Individual Final Report:

Comparative Sentiment and Topic Analysis of Threads and Twitter Reviews

Prepared for: Prof. Ning Rui

Prepared by: Haeyeon Jeong

December 05, 2025

I. Introduction

This project, "Comparative Analysis: Threads and Twitter Reviews," is a comprehensive study focused on analyzing user feedback toward two major microblogging platforms, Threads and Twitter (now X), during the critical launch period of Threads in July 2023. The core objective was to extract comparative insights regarding user sentiment and thematic concerns using a joint pipeline of classical and advanced Natural Language Processing (NLP) techniques.

The shared work was executed in three major phases: Data Acquisition and Preprocessing, Topic Modeling, and Sentiment Classification.

Table 1
Individual Contribution

Phase	Shared Work	My Individual Contribution
I. Data Acquisition & Preprocessing	Dataset selection and harmonization.	Sourcing Datasets, Designing and Executing the Data Cleaning Pipeline for both platforms.
II. Sentiment Classification	Defining the classification strategy (baseline vs. transformer).	Developing and Training the DistilBERT Model for both Threads and Twitter datasets (Hyperparameter tuning was done by other teammate)
III. Topic Modeling	Defining the modeling strategy (LDA, NMF, BERTopic).	Implementing the BERTopic Topic Modeling approach for sentiment-aware theme extraction.

The comparative analysis aims to address four research questions regarding sentiment distribution, VADER reliability, model performance (classical vs. transformer), and topic interpretability across both platforms.

II. Description of Individual Work

1. Algorithms and Background

My individual work focused on the critical areas of data preparation and implementing the two models in our pipeline: the DistilBERT transformer for sentiment classification and BERTopic for topic modeling.

A. DistilBERT for Sentiment Classification

I used DistilBERT as classifier to demonstrate the superiority of contextual models over traditional sparse feature methods like TF-IDF + Logistic Regression. DistilBERT is a smaller, faster, and lighter version of the BERT (Bidirectional Encoder Representations from Transformers) model.

Background and Equations: DistilBERT utilizes the Transformer architecture and the self-attention mechanism to understand the context of every word in relation to all other words in the sentence. The foundation of this architecture is the multi-head attention mechanism. The output of the scaled dot-product attention for a query Q, key K, and value V is given by the standard Transformer equation (Vaswani et al., 2017):

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

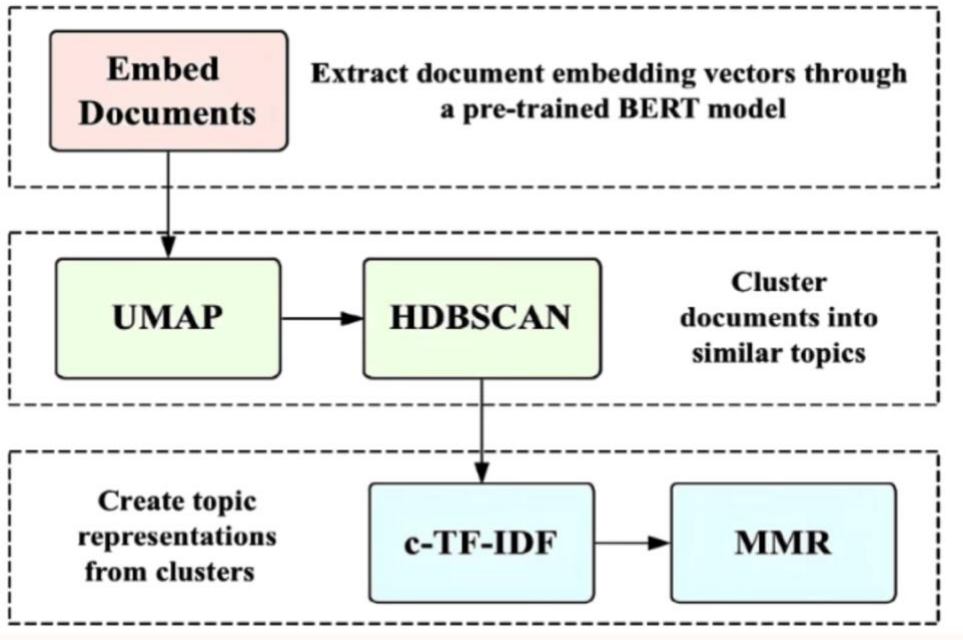
where d_k is the dimension of the key vectors. This capability allows the model to capture complex semantic and syntactic relationships necessary for interpreting short, informal social media text.

