Sentiment Analysis of Product Reviews: Comparing Naïve Keyword Matching with DistilBERT-Based Classification🤗

Gabriel Hao Wei Yap (2024148069)

1 Introduction

- **Task description:** I am building an automated sentiment analysis system that can automatically read product reviews and classify them as either positive or negative. In other words, the AI will determine if the commenter liked or disliked the product, similar to how a person would quickly scan a review to gauge the overall sentiment.
- **Motivation:** This project addresses a practical problem relevant to both businesses and consumers. Companies can use automated sentiment analysis to quickly process thousands of reviews, identifying common complaints and popular features without manual effort. For shoppers, it helps highlight helpful feedback, making product decisions easier and more informed. Additionally, sentiment analysis serves as an accessible introduction to AI, focusing on a tangible task which is understanding human language that bridges everyday communication with machine learning.
- **Input / Output:**
  - Input: The system accepts raw text from product reviews in natural language.
    Example Input: "The camera quality is outstanding, but the battery life is disappointingly short."
  - Output: The system produces a simple classification label.
    Example Output: NEGATIVE (or POSITIVE)
- **Success criteria:** The system will be considered successful if it achieves the following key objectives:
  - **Primary Performance Goal:** The AI pipeline must significantly outperform the naïve baseline model, demonstrating clear advantages in understanding complex language.
  - **Quantitative Targets:**
    - Achieve accuracy substantially above random guessing (50%)
    - Maintain balanced performance across both positive and negative sentiment classes
    - Show at least 5–10% accuracy improvement over the baseline approach
  - **Qualitative Expectations:**
    - Correctly handle challenging linguistic constructions including negations, mixed sentiments, and nuanced expressions
    - Demonstrate contextual understanding beyond simple keyword matching
    - Make sensible predictions that align with human interpretation of review sentiment

2 Datasets

- **Source:** The dataset used is `m-ric/amazon_product_reviews_datafiniti` from Hugging Face Datasets. It contains Amazon product reviews along with metadata such as review text, ratings, brand, and categories.
- **Total examples:** The original dataset contained 6,000 examples. After preprocessing and balancing, 300 examples were used for this project, with 150 positive and 150 negative reviews.
- **Train/Test split:** The processed dataset was divided using a 70/30 split:
  - Training set: 210 examples (70%)
  - Test set: 90 examples (30%)

  Stratified sampling was applied to maintain the same class distribution in both splits, ensuring approximately equal numbers of positive and negative reviews.
- **Preprocessing steps:**
  1. **Data Cleaning & Filtering:** Removed reviews with missing text or ratings, filtered out neutral 3-star reviews, removed duplicates, and excluded very short reviews (less than 10 characters).
  2. **Sentiment Labeling:** Positive sentiment (label=1) was assigned to 4- and 5-star reviews, negative sentiment (label=0) to 1- and 2-star reviews.
  3. **Text Preprocessing:** Converted all text to lowercase, stripped leading/trailing whitespace, and balanced the dataset to ensure equal numbers of positive and negative examples.
  4. **Feature Selection:** Only the review text and rating columns were used; other metadata was discarded for the core classification task.
  5. **AI Pipeline Specific Preprocessing:** For the DistilBERT model, tokenization was applied using the DistilBERT tokenizer, text was truncated to 128 tokens, padded to a consistent length, and [CLS] token embeddings were extracted as sentence representations.
- **Class Distribution:** The final processed dataset maintained perfect class balance with 150 positive and 150 negative reviews, ensuring that evaluation metrics such as accuracy are meaningful and not skewed by class imbalance.

3 Methods

This section includes both the naïve baseline and the improved AI pipeline.

3.1 Naïve Baseline

- **Method description:** The naïve baseline operates on a simple keyword-matching principle. It uses two predefined lists of sentiment-bearing words:
  - Positive words: {“great”, “good”, “excellent”, “love”, “awesome”, “perfect”, “best”, “recommend”}
  - Negative words: {“bad”, “terrible”, “awful”, “hate”, “worst”, “waste”, “broken”, “disappointed”}

  The classification process is as follows:
  1. Convert each review text to lowercase.

