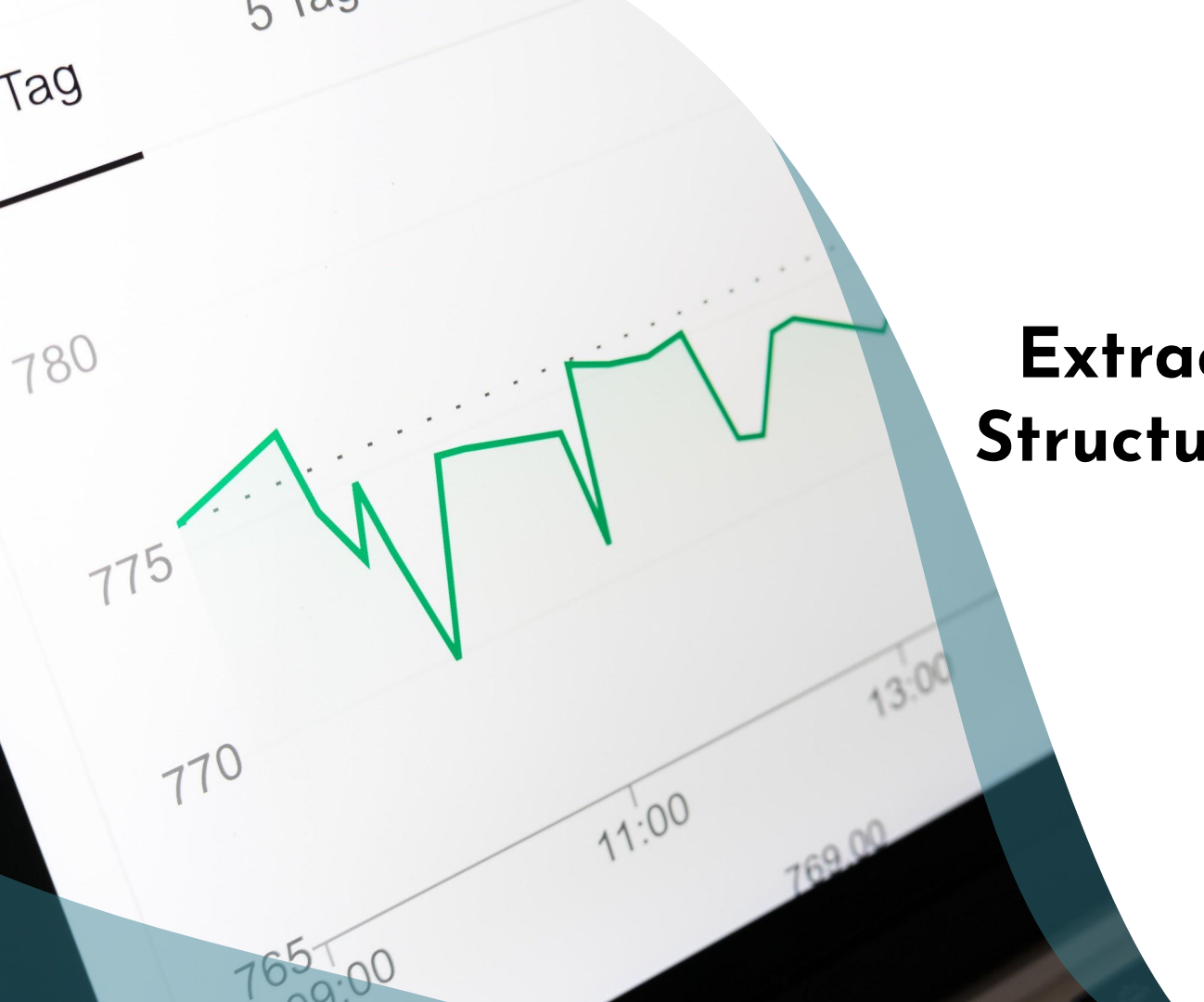


Extracting Graphical Structures from mixed Data Sources

JP Morgan

Aline Niyonsaba
Eric Ngabonzima
Ernest Kufuor
Ryan Harty

Advisor: Moise Busogi

Problem Definition

JPMorgan is looking for a way to identify communities of companies, as well as relationships between companies without having to guess at them by hand. They would like a more rigorous technical approach to figuring out which companies are related in order to guide processes like investment strategy and fraud detection.