B. BERTopic for Topic Modeling

While traditional topic models (LDA, NMF) rely on word co-occurrence counts, BERTopic leverages deep learning embeddings, making it highly effective for short, noisy text (Xenoss, 2025).

BERTopic Architecture: BERTopic follows a modular, three-step process that utilizes modern embedding and clustering techniques. By default, the main steps are Sentence-BERT, UMAP, HDBSCAN, and c-TF-IDF run in sequence.

Figure 1
BERTopic Architecture



Note. This figure illustrates the modular three-step BERTopic pipeline: Embedding, Clustering, and Topic Representation. Adapted from BERTopic Documentation (n.d.).

1. Embedding: Documents are embedded using a pre-trained BERT model (specifically, Sentence-BERT (S-BERT)) to extract dense vector representations that capture semantic meaning.

2. Clustering: Techniques like UMAP (Uniform Manifold Approximation and Projection) are first applied to reduce the dimensionality of the embeddings. Subsequently, HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) is used for density-based clustering to group semantically similar documents into topics and identify outliers.
3. Topic Representation: Class-based TF-IDF (c-TF-IDF) is used to extract the most descriptive keywords for each cluster, quantifying the importance of each word within the cluster. This yields coherent topic representations.

III. Detailed Description of Individual Work

A. Data Sourcing and Cleaning Pipeline

My first contribution was sourcing the datasets from Kaggle and developing the unified preprocessing pipeline.

The analysis was conducted using a joint pipeline of classical and deep learning models.

1. Sourcing and Harmonization: I selected the Threads (Kaggle, 2023) and Twitter (Kaggle, 2023) datasets, ensuring temporal alignment in July 2023 and comparable size.
2. Data Cleaning Pipeline: I implemented the unified, sequential cleaning pipeline on both datasets, which included:
    - Duplicate Removal: Removed raw text duplicates and long-review duplicates (>15 words).
    - Brand Normalization: Standardized platform names (e.g., "twt," "x" → "twitter"; "treads" → "threads").

- Text Normalization: Lowercased text, removed URLs, emojis, special characters, and normalized whitespace.
- Stop-words & Lemmatization: Removed English stopwords but retained negations ("not," "never"), and lemmatized words
- Minimal Language Filtering: Used langdetect to remove frequently misdetected languages, keeping English and "unknown" content.

The final dataset sizes were 29,646 for Threads and 29,611 for Twitter.

B. DistilBERT Sentiment Classification

I was responsible for implementing and training the initial DistilBERT model for sentiment classification (before hyperparameter tuning).

1. Training Setup: I loaded the pre-trained DistilBERT model and configured the training process using the Hugging Face Trainer framework. The model utilized the Adam optimizer and CrossEntropy Loss.
2. Hyperparameters (Initial Configuration): I ensured the model was correctly loaded and trained using the VADER pseudo-labels on the cleaned datasets for sentiment classification.
    - Threads:

      - num_train_epochs=3

      - learning_rate=2e-5

      - per_device_train_batch_size=16

      - per_device_eval_batch_size=16

      - weight_decay=0.01

❿ Twitter:

- num_train_epochs=3
- learning_rate=1e-5
- per_device_train_batch_size=16
- per_device_eval_batch_size=16
- weight_decay=0.01

C. BERTopic Topic Modeling

I was responsible for implementing the BERTopic model to extract coherent, sentiment-aware themes.

