PharmaFeel

Sentiment Analysis on Patient Drug Reviews

Yuval Elisha
Nikol Jabotinski

Project Description

Motivation: Understanding patient experiences with medications through unstructured text analysis.

Relevant NLP Task: Sentiment Analysis using LLMs.

Preprocessing: cleaning, tokenization, stopword removal.

Embedding: Sentence-BERT or OpenAI embeddings.

Classification: LLM zero-shot classifier and logistic regression baseline.

Evaluation Metrics:

- Precision, Accuracy for sentiment classification.
- Comparing LLM results to Lexicon based Sentiment Analysis.

Previous Work

Sentiment Analysis on Drug Reviews is a common task..

<u>Drugs 💉 Review - Sentiment 📊</u>

- Data: The Drug Review Dataset - UCI Machine Learning Repository.
- Classifying to positive and negative reviews.
- Vectorization: TF-IDF.
- Machine Learning models: LightGBM.
- Evaluation Metrics: Accuracy, Precision, Recall and F1 score.
- Accuracy: 90%.

<u>Deep learning based sentiment analysis of drug reviews</u>

- Data: The Drug Review Dataset - UCI Machine Learning Repository.
- Classifying to positive, neutral and negative reviews.
- Using Tokenization.
- LTSM (Long Short-Term Memory) model with embeddings.
- Accuracy: 58%.

Previous Work

<u>BERT-based language model for accurate drug adverse event extraction from social media: implementation, evaluation, and contributions to pharmacovigilance practices</u>

- Data: ADE-Corpus-V2, SMM4H dataset.
- Extracting side effects from social media posts.
- Removal of characters in text and tokenization for embeddings.
- Using RRF + TF-IDF and ARF.
- Fine tuning a BERT based uncased model with the dataset ADE-Corpus-V2.
- F1 score: 0.9 for internal categories, 0.8 for external validation.

Project Pipeline

1. Preprocessing

<u>Input:</u> Raw patient review (unstructured text)

Steps:

- Text cleaning: remove HTML, punctuation, special characters
- Lowercasing
- Tokenization
- Stopword removal

<u>Output:</u> Cleaned text (tokens)

Project Pipeline

2. Embedding / Feature Extraction

Input: Cleaned text.

Techniques:

- Sentence-BERT – pre-trained model that generates sentence embeddings.
- OpenAI Embeddings – high-dimensional semantic representation.

Output: Embedding vector representing semantic meaning of each review.

Metrics at this stage: Cosine similarity (optional, for analysis).

Project Pipeline

3. Classification (Modeling)

<u>Input:</u> Sentence embeddings

Approaches:

- Zero-shot classification.
- Few-shot classification.
- Logistic Regression or Random Forest on top of embeddings (for supervised baseline).
- LSTM - in order to compare existing projects.

<u>Output:</u> Predicted sentiment class → Positive / Neutral / Negative

Evaluation Metrics:

- Per-class Precision, Recall, F1-score.
- Overall Accuracy

Project Pipeline

4. Error Analysis & Refinement

Compare model output to:

- Lexicon-based baseline.
- Ratings (1–10).

Analyze misclassifications by class.
Adjust prompts or retrain supervised classifier with new features.

Exploration and Baseline

Dataset:

- 280K drug reviews.
- Fields: Drug Name, Condition, User, Date, Rating, Content.
- Sentiment labels derived from rating: Positive (7–10), Neutral (4–6), Negative (1–3).
- Mean review length: 91 ± 50 words.

