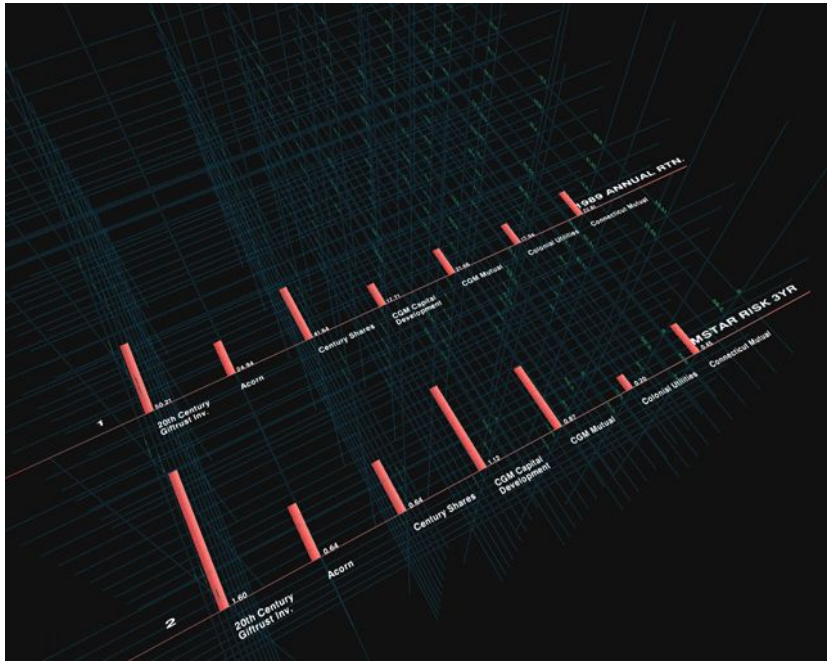
Sorcero Data Visualization

Forest Tennant, Michael Fahr, Bryce Mendenhall, Pao Yang, Rafael Carmen

Problem

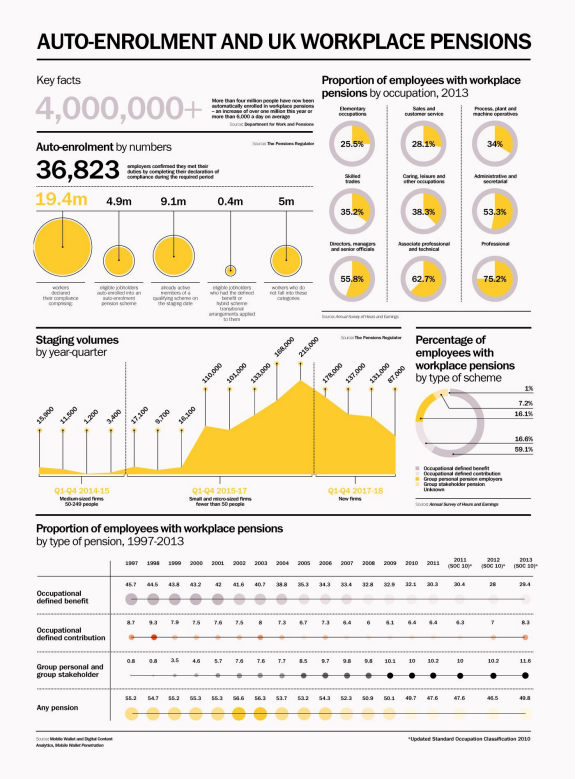
- Corporations generate large volumes of data
  - Information about documents
  - How users interact with these documents
- As the amount of data grows, it becomes progressively more difficult to visualize core information
- It is important for companies to have access to key performance indicators (KPI)
  - KPI: Key indicators of progress towards an intended result
  - In our project, KPIs mean heavily used topics or words within the document to be visualized
- Visualized solutions are to help quickly analyze data



http://www.inventinginteractive.com/2010/02/01/information-landscapes/

Objective

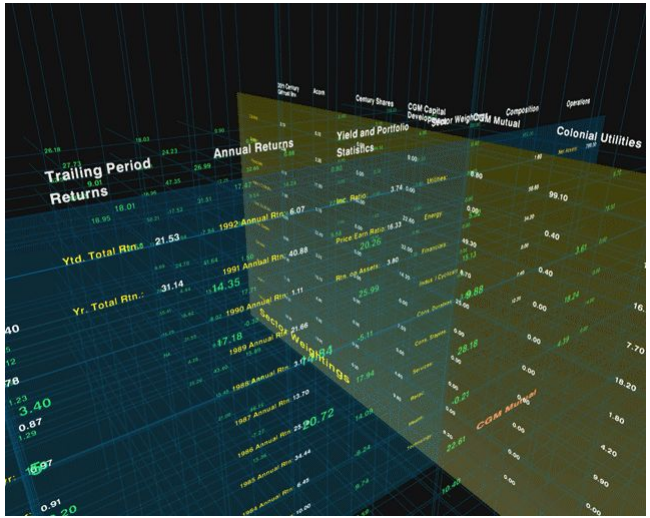
- Receive and process large volumes of data
- Use the data to provide
  - Meaningful visualization
  - Means of navigation
  - Insights into key performance indicators (KPIs)