1. Implementation: I applied BERTopic separately to the positive, neutral, and negative sentiment subsets for both Threads and Twitter.
2. Insight Extraction: The insights derived from BERTopic were crucial for comparative analysis, showing that Threads' negativity focused on functional issues (e.g., app, crashing, bug) while Twitter's negativity was ideological/leadership-focused (e.g., elon, ruined, sucks, worst).

IV. Results

My experiments focused on model performance comparison and advanced topic discovery.

A. Sentiment Classification Performance (DistilBERT vs. Baseline)

The DistilBERT model consistently outperformed the traditional TF-IDF + Logistic Regression baseline , proving that contextual embedding models are necessary for short, complex social media text.

Table 2
Classification Model Performance Comparison (Before DistilBERT Hyperparameter Tuning)

Model	Threads Accuracy	Threads F1	Twitter Accuracy	Twitter F1
TF–IDF + Logistic Regression	0.74	0.69	0.84	0.83
DistilBERT	0.84	0.81	0.92	0.92

Explanation:

- Superiority of DistilBERT: The transformer model yielded substantial performance gains, with the performance gap being larger on Twitter (Accuracy: 0.92 vs 0.84).
- Platform Difference: DistilBERT achieved its highest performance on Twitter (Accuracy: 0.92) because Twitter's concise and emotionally explicit posts allowed for easier contextual modeling.

B. BERTopic Coherence Scores (Topic Interpretability)

BERTopic model I implemented achieved the best overall coherence on Twitter (Average 0.52) and strong coherence on Threads (Average 0.53), validating its effectiveness.

Table 3
Topic Model Coherence Comparison

Model	Threads (Avg. Coherence Score)	Twitter (Avg. Coherence Score)
LDA	≈ 0.52	≈ 0.50
NMF	≈ 0.46	≈0.43
BERTopic	≈ 0.53	≈ 0.52

Explanation:

- BERTopic's superior coherence demonstrated the benefit of using contextual embeddings to group documents based on semantic similarity , resulting in key insights unavailable to count-based models.

The BERTopic analysis reveals stark differences in the thematic drivers of sentiment between the two platforms. The keyword weights shown below confirm that Threads users focus on functional issues, while Twitter users focus on policy and leadership controversy.

