Project Overview

This project aims to develop a contextual multi-armed bandit (CMAB)-based crawler to explore Wikipedia pages and identify those most relevant to a given subject.

The system is composed of five core Python files, each playing a unique role:

- **main.py**: Runs the entire crawler system, coordinating the epsilon-greedy and UCB agents to manage exploration and decision-making.
- **webpage_crawler.py**: Contains the WikipediaCrawlerWithCMAB class, which handles the core crawling functions and navigates Wikipedia links.
- **cmab_epsilon_greedy.py** and **cmab_ucb.py**: Define the two CMAB agents that facilitate learning. The first file implements an epsilon-greedy agent, while the second uses an Upper Confidence Bound (UCB) agent.
- **plot.py**: Includes functions to visualize and save the results, allowing for easy comparison of each strategy's performance.

Key Features:

Content Extraction: The crawler uses BeautifulSoup to pull content from Wikipedia pages and find links within each page, allowing it to explore further and build a network of related pages.

Hybrid Search Strategy: To ensure thorough exploration, I developed a hybrid search strategy that combines Breadth-First Search (BFS) and Depth-First Search (DFS). This combination allows the crawler to balance wide-ranging exploration with deeper dives into promising links, providing a more effective search across diverse areas while also focusing on relevant content.

Memory Management: To prevent memory overload and terminating the method on time, I implemented a control to ensure the crawler doesn’t reach full memory capacity.

Similarity Check: As crawler explores, it checks the cosine similarity between the subject and the topic and content of each page it encounters, using a text embedding model from Hugging Face called paraphrase-MiniLM-L6-v2. This compact transformer captures the context of phrases efficiently, making it suitable for similarity comparisons between short text passages.

CMAB Agents: The system includes two CMAB agent types:

- **Epsilon-Greedy:** This agent mostly selects pages with high past rewards but will occasionally explore random pages to avoid missing potentially valuable links.
- **UCB:** This agent strikes a balance between exploring new pages and leveraging known ones with high potential, often resulting in bolder choices aimed at maximizing cumulative reward.

Reward Function Testing: After testing various reward functions, I selected the one that provided the best cumulative reward while maintaining effective exploration. This optimized reward function has proven the most effective for the crawler's learning and exploration balance.

Results

- **Epsilon-Greedy** strategy is significantly outperforming **UCB** in terms of cumulative reward over time.
- Epsilon-Greedy performs better in selecting relevant pages based on similarity to the target.

Challenges Faced

1. **Hyperparameter Tuning**: Finding the optimal parameters for each agent required extensive testing and adjustments.
2. **Efficient Link Exploration**: Balancing high-speed navigation with accuracy was challenging, and achieving an efficient yet comprehensive exploration strategy required significant refinement.
3. **Embedding Model Selection**: Choosing an embedding model that was both effective and efficient was critical to performance.

4. **Long Execution Time**: The crawler's long runtime hindered hyperparameter tuning and evaluation, posing challenges in achieving optimal results quickly.