A picture containing text

Description automatically generated

DSE 203 - Knowledge Graph Report

NFL Players with Chronic Traumatic Encephalopathy

(CTE)

Group #5 ● Ramona Henry, Alejandro Hohmann, Cammilus Perera ● DECEMBER, 2022

Table of Contents

[Section 1: Overview 3](#_Toc121637179)

[1.1 Context 3](#_Toc121637180)

[1.2 Objective 3](#_Toc121637181)

[1.3 Scope of Work 3](#_Toc121637182)

[Section 2: Data Sources 4](#_Toc121637183)

[Section 3: Data Sources 1 & 2 Integration Strategy 6](#_Toc121637184)

[Section 4: Data Source 3 Named Entity Recognition (NER) & LDA Topic Modeling 13](#_Toc121637185)

[Section 5: Data Sources 2 & 3 Integration Strategy 17](#_Toc121637186)

[Section 6 Knowledge Graph Implementation Strategy 19](#_Toc121637187)

[Section 7 Analytical Queries 19](#_Toc121637188)

[Section 8 Project Files 20](#_Toc121637189)

[Section 9 Revision History 20](#_Toc121637190)

# Section 1: Overview

## 1.1 Context

*Many former National Football League (NFL) players have been diagnosed with or have had chronic traumatic encephalopathy, or CTE. A definitive diagnosis so far can be made only post-mortem. However, an increasing number of former players are reporting symptoms of CTE.*

*According to 2017 study on brains of deceased gridiron football players, 99% of tested brains of NFL players, 88% of Canadian Football League (CFL) players, 64% of semi-professional players, 91% of college football players, and 21% of high school football players had various stages of CTE. However, this study had several limitations, including possible selection bias as families of players with symptoms of CTE are far more likely to donate brains to research than those without signs of the disease. Despite the limitations, the study still showed that CTE is far more common than once believed.*

## 1.2 Objective

*Organize data from multiple sources, about entities of interest associated with CTE in the NFL, and forge connections between them to represent a network of real-world entities—i.e. objects, events, situations, or concepts—and illustrates the relationship between them in a graph database and visualized as a graph model.*

## 1.3 Scope of Work

*Integrate data from three different sources which combines a mixture of structured, semi-structured, and unstructured data to construct a knowledge graph per below provided project requirements for UCSD’s DSE203 final project.*

***Project Requirements:*** *Your knowledge graph need not be humongous, but it should have a reasonable size (nodes + edges)*

* *100 is too low, 100k is high*
* *The number of edge labels should not be less than 7*
* *The number of node labels should not be less than 7*

*Your methods should include some degree of information extraction from text and some degree of value matching*

*Try to make your use cases realistic. Your example queries on the knowledge graph*

* *Should have some degree of complexity*
* *May include some analytical operations on graphs that are permitted by Neo4J (and Python if needed)*

# Section 2: Data Sources

*The project team combined a mixture of datasets from the following publicly available data sources.*

***Data source 1: Kaggle - NFL Statistics***

*\*\*Note: The project team only utilized the Basic\_Stats.csv dataset from data-source-1.*

*Data Variety - Structured data.*

*Data Source URL:* [*https://www.kaggle.com/datasets/kendallgillies/nflstatistics*](https://www.kaggle.com/datasets/kendallgillies/nflstatistics)

*Download URL:* [*https://github.com/mona-jandro-camm/dse203/tree/main/Datasets*](https://github.com/mona-jandro-camm/dse203/tree/main/Datasets)

*DDL Schema:*

*Graphical user interface

Description automatically generated with low confidence* Graphical user interface, text

Description automatically generated

***Data source 2: Wikipedia - NFL players with chronic traumatic encephalopathy***

*Data Variety - Semi structured and unstructured data.*

*Data Source URL:* [*https://en.wikipedia.org/wiki/List\_of\_NFL\_players\_with\_chronic\_traumatic\_encephalopathy*](https://en.wikipedia.org/wiki/List_of_NFL_players_with_chronic_traumatic_encephalopathy)

*Graphical user interface, text, application

Description automatically generated*

*Graphical user interface, text

Description automatically generated with medium confidence*

***Data source 3: UCSD’s AWESOME Database - US Newspaper Articles***

*Data Variety- Structured and unstructured data.*

*Table name: usnewspaper*

*Connection details:*

*host - awesome-hw.sdsc.edu*

*database name - postgres*

*DDL Schema:*

*Text

Description automatically generated* *Text

Description automatically generated*

# Section 3: Data Sources 1 & 2 Integration Strategy

***Method:***

*Extract NFL CTE affected player names from Wikipedia page and perform player name matching with NFL basic statistics dataset obtained from Kaggle. To extract the Wikipedia player names, the project team utilized Python’s BeautifulSoup library, and for the name matching between both datasets,* py\_entitymatching’s *attribute blocker EM process was utilized.*

