This work, I have done during the first week of a STEM research, focuses on preparing and anonymizing medical and mental health datasets for use in natural language processing (NLP) applications. The primary objective was to process datasets (MentalChat16K and MedQuAD) from Hugging Face, anonymize sensitive information (e.g., PHI - Protected Health Information), and preserve medical terms, organizations, and locations to maintain research value. The pipeline integrates advanced NLP tools (SciSpaCy, medSpaCy, SentenceTransformers) with regex-based anonymization to ensure strong de-identification while addressing mental health community-specific patterns.

The script took approximately 9 hours to develop, including research on which model to use. An additional 11 hours were dedicated to building the anonymization logic. Initial attempts with two to three other models (e.g., general-purpose NER models like spaCy’s en\_core\_web\_sm and BioBERT-based models) were unsuccessful due to poor performance on biomedical or PHI-specific tasks but I eventually found en\_ner\_bc5cdr\_md from SciSpaCy and medSpaCy, which worked well. I also used SentenceTransformer for context classification. It took around 2.5 hours to anonymize the data using these models, along with regex validation and context-aware processing to handle sensitive data responsibly.

The resulting pipeline produces anonymized datasets in JSONL and CSV formats, split into training, validation, and test sets, along with detailed statistics.

**GitHub Repository**: <https://github.com/I-VAGAT/STEMRESEARCH>

**Objectives**

* **Data Loading**: Load MentalChat16K and MedQuAD datasets from Hugging Face.
* **Data Cleaning**: Remove duplicates, missing values, and invalid records.
* **Anonymization**: Use SciSpaCy for biomedical Named Entity Recognition (NER), medSpaCy for PHI de-identification, and regex patterns for additional sensitive data detection.
* **Context Awareness**: Implement context classification using SentenceTransformers to distinguish between medical and personal contexts, preserving relevant medical terms.
* **Validation**: Validate question-answer pairs for quality and length constraints.
* **Data Splitting**: Split data into training (80%), validation (10%), and test (10%) sets.
* **Output Generation**: Save anonymized data in JSONL format for fine-tuning and CSV for analysis, along with detailed statistics.

**Methodology**

The pipeline combines multiple NLP and machine learning techniques to achieve robust anonymization and data preparation:

1. **Model Loading**:
   * **SciSpaCy (en\_ner\_bc5cdr\_md)**: Used for biomedical NER to identify diseases and chemicals.
   * **medSpaCy**: Employed for clinical PHI de-identification (e.g., names, locations).
   * **SentenceTransformer (all-MiniLM-L6-v2)**: Used for context classification by generating embeddings for text windows.
2. **Regex Patterns**:
   * Patterns for detecting emails, phone numbers, SSNs, patient IDs, usernames, IP addresses, credit cards, dates, and medical record numbers.
   * Added validation to ensure patterns are syntactically correct before use.
3. **Context Classification**:
   * Pre-computed embeddings for medical and personal contexts.
   * Cosine similarity used to classify text windows as medical, personal, or neutral, guiding anonymization decisions.
4. **Anonymization Logic**:
   * Preserves medical terms (e.g., "depression," "melatonin"), organizations (e.g., "NIH"), and locations (e.g., "Boston") in medical contexts.
   * Anonymizes personal names, organizations, and locations in non-medical contexts using placeholders like [PERSON] or [ANONYMIZED].
5. **Data Processing**:
   * Combines MentalChat16K and MedQuAD datasets.
   * Applies cleaning, anonymization, and validation.
   * Splits data into train/validation/test sets with stratification by source.
   * Formats data as JSONL for fine-tuning and CSV for analysis.
6. **Statistics and Logging**:
   * Computes dataset statistics (e.g., record counts, question/answer lengths).
   * Logs processing steps and errors to a file, suppressing repetitive regex errors.

**Code Structure and Explanation**

The script is modular, with functions for specific tasks to enhance maintainability and reusability. Below is a breakdown of key components:

**1. Imports and Setup**

* Libraries: pandas, spacy, medspacy, re, datasets, sklearn, sentence\_transformers, torch, numpy, tqdm, json, logging, os, datetime.
* Logging configured to write to data\_preparation.log with timestamps.

**2. Model Initialization**

* Loads SciSpaCy (en\_ner\_bc5cdr\_md) for biomedical NER.
* Loads medSpaCy for PHI de-identification.
* Loads SentenceTransformer (all-MiniLM-L6-v2) with CUDA support if available.
* Defines lists of medical terms, organizations, and locations to preserve.

**3. Regex Validation and Testing**

* validate\_regex\_patterns(): Ensures regex patterns are syntactically correct.
* test\_regex\_patterns(): Tests patterns on sample text to verify functionality.
* safe\_regex\_search(): Applies regex with error handling and limited logging.