2. Count occurrences of positive and negative keywords in the text.
3. Compare the counts: if positive count is larger than negative count, predict positive sentiment; otherwise predict negative sentiment.
4. Return the binary classification (0 for negative, 1 for positive).

  This approach performs classification without considering context, word order, or semantic meaning.

- **Why naïve:** This baseline is considered naïve because it lacks context understanding and cannot interpret words in context. For example, it would treat "not good" as positive simply by detecting the word "good", missing the negation. It also has a limited vocabulary of only 16 manually curated words and cannot recognize synonyms, related terms, or domain-specific language. Additionally, it operates at the surface level with no semantic comprehension, meaning it cannot grasp meaning, sarcasm, irony, or nuanced expressions. Unlike machine learning models, it cannot learn or improve with more data, nor adapt to different domains without manual intervention.

- **Likely failure modes:**
  - **Negation and Contextual Language:**
    - Reviews containing negations can lead to incorrect predictions, e.g., "This product is not good" may be classified as POSITIVE due to the presence of the word "good".
    - Subtle expressions, such as "I don't hate this product actually," may be misclassified as NEGATIVE because the model detects the word "hate" without understanding context.
  - **Mixed and Complex Sentiments:**
    - Reviews expressing contrasting opinions, such as "Great camera but terrible battery," may result in misclassification depending on which sentiment words appear more frequently.
    - Phrases like "It's good for the price" may be incorrectly predicted as POSITIVE, as the baseline fails to capture qualifiers or limitations.
  - **Sarcasm and Irony:**
    - Sarcastic statements, e.g., "Just what I needed - another broken device," are likely to be misclassified as NEGATIVE when literal interpretation is applied.
    - Similarly, "Great, another product that doesn't work" may be incorrectly labeled as POSITIVE due to the presence of the word "great".
  - **Vocabulary Limitations:**
    - Reviews that use domain-specific terms not included in the keyword lists, such as "The lens is blurry" or "The fabric is scratchy," cannot be classified reliably.
  - **Comparative Statements:**
    - Phrases that indicate relative judgment, such as "Better than expected" or "Worse than the old version," often remain ambiguous and are not handled correctly by the baseline.

3.2 AI Pipeline

The AI pipeline uses Sanh et al. [1], implemented via the Hugging Face Transformers library [2].

- **Models used:**
    - **DistilBERT** (`distilbert-base-uncased`) for converting text into numerical embeddings.
    - **Logistic Regression** classifier for making the final sentiment decision.
- **Pipeline stages:**
    1. **Text Cleaning:** Filter the raw dataset by removing duplicate reviews, discarding very short texts, and creating a balanced set with equal positive and negative examples.
    2. **Tokenization:** Use the DistilBERT tokenizer to split each review into sub-word tokens, then pad or truncate all reviews to a fixed length of 128 tokens.
    3. **Embedding Generation:** Pass tokenized text through the pre-trained DistilBERT model and extract the [CLS] token's hidden state, producing a 768-dimensional vector representing the semantic content of the review.
    4. **Classification:** Train a Logistic Regression classifier on the embeddings from the training set to separate positive from negative reviews.
    5. **Evaluation:** Predict labels on the held-out test set and compute accuracy, precision, recall, and F1-score to assess performance.
- **Design choices and justification:**
    - DistilBERT was chosen as the core language model because it efficiently captures language context and semantic meaning, outperforming simple keyword-based approaches while remaining computationally lightweight.
    - The [CLS] token is used to represent the overall meaning of each review, as it is optimized for classification tasks.
    - Logistic Regression was selected as the classifier for its simplicity and speed, making it suitable for use on top of pre-trained embeddings without requiring extensive training.
    - This approach balances strong language understanding from a pre-trained model with the simplicity and efficiency of a classic classifier, enabling effective sentiment analysis without heavy computational costs.

