authorship edges.
- Analyze Engine: A tool that derives specific statistics and information from the Graph.

5.2. Requirements

1. Functional Requirements

1. User Requirements
    1. Managing Account
        1. Users shall be able to signup via email, username, and password
        2. Users shall be able to log in to their account using email and password
        3. Users shall be able to log out of their account
        4. Users shall be able to delete their account
    2. Searching
        1. Paper Search
            1. Users shall be able to search papers with a keyword
            2. Users shall be able to filter the paper search results by publishing date
            3. Users shall be able to filter the paper search result by inbound citation count
            4. Users shall be able to filter the paper search result by outbound citation count
            5. Users shall be able to filter the paper search results by field of study
            6. Users shall be able to sort the paper search results by publishing date
            7. Users shall be able to sort the paper search results by inbound citation count
            8. Users shall be able to sort the paper search results by outbound citation count
    3. Managing Catalogs
        1. Catalog Base
            1. Users shall be able to create a catalog base
            2. Users shall be able to add papers to a catalog base
            3. Users shall be able to remove papers from a catalog base
            4. Registered users shall be able to save a catalog base for later use
            5. Registered users shall be able to delete a catalog base

2. Catalog Extension
    1. Users shall be able to add all papers written by the authors in a catalog base to its catalog extension
    2. Users shall be able to add all papers that cited the papers in a catalog base to its catalog extension
    3. Users shall be able to add all papers that are cited by the papers in a catalog base to its catalog extension
    4. Users shall be able to remove papers from a catalog extension
    5. Users shall be able to move papers from a catalog extension to its catalog base
    6. Users shall be able to save a catalog extension for later use
    7. Users shall be able to delete a catalog extension
4. Using Graphs
    1. Users shall be able to build a graph of a catalog
    2. Users shall be able to view a graph
    3. Users shall be able to analyze a graph
5. Viewing Graphs
    1. Users shall be able to filter the type of nodes visible
    2. Users shall be able to filter the type of edges visible
6. Analyzing Graphs
    1. Users shall be able to calculate graph measures of the graph
    2. Users shall be able to calculate graph measures of a particular node in the graph

2. System Requirements
    1. The system shall delete the graph from the graph database after the analysis session is over

2. Non-Functional Requirements
    1. Availability
        1. The application shall be available as a website browsable by a web browser
        2. The application shall be dockerized

    3. The application shall be deployable to a configurable server
    4. The application shall support the English characters
2. Reliability
    1. The application shall always run if not intentionally shut down
    2. The application shall be scalable if needed

6. DESIGN

6.1. Information Structure and System Design

Our main approach in the design of the project is to divide the project into 3 main parts in terms of the action flow, **Search & Select**, **Graph Process**, and **Analyze Graph**. To search for information about articles, authors, and citations, Semantic Scholar Academic Graph (S2AG) API is used.

Core Application is responsible for the process from a given query to the final article list as an input for the Graph Engine. With the given query, an S2AG query is built and parsed to make data useful for the next steps. External data sources S2AG API ordered them with respect to relevance. The output of the Search Engine is always a list of articles after filtering and extending actions applied.

Neo4j Graph Database is a persistent database that handles adding, deleting, and updating operations on the catalog bases and extensions of users. A bunch of CYPHER queries is responsible for the integrity of the graph database.

Analyze Engine derives the metrics of a given graph with some Cypher Queries. These metrics are:

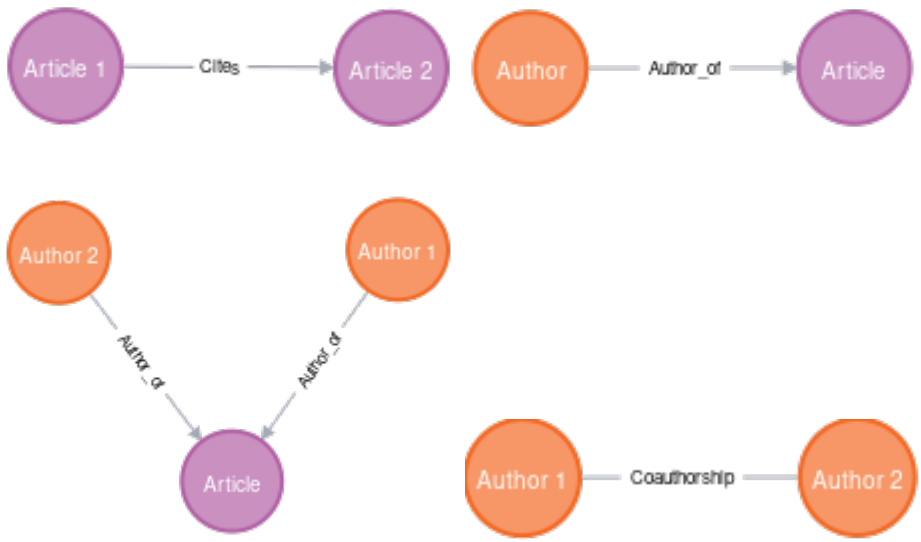
- Betweenness Centrality
- PageRank
- ArticleRank
- Eigenvector Centrality
- Degree Centrality
- Indegree Centrality
- Outdegree centrality
- Closeness Centrality

