**Data Science Project Report: News Article Analysis**

**Project Overview**

**Objective**

The primary objective of this project was to develop a comprehensive data processing application that can:

* Scrape news articles from URLs
* Extract named entities (persons and organizations)
* Perform sentiment analysis
* Store processed information in a database
* Provide a user-friendly interface for interaction

**Implementation Details**

**1. Data Scraping**

The project implemented a robust web scraping mechanism using Python libraries:

* **Libraries Used**:
  + requests for fetching web content
  + BeautifulSoup for HTML parsing

**Key Function: fetch\_article(url)**

python

Copy

def fetch\_article(url):

try:

response = requests.get(url)

soup = BeautifulSoup(response.content, 'html.parser')

article\_text = " ".join([p.text for p in soup.find\_all('p')])

return article\_text

except Exception as e:

print(f"Error fetching article: {e}")

return None

**2. Named Entity Recognition (NER)**

**Approach**

Two NER strategies were explored:

1. **Pre-trained Model**: Using spaCy's en\_core\_web\_sm model
2. **Custom NER Model**: Training a specialized model for enhanced entity extraction

**Custom Model Training Process**

* Created a training dataset with annotated examples
* Used spaCy's training framework
* Implemented checkpointing during model training
* Saved and exported the custom NER model

**Entity Extraction Function**

python

Copy

def extract\_entities(text):

doc = nlp(text)

entities = {"PERSON": [], "ORG": []}

for ent in doc.ents:

if ent.label\_ in ["PERSON", "ORG"]:

entities[ent.label\_].append(ent.text)

return list(set(entities["PERSON"])), list(set(entities["ORG"]))

**3. Sentiment Analysis**

**Methodology**

* Utilized VADER (Valence Aware Dictionary and sEntiment Reasoner) Sentiment Analyzer
* Classifies sentiment into three categories: Positive, Negative, Neutral

**Sentiment Analysis Function**

python

Copy

def analyze\_sentiment(text):

score = analyzer.polarity\_scores(text)

if score['compound'] > 0.05:

return "Positive"

elif score['compound'] < -0.05:

return "Negative"

else:

return "Neutral"

**4. Database Storage**

**Database Schema**

* Used SQLAlchemy for Object-Relational Mapping (ORM)
* SQLite as the database backend
* Created an Article model to store:
  + URL
  + Article Text
  + Extracted Entities
  + Sentiment Classification

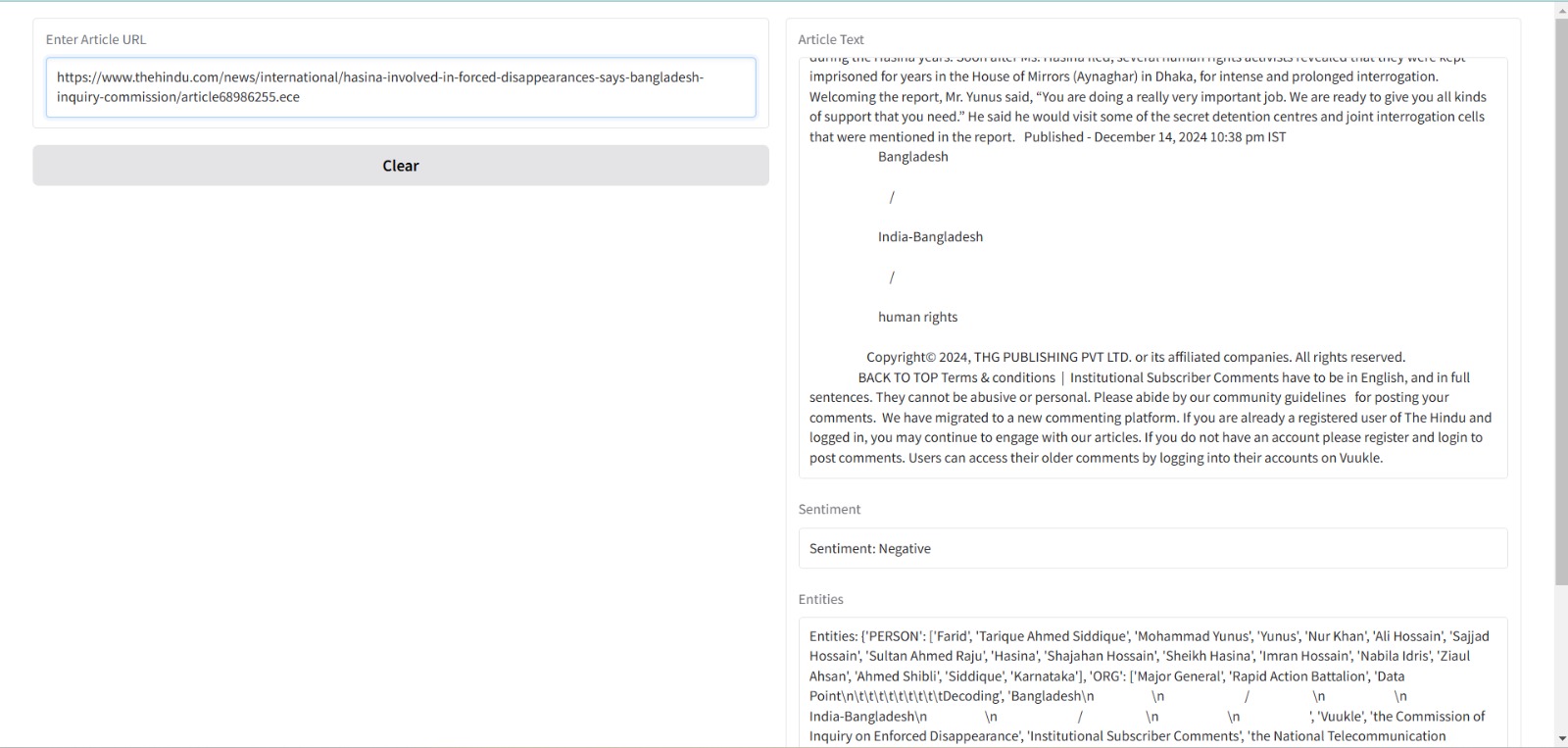
**5. User Interface**

**Gradio Interface**

* Developed an interactive web interface
* Allows users to input article URLs
* Displays:
  + Article Text
  + Extracted Entities
  + Sentiment Analysis Results

**Challenges and Limitations**

1. **Entity Extraction Accuracy**
   * Dependence on pre-trained models
   * Variability in entity recognition across different text types
2. **Sentiment Analysis Nuances**
   * VADER's limitations in understanding complex contextual sentiments
   * Potential misclassification in nuanced text
3. **Scalability Considerations**
   * Current implementation processes one article at a time
   * Limited error handling for complex web scraping scenarios



**Future Improvements**

1. Expand training data for custom NER model
2. Implement more sophisticated sentiment analysis techniques
3. Add support for multiple languages
4. Enhance error handling and robustness
5. Implement distributed processing for multiple articles

**Conclusion**

The project successfully demonstrates a end-to-end data processing pipeline for news article analysis, integrating web scraping, named entity recognition, sentiment analysis, and database storage with a user-friendly interface.

**Technologies Used**

* Python
* spaCy
* VADER Sentiment Analyzer
* SQLAlchemy
* Gradio
* BeautifulSoup
* Requests

**Code Repository**

<https://github.com/shivbhagat03/Project>