SD DOMBO UNIVERSITY OF BUSINESS AND INTEGRATED DEVELOPMENT STUDIES

FACULTY OF INFORMATION, COMMUNICATION AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

GROUP 7 – AI ESSENTIALS

COURSE LECTURER: DR. NANAYAW AFRIYIE

SENTIMENT ANALYSIS ON AMAZON REVIEWS

Introduction

Sentiment analysis is a crucial NLP task that helps businesses understand customer opinions from textual data. This project analyzes Amazon product reviews using three different sentiment analysis techniques:

1. **VADER (Valence Aware Dictionary and sEntiment Reasoner)** – A rule-based model optimized for social media text.
2. **RoBERTa (Robustly optimized BERT approach)** – A transformer-based model fine-tuned for sentiment analysis.
3. **ALBERT (A Lite BERT)** – A lightweight but powerful variant of BERT.

The goal was to compare their performance in classifying sentiment and determine which model best aligns with human-assigned review scores (1-5 stars).

Dataset Overview

- ProductId: Unique identifier for the product.

- UserId: Unique identifier for the user.

- ProfileName: Name of the user profile.

- HelpfulnessNumerator: Number of users who found the review helpful.

- HelpfulnessDenominator: Total number of users who indicated whether the review was helpful.

- Score: Rating given by the user (ranging from 1 to 5).

- Time: Timestamp of the review.

- Summary: Summary of the review text.

- Text: The actual review text.

Exploratory Data Analysis (EDA)

- **Review Distribution**:
  - Most reviews were highly positive (4-5 stars).
  - Skewness (-1.74): Strong left skew, indicating bias toward high ratings.
  - Kurtosis (1.86): Platykurtic (flatter than a normal distribution), meaning fewer extreme low/high ratings.

- **Correlation Analysis**:
  - HelpfulnessNumerator vs. HelpfulnessDenominator (0.89): Strong correlation, suggesting they measure similar aspects.
  - Score vs. Helpfulness (-0.00 to -0.19): Near-zero correlation, meaning review rating does not predict helpfulness.
  - Time vs. Helpfulness (-0.19 to -0.24): Slightly negative, possibly because older reviews had more time to accumulate helpful votes.

Methodology

Preprocessing

- **Tokenization & POS Tagging** (using NLTK):
  - Example:

```
tokens = nltk.word_tokenize("This oatmeal is not good.")

# Output: ['This', 'oatmeal', 'is', 'not', 'good', '.']
```

  - Part-of-speech tagging helped in understanding grammatical structure.

Sentiment Analysis Models

A. VADER (Rule-Based)

- **Strengths**:
  - Works well with informal language (e.g., "This product is NOT good!").
  - Provides compound scores (ranging from -1 to +1).
- **Implementation**:

```
from nltk.sentiment import SentimentIntensityAnalyzer

sia = SentimentIntensityAnalyzer()

sia.polarity_scores("I am so happy!")

# Output: {'neg': 0.0, 'neu': 0.334, 'pos': 0.666, 'compound': 0.6115}
```

B. RoBERTa (Transformer-Based)

- **Strengths**:
  - Pretrained on X (formerly Twitter) data, making it robust for short text.
  - Uses neural networks for deep contextual understanding.
- **Implementation**:

```
from transformers import AutoTokenizer, AutoModelForSequenceClassification

MODEL = "cardiffnlp/twitter-roberta-base-sentiment"

tokenizer = AutoTokenizer.from_pretrained(MODEL)

model = AutoModelForSequenceClassification.from_pretrained(MODEL)
```

C. ALBERT (Lightweight BERT)

- **Strengths**:

    - More efficient than BERT but maintains high accuracy.
    - Introduced a neutral category (since ALBERT is binary by default).
- **Implementation**:

```
ALBERT_MODEL = "textattack/albert-base-v2-imdb"

albert_tokenizer = AutoTokenizer.from_pretrained(ALBERT_MODEL)

albert_model =
AutoModelForSequenceClassification.from_pretrained(ALBERT_MODEL)
```

Results & Model Comparison

Performance Metrics

VADER Performance		**RoBERTa Performance**		**ALBERT Performance**	
Accuracy	83%	Accuracy	86%	Accuracy	87%
Precision	80%	Precision	86%	Precision	83%
Recall	83%	Recall	86%	Recall	87%
F-1 Score	81%	F-1 Score	86%	F-1 Score	84%

Key Observations

- RoBERTa & ALBERT outperformed VADER due to their contextual understanding.
- VADER struggled with sarcasm & complex sentences (e.g., "Great, just what I needed... not.").
- ALBERT was slightly better than RoBERTa in accuracy but had lower precision, meaning it was more lenient in classifying negatives.

Confusion Matrices

- **VADER**:
  - Misclassified 23 negative reviews as positive.
  - Struggled with neutral sentiment.
- **RoBERTa**:
  - Better at detecting positives (376 correct).
  - Still had issues with neutral reviews.
- **ALBERT**:
  - Best at identifying positives (387 correct).
  - Performed poorly on neutral labels (likely due to binary pretraining).

Visualizations

Sentiment Distribution by Score

- Bar plots showed that higher star ratings (4-5) had higher compound sentiment scores.
- Word clouds highlighted frequent terms in positive/negative reviews.

Correlation Heatmap

- Confirmed that helpfulness metrics were strongly correlated, while score was independent of time & helpfulness.

Challenges & Solutions

Challenge	Solution
Memory constraints	Used only 500 samples instead of full dataset.
ALBERT’s binary output	Added a neutral threshold (neutral_threshold=0.05).
Runtime errors in long texts	Used try-except blocks to skip problematic reviews.

Conclusion & Future Work

Findings

- Transformer models (RoBERTa, ALBERT) outperformed VADER in accuracy.
- ALBERT was the best overall, but RoBERTa was more precise.
- VADER is still useful for quick, rule-based sentiment checks.

CONCLUSION

For high accuracy, RoBERTa/ALBERT are superior, but VADER remains a good choice for fast, lightweight sentiment analysis. Businesses can use these insights to automate review analysis and improve customer experience.