Background

- Key Concepts
  - Natural Processing Language (NLP)
    - Focuses on how computer interpret and analyze natural language data
    - Examples
      - Speech recognition
      - Text-to-speech
  - Matplotlib
    - A Python library that allows for data to be graphed

- Related Work
  - MATLAB
  - IBM Watson
  - Information Landscapes



http://www.inventinginteractive.com/2010/02/01/information-landscapes/

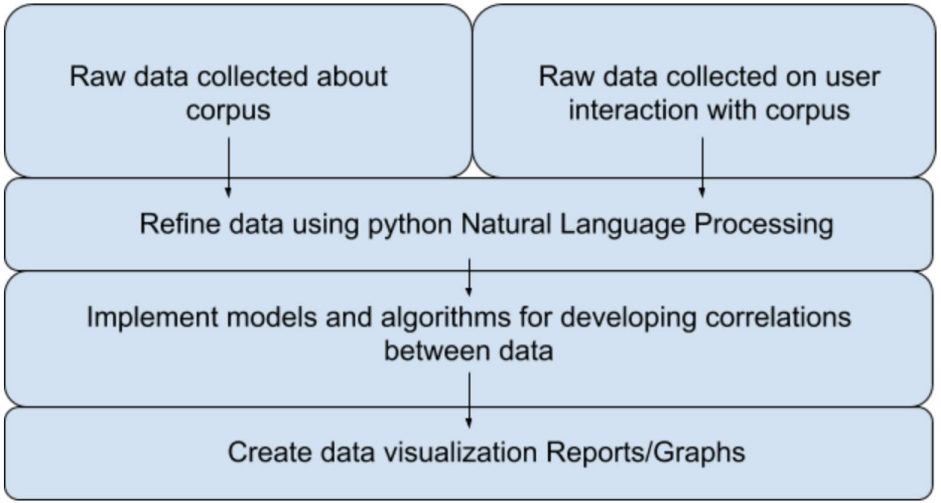
Data Visualization Project

- This project was chosen by our group due to our interest in using natural language processing and other statistical means to gather information from documents

- Our group is also interested in using Python, an extremely popular programming language that is currently used in a variety of industries

Design Overview

- Design Goals
  - Store the data
  - Group data into similar data sets
  - Provide additional means of navigation through data sets
  - Generate a communicable visual context based on the data provided
  - Show significant results with insights into KPIs
- High Level Architecture
  - Two sets of data
    - One set of data from corpus
    - One set of data from user interaction with the corpus
  - Use algorithms to sort the data
  - Run these two sets of data through a visualization tool to collect and process the data
  - Generate a straightforward visual based context on the data provided



Tasks & Schedules

Tasks

1. Explore and understand the background of data visualization and natural language processing.
2. Research other modern implementations to get an idea of other approaches.
3. Finalize architecture design and language of implementation.
4. Develop code to process the large volumes of data.
5. Create an algorithm for grouping/sorting the data into related fields.
6. Use the data to provide a meaningful visual context that suits the data.
7. Finalize the program by testing the application on multiple large sets of data
8. Document the final results.

Tasks	Dates
1. Do some research and understand the background of data visualization and natural language processing.	1/13-1/20
2. Research other modern implementations to receive an idea of other approaches.	1/21-1/27
3. Finalize architecture design and language of implementation.	1/28-2/10
4. Develop code to intake the large volumes of data.	2/11-2/24
5. Create an algorithm for grouping/sorting the data into related fields.	2/24-3/9
6. Use the data to provide a meaningful visual context that suits the data.	3/9-3/23
7. Finalize the program by testing the application on multiple large sets of data	3/23-4/6
8. Document the final results.	4/7-4/21

Final Deliverables

- Design Documentation:
  - Information on programming languages and libraries used in the implementation
  - Design process for implementing NLP with database
  - Features for the final graphical user interface developed

- Python code:
  - Analyzing data and creating relations
  - Visualizing the document and included relations

Data File Statistics and NLP

Multiple different CSV were created for relating information in the data file:

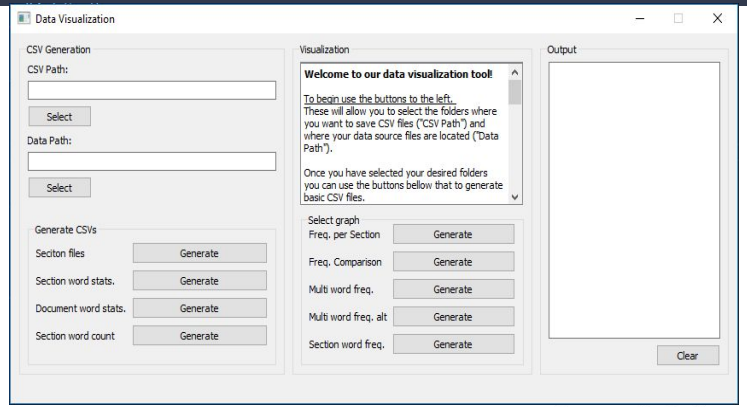
- Frequency of:
  - Words per section
  - Words per document
  - Important words per section
  - Important words per document
  - Nouns in the document
- Total word count:
  - Each section
  - Document
- Section of figures in the document

External libraries were used with Python for creating the final CSV documents:

- Natural Language Toolkit
  - Tokenize text
  - Group parts of speech
  - Remove stopwords
- Pandas
  - Ordering data
  - Creating CSV files

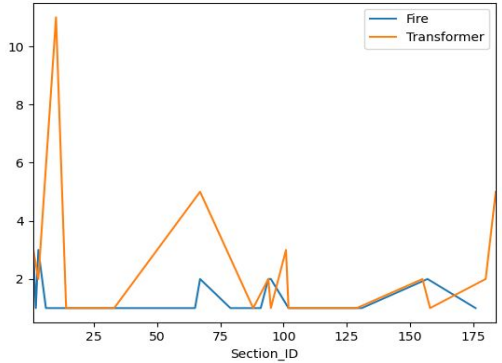
Results

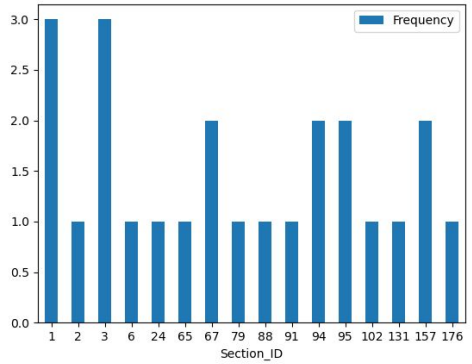
- GUI
  - Provides easy access for user to store CSV and find data path
  - Intuitive to understand
  - Instructions on how to use program in viewbox
- Generate CSVs
  - Four types of CSV files
- Select Graph
  - Five types of graphs



Multi Word Frequency

Frequency Per Section





Future Work

- Statistics and Natural Language Processing
  - Develop additional relations in the data
  - Create a better way for saving data relations from large volumes of data
  - Generate relations between multiple files of data

- GUI
  - Add a process for navigating the full data document
  - Include pre existing figures and graphs from the data document
  - Construct an area for obtaining a quick summary of the document
  - Error checking for data and CSV files

References & Key Personnel

[1] “Tableau: Business Intelligence and Analytics Software.” Tableau Software, www.tableau.com/.

[2] “MATLAB.” *MathWorks*, www.mathworks.com/products/matlab.html.

[3] “Smart Data Analysis and Visualization.” *Watson Analytics*, www.ibm.com/watson-analytics?lnk=hmhm.

[4] “Muriel Cooper: Information Landscapes.” *Inventing Interactive*, www.inventinginteractive.com/2010/02/01/information-landscapes/.

Michael Fahr – Fahr is a senior Computer Science major in the Computer Science and Computer Engineering department at the University of Arkansas. He has completed Software Engineering and Programming Paradigms.

Forrest Tennant - Tennant is a senior Computer Engineering major in the Computer Science and Computer Engineering department at the University of Arkansas Fayetteville. He has completed Software Engineering, Programming Paradigms and Programming Foundations I/II.

Bryce Mendenhall – Mendenhall is a senior Computer Engineering major in the Computer Science and Computer Engineering department at the University of Arkansas. He has completed Programming Foundations I/II, Software Engineering, and Programming Paradigms.

Rafael Del Carmen – Del Carmen is a senior Computer Engineering major in the Computer Science and Computer Engineering department at the University of Arkansas. He has completed. He has completed Cloud Computing, Programming Foundations I/II, Programming Paradigms, and Software Engineering.

Pao Yang – Yang is a senior Computer Science/Computer Engineering major in the Computer Science and Computer Engineering department at the University of Arkansas. He has completed Programming Foundations I/II, Software Engineering, and Programming Paradigms.