DSAA 5002 - Data Mining and Knowledge Discovery in Data Science

Project

50015940 Jiaxiang Gao

The below report briefly describes my process. For a detailed view of the process, you can check the Code section, which clearly introduces each step using markdown.

All the code for this project can be found in this GitHub repository:

Github: HKUSTGZ_DSAA5002_project_Financial_Text_Analysis_Knowledge_Graph

Additionally, due to file upload limitations, you can find the complete project, including files saved during the process, at this OneDrive link:

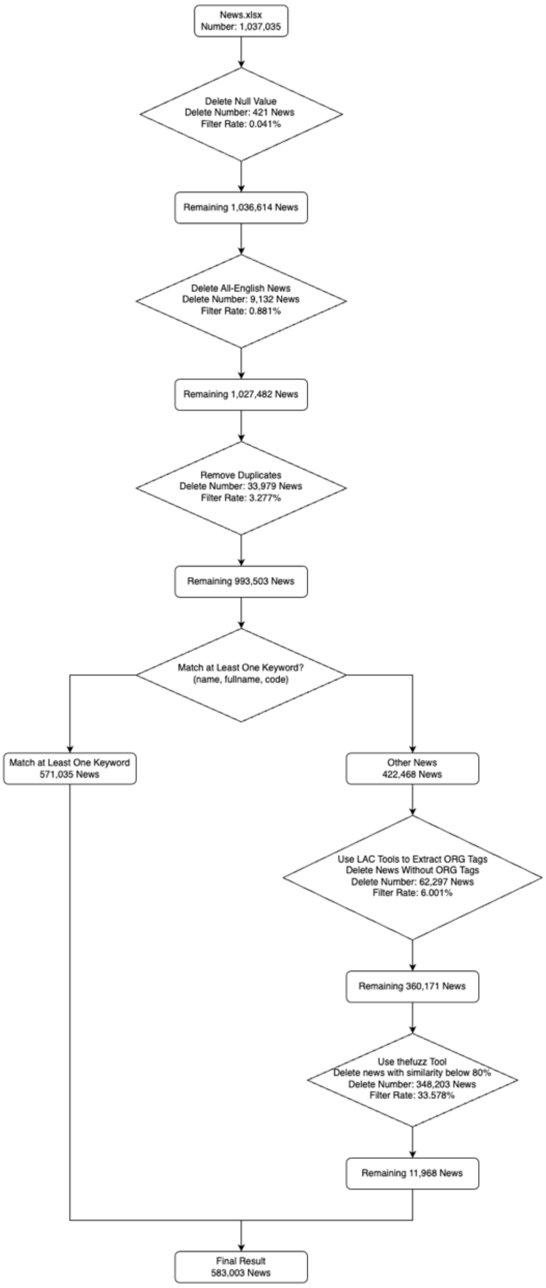
OneDrive: HKUSTGZ_DSAA5002_project_Financial_Text_Analysis_Knowledge_Graph

Q1: Data Preprocessing - Noise Removal

Here's a specific description of my data processing for Q1:

Original news data: 1,037,035.

1. Deleted the NewsContent column.
2. Removed null values.
   - Data deleted in this process: 421
   - Remaining data: 1,036,614
   - Filter rate: 0.041%
3. Removed non-Chinese news.
   - Data deleted in this process: 9,132
   - Remaining data: 1,027,482
   - Filter rate: 0.881%
4. Removed special characters and numbers from the news.
5. Deduplicated the news.
   - Data deleted in this process: 33,979
   - Remaining data: 993,503
   - Filter rate: 3.277%



6. Used regular expressions to remove certain special characters from the A-share list's name and divided the dataset into complete matches and other news based on whether at least one of the name, fullname, and code appeared in the news. Special characters: r'*?ST', r'^PT', r'^S', r'B 股$', r'B$', r'A$'
    - Complete match news count: 571,035
    - Other news count: 422,468

7. Used Baidu's LAC library to further process other news, extracting the ORG tag and deleting news that did not extract the ORG tag. [1][2]
    - Data deleted in this process: 62,297.
    - Remaining data: 360,171
    - Filter rate: 6.001%

