AI based Web scraping and Knowledge Graph

github Link: https://github.com/gvrstk/AIwebScraping/

Tarun Kumar Gotety – 45223015
Mayur Chobhe – 43364021
Mustafa Mamawala – 45049566
Dhruwa Ranjan – 45042728

GB/ GF: Enterprise Technology/ Colleague Experience IT

Abstract:

In today's fast-paced financial landscape, understanding customers comprehensively is crucial for financial institutions. Our project, "Risk Assessment using AI-Powered Web Data Scraping," addresses this need by creating a 360-degree view of the Risk Assessment through the integration of external publicly available information. Leveraging advanced AI and ML techniques, our solution aims to Scrape regulatory updates, policy changes, and compliance requirements from regulatory websites and financial authorities. Build a knowledge graph that connects regulations, compliance rules, banking processes, and relevant stakeholders to assist in maintaining regulatory compliance aiding financial institutions in making informed decisions.

Solution Overview:

1. Data Collection:

We will develop a web scraping engine capable of collecting relevant financial news and information about customers from publicly available sources such as RBI (Reserve Bank of India). This will include news articles, press releases, and social media posts.

2. Data Integration:

Our solution will integrate the collected external data with regulation and compliance details. This combined dataset will serve as the foundation for financial insights.

3. Natural Language Processing (NLP):

Utilizing NLP techniques, we will process the textual data extracted from news articles and other sources. Named Entity Recognition (NER) and sentiment analysis will help identify key entities, events, and sentiment related to Financial Institutions.

4. Feature Engineering:

We will engineer a diverse set of features, including financial institution, regulatory data, transaction patterns, external sentiment scores, and financial market indicators. In future, these features will enrich the customer profiles and contribute to risk assessment and spending potential prediction.

Solution Overview contd:

5.Risk Profile Calculation:

By analysing historical financial data, external sentiment, and market indicators, we will build a predictive model to calculate the risk profile of any financial institution. This model will assess the likelihood of default, bankruptcy, or other financial risks.

6. Visualization and Reporting:

We will create interactive dashboards that provide an intuitive visualization risk profiles and compliance requirements from regulatory websites . These visualizations will empower decision-makers with actionable information.

Expected Outcomes:

1. Comprehensive Insights:
Our solution will provide a holistic view of regulatory updates, policy changes, and compliance requirements from regulatory websites and financial authorities. Build a knowledge graph that connects regulations, compliance rules, banking processes, and relevant stakeholders to assist in maintaining regulatory compliance within the bank

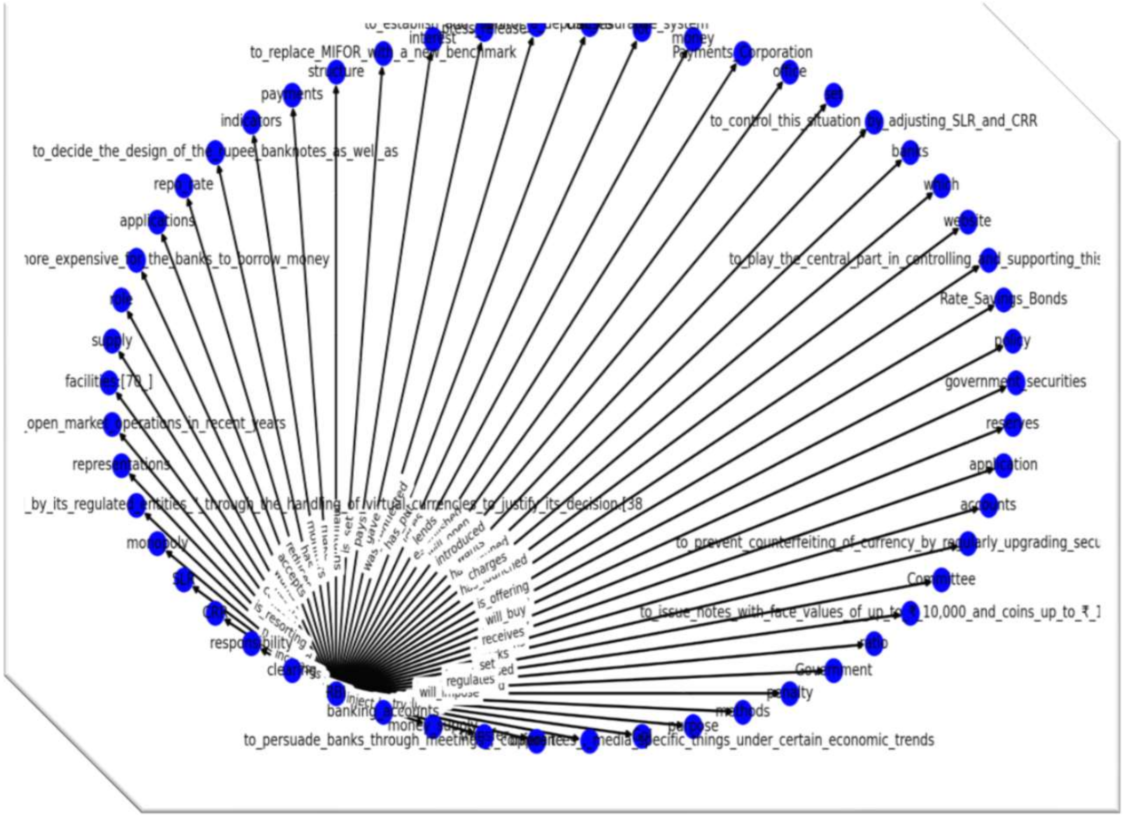
2. Enhanced Risk Management:
The AI-powered risk assessment model will assist in identifying potential financial risks associated with stakeholders, leading to more effective risk management strategies.

3. Informed Decision Making:
Predictions of spending potential will empower institutions to offer targeted products, improving customer satisfaction and loyalty.

4. Competitive Advantage:
By leveraging cutting-edge AI technologies for web data scraping and analysis, financial institutions can gain a competitive edge in understanding customers and tailoring their services.

5. Ethical Considerations:
We recognize the importance of privacy and data security. Our solution will adhere to all relevant data protection regulations and guidelines.

Sample Knowledge Graph

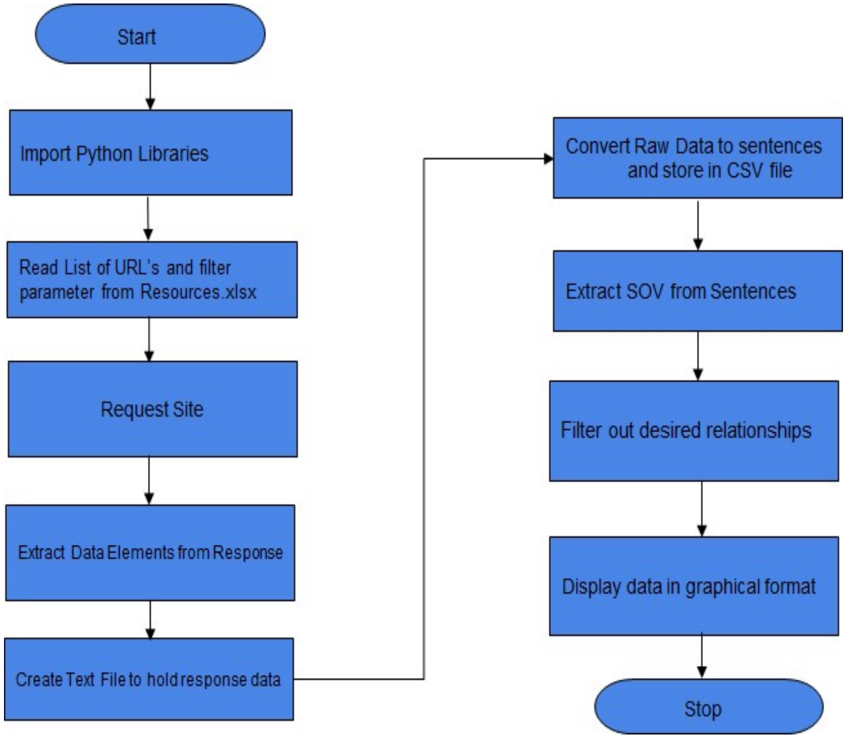

interest
to_replace_MIFOR_with_a_new_benchmark
structure
payments
indicators
to_decide_the_design_of_the_rupee_banknotes_as_well_as
repo_rate
applications
iore_expensive_for_the_banks_to_borrow_money
role
supply
facilities[70_]
open_market_operations_in_recent_years
representations
_by_its_regulated_entities_'_through_the_handling_of_virtual_currencies_to_justify_its_decision[38
monopoly
SLR
CRR
responsibility
clearing
RBI
banking_accounts
money_supply
to_persuade_banks_through_meetings
media_specific_things_under_certain_economic_trends
money
Payments_Corporation
office
to_control_this_situation_by_adjusting_SLR_and_CRR
banks
which
website
to_play_the_central_part_in_controlling_and_supporting_this
Rate_Savings_Bonds
policy
government_securities
reserves
application
accounts
to_prevent_counterfeiting_of_currency_by_regularly_upgrading_secu
Committee
to_issue_notes_with_face_values_of_up_to_₹_10,000_and_coins_up_to_₹_1
ratio
Government
penalty
methods
purpose
accepts
is_resorting
is_set
pays
gave
has_pa
lends
introduced
charges
has_launched
is_offering
will_buy
receives
set
regulates
will_impose
inject

This Knowledge Graph depicts various relationships between the financial entities within RBI (Reserve Bank of India)

This graph is generated using NetworkX with circular layout and labeled nodes and displayed using Matplotlib.

Pseudo Code

```
Import necessary libraries.
Define the current directory and load the English language model for spaCy.
Define a function readFile() to read data from an Excel file.
Define a function getData(link) to make an HTTP GET request to a given URL.
Define a function extractHTML(resp) to extract text from the HTML content using BeautifulSoup.
Read data from an Excel file using readFile().
Loop through the rows of the Excel file.
    a. Extract topic, URL, and filters from the Excel row.
    b. Make an HTTP GET request to the URL using getData().
    c. Extract text from the HTML content using extractHTML().
    d. Write the extracted text to a raw output file.
    e. Tokenize the text into sentences using spaCy.
    f. Write the tokenized sentences to a CSV output file.
    g. Extract subject-verb-object triples using textacy.
    h. Process the triples and create nodes and relations for the knowledge graph.
Create a directed graph using NetworkX.
Add nodes and edges to the graph based on the extracted relations.
Draw the graph using NetworkX with circular layout and labelled nodes.
Display the graph using Matplotlib.
```



Start

Import Python Libraries

Read List of URL's and filter parameter from Resources.xlsx

Request Site

Extract Data Elements from Response

Create Text File to hold response data

Convert Raw Data to sentences and store in CSV file

Extract SOV from Sentences

Filter out desired relationships

Display data in graphical format

Stop

Current State Design and Enhancements to be made to the model:

We have provided a script for web scraping and extracting data from URLs in an Excel file, and then generating a simple directed graph using the NetworkX library to represent relationships between entities.

But, there are several areas that we would want to consider improving and expanding upon to achieve a more comprehensive 360-degree customer view:

1. Modularization and Code Structure: While We have included functions like `readFile()` and `getData()`, it might be beneficial to further modularize our code into functions or classes for better organization, reusability, and maintainability.

2. Data Integration: We're scraping data from various URLs, but for a 360-degree customer view, We'd need to integrate this external data with internal customer data, which is not yet shown in the published code.

3. Advanced Text Processing and NLP: We are currently extracting sentences and performing basic subject-verb-object triple extraction. However, a comprehensive customer view may involve more advanced NLP techniques like sentiment analysis, entity recognition, and topic modelling to gain deeper insights from the extracted text.

4. Feature Engineering and Machine Learning: To calculate risk profiles and spending potential, we will likely need to incorporate machine learning models that can analyze the integrated data and provide predictions based on various features.

5. Visualization and Interpretation: While we are creating a directed graph, We might have to explore more advanced visualization techniques and tools to represent complex relationships and insights derived from the integrated data.

6. Error Handling and Robustness: Production-level code should include error handling mechanisms, such as handling connection issues, HTTP errors, and exceptions that might occur during scraping and data processing.

7. Data Privacy and Security: Ensure that We're handling sensitive financial data responsibly and in compliance with data protection regulations.

8. Data Storage and Retrieval: For a comprehensive solution, We might need to store the extracted and processed data in a structured manner, possibly using a database, to enable efficient retrieval and analysis.

9. Documentation: Consider adding comments and documentation to explain each step of our code and the purpose of different functions. This will make it easier for others (and our future self) to understand and work with the code.

10. Testing and Validation: Thoroughly test our code with different scenarios and validate the results to ensure that the data extraction, processing, and visualization are accurate and meaningful.

11. Scalability: Consider how the current code will scale as more data is added. Optimize our code to handle larger datasets efficiently.

Conclusion:

"Integrated Regulatory and Risk Assessment using AI-Powered Web Data Scraping" is a powerful solution that aligns with the theme of Data Con. It addresses the challenge of creating a 360-degree view by harnessing AI-driven web scraping and data analysis techniques. This project has the potential to revolutionize how financial institutions understand and amend policies or regulations, ultimately leading to better governance of a bank and informed decision-making.

Source code available on: https://github.com/gvrstk/AIwebScraping/

Thank You