DSAA 5002 - Data Mining and Knowledge Discovery in Data Science (Fall Semester 2023)

Project Report

Task 1 Data Preprocessing and Analysis

Question 1 (20 marks): Data Preprocessing - Noise Removal

1. noise removal strategies (15 marks):

- Remove **punctuation** and special **characters** (eg. 。 ， ？ ！)
- Remove **numbers** and **alphbet** (eg. A, b, 1000)
- Remove **stop words** from nltk.corpus (such as "的" , "是", "和", and so on)
- Tag the words by pos_tag and **filter out nouns** after jieba.lcut

(because of the company names for similarity matching, which are typically nouns, I consider adjectives and verbs as noise to reduce text length and improve matching efficiency.)

- Remove **specific words** defined by myself (repetitive and meaningless words: "股份","有限公司", "集团")

"法院经开庭审理查明，2000 年至 2004 年期间，被告人张恩照利用其担任原中国建设银行副行长、行长，中国建设银行股份有限公司董事长的职务便利，为他人牟取利益，多次非法收受他人给予的款物共计人民币 400 余万元。案发后，赃款、赃物已全部退缴。 法院认为，被告人张恩照身为国家工作人员，利用职务上的便利，为他人谋取利益，非法收受他人财物，其行为已构成受贿罪，受贿数额特别巨大。鉴于张恩照因其他违纪问题被审查后，主动交代了有关部门不掌握的本案受贿事实，属于自首，且赃款、赃物已全部退缴，对张恩照依法可从轻处罚。法院遂依法以受贿罪判处张恩照有期徒刑 15 年。 "

(length: 271)

"记者田雨银行董事长受贿案一审受贿罪法院被告人银行行长行长银行董事长职务利益款物人民币赃款赃物法院被告人国家人员职务利益财物受贿罪数额部门事实赃款赃物法院受贿罪"

(length: 79)

But some words (both nouns and verbs) in company name may be removed like "建设", so I add the **news tittle** to preprocessed_content. eg.

"记者田雨银行董事长受贿案一审受贿罪法院被告人银行行长行长银行董事长职务利益款物人民币赃款赃物法院被告人国家人员职务利益财物受贿罪数额部门事实赃款赃物法院受贿罪"

"建设银行原董事长张恩照一审被判 15 年记者田雨银行董事长受贿案一审受贿罪法院被告人银行行长行长银行董事长职务利益款物人民币赃款赃物法院被告人国家人员职务利益财物受贿罪数额部门事实赃款赃物法院受贿罪"

2. filter rates of each strategy (5 marks):

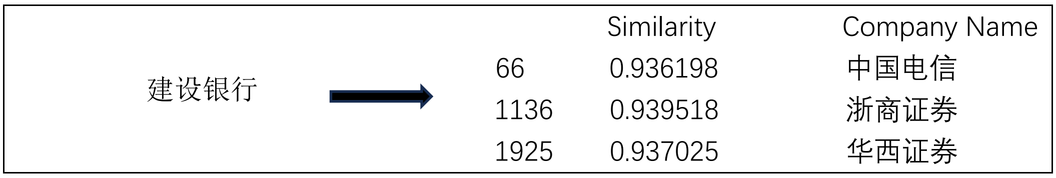
$$fiter\ rate = \frac{\#\ fitered\ news}{\#\ totel\ news}$$

(# totel news = 1037035)

- solution 1: bert-base-chinese+cosine_similarity ✗

$$similarity(A,B) = \frac{A \cdot B}{|A| \times |B|}$$

Due to the fact that the company corresponding to the maximum similarity score is not the correct company name mentioned in the text, it is speculated that "bert-base-chinese" may not be suitable for vectorization of financial news. Therefore, it is necessary to change the model.



			Similarity	Company Name
建设银行	→	66	0.936198	中国电信
		1136	0.939518	浙商证券
		1925	0.937025	华西证券

- solution 2: key-word matching ✓

Match the company name (company_name) and full company name (company_fullname) for each news content and add them to a new dataframe.

✧ Advantages: It can quickly identify most news articles that contain company names.

✧ Disadvantage: It may overlook abbreviations of company names.

$$fiter\ rate = \frac{467426}{1037035} \approx 45.07\%$$

- solution 3: bert-wwm+cosine_similarity ✓

Modify the model to "bert-wwm" and set the cosine similarity threshold to 0.83, selecting the maximum similarity score greater than the threshold as the corresponding "Explicit_Company"

$$fiter\ rate = \frac{577432}{1037035} \approx 55.68\%$$

Question 2(30 marks): Data Analysis - Text Knowledge Mining

1. the strategies (10 marks):

- solution 1: Financial Sentiment Analysis Dictionary ✗

Load the sentiment dictionary and determine the sentiment based on the count of positive and negative words.

To load **two different**[1][2] sentiment dictionaries, make predictions separately, merge the dictionaries, and then make predictions again.