- Harmonic Centrality

and their changes with respect to time.

The papers we have read and used for detailed mathematical explanation are given in **References** chapter.

The graph structure can be demonstrated as:



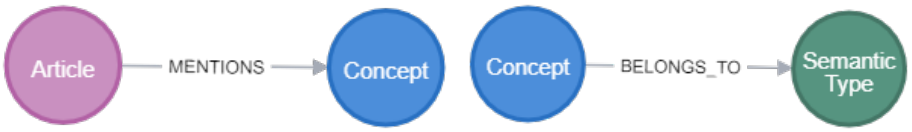
The co-authorship edge is inferred from two authorship edges ending at the same article node.

In addition to the core application, we have experimented with how we can connect these research graphs to various ontologies for an extended knowledge representation.

For our experimentation, we have decided to try to connect a graph we created using articles in the CORD-19 dataset which have "pathology" keyword in their abstracts. After the articles are selected, we processed these abstracts using CogStack's MedCAT [7], a named entity recognizer that returns the Unified Medical Language System (UMLS) [8] descriptions of the recognized concepts in the abstract. Then,

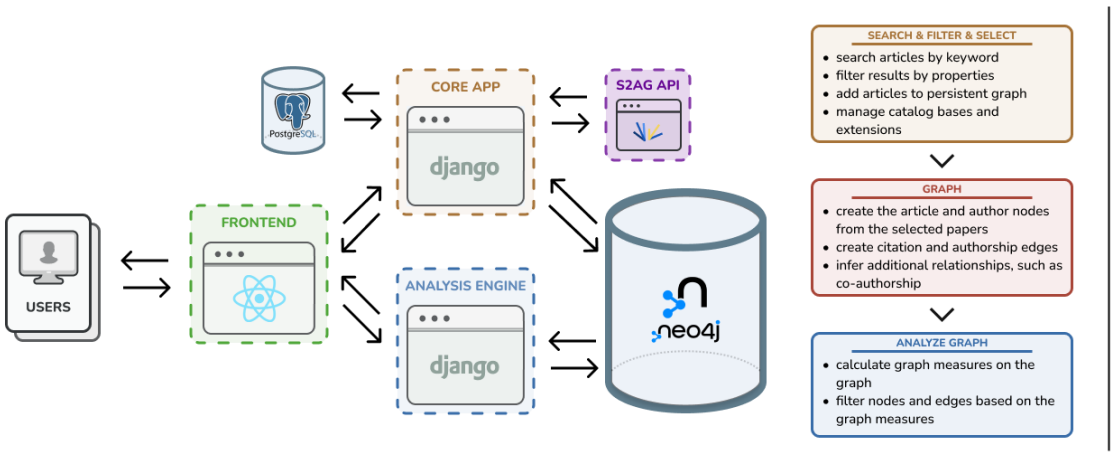
created and connected these concepts to the articles.

The structure the semantic network incorporates into the graph is demonstrated below:



6.2. Information Flow

The flow from input query to metrics is demonstrated via the diagram below.



A user can either use the frontend application to send requests to the core API or send requests to the core API themselves. The figure above shows the architecture for the former use case. The user first enters a query to retrieve articles. Then, the Core Application retrieves the relevant articles to the query from S2AG API. Also, users can manage catalog bases and extensions. Adding search results to the catalog base, and creating extensions with some options is one of the actions users can perform. All these actions can create different catalogs of users. Core Applications uses 2 databases. Relational Database is only responsible for Authorization and user-related operations. Neo4j Graph Database is the persistent database we have maintained. All users have a single node in the database. Their catalog bases and extensions are also connected to them. When a user adds an article to the catalog base, if this article exists on the graph, we create an edge between the catalog base and the article, otherwise, the article is created with all the citations, references, and authors. Also, articles can be either connected or not connected to these catalog bases. Authors, on the other hand, are always connected to their articles.