***Technology Stack:***

*The following technology stack was primarily used to integrate data sources 1 & 2.*

* Python (Jupyter Notebooks)
* SQL (Postgres SQL, for Analysis only)
* Python Libraries

Graphical user interface, text

Description automatically generated

***Data Integration Steps:***

1. *Load structured CSV dataset from data source #1 - Kaggle into a DataFrame via py\_entitymatching’s read\_csv\_metadata method.*

*Sample Code Snippet:*

**

1. *Extract semi-structured and unstructured data from data source #2 - Wikipedia. The data extraction was done via Python’s Wikipedia and BeautifulSoup libraries.*

*\*\*Note: BeautifulSoup provides additional drill-down functionality to get at specific HTML tags in comparison to Python’s Wikipedia library.*

*Sample Code Snippet:*

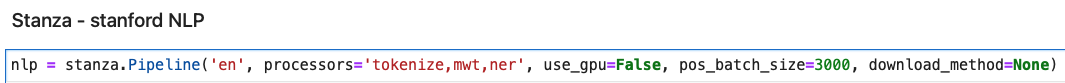
*Graphical user interface, application

Description automatically generated*

1. *Perform Named Entity Recognition (NER) and extract “PERSON” entities with the help of Python’s Stanza library for data source #2 - Wikipedia. The “PERSON” entity data was placed into separate data/lists structures based on each header category they are listed in data source #2. Those five different “PERSON” header categories are listed below.*

* *Players Affected Wiki Section*
* *Former Players affected with CTE Wiki Section*
* *Deceased players suspected of having had CTE Wiki Sction*
* *Living former players diagnosed with CTE or ALS or reporting symptoms consistent with CTE or ALS Wiki Section*
* *Former players listed as plaintiffs in lawsuits against the NFL for concussion-related injuries received after Wiki playing Section*

*Sample Code Snippet:*

**

Graphical user interface, text, application

Description automatically generated

1. Text normalization and pre-processing to clean player names, lower and remove punctuations from text.

*Sample Code Snippet:*

Graphical user interface, text, application, email

Description automatically generated

1. Combine the five different header categories data structures which contain NFL player names from data source #2 into one single DataFrame while creating following groups for each header category.

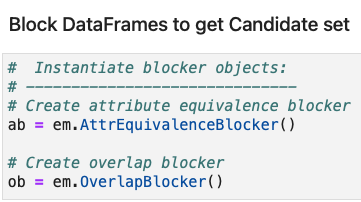
|  |  |
| --- | --- |
| **Wiki Header Section** | **CTE Category Group** |
| Players Affected Wiki Section | affected\_players |
| Former Players affected with CTE Wiki Section | pm\_former\_players |
| Deceased players suspected of having had CTE Wiki Sction | suspected\_deceased\_players |
| Living former players diagnosed with CTE or ALS or reporting symptoms consistent with CTE or ALS Wiki Section | cte\_als\_former\_players |
| Former players listed as plaintiffs in lawsuits against the NFL for concussion-related injuries received after Wiki playing Section | players\_nfl\_lawsuits |

*Sample Code Snippet:*Graphical user interface, application

Description automatically generated

1. *Perform entity matching Attribute Blocking on data source #1 & #2 player names and create a new DataFrame.*

*Sample Code Snippet:*

**

*Table

Description automatically generated with medium confidence*

1. *Perform Named Entity Recognition (NER) and extract “ORG” entities with the help of Python’s Stanza library for data source #2 - Wikipedia. The “ORG” entity data was placed into separate DataFrame in order to map NFL player names to an organization entity.*

*Sample Code Snippet:*

Text

Description automatically generated with medium confidence

1. The blocked/matched NFL player names from data sources #1 & #2 are mapped to their parent “ORG” entity, i.e. NFL, and the graph model direction, i.e. partent-to-child, attribute is added to DataFrame created in step #6.

*Sample Code Snippet:*

Graphical user interface, table

Description automatically generated

1. Construct and unit-test Neo4j knowledge graph to for data source #1 & #2.

Basic Rules logic: The nodes are simply players and their parent “ORG” entity NFL.

The edges are the five different CTE header categories from data source #2.

