Project Methodology

Parnavi Sen,Navneet Parab

March 2025

1 Purpose of the Methodology

This methodology is designed to perform sentiment analysis on Reddit data, focusing on athletic apparel and technology companies. By leveraging scraped social media data, the approach aims to extract insights into public perception, brand sentiment, and emerging trends.

The primary objective is to understand consumer sentiment in real time, allowing businesses to make informed decisions about marketing strategies, product development, and customer engagement. Reddit provides a rich source of unfiltered discussions, making it an ideal platform to gauge public perception and brand sentiment dynamically.

To ensure a comprehensive sentiment analysis, we compare multiple approaches, including traditional machine learning and deep learning models. Each method offers unique advantages—some are quick and interpretable, while others are more sophisticated and capable of capturing nuanced sentiment expressions. By conducting this comparison, we aim to evaluate the trade-offs between efficiency, accuracy, and complexity, ultimately determining the most effective approach for sentiment classification.

This multi-model comparison allows us to optimize sentiment detection, ensuring that businesses can rely on accurate and actionable insights. Understanding how different techniques perform on unstructured, real-time text data helps in refining sentiment analysis methodologies for broader applications in consumer insights and brand perception monitoring.

2 Problem Statement

2.1 Defining the Problem

The problem at hand falls under the category of sentiment analysis, a subset of text classification in Natural Language Processing (NLP). Specifically, this project aims to analyze sentiment from user-generated content on Reddit, focusing on discussions related to athletic apparel and technology companies. Sentiment analysis is crucial for understanding consumer perception, brand reputation, and market trends.

To achieve this, we compare multiple machine learning approaches, including traditional lexicon-based models, unsupervised clustering techniques, and advanced deep learning models. This multi-model evaluation allows us to determine the most effective method for handling unstructured, real-time social media data and extracting meaningful sentiment insights.

2.2 Significance

Understanding sentiment from large-scale social media data is valuable for both research and industry applications:

- **Industry Impact**: Businesses can leverage sentiment analysis for brand monitoring, targeted marketing, and crisis management. Insights from Reddit discussions help companies track emerging trends and customer satisfaction levels.

- **Academic Contribution**: Our approach extends existing research by comparing traditional and modern machine learning techniques. Unlike prior studies that rely on feature engineering or rule-based sentiment models, our methodology integrates Large Language Models (LLMs) such as BERT and LLaMA, which excel in contextual sentiment understanding.

- **Technical Challenges Addressed**: Many existing sentiment analysis methods struggle with sarcasm, evolving slang, and context-dependent sentiment shifts. Our approach aims to overcome these issues by utilizing advanced transformer-based embeddings and improving the interpretability of sentiment scores.

2.3 Related Work

Our work builds on several existing research efforts in sentiment analysis:

- **Traditional ML Models**: Studies like "Sentiment Analysis of User-Generated Twitter Updates" use Naïve Bayes and Maximum Entropy, which are effective but lack contextual understanding. Our approach enhances these methods by integrating transformer models like BERT and fine-tuned LLMs for deeper sentiment comprehension.

- **Hybrid Approaches**: Prior work on "Sentiment Analysis of Twitter Data" combines syntactic and sentiment-specific features but struggles with scalability due to computationally expensive techniques. Our model eliminates the need for extensive feature engineering by leveraging pre-trained deep learning models.

- **Reddit-Specific Sentiment Analysis**: Research such as "Reddit Sentiment Analysis" relies on tools like VADER and CNNs but is limited in adaptability. Our approach introduces automated preprocessing, transformer-based embeddings, and multiple sentiment classification strategies to improve accuracy and scalability.

