Eskalate NLP Interview: Brief Report

Candidate:Melkamu Tesema
Title: Enhanced NLP System for News Extraction, Summarization, and Agentic Design

1. Introduction & Objective

This project presents a comprehensive NLP pipeline that handles real-world news data through extraction, summarization, and AI agent design. It uses the News Category Dataset and demonstrates practical applications of regex, SpaCy, TextRank, and T5 transformer models. The end-goal is to conceptualize an agent capable of delivering concise news briefs based on user queries.

2. Data Preparation & Exploratory Analysis

Sample: 5,000 articles randomly selected from 200k+ for efficiency.

Preprocessing Steps:

- **Lowercasing**
- **Tokenization**
- **Stopword Removal**
- **Lemmatization** (via SpaCy)

Key EDA Outputs:

- **Category Distribution**: Visualized with bar plots.
- **Word Frequency**: WordCloud and bar chart of most frequent terms.
- **Named Entity Frequency**: Top types (PERSON, ORG, GPE), with 'Donald Trump', 'Obama' among top PERSONs.

---

3. Information Extraction & Summarization

A. Entity & Information Extraction

- **Regex Dates**: Fast extraction of `YYYY-MM-DD` patterns.
- **SpaCy NER**: Identifies context-aware named entities like people, locations, organizations.

B. Summarization

- **TextRank (Extractive)**: Selects top-ranked original sentences. Fast and reliable.
- **T5 (Abstractive)**: Rewrites and summarizes text using transformers.

Showcase Example:

Headline: Biden At UN To Call For Ukraine War Accountability
Entities: Biden, UN, Ukraine, Russia, Joe Biden
T5 Summary: president joe biden is expected to sharply condemn Russia's war against Ukraine at the U.N. General Assembly.

---

4. Agentic System Design: News Aggregator Agent

A. Scenario

Use Case: A journalist or policy analyst wants a brief on a specific topic (e.g., "interest rate hikes").

B. Agent Architecture

Goal: Deliver a topic-specific report with summaries and named entities.

Tools:

- `DocumentSearcher(topic, date_range)`: Filters relevant documents.

- `Summarizer(document_text)`: Applies T5 summarization.

- `InformationExtractor(document_text)`: Extracts entities using SpaCy.

Planning Strategy:

```
def news_aggregator_agent(query):
    topic, date = parse_query(query)
    articles = DocumentSearcher(topic, date)
    memory = {'summaries': [], 'entities': set()}

    for a in articles:
        memory['summaries'].append(Summarizer(a['text']))
        ents = InformationExtractor(a['text'])
        memory['entities'].update([e for e, label in ents if label in ['PERSON', 'ORG', 'GPE']])

    return synthesize_report(memory)
```

Final Report Structure:

- Summaries from T5

- List of people, organizations, locations

Memory:

- **Short-Term**: Task-specific memory (summaries, entities)

- **Optional Long-Term**: Store user preferences and past queries for personalization

5. Evaluation & Results

Qualitative Results:

Method	Output Quality
Regex	Precise but rigid
SpaCy NER	Accurate contextual entity recognition

TextRank	Factual but sometimes disjointed
T5	Human-like, fluent, better for briefings

Example Output:

Original: Biden Says U.S. Forces Would Defend Taiwan If China Invaded.
T5 Summary: u.s. forces would defend taiwan if china invaded. president vows as tensions with china rise.

---

6. Challenges & Future Work

Challenges:

- T5 requires truncation and is computationally heavy
- SpaCy misses some nested or rare entities

Enhancements:

- Use vector search for semantic DocumentSearcher
- Add long-term memory with FAISS or Redis
- Deploy as an interactive web-based assistant

---

7. Files and Structure

- `Eskalate_NLP_Interview.ipynb`: Full notebook with code, outputs, visualizations
- `agent.md`: Modularized agent logic (optional extension)

Thank you for the opportunity!