*Sample Code Snippet:*

Text

Description automatically generated

Chart

Description automatically generated

# Section 4: Data Source 3 Named Entity Recognition (NER) & LDA Topic Modeling

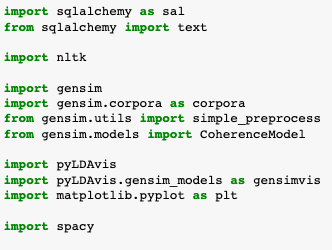
***Method:***

*Extract NFL CTE affected player names and keywords from USNewspaper articles using named entity recognition (NER). The project team created a LDA topic model in order to build further knowledge about each player from the unstructured data available in the newspaper articles. The team utilized various NLP Python libraries as outlined below including Spacy for NER and gensim for the LDA topic modeling.*

***Technology Stack:***

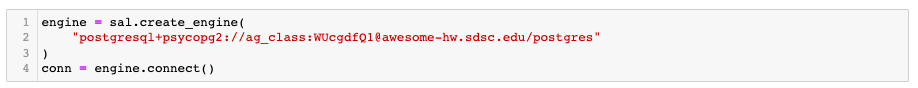
*The following technology stack was primarily used to extract information from data source 3.*

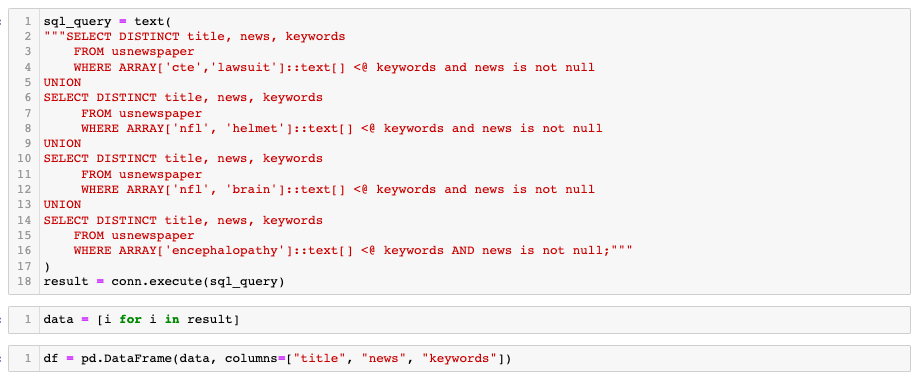
* Python (Jupyter Notebooks)
* SQL (Postgres SQL)
* Python Libraries



***Steps:***

1. *Load unstructured dataset from data source #2 - USnewspapers into a DataFrame using sqlalchemy and Pandas.*

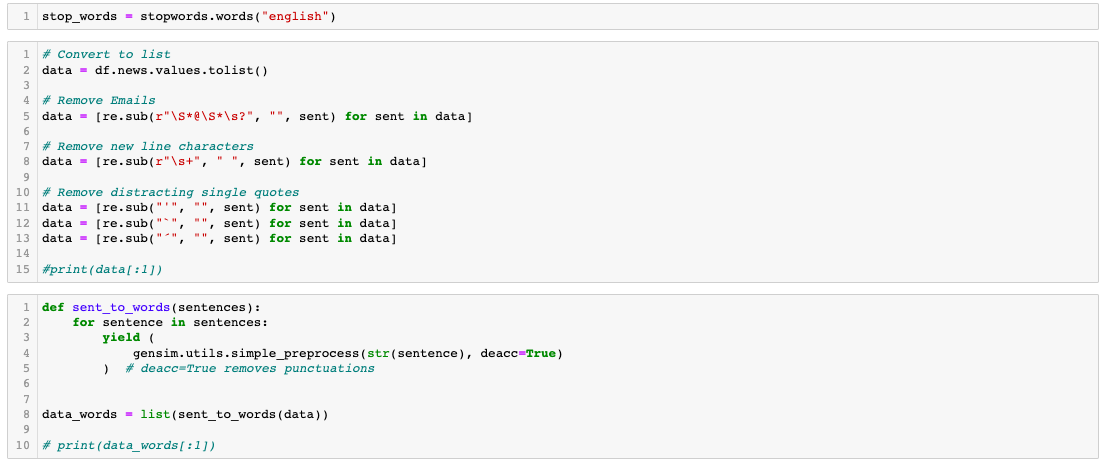
**

**

1. *Use Spacy to extract named entities of type “ORG” and “PERSON”.*

**

1. *Prepare the dataset for LDA topic model using a combination of re, Spacy, and gensim.*

