Task 1 Q1 - noise removal strategies and result descriptions

Text coloured in **blue** is saved in the folder 'others/checkpoints'.

(a) We previewed the features of a_share_list. We select the feaures 'company name', 'fullname' and 'code' as the threshold for relevance. For news_data, we selected the Title and Newscontent for text mining.

```
print(a_share_list[:1])
>>> [{'name': '邵阳液压', 'fullname': '邵阳维克液压股份有限公司', 'code': '301079',
'location': '深圳证券交易所', 'time': '2021-10-19'}]
```

	NewsID	Title	NewsContent	NewsSource
0	1	建设银行原董事长张恩照一审被判15年	本报记者 田雨 李京华 中国建设银行股份有限公司原董事长张恩照受贿案3日一审宣...	中国证券报

Fig 1. Preview of a_share_list (top) and news_data (bottom).

We perform text preprocessing to clean raw text data. Here I combined the title with newscontent for easier manipulation. This is cleaned with the following considerations:

- remove all Chinese punctuations;
- remove stop words with reference to the document from *jieba* package[1];
- remove HTML tags with string starts with 'http' or 'https';
- convert all traditional to simplified Chinese characters using zhconv package[2].

We consider multiple approaches for noise removal and provide reasons for the better option.

1. The first approach is to identify exact matches with the terms in a-share list, including company's name or in full, and stock code, since a mentioned name must be relevant. Saved as def_data.csv. This filters out approximately half the data.

We analyse the remaining 550326 data without exact matches (def_data.csv) using similarity-based measures.

2. We attempted but later discarded the approach of using TFIDF vectorizer. TF-IDF is frequency-based for evaluating word importance, it lacks the ability to consider broader context and cannot capture the sparse representations of abbreviations. The process is written in task1_code. In the following approaches, we use the pre-processed data tfidf_data_sorted.csv, where I once used jieba.posseg to filter out the nouns for quicker text mining. (Fig 2)

标签	含义	标签	含义	标签	含义	标签	含义
n	普通名词	f	方位名词	s	处所名词	t	时间
nr	人名	ns	地名	nt	机构名	nw	作品名
nz	其他专名	v	普通动词	vd	动副词	vn	名动词
a	形容词	ad	副形词	an	名形词	d	副词
m	数量词	q	量词	r	代词	p	介词
c	连词	u	助词	xc	其他虚词	w	标点符号
PER	人名	LOC	地名	ORG	机构名	TIME	时间

Fig 2. Screenshot of parts of speech tags from jieba.posseg. We select the tags that start with 'n', and ORG.

3. We used pretrained BERT model for similarity-based mining of short forms. We used a smaller pretrained Chinese transformer AutoModel *ckiplab/albert-base-chinese*[3] with its corresponding pretained BertTokeniser *bert-base-chinese* with 11M parameters w.r.t. GitHub instructions[4], due to constrained environment. We tokenise and convert company names, full names, and company code of the a_share_list, together with tokenized_sequence.pt separately into embeddings. To

[1] https://github.com/fxsjy/jieba
[2] https://github.com/gumblex/zhconv
[3] https://huggingface.co/ckiplab/albert-base-chinese
[4] https://github.com/ckiplab/ckip-transformers

select the related rows, we compare each word with the a_share_embeddings.pt to measure the cosine similarity. The similarity values higher than threshold is labelled as related, in related_info.pkl and related_info2.pklthe corresponding news and a_share indices recorded for later inference.

- To document the process, we tried setting threshold = 0.9 results in only 2 companies considered as relevant.
- Another attempt with threshold = 0.7 results in around 2.5% higher coverage of the whole data.

```
Count of True values in related_list (threshold = 0.9): 1384
Count of True values in related_list2 (threshold = 0.7): 27366
```

Fig3. Output snippet of news count increase using BERT model with threshold 0.9 and 0.7.

(b) We introduce the filter rate as: filter rate = filtered news / total news for each approach.

Filter approach	News count	Filter rate
Exact match	486709	0.469327457607506
TF-IDF (discarded)	-	-
BERT (threshold 0.9)	488093	0.4706620316575622
BERT(threshold 0.7)	514075	0.4957161522995849

Task 1 Q2 – sentiment analysis and format submission of Task.xlsx

To be on the safe side, we use the dataset with exact matches with 100% cleaning accuracy for Q2.