Project Objectives

01

Store unstructured news article data about various companies in a structured format.

02

Identify relationships and gain insights between companies from news articles content.

03

Include other data sources such as stock price data and SEC reports to reveal additional relationships between companies.

04

Publicize results order to advance knowledge in this field.

Planned Deliverables

- Python package that can extract different entities from news articles and build graphs from them.
- Research paper or medium article
- Project final report

Prototype

Rudimentary Prototype

Prototype

Rudimentary Prototype

4. Remove pronouns

```
[ ]
    pronouns = ['I', 'You', 'It', 'He', 'She', 'We', 'They']
    suffixes = ["", "'m", "'re", "'s", "'ve", "'d", "'m", "'re", "'s", "'ve", "'d", "m", "re", "s", "ve", "d"]

    contraptions = [(p, s) for p in pronouns for s in suffixes]

    df_contraptions = pd.DataFrame(contraptions, columns=['pronoun', 'suffix'])

    df_contraptions['contraption'] = df_contraptions.apply(lambda x: x['pronoun'] + x['suffix'], axis=1)

    contraptions = df_contraptions.contraption.values
```

4. Define NER function

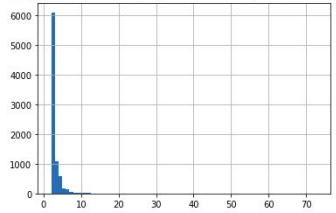
```
# define function

def get_ner_data(paragraph):
    '''
    - function to extract named entities from a paragraph
    - returns two data frames:
        - the first is a dataframe of all unique entities (persons and orgs)
        - the second is the links between the entities
    '''

    # remove newlines and odd characters
    paragraph = re.sub('\r', '', paragraph)
    paragraph = re.sub('\n', ' ', paragraph)
    paragraph = re.sub("'s", '', paragraph)
    paragraph = re.sub("“", '', paragraph)
    paragraph = re.sub("”", '', paragraph)
```

Prototype

Rudimentary Prototype

```
[ ] df_links.reset_index(drop=True, inplace=True)
    df_links['weight'] = df_links['count']
    df_links['weight'].hist(bins=max(df_links['weight']))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe5c7f3fd68>
```




```
[ ] df_links[['from', 'to', 'weight']].sort_values('weight', ascending=False).head(10)
```

	from	to	weight
5527	Johnson	NHS	73
4545	Hancock	NHS	36
3211	Dyson	Johnson	32
1841	Charles	NHS	29
4262	Government	NHS	26
4529	Hancock	Johnson	26
4688	Harry	Meghan	26
5604	Johnson	Sunak	24
2466	Cuomo	Trump	23
6999	NHS	UK	23

Timeline

Python Package

March — April

Week 1

Build data scraping functionality

Week 2

Implement graph functionality, using data from single source

Week 3

Build community detection functionality

Build graph evaluation functionality

Week 4

Build graph evaluation functionality

Build visualization functionality

Timeline

Journal Paper

Week 1

Read related journal papers

Week 2

Read related journal papers

Write a literature review section

Week 3

Write a literature review section

Week 4

Write a methodology and implementation section

Stakeholders and their Expertise

CMU Africa

- Eric Umuhoza
  - CMU Engineering faculty, experience in Big Data Analysis
- Moise Busogi
  - CMU Engineering faculty, experience in ML

JP Morgan

- Samuel Assefa
  - Experienced in AI/ML, practiced Data Scientist
- Parisa Hassanzadeh
  - Engineering background, experienced in ML and Graph Methods
- Srijan Sood
  - ML degree, experienced in Graph Methods

THANKS!

Any Questions?