8. Used the thefuzz library to calculate text similarity between the extracted ORG tag and the name, deleting news with similarity below 80%. [3]
    - Data deleted in this process: 348,203.
    - Remaining data: 11,968
    - Filter rate: 33.578%

9. Merged the news obtained with the complete match news from step 7 to get the result.
    - Final data count: 583,003

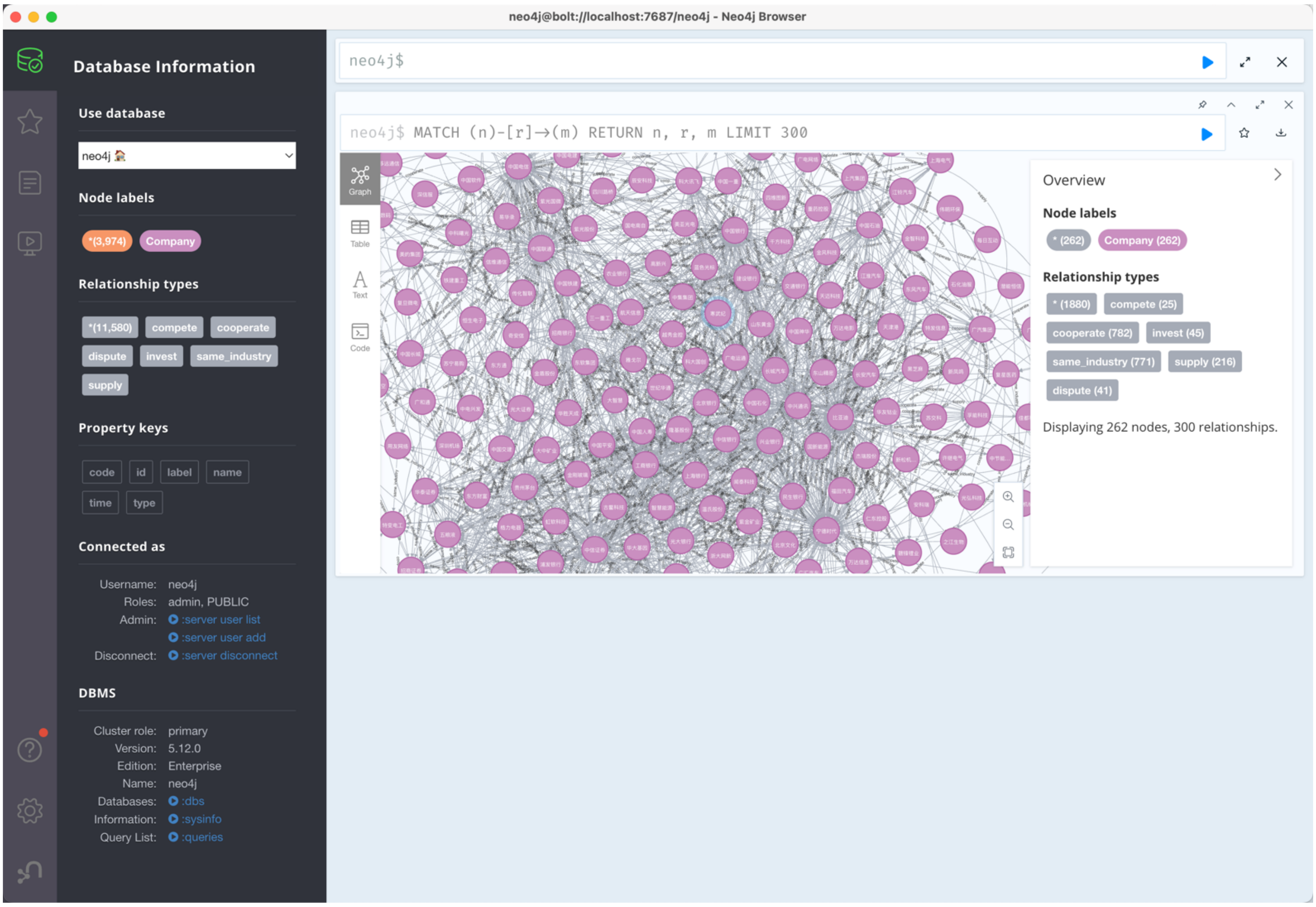
Q2: Data Analysis - Text Knowledge Mining