With our aim for creating the Task1.xlsx format, the 'NewsID' and 'NewsContent' is extracted. To state the 'Explicit_Company', this has been done in Q1, where for each matched feature, we append the company 'name' (instead of 'fullname' w.r.t. *sample_Task1.xlsx*) distinctly separated by English comma.

As for the column 'labels', we approach our sentiment analysis using *a bert-base-chinese-finetuning-financial-news-sentiment-v2* from HuggingFace[5], from the contributor *hw2942*. This BERT model is finetuned on Chinese financial news specialised for text classification, with type *BertForSequenceClassification*. Compared to my layman-labelled data which has higher risk of introducing error, a fine-tuned model is advantageous as it allows adaptation to specific tasks, domain-specific vocabulary, leading to improved performance than non-fine-tuned models. The transfer learning capability of pre-trained BERT models enhances their efficiency in real-world applications, enabling quicker deployment.

Upon parsing configuration, it has 102M parameters with 21128 tokeniser, max 512 tokens; regarding the model output *label2id*, it classifies the sentence into "Negative": 0, "Neutral": 1 and "Positive": 2.

The sentiment polarity is determined by preprocess the text with the same method as Q1, then parse the tokens into the model. The labels are adjusted to 'Negative' as 0 and 'positive' as 1. To handle the class for 'neutral', the polarity is determined based on ranking the second highest SoftMax output probability. This aims to infer a nuanced polarity for neutral sentiments for relative confidence.

[5] https://huggingface.co/hw2942/bert-base-chinese-finetuning-financial-news-sentiment-v2/tree/main

We will address an issue on relevance of sentiment polarity on the mentioned A_shares. Analysing a single sentence allows a focused understanding of immediate sentiment, useful for extracting concise market signals. However, assessing sentiment across the entire text provides a holistic perspective, capturing nuanced sentiments that might be missed in isolated sentences (Fig. 4) .

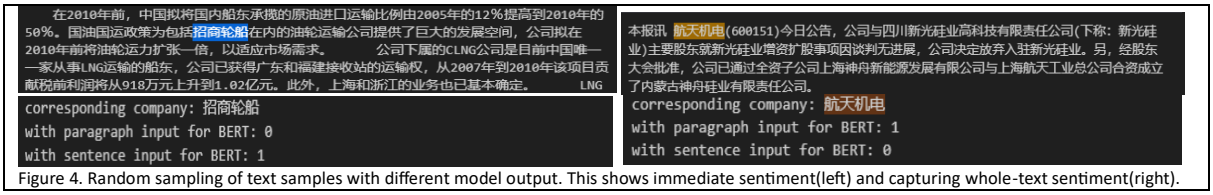

Figure 4. Random sampling of text samples with different model output. This shows immediate sentiment(left) and capturing whole-text sentiment(right).

Therefore, we have further extended our data preprocessing approach to extract the relevant sentence between full stops as input for the BERT model, shown in section 2.2-3 in Task1_code.ipynb. This gives quite a different output distribution (Fig 5).Since we are evaluating the impact of each news article on the A-share listed companies, we selected the data frame with the single sentene text input as xlsx for submission.