Figure 4
BERTopic Topics and Keyword Weights for Positive Threads Review

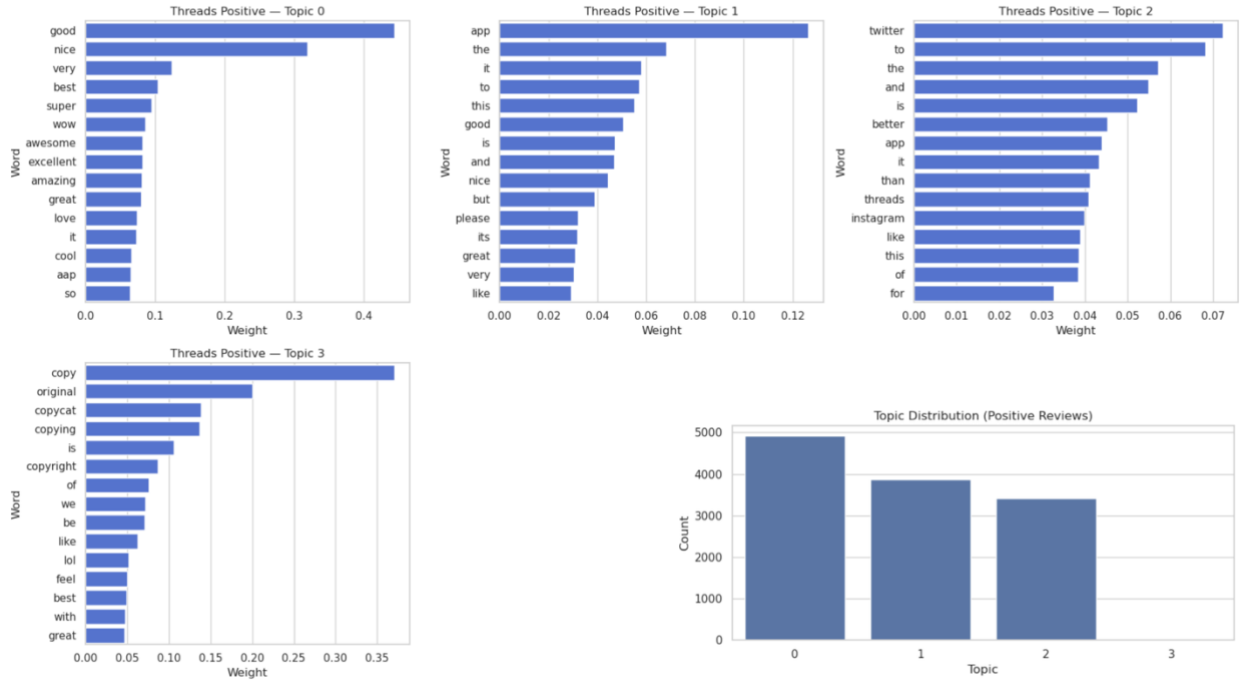


Figure 5
