Final-Project-Group4

Sentimental Analysis of Amazon Electronics Reviews

Team Members:

Chaya Chandana Doddaiggaluru Appajigowda

Ramana Bhaskar Kosuru

Adam Stuhltrager

Overview of the Project:

The project focuses on advanced Natural Language Processing (NLP) techniques to develop a sentiment analysis system for Amazon Electronics product reviews. We selected this problem because understanding customer sentiment is critical for both manufacturers and retailers in the rapidly evolving electronics market. By automatically classifying reviews into negative, neutral, and positive sentiments, businesses can gain actionable insights about product reception without manual analysis of thousands of reviews.

Electronics reviews are particularly interesting for NLP analysis because they often contain technical jargon, complex opinions about multiple product features, and comparative references to other products. This complexity makes them an excellent candidate for applying advanced NLP techniques beyond simple binary classification.

Dataset Source:

Amazon Electronics Reviews - https://mcauleylab.ucsd.edu/public_datasets/data/amazon_2023/raw/review_categories/Electronics.jsonl.gz. This comprehensive dataset contains millions of product reviews with various fields including:

- Star ratings (1-5)
- Review text
- Review titles

- Helpful votes
- Verified purchase indicators

For sentiment classification:

- Negative: 1-2 stars
- Neutral: 3 stars
- Positive: 4-5 stars

This approach provides a more nuanced analysis than binary classification while maintaining clear class boundaries.

Models Used:

1. **Baseline Model**: A classical approach using TF-IDF vectorization with Multinomial Naive Bayes classification.
2. **CNN Text Model**: A convolutional neural network with multiple filter sizes for capturing different n-gram patterns.
3. **RNN Text Model**: A recurrent neural network leveraging bidirectional architecture to capture context in both directions.
4. **LSTM Model**: Long Short-Term Memory network to better handle long-range dependencies in review text.

The deep learning models will require customization of architectures to handle the specific characteristics of product reviews, particularly in terms of sequence length and embedding dimensions.

Framework and Libraries:

- **PyTorch**: For implementing all deep learning models due to its flexibility and intuitive design.
- **Scikit-learn**: For the baseline model, train-test splitting, and evaluation metrics.
- **NLTK**: For text preprocessing including tokenization and stopword removal.
- **Pandas/NumPy**: For efficient data manipulation and numerical operations.
- **Matplotlib/Seaborn**: For visualization of results and model performance.
- **tqdm**: For progress tracking during data processing and model training.

Code Structure:

1. **Text Preprocessing**: Cleaning the review text by converting to lowercase, removing punctuation and numbers, and eliminating stopwords.
2. **Tokenization**: Breaking the text into words or tokens for model processing.
3. **Vocabulary Building**: Creating a mapping from tokens to indices with handling for out-of-vocabulary words.
4. **Feature Extraction**:
    - For baseline: TF-IDF vectorization with n-grams.
    - For deep learning models: Learning word embeddings from scratch.
5. **Sentiment Classification**: Training models to predict the three sentiment classes.

Performance Evaluation:

1. **Accuracy**: Overall correctness of predictions.
2. **Precision, Recall, and F1-Score**: Using weighted averages to account for potential class imbalance.
3. **Confusion Matrix**: To visualize model performance across different sentiment classes.
4. **Loss Curves**: To track training and validation loss over epochs to monitor for overfitting.

The performance of all four models to determine which approaches work best for this specific task, considering both accuracy and computational efficiency.