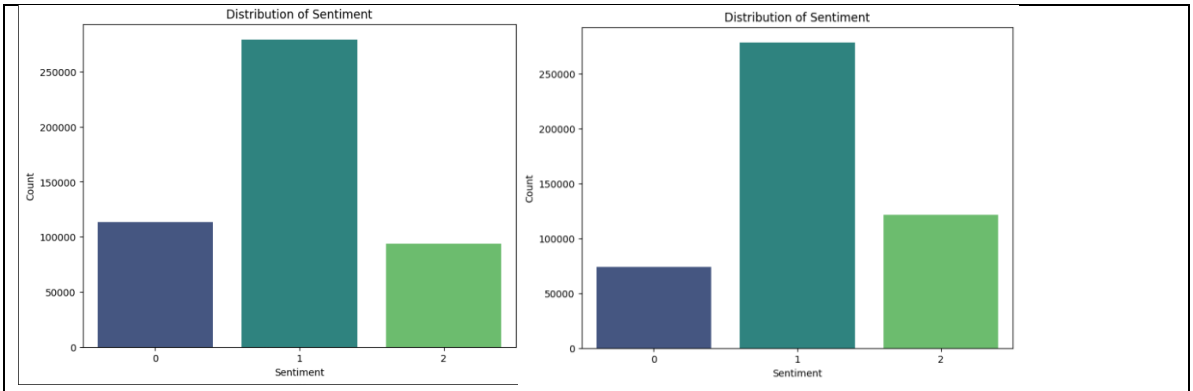

Fig 5. Output distribution with fine-tuned BERT with whole-text processing(left) and single-sentence processing (right).

Task 2 Q3: Using Python and Neo4j database for knowledge graph plotting

Documentation

1. We install the Neo4j Desktop (v1.5.9 into local drive, with the input files in KnowledgeGraph under the file section for supplementary. With proper explicit IP address, username and password configurations, we opened the Neo4j browser *bolt://localhost:7687* for visualisation (Fig6).

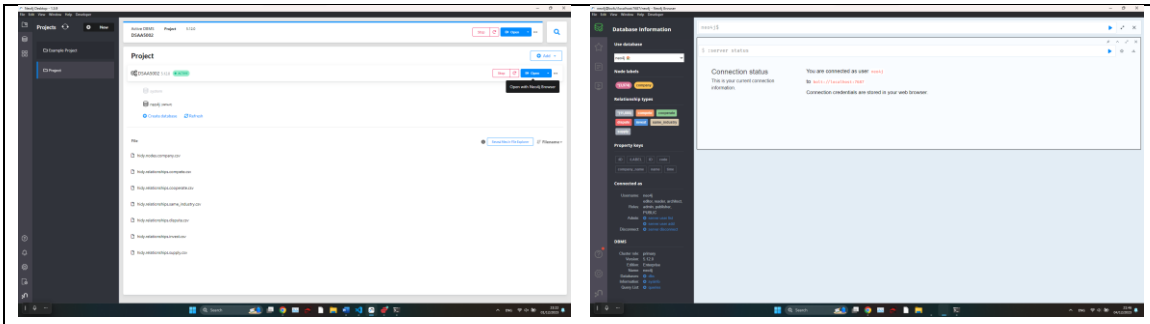

Fig 6. Screenshots showing Neo4j desktop (left) and Neo4j browser with information panel (right)

2. In VSCode, we imported *neo4j* package for initialising the GraphDatabase Python driver by parsing the uri and credentials.

3. The csv files were read using pandas. Next, we created 2 functions for creating company nodes and relationships by writing query in cypher using 'create' for company nodes. As for relationships, they were parsed by matching startID and endID of the companies. Additionally, we create the 'time' property when the column exists for each df. For graph construction we open a driver session.

Visualisations

1. We visualized all 3974 company nodes with 11666 edges / relationships (Fig.7).
2. 43 companies have competing edges (orange), and the majority that are clustered together are in the same industry (blue edges) (Fig. 8). Among them 4 has disputes, including 赣锋锂业 and 宁德时代; 金盾股份 and 世纪华, with edges shown in red (Fig. 9). Therefore, companies that share edge of the same industry does not necessarily drive a positive impact.

MATCH (n:Company) RETURN n;	MATCH p=()-[r:compete]->() RETURN p
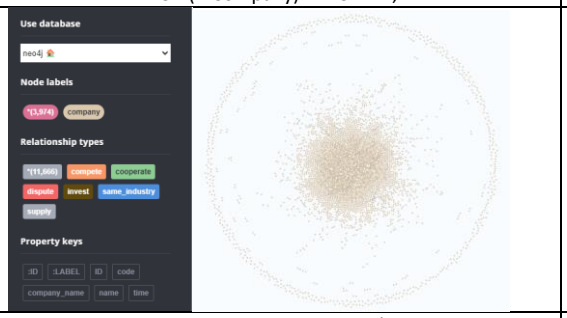	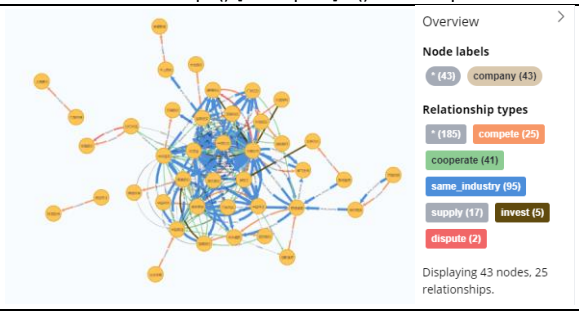
Fig 7. Graph of all company nodes with 6 edge / relationship types	Fig 8. Graph showing 43 competing companies