BERTopic Topics and Keyword Weights for Neutral Threads Review

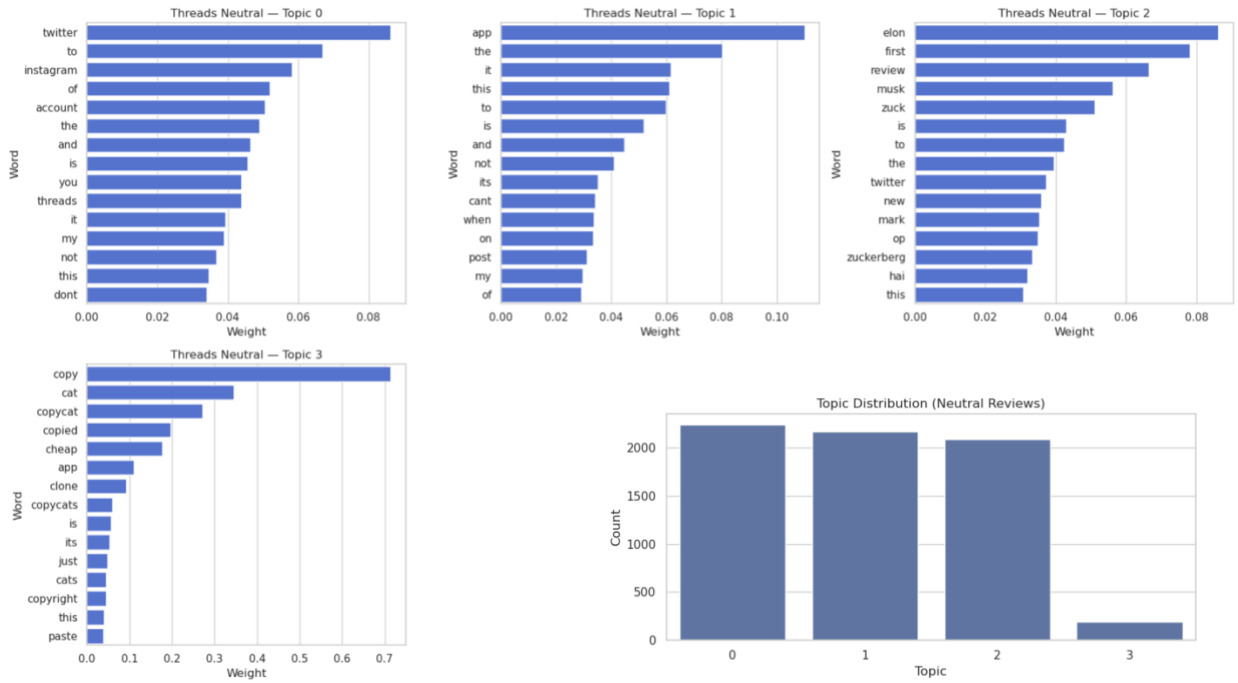
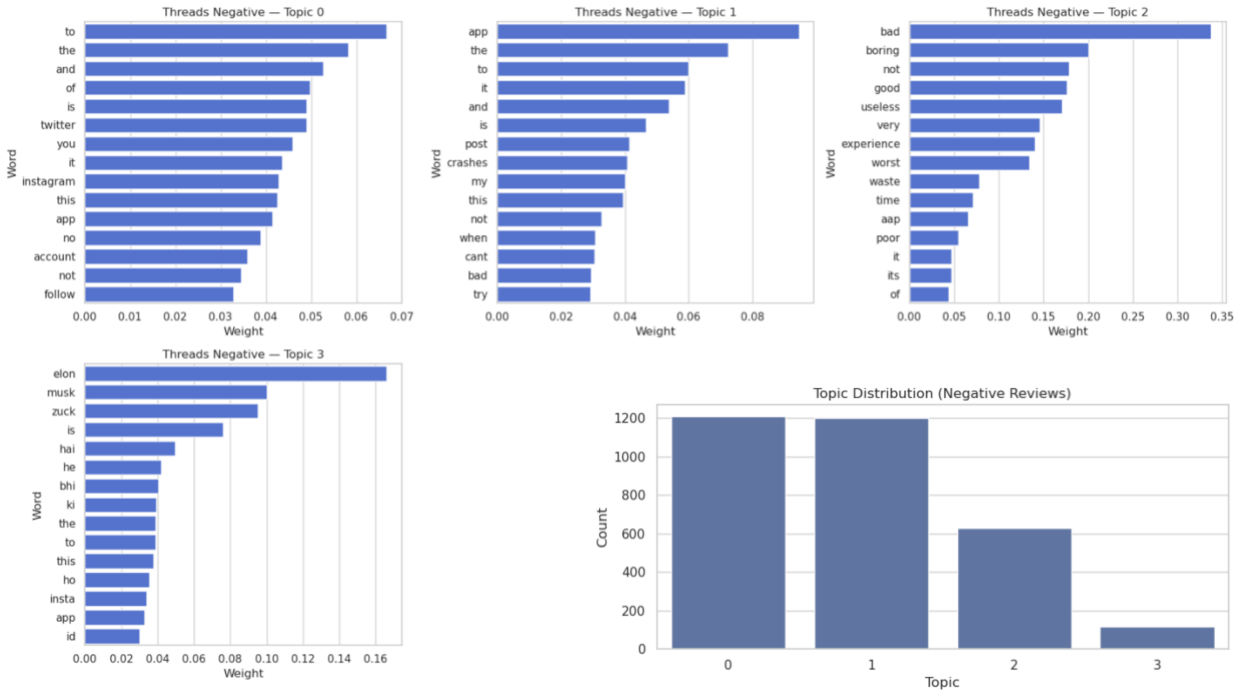