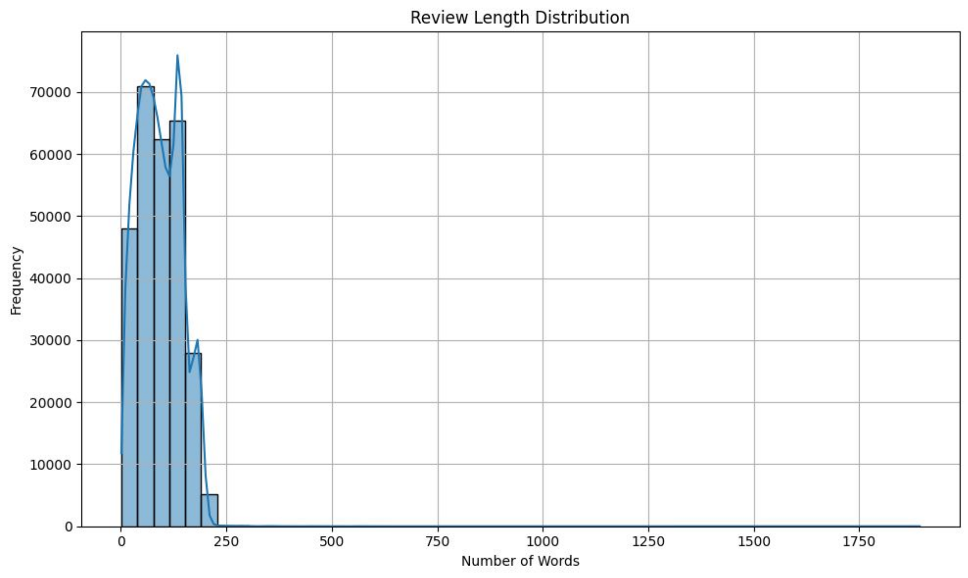
Baseline:

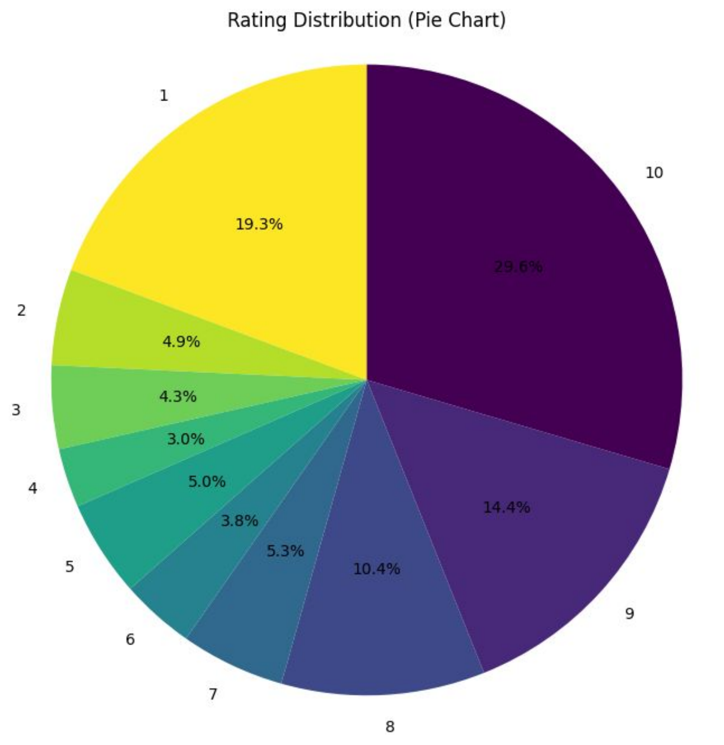
- Preprocessing: Lowercasing, removing punctuation, short words, stopwords.
- Embedding: Sentence-BERT (MiniLM-L6-v2) to generate vectors.
- Classifier: Logistic Regression trained on embeddings.
- Train/test split: 80/20.
- Training only 10,000 reviews.

Exploration and Baseline

Results:

- Accuracy = 0.62
- Best performance on positive/negative classes.
- Neutral class remains challenging.





Conclusions

Findings so far:

- Most reviews are short and focused, making them suitable for sentence-level embeddings (e.g., SBERT).
- Sentiment distribution is imbalanced.
- Neutral reviews are hard to classify.

Next steps:

- Using zero-shot classification to handle ambiguous language and context better.
- Evaluate few-shot or prompt-based approaches.
- Explore class balancing strategies (e.g., class weights, oversampling).
- Add explainability techniques (e.g., SHAP, LIME) to interpret what the model learns.

“The medicine was helpful but gave me headaches.” → Negative

Thank you!

Any questions?