LLM-Based Text Embeddings for Graph Link Prediction

Andrew Sun (ACS22) *Henry Xue (HHX1)* *Nathan Calzat (NGC5)*

Objective

The goal of this project is to integrate LLM-based text embeddings with a graph-based link prediction method and evaluate its effectiveness in predicting relationships between nodes. By leveraging large-scale web data from the Common Crawl dataset, the project aims to enhance link prediction models by incorporating semantic text information extracted from web pages.

Dataset

Common Crawl

Description: Open-source repository of web crawl data (∼1 billion pages).

Content: HTML, metadata, and raw text extracted from web pages.

Available Data Formats: *Web Archive* (full page snapshots), *Web Archive Text* (page metadata), *Web Extracted Text* (extracted raw text from HTML).

Accessibility: Hosted on Amazon S3 (commoncrawl bucket). Can be accessed via AWS CLI, Spark, or Python libraries like warcio.

Methodology

Link-prediction method

The core of our project involves constructing a graph from the Common Crawl dataset. We can initially establish links based on existing hyperlinks and then augment these nodes with dense text embeddings that capture the context of the content. This graph provides the foundation for our link prediction method, which aims to identify missing and potential new relationships between web pages. To predict links, we can frame the problem as a binary classification task, where each pair of nodes is evaluated to determine the likelihood of an edge between them. By comparing predicted links with the actual web structure, we can evaluate the model's effectiveness, ultimately aiming to enhance our link prediction with LLM-based embeddings.

Pre-trained LLM

We'll compare and contrast two main different strategies to generate the embeddings for the web pages:

(1) Using open-source LLMs that have been fine-tuned to generate text embeddings—such as Sentence Transformer variants (SBERT)—to directly convert preprocessed, tokenized chunks of webpage text into fixed-length, semantically rich vectors.

(2) Extracting hidden state representations from open-source SOTA generative LLMs (e.g., models like Llama or DeepSeek) where we process the entire webpage's text and retrieve hidden layer representations from specific layers (will also be a point of exploration).

Most SOTA LLMs have a large enough context length, so we will be able to generate a single, unified embedding for the pages. If the hidden layer activations we extract are too high-dimensional, we'll look into dimensionality reduction techniques as well, like PCA or autoencoders. Through examining these methods on generating semantically rich text embeddings using LLMs, we'll be able to examine their effectiveness on graph-based link prediction.