Figure 6
BERTopic Topics and Keyword Weights for Negativel Threads Review



Explanation (Threads):

- Positive topics are straightforward. Topic 0 contains general praise (nice, very, best, amazing), and Topic 2 highlights positive comparisons (twitter, better, than instagram) (Figure 4).

- Neutral sentiment is defined by comparison and observation. Topic 0 focuses on platform comparison (twitter, instagram), and Topic 3 explicitly discusses the copy, cat, copycat, copied narrative, indicating neutral commentary on the app's derivative nature (Figure 5).

- The negative topics are dominated by complaints related to stability and features. Topic 1 focuses on crashes and errors (app, post, crashes), while Topic 2 reflects generalized dissatisfaction (bad, boring, useless, worst) (Figure 6).

Figure 7

BERTopic Topics and Keyword Weights for Positive Twitter Review

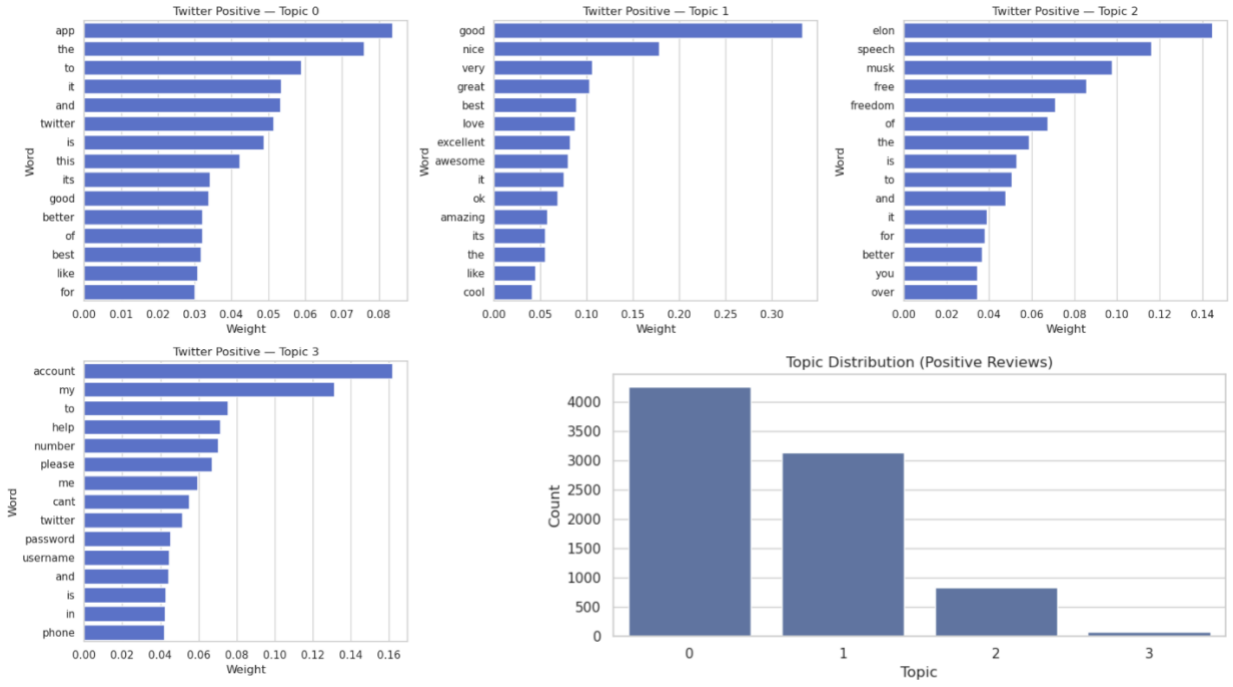


Figure 8

BERTopic Topics and Keyword Weights for Neutral Twitter Review

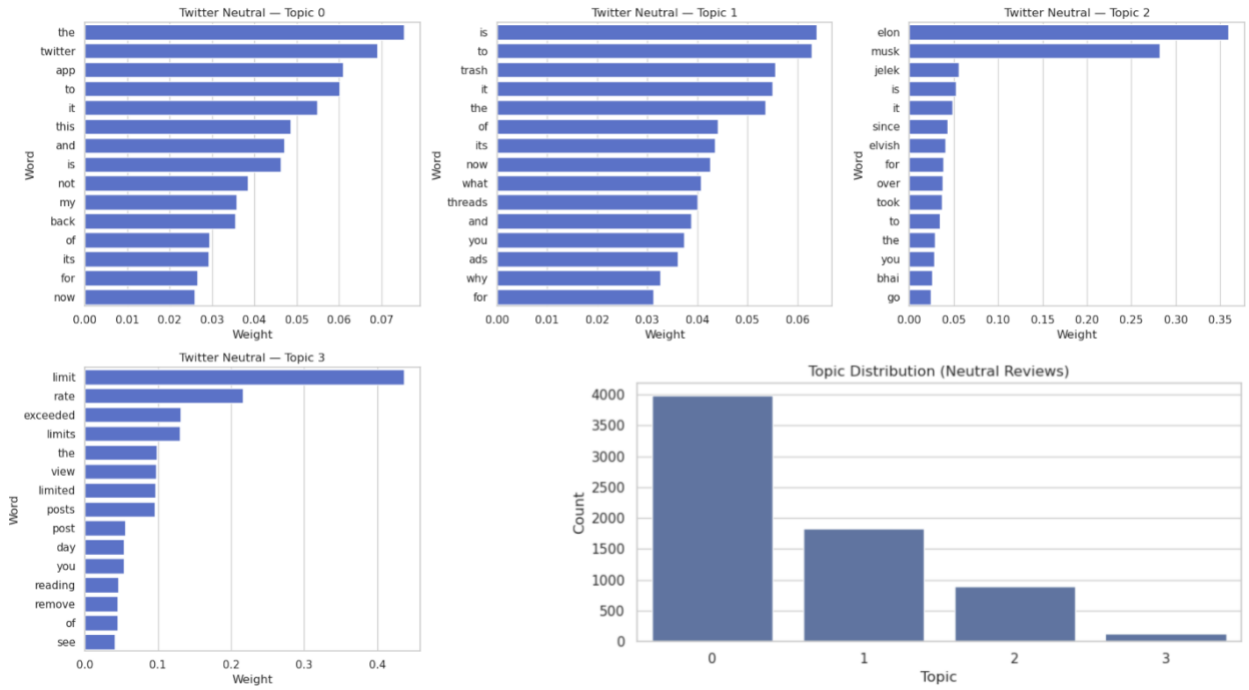
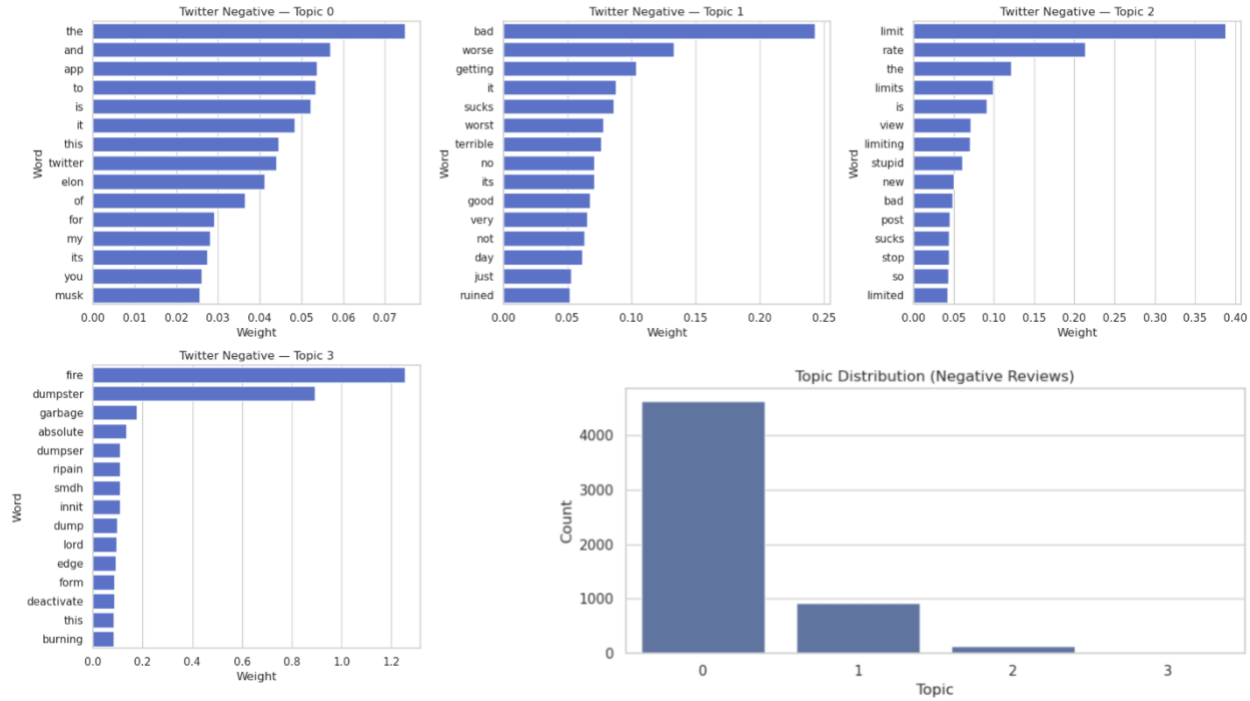