3. Among the competing companies, 6 of them do share other edges with non-opposing effects (cooperate, invest and supply), where these edges coexist and are bidirectional. (eg. 广发证券 compete with, and brings benefits to 浦发银行). (Fig. 10) This means opposing and non-opposing effect can overlap in question 4.

MATCH p=()-[r:compete]->() RETURN p	MATCH (a:company)-[r1:compete]-(b:company) MATCH (a)-[r2]-(b) WHERE type(r2) IN ['cooperate', 'invest', 'supply'] WITH a, b, COLLECT(r2) AS relationships FOREACH (rel IN relationships | CREATE (a)-[:cooperate_invest_supply]->(b) DELETE rel) RETURN a, b;
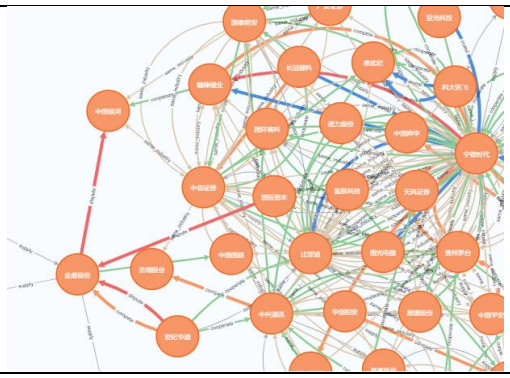	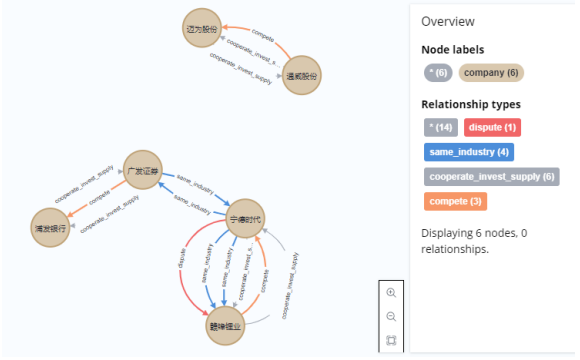
Fig 9. Graph of 4 companies with compete edges also has disputes	Fig 10. Coexisting competing and non-opposing edges between 2 companies

4. The company with the most cooperate edges is 宁德时代 (Fig. 11). Its directly connected nodes involve 103 companies. This shapes an interconnected network with 268 edges (Fig. 12), with 393 edges declaring the same industry.
5. The top suppliers are 宁德时代(22 edges), 中国电信(18 edges), 比亚迪(15 edges), 中兴通讯(9 edges), 中信证券(9 edges) , where they are interconnected as well. Their interconnections makes up 481 nodes with 2403 edges (Fig. 13)
6. 481 companies are involved in 439 disputes. Among the companies with dispute edges, the rank with highest number of disputes: 乐视网 (8 edges), ST 康得(6 edges), 东方财富(6 edges), 中信证

券(5 edges) and 申万宏源 (5 edges). (Fig. 14)

```
MATCH (node:company)-[r:cooperate]->(otherNode)
WITH node, COUNT(r) AS incomingCount, COLLECT(r) AS relationships
ORDER BY incomingCount DESC
LIMIT 1
RETURN node, relationships;
```