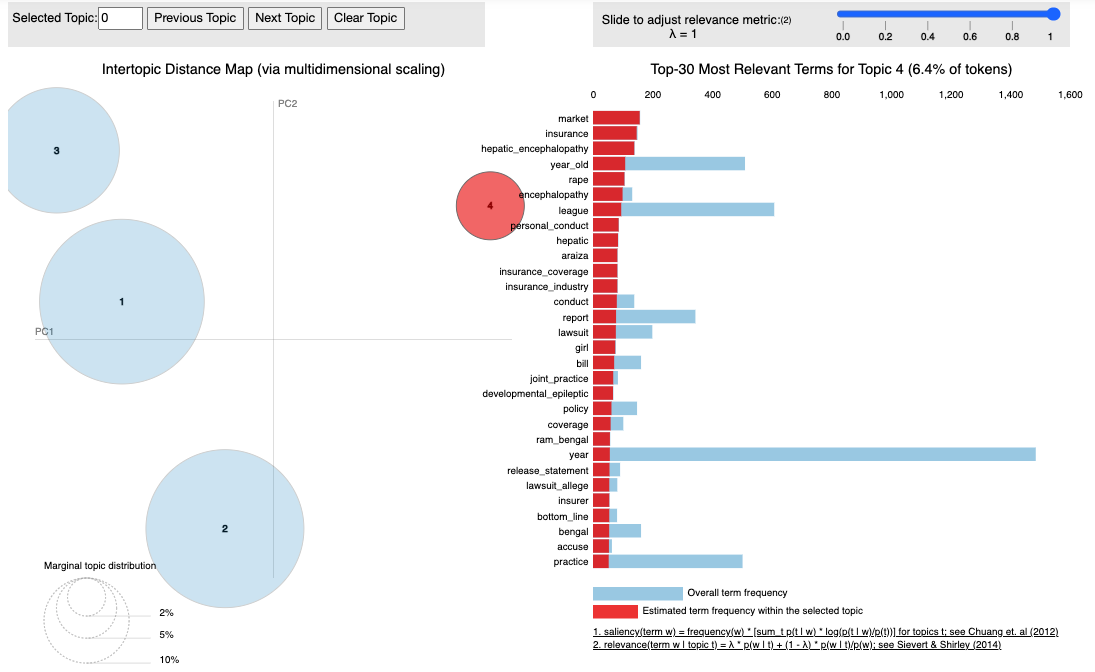
**

**

**

1. *Build LDA topic model with 4 topics and display it, focusing on one topic of interest in particular, that is, a topic with words associated with “personal conduct” issues.*

**

**

# Section 5: Data Sources 2 & 3 Integration Strategy

We found articles, where the list of players from data source 1 and 2 were mentioned and we created an additional set of nodes and relationships for the knowledge graph.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated with medium confidence

# Section 6 Knowledge Graph Implementation Strategy

Used command line to import node and relationship CSV files that were created in the Python notebook.

Import node command:

./bin/neo4j-admin import --force --multiline-fields=true --nodes=./import/CTE\_Nodes.csv --nodes=./import/CTE\_NEWS\_Nodes.csv --relationships=./import/CTE\_Relations.csv --relationships=./import/CTE\_NEWS\_Relations.csv

Graphical user interface, text, application

Description automatically generated

# Section 7 Analytical Queries

MATCH p=()-[r:MENTIONED\_IN|affected\_players]->() RETURN p LIMIT 25

MATCH p=()-[r:MENTIONED\_IN|suspected\_deceased\_players]->() RETURN p LIMIT 25

MATCH p=()-[r:cte\_als\_former\_players|MENTIONED\_IN ]->() RETURN p LIMIT 25

MATCH p=()-[r:suspected\_deceased\_players|MENTIONED\_IN ]->() RETURN p LIMIT 25

# Section 8 Project Files

***GitHub Repository***

<https://github.com/mona-jandro-camm/dse203>

**Project Slides**

https://docs.google.com/presentation/d/161il7paAIHnHalWCMS4KxR9bqgB1HZ0-Ui0PB2FHUZg/edit?usp=sharing

# Section 9 Revision History

| **Version** | **Date** | **Name** | **Description** |
| --- | --- | --- | --- |
| 1.0 | 12-6 | Camm Perera | Initial Document Creation |
| 1.1 | 12-7 | Mona Henry | Data Source 3 integration |
| 1.3 | 12 -10 | Mona Henry | Update Knowledge Graph section |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |