**PROJECT REPORT**

**BIOMEDICAL BIG DATA KNOWLEDGE BASE**

**Niteesh Nitin Singh**

**Overview**

This project involves creating an NLP-based information extraction pipeline to process collections of biomedical articles. The modified code extracts linguistic features from documents, stores the processed information in a JSON file, and then loads it into a MongoDB database for structured storage and retrieval.

**Objectives**

1. **Information Extraction (IE)**: Utilize NLP models to extract tokens, part-of-speech (POS) tags, named entities (NER), and lemmas for each sentence. OpenIE is used to generate triples, which capture sentence-level relations.
2. **Data Export to JSON**: Store the extracted information in JSON format as an intermediary step before uploading it to MongoDB.
3. **Database Storage**: Load the JSON files into MongoDB collections for efficient data management and retrieval.

**Tools and Libraries**

* **Stanza**: Employed with multiple biomedical models for specific biomedical entity recognition.
* **Stanford OpenIE**: Extracts triples representing relational information in both scripts.
* **MongoDB**: Stores the processed JSON data for queryable structured storage.
* **Python Libraries**: *pandas* for handling CSV data, and *xml.etree.ElementTree* for parsing XML files.

**Script Details and Execution - Processing Biomedical Articles**

1. **Biomedical Model Loading with Stanza**:
   * Initialized separate Stanza pipelines for each biomedical model (e.g., *bc5cdr, bionlp13cg*, etc.) to recognize various biomedical entities.

A computer screen shot of a program code

Description automatically generated

1. **XML Parsing and Document Processing**:
   * Parsed *pubmed24n1220.xml*, extracting the PubMed ID, title, and abstract from each article.

A computer screen shot of a program

Description automatically generated

* + Each abstract was processed to extract tokens, POS tags, lemmas, biomedical entities, and triples. The processed data was structured as JSON entries for each article.

A computer screen shot of a program

Description automatically generated

A computer screen shot of a computer code

Description automatically generated

1. **Saving and Storing in MongoDB**:
   * Saved the processed data to *biomedical\_knowledge\_base.json.*

A screen shot of a computer

Description automatically generated

* + Reloaded the JSON file and inserted its contents into the *MedicalKnowledgeBase* collection within the *BIOMEDICAL\_ARTICLES* MongoDB database.

A screen shot of a computer program

Description automatically generated

**Results**

1. **Code Executions**:

A screenshot of a computer program

Description automatically generated

1. **MongoDB Collections**:

A screenshot of a computer

Description automatically generated

1. **JSON Exports**:

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

**Conclusion**

This project demonstrates a comprehensive NLP pipeline and data storage workflow for unstructured text. Storing the processed data in JSON files before MongoDB insertion provided a versatile format for further processing or backup. By utilizing *Stanza* with MongoDB, the project showcases a scalable approach to managing large text datasets across biomedical domains.