**Documentation: Text Analysis and Visualization of Medical Reports**

**Project Overview**

This project aims to analyze a dataset of medical reports containing various textual attributes and to visualize the relationships between important word pairs identified from the analysis. The steps include text preprocessing, embedding generation, similarity computation, and visualization.

**Dataset Description**

* **Attributes**:
  + **Report Name**: The title of the medical report.
  + **History**: A brief medical history of the patient.
  + **Observation**: Detailed observations from medical imaging or tests.
  + **Impression**: The concluding thoughts or findings based on the history and observations.
* **Dataset Size**: The dataset consists of **330 records**.

**Methodology**

**1. Text Preprocessing**

**Objective**: Clean the text data to enhance the quality of analysis.

* **Steps**:
  + **Tokenization**: Split the text into individual words.
  + **Lowercasing**: Convert all text to lowercase to maintain uniformity.
  + **Stop Words Removal**: Eliminate common words (e.g., "the," "is," "and") that do not contribute significant meaning.
  + **Stemming**: Reduce words to their base or root form (e.g., "running" to "run").
  + **Lemmatization**: Further refine words to their dictionary form (e.g., "better" to "good").
* **Tools Used**:
  + nltk: Natural Language Toolkit for text preprocessing.

**Assumptions**:

* The use of both stemming and lemmatization is intended to provide a balance between reducing word forms and preserving their meanings.

**2. Embedding Generation**

**Objective**: Convert the cleaned text data into numerical vectors that capture semantic meanings.

* **Method**:
  + Used a pre-trained transformer model (BERT) to generate embeddings for the processed text.
  + **Mean Pooling**: Averaged the token embeddings to obtain a single vector representation for each report.
* **Tools Used**:
  + transformers: Hugging Face library for loading pre-trained models.

**Assumptions**:

* The BERT model is sufficient for capturing the contextual meaning of the medical text given its success in various NLP tasks.

**3. Similarity Computation**

**Objective**: Identify the relationships between words based on their embeddings.

* **Method**:
  + Calculated cosine similarity between unique word pairs derived from the processed text to quantify their semantic similarity.
* **Tools Used**:
  + sklearn.metrics.pairwise: For computing cosine similarity.

**Assumptions**:

* Words that are closer in the embedding space are assumed to have similar meanings or contexts in the medical domain.

**4. Visualization**

**Objective**: Create visual representations of the top word pairs based on similarity for easier interpretation.

* **Methods**:
  + **Basic Visualization**: Created a bar plot using matplotlib to show the top word pairs and their similarities.
  + **Interactive Visualization**: Developed scatter plots and network graphs using plotly to provide a dynamic view of word relationships.
* **Tools Used**:
  + matplotlib: For static visualizations.
  + plotly: For creating interactive visualizations.

**Assumptions**:

* Visualizations are assumed to aid in better understanding the relationships between terms and provide an engaging way to explore the data.

**Challenges and Solutions**

* **Memory Management**: Encountered issues with GPU memory while fine-tuning models. This was managed by reducing batch sizes and ensuring proper memory allocation.
* **Data Quality**: Assumed that the textual data provided in the dataset was cleaned and relevant for the analysis. If noise or irrelevant data exists, preprocessing would need adjustments.

**Conclusion**

The approach effectively combines natural language processing techniques and modern deep learning methods to analyze medical reports. The visualization steps enhance interpretability and provide valuable insights into the data.