Analyze Engine uses Neo4j graph database to extract meaningful information for users. The users interact with both Core Application and Analysis Engine through either the

frontend application or the API provided

For the ontology connections, first, the abstracts for the selected articles are passed to the named entity recognizer, and the articles themselves are passed to the core application. Then, the module for creating semantic connections waits for the core application to be done and creates the semantic connections.

6.3. User Interface Design

For UI Design:
https://drive.google.com/drive/folders/1q4ur96v7syL0IDudRcpPbPhevtjXhKUG?usp=sharing

7. IMPLEMENTATION AND TESTING

7.1. Implementation

7.2. Testing

Testing a software project is very important for CI/CD. We have implemented the unit tests of the application and tested the project with them after every time we have implemented a new improvement or bug fix. The following unit tests are provided:

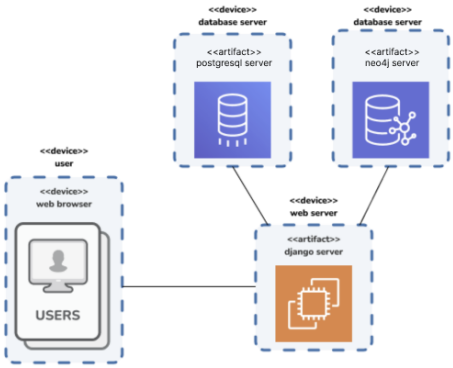
- Test for successful search
- Test for successful catalog base retrieval
- Test for failed catalog base retrieval
- Test for successful catalog base creation
- Test for failed catalog base creation
- Test for successful catalog base deletion
- Test for failed catalog base deletion

All the tests implemented are available on the public repository.

Repository for the API:
https://github.com/yusuferdemnacar/sonar-backend

Repository for the user interface application:
https://github.com/mehmetemreakbulut/sonar-frontend

7.3. Deployment



The Django server is deployed in an EC2 instance using Amazon Web services. Also, the Postgresql server is deployed in another EC2 Server. Digital Ocean Droplet is used for Graph Database Neo4j Server. The Django server is communicating with both of the database servers.

The links can be changed. Application User Interface and Backend link available at:
`https://docs.google.com/document/d/1uMZn04jNFVIrvc1XzXi83r7cKm18zCW4VQJ2dSAKvKQ/edit?usp=sharing`

8. RESULTS

We have revised the design and implementation of the SONAR application. The application now provides a complete API that performs the tasks described in the previous sections.

We have implemented a user interface that provides users with an easier way to access the functionalities of the API.

We have investigated the possible connections that can be constructed between other graph-structured knowledge representations and our research graphs and created interfaced graphs for proof of concepts.

9. CONCLUSION

Overall, after 2 semesters' worth of work and many meetings, we can confidently assert that we have a complete project structure in architecture, design, implementation, and documentation with respect to the plans made at the start of the CmpE491 course.

In addition to the application, we have made great efforts to research and uncover possible future directions for the project such as creating connections between semantic representations and the research graphs.

As a conclusion, we have completed all the processes that we mentioned in the report, aligning with our advisors.

10. REFERENCES

[1] L. Rossoni, "Models and methods in social network analysis," 2006.

[2] S. Wasserman and K. Faust, "Social Network Analysis: Methods and Applications," 1994.

[3] W. Fan, "Graph pattern matching revised for social network analysis," 2012.

[4] K. Lo, L. L. Wang, M. Neumann, R. M. Kinney, and D. S. Weld, "S2ORC: The Semantic Scholar Open Research Corpus," 2020.

[5] H. Rim, Y. Lee, and S. Yoo, "Polarized public opinion responding to corporate social advocacy: Social network analysis of boycotters and advocators," Public Relations Review, vol. 46, p. 101869, 2020.

[6] A. Abbasi and J. Altmann, "On the Correlation between Research Performance and Social Network Analysis Measures Applied to Research Collaboration Networks," 2011 44th Hawaii International Conference on System Sciences, pp. 1–10, 2011.

[7] Z. Kraljevic et al., "Multi-domain Clinical Natural Language Processing with MedCAT: the Medical Concept Annotation Toolkit," Artificial intelligence in medicine, vol. 117, p. 102083, 2020.

[8] O. Bodenreider, "The Unified Medical Language System (UMLS): integrating biomedical terminology," Nucleic acids research, vol. 32 Database issue, pp. D267-70, 2004.