4 Experiments and Results

4.1 Metrics

I focused on standard classification metrics to thoroughly evaluate both models:

- **Accuracy:** The overall percentage of reviews classified correctly. This metric is appropriate because our dataset is balanced with equal positive and negative reviews, making overall correctness a meaningful measure of general performance for sentiment classification.
- **Precision:** For each class, the proportion of predicted positives or negatives that are actually correct. This ensures that when our model labels a review as positive, it is likely to truly be positive.
- **Recall:** For each class, the proportion of actual positives or negatives that were correctly identified. This ensures that our model catches most negative reviews.
- **F1-Score:** The harmonic mean of precision and recall, providing a single balanced score that accounts for both false positives and false negatives. This metric is particularly valuable for sentiment analysis because it prevents the model from favoring one sentiment class over the other, ensuring fair performance across both positive and negative reviews.

4.2 Results

Metric	Negative	Positive	Overall
Accuracy	-	-	75.6%
Precision	0.69	0.87	-
Recall	0.91	0.60	-
F1-Score	0.79	0.71	-

Table 1: Performance of the Naïve Baseline model.

Metric	Negative	Positive	Overall
Accuracy	-	-	85.6%
Precision	0.85	0.86	-
Recall	0.87	0.84	-
F1-Score	0.86	0.85	-

Table 2: Performance of the AI Pipeline model.

Method	Accuracy	Precision	Recall	F1-Score
Naïve Baseline	75.6%	78.0%	75.5%	75.0%
AI Pipeline	85.6%	85.5%	85.5%	85.5%

Table 3: Comparison of overall performance metrics for Naïve Baseline and AI Pipeline.

4.3 Example Cases (Error Analysis)

- **Example 1: Complex Negation**
  Text: "it's not bad for the novelty of alexa. might as well get the echo. sound it not bad and i'm no audiophile. might be good when my echo dots come in. is it worth it? there are better bluetooth speakers out there."
  True Rating: 4 stars → **True Label:** POSITIVE
  Baseline Prediction: NEGATIVE (Wrong prediction)
  AI Prediction: POSITIVE (Correct prediction)
  Discussion: The baseline sees "bad" and calls it negative, but misses that "not bad" actually indicates a positive sentiment. The AI correctly interprets "not bad" as positive.

- **Example 2: Praising the Idea, Criticizing the Result**
  Text: "the tap is a great concept, i love my echo so a portable one was that much better. however the sound is pretty terrible, no bass and can be scratchy.. it was a good concept just poor execution."
  True Rating: 2 stars → **True Label:** NEGATIVE
  Baseline Prediction: POSITIVE (Wrong prediction)
  AI Prediction: NEGATIVE (Correct prediction)
  Discussion: The baseline relies on keywords like "great" and "love", missing the impact of "however" and "poor execution". The AI captures the overall negative sentiment correctly.

- **Example 3: Cautiously Satisfied**
  Text: "i was sick and tired of my duracells leaking so i bought these. only one month in, so it's hard to know if they leak, but so far so good. no complaints."
  True Rating: 5 stars → **True Label:** POSITIVE
  Baseline Prediction: POSITIVE
  AI Prediction: NEGATIVE
  Discussion: The baseline correctly predicts positive based on "good". The AI misclassifies

due to focusing on early complaints and cautious tone. This demonstrates that the AI pipeline can still make errors even when the baseline gets it right.

5 Reflection and Limitations

Both models performed better than expected because the dataset was artificially clean, making classification unusually straightforward. The naïve baseline's 75.6% accuracy shows the task was simplified, while the AI's 85.6% reflects the lack of real-world noise and nuance. Embedding generation was slower than anticipated, making real-time processing impractical. Setting up the comparison between models was trickier than expected because of variable scoping issues as I had to carefully manage naming and ensure evaluation data flowed correctly between the baseline and AI pipeline stages.

The chosen metrics measured correctness but not understanding, failing to reflect struggles with nuance or context. Accuracy doesn't reveal if the AI truly understood language or just succeeded on easy cases. The project was more a basic demonstration that shows something can work in a simple, controlled setting.than a robust system. With more time, I would fine-tune DistilBERT directly on the sentiment task. I would also test on larger, messier datasets to expose real limitations. Finally, I would implement confidence scoring to flag uncertain predictions for human review, improving real world reliability.

References

[1] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[2] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, 2020.