Figure 9

BERTopic Topics and Keyword Weights for Negative Twitter Review



Explanation (Twitter):

- Positive topics are varied. Topic 2 highlights ideological satisfaction (elon, speech, musk, free, freedom), a thematic driver absent on Threads. Topic 3 shows account support requests (account, help, please, number), which were a small segment pseudo-labeled as positive by VADER (Figure 7).

- Neutral topics are tied to specific policies and leadership mentions. Topic 2 is a leadership mention (elon, musk, zuck), and Topic 3 mentions rate limits (limit, rate, exceeded) without strong sentiment (Figure 8).

- Twitter negativity is highly polarized and policy-driven. Topic 1 reflects intense personal rejection (bad, worse, sucks, worst, ruined). Topic 2 clearly focuses on policy issues (limit, rate, exceeded, limits) (Figure 9)

Summary and Conclusions

Summary of Results

My work established a robust data foundation and implemented the project's most advanced analytical models. The DistilBERT classification confirmed that transformer architectures provide a necessary performance boost for sentiment analysis (up to 92% accuracy for Twitter before hyperpameter tuning). The BERTopic implementation showed that Threads' issues were functional (missing features, stability), while Twitter's were ideological and political (leadership, rebranding).

Lesson Learned

The most crucial lesson learned was the practical difference between traditional and embedding-based NLP. The substantial performance gap between Logistic Regression and DistilBERT

highlighted that simple word-counting techniques are obsolete for nuanced, informal text. I learned that meticulous data preprocessing directly impacts model performance. I also learned the value of a well-defined hyperparameter tuning strategy (executed by the team) in maximizing the predictive power of complex transformer models.

Suggested Improvements

- Manual Labeling: The reliance on VADER pseudo-labels limited the models' potential. VADER struggles with mixed or nuanced sentiment, and these pseudo-labels introduced noise into model training.

- Future Work: Explore alternative topic extraction models such as Top2Vec and investigate multi-label classification for reviews expressing multiple themes.

Business Insights

The comparative analysis reveals distinct strategic imperatives for both Threads and TwitterL

For Threads (New Products): New products must prioritize delivering a stable core experience and essential functionality first. The immediate focus should be on platform stability and rapid feature parity to convert early adopters.

For Twitter (Major Updates): Major updates or rebranding efforts require careful, transparent communication to maintain user trust. Addressing policy controversies (like rate limits) and managing the rebranding narrative are critical, as these ideological topics are core drivers of negative sentiment.

Code Calculation

The percentage of code found or copied from the internet is calculated using the formula provided for the project. The percentage of the code found or copied from the internet is approximately 39%.

References

BERTopic. (n.d.). *The BERTopic Algorithm*. BERTopic Documentation. Retrieved from https://bertopic.com/

Grootendorst, M. (2022). **BERTopic: Neural topic modeling with a class-based TF-IDF procedure**. *arXiv preprint arXiv:2203.05794*. *https://arxiv.org/abs/2203.05794*

Kaggle. (2023). *Threads: An Instagram App Reviews* [Data set]. Kaggle. https://www.kaggle.com/datasets/saloni1712/threads-an-instagram-app-reviews

Kaggle. (2023). *2 Million X (Twitter) Google Reviews* [Data set]. Kaggle. https://www.kaggle.com/datasets/bwandowando/2-million-formerly-twitter-google-reviews

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). **Attention is all you need**. In *Advances in neural information processing systems* (Vol. 30). Curran Associates, Inc.

Xenoss. (2025, April 7). **Topic modeling techniques: LDA, NMF, Top2Vec & BERTopic**. Xenoss Blog. https://xenoss.io/blog/topic-modeling-techniques-comparison