When the count of positive words is equal to the count of negative words, it is considered as a **neutral sentiment**.

- solution 2: Financial Sentiment Analysis Dictionary + training a **classification model** ✗

Use the positive and negtive outcomes above to train a classification model. Preprocess the data like Q1 and make a pipeline. (Tips: need chinese_tokenizer to tokenize the text.)

The classifier parameters are as follows:

- L2 penalty (e.g., Ridge)
- 10 iterations per fit (remember, logistic regression has no closed form solution for the betas!)
- 5-fold cross-validation
- Random state of 0 (the fitting can be stochastic)

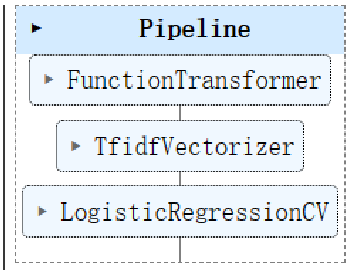


Fig.1. the pipeline to do classification

Due to less distinct features learned, the accuracy is only 0.473. Despite multiple changes and adjustments, the **accuracy** did not improve.

- solution 3: FinBERT 1.0[3]+inancial Sentiment Analysis Dictionary ✓

To load the FinBERT model and classify the neutral data from solution 1. The positive-negative ratio of solution 3 is **11.01**. And the positive-negative ratio of solution 1 is 7.15.

✧ Issue: The dictionary only considers greater-than and less-than relationships without considering error redundancy.

Improvement: Introduce a certain degree of redundancy (e.g. 3) to generate more neutral predictions.

✧ Issue: Only predicting on neutral sentiment.

Improvement: Fine-tune the model to improve its performance on sentiment categories.

Task 2 Application of Knowledge Graph

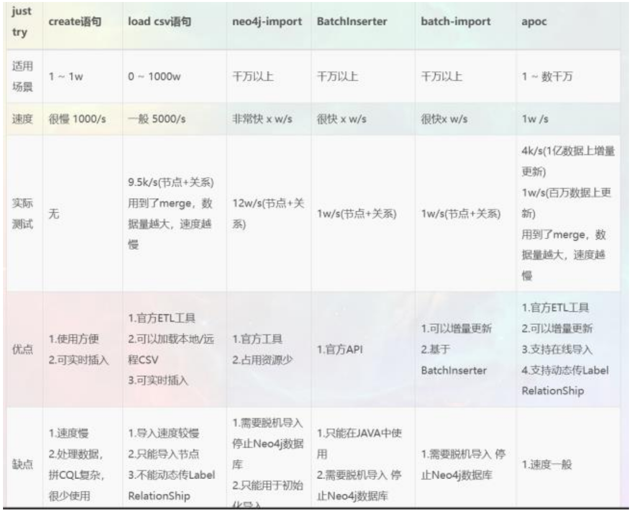
Question 3: (10 marks) Constructing a Knowledge Graph

1. the knowledge graph (10 marks):

- solution 1 (colab): ✗

Initially, I want to install the Neo4j Python driver and connect to a remote Neo4j database directly in Colab without installing Neo4j locally. But I encountered an **infinite loop**. After the initial connection, I need to modify the initial password, but the password can only be changed through a command after logging in. Without modifying it, I cannot log in. Therefore, I have decided to install and run Neo4j locally.