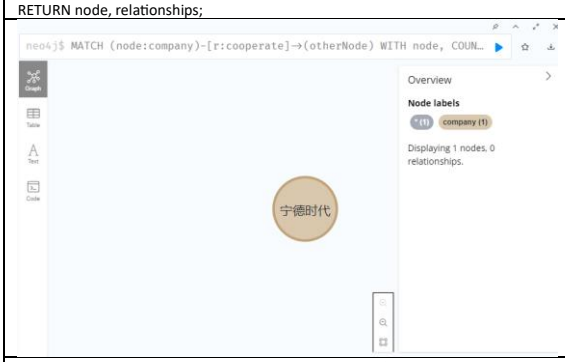

Fig 11. Graph of a single node company 宁德时代 with highest cooperation edges

```
MATCH (c:company {code: "300750.SZ", company_name: "宁德时代", ID: 743})-[r]->(connectedNode)
RETURN c, r, connectedNode;
```

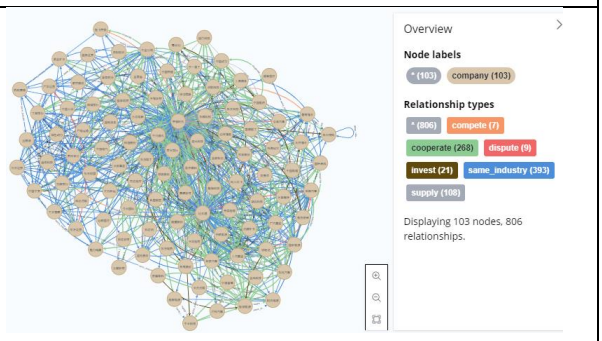

Fig 12. Graph showing directly connected nodes from 宁德时代 and their respective interconnections

```
MATCH (supplier:company)-[r:supply]->(customer:company)
WITH supplier, COUNT(r) AS outgoingSupplyCount
ORDER BY outgoingSupplyCount DESC
LIMIT 5
RETURN supplier, outgoingSupplyCount
```

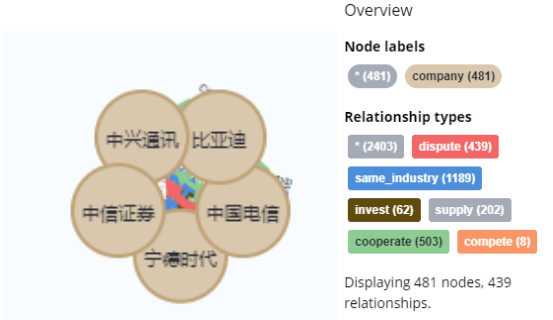

Fig 13. Graph showing top 5 companies with supplying edges

```
MATCH (entity:company)-[r:dispute]->()
WITH entity, COUNT(r) AS disputeCount
WHERE disputeCount > 1
RETURN entity, disputeCount
ORDER BY disputeCount DESC
LIMIT 5;
```

entity	disputeCount
(:company {code: "300104.SZ",company_name: "乐视网",ID: 2678})	8
(:company {code: "002450.SZ",company_name: "*ST康得",ID: 1101})	6
(:company {code: "300059.SZ",company_name: "东方财富",ID: 2176})	6
(:company {code: "600030.SH",company_name: "中信证券",ID: 3564})	5
(:company {code: "000166.SZ",company_name: "申万宏源",ID: 3330})	5

Fig14. Table of companies with ranked dispute counts

Task 2 Q4: Knowledge-Driven Financial Analysis

We first build the database for identifying synergising effect and opposing effect, or both (since there are companies that are both involved). We begin by segregating the dataset into two distinct scenarios, concatenating them separately while excluding the time column whenever it appears. Here are the cases:

1. Relation_oppo_dict = df_compete + df_dispute
2. Relation_tgt_dict = df_invest + df_supply + df_cooperate + df_same_industry.

Then, we reverse the column names, ensuring distinctiveness, and construct dictionaries to denote associations with other companies.

We separate two cases for labelling with dictionaries Implicit_Positive_Company and Implicit_Negative_Company. We read the explicit companies separated by comma for mapping and append. When label = 0, opposing company will be appended for the formal case with respect to each explicit company, whereas the latter one would be the related industry in terms of investment, supply, cooperation, or in the same industry. Vice versa for label = 1. This is submitted as Task2.xlsx.