PROJECT PROPOSAL FOR CS 839

Effect of Tokenizers and String Matching Algorithms on Identifying Similar Documents

Divy Patel, Sujay Chandra Shekara Sharma, Venkata Abhijeeth Balabhadruni
Team ID: 16

1) Problem Description

Document similarity is a fundamental task in various domains such as information retrieval, natural language processing, plagiarism detection, and recommendation systems. Text tokenization forms the building blocks for implementing and achieving applications in such domains. Understanding how different tokenization methods and string-matching algorithms affect the accuracy and efficiency of identifying similar documents is important. This exploration is crucial for optimizing document similarity models for resource utilization and selecting the best algorithms to fulfill the user-query SLOs(Service Level Objectives) in the form of user-query accuracy, latency, cost and throughput requirements.

In this project, we will build a framework that takes input as a list of news headlines, a target headline and user-query SLOs. The output will be a list of similar headlines. First, the framework evaluates different combinations of tokenizers and String-Matching Algorithms (both rule-based and ML-based) based on a certain set of performance metrics and chooses the combination which best fulfills the user's SLOs. The tokenizer-algorithm combination obtained from the above program can then be considered an optimal combination given the query requirements and can be used to generate the output list of similar headlines. We will also explore how an open-source LLM model performs for this task.

2) Data

We plan to utilize the UCI Machine Learning News Aggregator Dataset, which comprises over 400,000 entries. Each entry contains a headline along with an alphanumeric ID corresponding to the news story it addresses. Altogether, there are approximately 7,000 distinct news stories referenced by these headlines.

3) Performance

We will assess our model's performance using the following metrics:

- Accuracy
- Precision
- Recall
- F1 Score
- Task Runtime
- CPU and Memory utilization

4) Challenges

- Different tokenization methods may have varying effectiveness in capturing semantic similarities between headlines. Choosing the appropriate tokenization strategy can be challenging.
- Selecting the right string-matching algorithms and tuning their parameters (like the value of N in N-gram) can be challenging. Rule-based algorithms may not capture complex semantic relationships, while machine learning-based approaches may require significant computational resources/training data.

5) Hardware and Software

We believe specialized hardware such as GPUs may not be necessary for this project. However, we intend to utilize the CS Lab InstGPU Cluster if the need arises.

6) Timeline

Date	Goal
March 25, 2024	Clean and prepare the dataset for further tasks.
April 8, 2024	Implement the different tokenization techniques and Rule-based/ML-based String-matching algorithms.
April 22, 2024	Test how various combinations of tokenizers and string-matching algorithms perform and make observations.

April 29, 2024	Explore how an open-source LLM model performs for this task.
May 6, 2024	Compile all results and observations into a project report.
Misc	Meet every Friday to discuss the project's progress.

7) Misc
N/A