- solution 2 (neo4j community): ✓



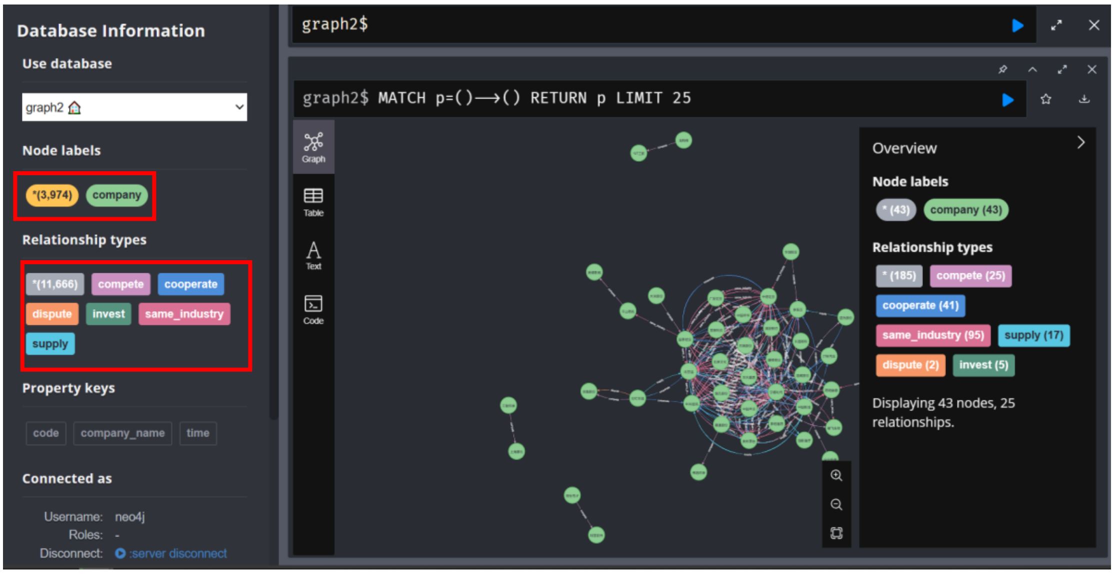
just try	create语句	load csv语句	neo4j-import	BatchInserter	batch-import	apoc
适用场景	1 ~ 1w	0 ~ 1000w	千万以上	千万以上	千万以上	1 ~ 数千万
速度	很慢 1000/s	一般 5000/s	非常快 x w/s	很快 x w/s	很快x w/s	1w /s
实际测试	无	9.5k/s(节点+关系) 用到了merge，数据量越大，速度越慢	12w/s(节点+关系)	1w/s(节点+关系)	1w/s(节点+关系)	4k/s(1亿数据上增量更新) 1w/s(百万数据上更新) 用到了merge，数据量越大，速度越慢
优点	1.使用方便 2.可实时插入	1.官方ETL工具 2.可以加载本地/远程CSV 3.可实时插入	1.官方工具 2.占用资源少	1.官方API	1.可以增量更新 2.基于BatchInserter	1.官方ETL工具 2.可以增量更新 3.支持在线导入 4.支持动态传Label RelationShip
缺点	1.速度慢 2.处理数据，拼CQL复杂，很少使用	1.导入速度较慢 2.只能导入节点 3.不能动态传Label RelationShip	1.需要脱机导入停止Neo4j数据库 2.只能用于初始化导入	1.只能在JAVA中使用 2.需要脱机导入 停止Neo4j数据库	1.需要脱机导入 停止Neo4j数据库	1.速度一般

Fig.2. different method to import[4]

1.Open the command prompt and navigate to the "bin" directory of Neo4j, Run the following command:

```
neo4j-admin database import full graph2
--nodes=company=E:/neo4j-community-5.14.0-windows/neo4j-community-5.14.0/import/hidy.nodes.company.csv
--relationships=compete=E:/neo4j-community-5.14.0-windows/neo4j-community-5.14.0/import/hidy.relationships.compete.csv
--relationships=cooperate=E:/neo4j-community-5.14.0-windows/neo4j-community-5.14.0/import/hidy.relationships.cooperate.csv
--relationships=dispute=E:/neo4j-community-5.14.0-windows/neo4j-community-5.14.0/import/hidy.relationships.dispute.csv
--relationships=invest=E:/neo4j-community-5.14.0-windows/neo4j-community-5.14.0/import/hidy.relationships.invest.csv
--relationships=same_industry=E:/neo4j-community-5.14.0-windows/neo4j-community-5.14.0/import/hidy.relationships.same_industry.csv
--relationships=supply=E:/neo4j-community-5.14.0-windows/neo4j-community-5.14.0/import/hidy.relationships.supply.csv --trim-strings=true
```

2.Open the Neo4j configuration file (conf/neo4j.conf) and modify the default database to "graph2"



```
graph2$
graph2$ MATCH p=()⟶() RETURN p LIMIT 25
```

Displaying 43 nodes, 25 relationships.

Fig.3. the knowledge graph

Question 4 (20 marks): Knowledge-Driven Financial Analysis

- Solution 1 (locally): match in knowledge graph ✓

Querying and setting rules for Implicit_Positive_Company and Implicit_Negative_Company in Neo4j knowledge graph. Calculate the time which is 7146.63 and the results is in Task2_2.

- Solution 2 (locally): match in the list ✗

Directly searching in the list, first find the corresponding company ID, then check if :START_ID and :END_ID are present, and then perform rule-based evaluation.

But I meet the **MemoryError!**

- Solution 3 (colab) : ✓

  So I decide to move the code in solution 2 to colab and it needs high RAM.

Solution 1: searching in the graph	Solution 3: Directly searching in the list
a.Match the name with nodes directly to get the relationship. b.slower (7017.72s)	a.Need to match name with ID first and match the relationship, and get ID (Implicit) again. b.faster (4670.64s)

Reference:

[1] 姚加权，冯绪，王赞钧，纪荣嵘，张维．语调、情绪及市场影响：基于金融情绪词典．管理科学学报，2021. 24(5), 26-46.

[2] MengLingchao/Chinese_financial_sentiment_dictionary: A Chinese financial sentiment word dictionary (github.com)

[3] FinBERT/README.md at main · valuesimplex/FinBERT (github.com)

[4] Neo4j 批量导入数据的几种方式 | 天道酬勤 (weikeqin.com)