**4. Context Classification**

* get\_context\_embeddings(): Pre-computes embeddings for medical and personal contexts.
* classify\_context(): Classifies text windows using cosine similarity with a threshold of 0.4.

**5. Anonymization Logic**

* is\_medical\_term(), is\_medical\_organization(), is\_medical\_location(): Check if entities should be preserved.
* is\_real\_person\_name(): Determines if a name is a real person’s name based on context.
* combined\_ner\_anonymize(): Combines SciSpaCy, medSpaCy, and regex for anonymization, preserving medical terms in medical contexts.

**6. Data Processing**

* load\_datasets(): Loads MentalChat16K and MedQuAD from Hugging Face.
* preprocess\_data(): Combines datasets, cleans data, applies anonymization, and validates records.
* validate\_pair(): Ensures question-answer pairs meet length and quality criteria.
* split\_data(): Splits data into train/validation/test sets with stratification.
* format\_for\_finetuning(): Saves data in JSONL format with metadata.

**7. Statistics**

* compute\_statistics(): Generates dataset statistics (e.g., record counts, lengths) and saves them as JSON.

**8. Main Function**

* Orchestrates the pipeline, calling all functions and saving outputs to combined\_medical\_data/.

**Challenges Addressed**

1. **Context Sensitivity**: Used SentenceTransformers to classify contexts, ensuring medical terms are preserved while personal information is anonymized.
2. **Mental Health Specificity**: Included terms like "reddit," "subreddit," and "throwaway" to handle mental health community data.
3. **Performance**: Used tqdm for progress tracking and optimized model loading with CUDA support.
4. **Error Handling**: Strong logging and exception handling to prevent pipeline crashes and provide detailed diagnostics.

**Output**

The pipeline produces the following files in the combined\_medical\_data/ directory:

* train.jsonl, validation.jsonl, test.jsonl: Anonymized data for fine-tuning.
* train.csv, validation.csv, test.csv: Raw data for analysis.
* dataset\_statistics.json: Detailed statistics (e.g., record counts, question/answer lengths).
* data\_preparation.log: Processing logs with error details.

**References**

1. **SciSpaCy**:
   * Neumann, M., et al. (2019). "ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing." *arXiv preprint arXiv:1902.07669*. <https://arxiv.org/abs/1902.07669>
   * [SciSpaCy Documentation](https://allenai.github.io/scispacy/) <https://allenai.github.io/scispacy/>
2. **medSpaCy**:
   * [medSpaCy Documentation](https://github.com/medspacy/medspacy) <https://github.com/medspacy/medspacy>
3. **SentenceTransformers**:
   * Reimers, N., & Gurevych, I. (2019). "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks." *arXiv preprint arXiv:1908.10084*. <https://arxiv.org/abs/1908.10084>
   * [SentenceTransformers Documentation](https://www.sbert.net/) <https://www.sbert.net/>
4. **Hugging Face Datasets**:
   * MentalChat16K: [ShenLab/MentalChat16K](https://huggingface.co/datasets/ShenLab/MentalChat16K) <https://huggingface.co/datasets/ShenLab/MentalChat16K>
   * MedQuAD: [lavita/MedQuAD](https://huggingface.co/datasets/lavita/MedQuAD) <https://huggingface.co/datasets/lavita/MedQuAD>
   * [Hugging Face Datasets Documentation](https://huggingface.co/docs/datasets/)
   * <https://huggingface.co/docs/datasets/index>
5. **Python Libraries**:
   * pandas: [pandas Documentation](https://pandas.pydata.org/) <https://pandas.pydata.org/>
   * spaCy: [spaCy Documentation](https://spacy.io/) <https://spacy.io/>
   * scikit-learn: [scikit-learn Documentation](https://scikit-learn.org/) <https://scikit-learn.org/>
   * PyTorch: [PyTorch Documentation](https://pytorch.org/) <https://pytorch.org/>
   * tqdm: [tqdm Documentation](https://tqdm.github.io/) <https://tqdm.github.io/>
6. **Regular Expressions**:
   * Python re module: [Python re Documentation](https://docs.python.org/3/library/re.html) <https://docs.python.org/3/library/re.html>

**Conclusion**

This pipeline demonstrates a strong approach to medical data preparation and anonymization, balancing privacy protection with the preservation of research-relevant information. The integration of SciSpaCy, medSpaCy, and SentenceTransformers, along with custom regex patterns, ensures accurate and context-aware processing. The script is well-documented, modular, and ready for further enhancements in subsequent research phases.