Data Cleaning

Preprocessing Documentation

This section outlines the data cleaning, normalization, and transformation steps performed during preprocessing. It also highlights how data integrity was maintained and ensured no information leaks between training and testing data.

Preprocessing Steps

1. Data Cleaning

The following cleaning steps were applied to the dataset:

- **Handling Missing Values**:
  - Missing values in text columns (e.g., `Study Title`, `Primary Outcome Measures`, `Secondary Outcome Measures`, `Criteria`) were replaced with empty strings (`''`).
  - Missing numerical values (e.g., `Enrollment`, `duration_years`) were replaced with `0`.
- **Text Cleaning**:
  - Converted all text to lowercase.
  - Removed punctuation and special characters using regex.
  - Removed stopwords using NLTK's English stopwords list.
  - Normalized whitespace (e.g., multiple spaces replaced with a single space).
- **Eligibility Criteria**:
  - Merged the `eligibilities.txt` file with the main dataset using the `NCT Number` and `nct_id` columns.
  - Extracted and cleaned the `criteria` column for use in similarity computation.

2. Feature Engineering

The following features were created or transformed:

- **Combined Text Features**:
  - Text columns (`Study Title`, `Primary Outcome Measures`, `Secondary Outcome Measures`, `Criteria`, `Conditions`, `Interventions`, `Brief Summary`) were combined into a single weighted text representation.
  - Weights were assigned to each column based on their importance (e.g., `Study Title` had a weight of 3.0, while `Brief Summary` had a weight of 1.8).

- **Duration Calculation**:
    - Calculated the duration of each trial in years using the `Start Date` and `Completion Date` columns.
- **Sentiment Analysis**:
    - Used `TextBlob` to compute sentiment scores for the `Brief Summary` column.

---

3. Normalization

- **Text Normalization**:
    - All text data was normalized by converting to lowercase, removing stopwords, and cleaning special characters.
    - Tokenization was performed using SpaCy's `en_core_web_sm` model.
- **Numerical Normalization**:
    - Numerical columns like `Enrollment` and `duration_years` were scaled to a range of 0 to 1 for consistency.

---

4. Data Integrity

- **No Information Leakage**:
    - The dataset was processed in a way that ensures no information leaks between training and testing data. Since this is a retrieval task (not a traditional train-test split), all data was treated as a single corpus for similarity computation.
    - No future information was used during preprocessing (e.g., dates were handled appropriately).
- **Consistency Checks**:
    - Verified that all `NCT Number` values were unique.
    - Ensured that merged columns (e.g., `criteria` from `eligibilities.txt`) aligned correctly with the main dataset.

Scripts for Preprocessing

Below are the key preprocessing steps implemented in the code:

1. Data Cleaning

```
# Fill missing values
df = df.fillna('')

# Clean text
def clean_text(text):
    text = str(text).lower()
    text = re.sub(r'[^\w\s]', ' ', text)  # Remove punctuation
    text = re.sub(r'\s+', ' ', text)      # Normalize whitespace
    words = text.split()
    text = ' '.join(word for word in words if word not in self.stop_words)
# Remove stopwords
    return text
```

2. Feature Engineering

```
# Combine text features with weights
def combine_features(row):
    text_parts = []
    for col, weight in self.text_columns.items():
        if col in row.index and row[col]:
            cleaned_text = self.clean_text(row[col])
            text_parts.extend([cleaned_text] * int(weight * 5))  # Apply
weights
    return ' '.join(text_parts)
```

Calculate duration

```
df['year'] = pd.to_datetime(df[start_date_col], errors='coerce').dt.year
df['duration_years'] = pd.to_datetime(df[completion_date_col],
errors='coerce').dt.year - df['year']
```

3. Sentiment Analysis

```
# Compute sentiment score
df['sentiment_score'] = df['Brief Summary'].apply(lambda x:
TextBlob(str(x)).sentiment.polarity)
```

Verification of Data Integrity

- **Unique Identifiers**:
  - Verified that all `NCT Number` values were unique and consistent across the dataset.

```
    assert df['NCT Number'].nunique() == len(df), "Duplicate NCT Numbers
found!"
```

- **No Future Data Leakage**:
  - Ensured that date columns (`Start Date`, `Completion Date`) were used only for calculating trial duration and not for any future-dependent operations.
- **Consistency in Merged Data**:

- After merging `eligibilities.txt` with the main dataset, verified that all `nct_id` values matched the `NCT Number` column.

```
    assert df['nct_id'].isnull().sum() == 0, "Mismatch in NCT IDs after merge!"
```

Conclusion

The preprocessing steps ensured that the dataset was clean, normalized, and ready for embedding generation and similarity computation. Data integrity was maintained throughout the process, and no information leaks were introduced. These steps are critical for ensuring the accuracy and reliability of the semantic grouping and retrieval system.