In this question, I used the IDEA-CCNL/Erlangshen-Roberta-110M-Sentiment model [4] to conduct sentiment analysis on the obtained news. Due to computational resource limitations, I divided the dataset into 12 parts, calculated the sentiment in parallel, and then merged them into the result.

Model	**ASAP-SENT**	**ASAP-ASPECT**	**ChnSentiCorp**
Erlangshen-Roberta-110M-Sentiment	97.77	97.31	96.61
Erlangshen-Roberta-330M-Sentiment	97.9	97.51	96.66
Erlangshen-MegatronBert-1.3B-Sentiment	98.1	97.8	97

According to the official documentation, the 110M Erlangshen-Roberta-110M-Sentiment model offers a good balance between performance and resource consumption. While it has fewer parameters compared to larger 330M and 1.3B models, this 110M model demonstrates quite high accuracy in terms of performance. Furthermore, the smaller model size translates to lower operational costs and faster processing speeds, which can alleviate resource constraints for me.

Q3: Constructing a Knowledge Graph



neo4j@bolt://localhost:7687/neo4j - Neo4j Browser

Database Information

Use database

neo4j

Node labels

*(3,974) Company

Relationship types

*(11,580) compete cooperate dispute invest same_industry supply

Property keys

code id label name time type

Connected as

Username: neo4j
Roles: admin, PUBLIC
Admin: :server user list
:server user add
Disconnect: :server disconnect

DBMS

Cluster role: primary
Version: 5.12.0
Edition: Enterprise
Name: neo4j
Databases: :dbs
Information: :sysinfo
Query List: :queries

```
neo4j$
```

```
neo4j$ MATCH (n)-[r]→(m) RETURN n, r, m LIMIT 300
```

