Movie Reviews Sentiment Analysis

Team ID: 24

Level	ID	Name
SC 4	2021170488	محمد مصطفي محمد السيد
SC 4	2021170345	عمار إبراهيم سمير قطورة
SC 4	2021170520	مصطفي خالد محمود محمد
SC 4	2021170493	محمود أحمد عبدالرحيم السيد
SC 4	2021170232	سلمي سعيد محمد طه
SC 4	2021170262	شهد عبدالمقصود علي أحمد

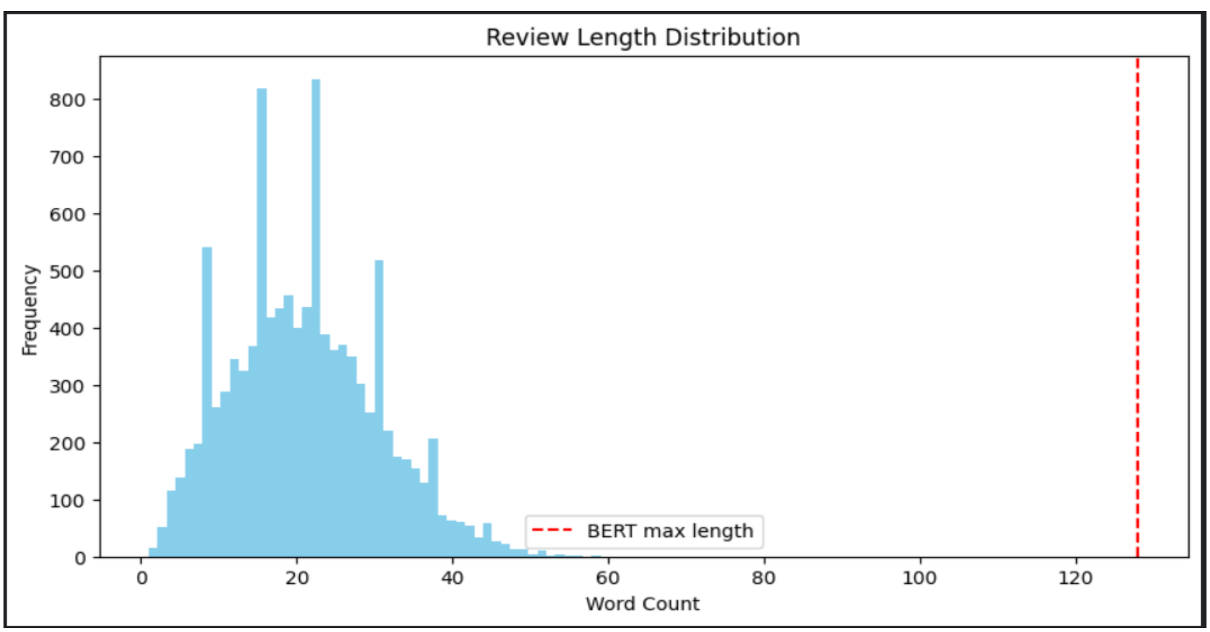
Overview

This project implements sentiment analysis on movie reviews using both traditional Machine Learning and BERT-based approaches. The goal is to classify movie reviews as either positive or negative.

Data

- **Source**: Movie reviews dataset from rt_polarity stored in `all_reviews.csv`
- **Structure**:
  - Size: **10,662** reviews
  - Features:
  - review: The movie review text
  - label: Sentiment label (1 for positive, 0 for negative)
- **Data Overview**:
  - No missing (null) values in the dataset
  - Some duplicate entries present (removed during preprocessing)
  - Balanced classes: 5,331 positive and 5,331 negative reviews
- **Text Characteristics**:
  - Review lengths are well within BERT's 128-token limit
  - Maximum review length: 59 tokens
  - Average review length: 21 tokens

Review Length Distribution



Data Preprocessing

Text Preprocessing Pipeline

1. **Clean Text**
    - Remove URLs
    - Remove special characters and numbers
    - Remove extra whitespace
2. **Character Normalization**
    - Normalize repeated characters (e.g., "goooood" → "good")
3. **N-gram Deduplication**
    - Remove duplicate n-grams for n=1,2,3
    - Process contiguous sequences
4. **Stop Words Removal**
    - Using NLTK stopwords
    - Preserving sentiment words:
        - not, no, never, very, really, too, so, much, many, few, little, hardly, barely, scarcely
5. **Word Normalization**
    - Primary approach: Lemmatization using NLTK's WordNetLemmatizer
    - Alternative tested: Stemming was also evaluated using NLTK's PorterStemmer
    - Comparison results showed no significant performance difference between stemming and lemmatization
    - Chose lemmatization for final implementation as it produces more readable words

Models

Machine Learning Approach

- **Vectorization**: TF-IDF with unigrams and bigrams
- **Models Used**:
    - Logistic Regression
    - Linear SVM
    - Random Forest (n_estimators=100)
- **Data Split**: 80% training, 20% testing

Classification Reports

```
Random Forest Classification Report:
              precision    recall  f1-score   support

    Negative       0.66      0.82      0.73      1062
    Positive       0.77      0.58      0.66      1071

    accuracy                           0.70      2133
   macro avg       0.71      0.70      0.70      2133
weighted avg       0.71      0.70      0.70      2133
```

```
Logistic Regression Classification Report:
              precision    recall  f1-score   support

    Negative       0.72      0.76      0.74      1062
    Positive       0.75      0.71      0.73      1071

    accuracy                           0.73      2133
   macro avg       0.74      0.73      0.73      2133
weighted avg       0.74      0.73      0.73      2133
```

```
Linear SVC Classification Report:
              precision    recall  f1-score   support

    Negative       0.74      0.77      0.76      1062
    Positive       0.76      0.74      0.75      1071

    accuracy                           0.75      2133
   macro avg       0.75      0.75      0.75      2133
weighted avg       0.75      0.75      0.75      2133
```

