**Data Preparation/Feature Engineering**

1. **Overview**

The data preparation and feature engineering phase is crucial for ensuring that the machine learning model is trained on clean, relevant, and well-structured data. Given that healthcare data can be noisy, incomplete, and complex, this phase plays an essential role in organizing the MIMIC-III dataset appropriately for training the Retrieval-Augmented Generation (RAG) model. This organization enhances both performance and accuracy in clinical decision support tasks. For this project, we are focusing specifically on the textbooks section of the MIMIC-III dataset, which allows us to minimize computational complexity, streamline processing, and improve model performance while still maintaining high-quality clinical knowledge for decision support.

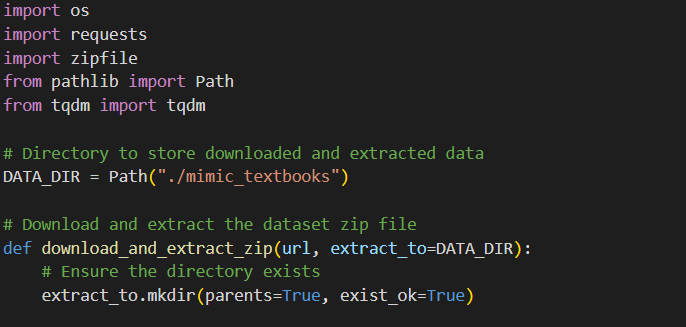
1. **Data Collection**

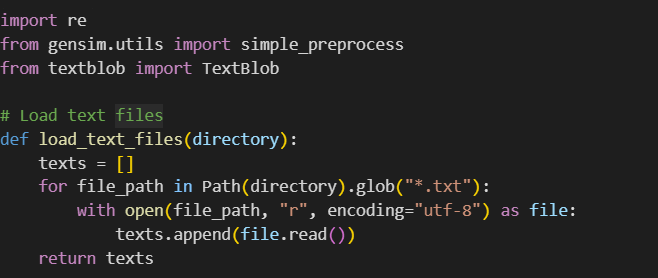
The MIMIC-III dataset is a unique, comprehensive resource combining various data sources to provide a detailed view of patient care in intensive care units (ICUs). This dataset supports research across a range of domains, from medical education to data science and machine learning, particularly in health informatics.

Here’s an overview of the core components of the MIMIC-III dataset and what each contributes:

* Patient histories from ICU: At its core, MIMIC-III contains detailed patient records of ICU visits, spanning multiple years and including various forms of medical informations
* **Medical textbooks:** MIMIC-III integrates medical textbooks, providing a structured knowledge base covering standardized medical information, such as treatment guidelines, disease descriptions, and symptomatology. This content enables users to simulate models that can access both factual, widely accepted medical knowledge and patient-specific information. The textbook component is useful for training systems to understand baseline medical standards and recommendations, which are critical for diagnostics and treatment decision support.
* **Wikipedia:** Some sections of MIMIC-III include summaries from Wikipedia articles relevant to medical terms, common diseases, and biological mechanisms. Wikipedia content is often up-to-date, with community-vetted articles that cover general knowledge on diseases, procedures, medications, and conditions. It provides context that aids in the interpretation of clinical data, enabling machine learning models to answer questions that might blend factual information with patient-specific data.
* **PubMed research articles:**PubMed articles incorporated into MIMIC-III serve as a scholarly source, providing peer-reviewed insights into medical research, recent advancements, and case studies. PubMed is rich in clinical trial data, meta-analyses, and reviews, offering insights into the latest findings in fields such as pharmacology, diagnostics, and therapeutic interventions. In MIMIC-III, these articles add a layer of depth, allowing models to reference cutting-edge research and apply recent findings in interpreting patient data.

We used the textbook data extracted from the MIMIC-III dataset specifically focusing on English medical textbooks to keep the scope manageable and improve performance. Text files are loaded using a simple directory-based approach that reads all .txt files. This targeted selection helps reduce computational complexity while still providing rich medical knowledge for training.

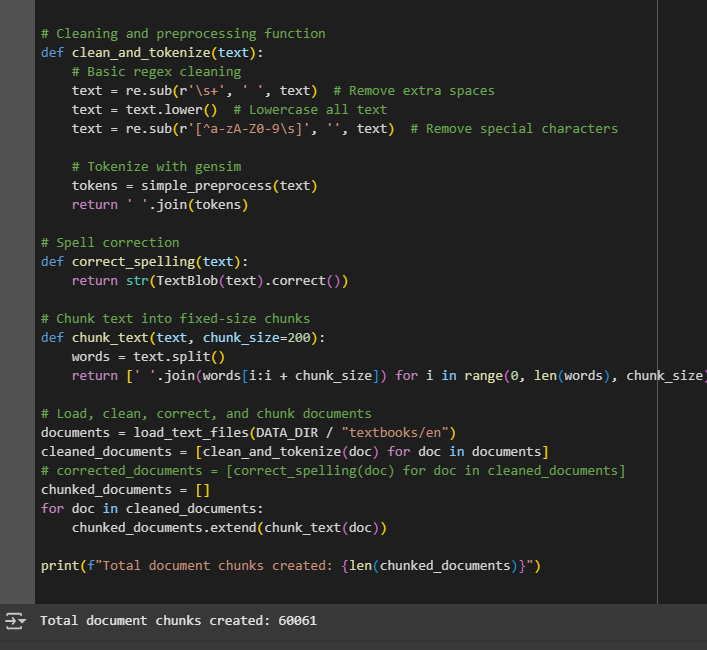




1. **Data Cleaning**

Data cleaning involves several steps:

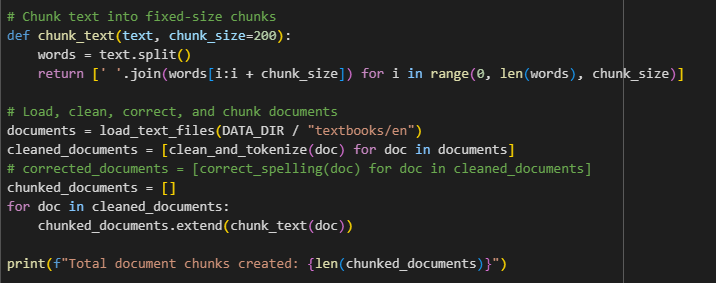
* **Whitespace removal**: Extra spaces are reduced to single spaces.
* **Lowercasing**: All text is converted to lowercase to ensure uniformity.
* **Special character removal**: Non-alphanumeric characters are stripped out to avoid unnecessary noise.
* **Tokenization**: Using Genism’s simple\_preprocess, text is split into clean, normalized tokens.
* **Spell correction** using **TextBlob** analyzes the text and returns a new string with corrected spelling.



1. **Exploratory Data Analysis (EDA)**

EDA involved basic checks such as:

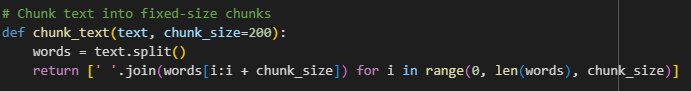
* Counting the number of text documents loaded.
* Verifying the effectiveness of the cleaning (e.g., absence of special characters).
* Analyzing the average length of documents before and after chunking to ensure manageable input sizes for the model.
* Printing the total number of chunks created, giving insight into dataset size post-processing.

****

1. **Feature Engineering**

Feature engineering in this context focused on chunking the cleaned text into fixed-size word groups:

* Each chunk contains approximately 200 words, making it easier for the retriever model to manage the context window efficiently.
* This technique ensures that each data sample remains within the optimal token limits for RAG models and LLMs, avoiding truncation issues.

****

Chunking maintains semantic coherence while fitting within model input limits.

1. **Data Transformation**

After cleaning and chunking:

* Text normalization: Already performed during cleaning (lowercasing, tokenization).
* Encoding: At this stage, no specific token encoding is applied manually because downstream models (like the retriever/FAISS) handle vectorization automatically.
* Spell correction was considered but not applied during transformation for efficiency reasons.

**Model Exploration**

1. **Model Selection**

We chose a **Retrieval-Augmented Generation (RAG)** model because:

* It effectively retrieves accurate textbook knowledge for clinical decision support.
* It ensures fact-based, explainable AI outputs by grounding responses in textbook facts.

**Strengths:**

* Better factual grounding.
* Faster training and inference by using a smaller, clean corpus (textbooks).

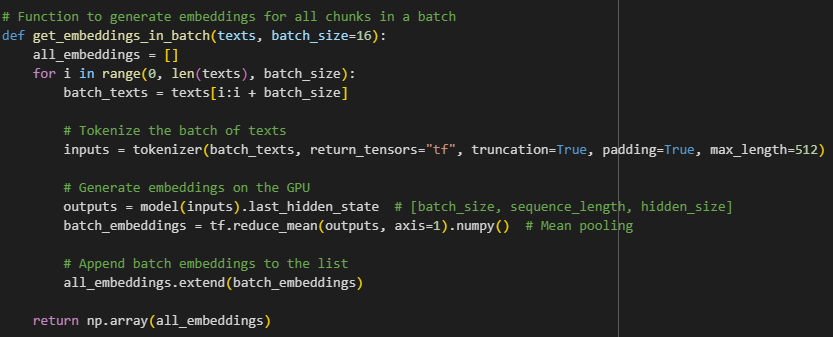
**Weaknesses:**

* Limited coverage compared to using full MIMIC-III (patient records).

1. **Model Training**

In this stage, the model is **not fine-tuned** — we are **using it as a feature extractor** to generate embeddings:

* Text chunks (around 200 words each) are tokenized and fed into the MiniLM model.
* The embeddings are obtained using **mean pooling** over the last hidden layer outputs.
* Batching was used (batch\_size=128) to efficiently process large numbers of chunks on the GPU.
* **Max length:** We truncated texts at 512 tokens during tokenization to fit the model’s input requirements.



1. **Model Evaluation**

Since the model was used only for embedding generation, traditional training evaluation (accuracy, loss) is not applicable here.

However, performance evaluation is still considered indirectly:

* Embedding Quality: We can later assess retrieval quality by how well the FAISS search (which will use these embeddings) retrieves semantically relevant textbook chunks.
* Efficiency Check: Confirmed that embeddings were generated successfully for all document chunks