Overview

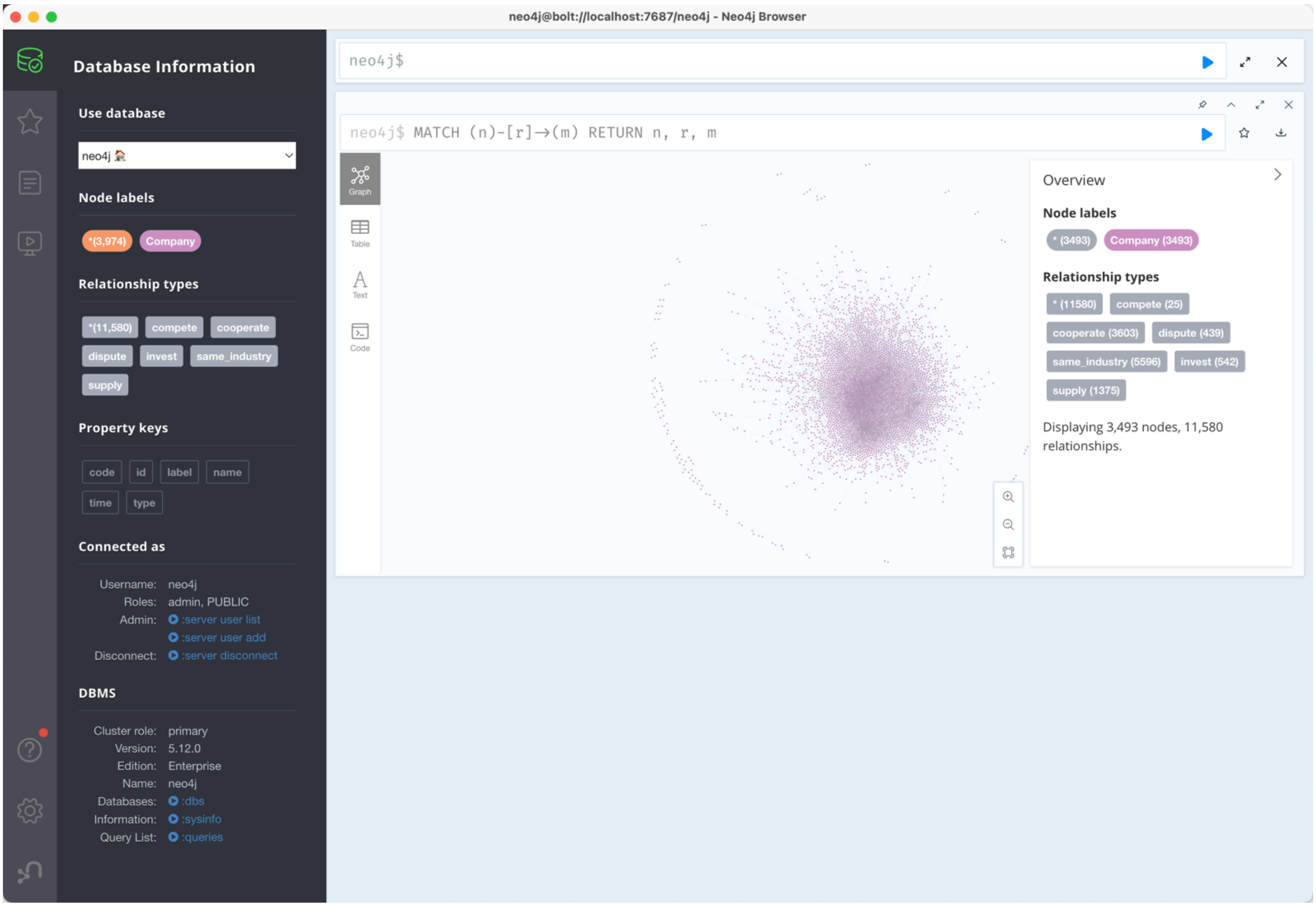
Node labels

* (262) Company (262)

Relationship types

* (1880) compete (25) cooperate (782) invest (45) same_industry (771) supply (216) dispute (41)

Displaying 262 nodes, 300 relationships.



neo4j@bolt://localhost:7687/neo4j - Neo4j Browser

Database Information

Use database

neo4j

Node labels

*(3,974) Company

Relationship types

*(11,580) compete cooperate dispute invest same_industry supply

Property keys

code id label name time type

Connected as

Username: neo4j
Roles: admin, PUBLIC
Admin: :server user list
:server user add
Disconnect: :server disconnect

DBMS

Cluster role: primary
Version: 5.12.0
Edition: Enterprise
Name: neo4j
Databases: :dbs
Information: :sysinfo
Query List: :queries

```
neo4j$
```

```
neo4j$ MATCH (n)-[r]→(m) RETURN n, r, m
```

Overview

Node labels

* (3493) Company (3493)

Relationship types

* (11580) compete (25) cooperate (3603) dispute (439) same_industry (5596) invest (542) supply (1375)

Displaying 3,493 nodes, 11,580 relationships.

References:

[1] Baidu. (2021). *Baidu/Lac Baidu NLP: Word segmentation, part of speech tagging, named entity recognition, word importance*. GitHub. https://github.com/baidu/lac

[2] Jiao, Z., Sun, S., & Sun, K. (2018). C*hinese Lexical Analysis with Deep Bi-GRU-CRF Network. arXiv preprint arXiv:1807.01882*. https://arxiv.org/abs/1807.01882

[3] Seatgeek. (2021). *Seatgeek/thefuzz: Fuzzy string matching in Python*. GitHub. https://github.com/seatgeek/thefuzz

[4] Idea-CCNL. (2022). *Idea-CCNL/erlangshen-roberta-110m-sentiment · hugging face*. IDEA-CCNL/Erlangshen-Roberta-110M-Sentiment · Hugging Face. https://huggingface.co/IDEA-CCNL/Erlangshen-Roberta-110M-Sentiment