BERT Model

- **Base Model**: textattack/bert-base-uncased-SST-2
- **Device**: CUDA/CPU auto-detection

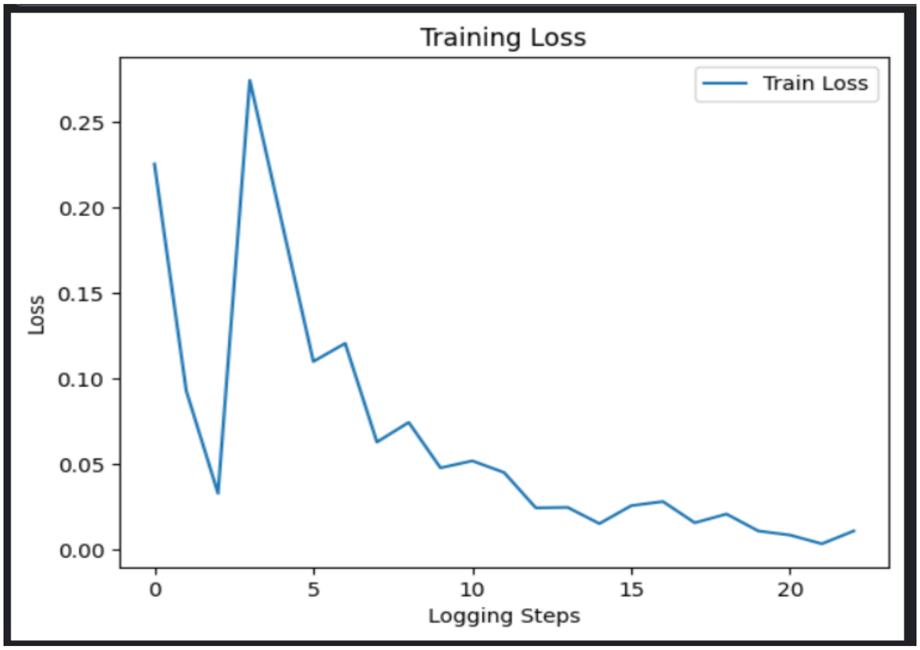
Parameters

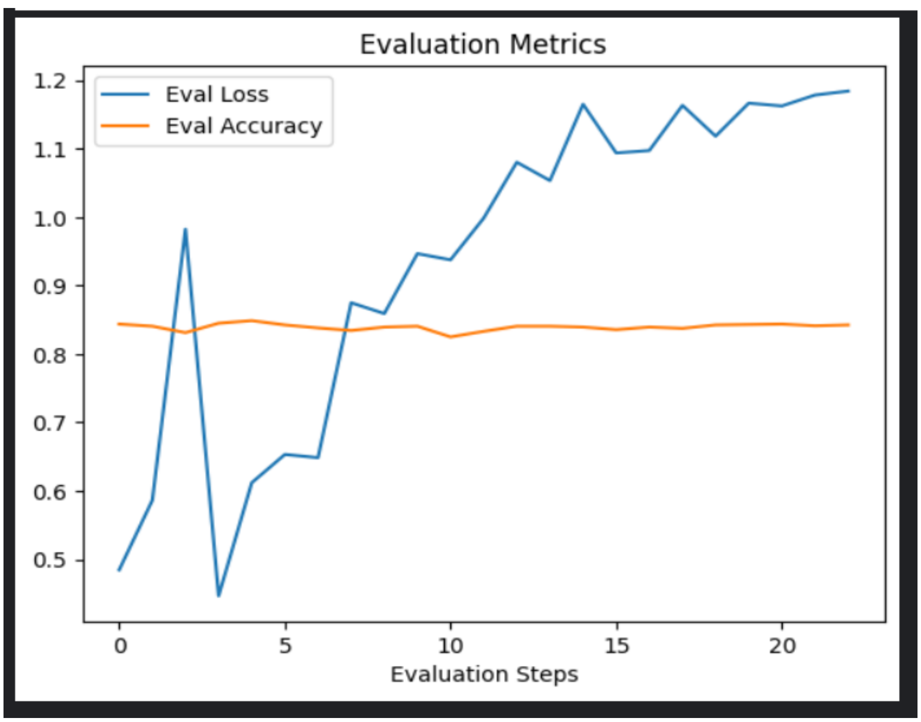
- BATCH_SIZE = 16
- MAX_LENGTH = 128
- LEARNING_RATE = 2e-5
- NUM_EPOCHS = 10
- PATIENCE = 2
- LOGGING_STEPS = 50

Training Configuration

- AdamW optimizer
- CrossEntropyLoss
- Data split:
- 70% training
- 15% validation
- 15% test
- Training arguments:
  - warmup_steps = 200
  - weight_decay = 0.01
  - gradient_accumulation_steps = 2

Training Progress





Results and Evaluation

Machine Learning Models Performance

- Evaluated using:
    - Accuracy
    - Precision
    - Recall
    - F1 Score

BERT Model Performance

```
Evaluating on validation set...
[100/100 00:06]
Validation Accuracy: 0.8487

Evaluating on test set...
Test Accuracy: 0.8313
```

- Validation Accuracy: 84%
- Test Accuracy: 83%