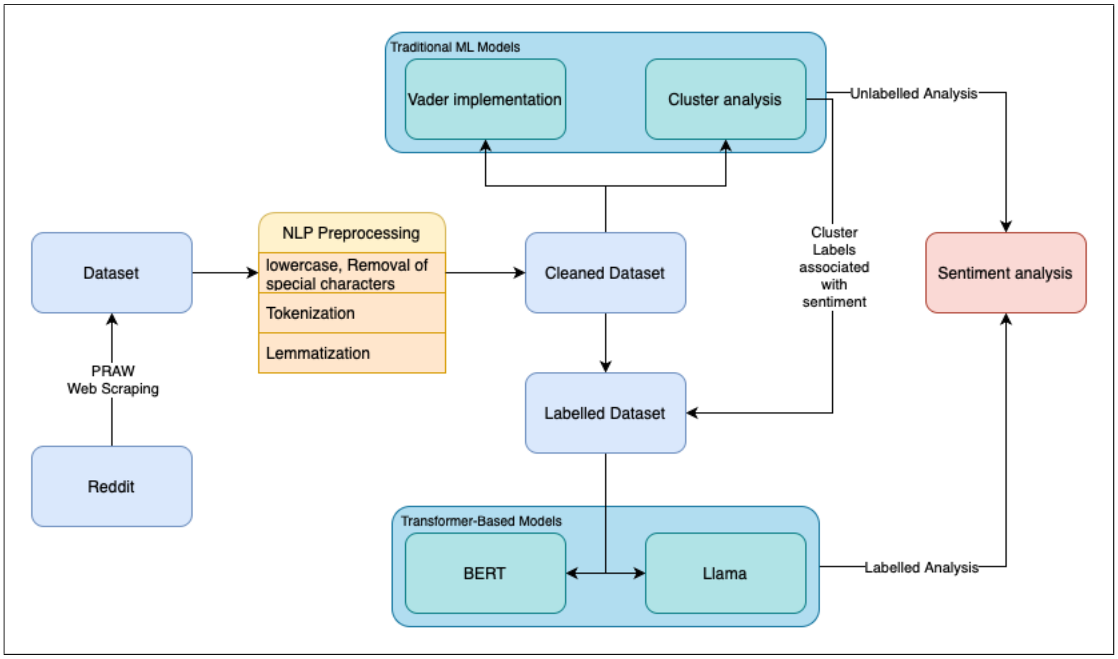

Figure 1: Project Workflow

By addressing these gaps, our methodology enhances sentiment detection across different domains, providing businesses and researchers with a robust and scalable solution for sentiment analysis on Reddit data.

3 Data Collection and Preparation

3.1 Data Sources

The dataset for this study is sourced from Reddit, using the PRAW (Python Reddit API Wrapper) for web scraping. This allows access to real-time discussions and public opinions. The dataset includes posts from two domains:

- Athletic Apparel Companies
- Technology Companies

Each dataset consists of approximately 11,000 data points, making a total of 22,000 records. The data includes structured elements such as timestamps, subreddit names, post titles, post body, comment counts, upvotes, and upvote ratios.

3.2 Data Description

Number of Samples: 11,000 records per dataset (Athletic Apparel & Technology Companies).
Feature Descriptions:

- **Subreddit**: The name of the subreddit where the post was made.
- **Timestamp**: When the post was created.
- **Title**: The headline of the post.
- **Body**: The content of the post (if any).
- **Comments**: Number of comments on the post.
- **Number of Upvotes**: Popularity measure based on community reactions.
- **Upvote Ratio**: Ratio of upvotes to total votes (upvotes + downvotes).

3.3 Preprocessing Steps

Text Clean-Up

- Standardize to all lowercase.
- Removal of special characters, numbers, and stopwords.

NLP Breakdown

- **Tokenization**: Splitting text into meaningful components.
- **Lemmatization**: Converting words to their root forms.
- **Emoji Processing**: Using the emoji package to convert emojis into text descriptions, preserving sentiment-related cues from emojis in social media posts.

BERT-Specific Processing

- We use the BERT tokenizer (bert-base-uncased) to tokenize text efficiently.
- The tokenizer splits text into subwords, ensuring that the model can handle out-of-vocabulary words.
- Padding is applied to ensure all sequences in a batch have the same length, preventing shape mismatches during model training.

By carefully collecting and processing this dataset, we ensure that sentiment analysis models are trained on high-quality, meaningful, and representative text data.

4 Selection of Machine Learning or LLM Models

4.1 Model Consideration

For this project, we evaluated a range of models to determine the most effective approach for sentiment analysis. We considered both traditional machine learning models and deep learning/transformer-based models:

- **Traditional Models**:
  - **VADER (Valence Aware Dictionary and Sentiment Reasoner)**: A lexicon-based model for quick sentiment classification.
  - **K-Means Clustering**: An unsupervised approach for identifying sentiment-based clusters within the data.
- **Deep Learning and Transformer-Based Models:**
  - **BERT (bert-base-uncased)**: A transformer-based model capable of capturing contextual sentiment.
  - **LLaMA**: A large-scale language model designed for nuanced sentiment understanding.

We considered these models due to their ability to handle unstructured text data, capture context, and improve sentiment classification accuracy. To determine the best-performing model(s), we used the following evaluation metrics:

- **Accuracy**: Measures overall correctness of sentiment classification.
- **Computational Efficiency**: Considers model inference time and scalability for large-scale sentiment analysis.

By comparing multiple models, we ensured a balanced trade-off between interpretability, computational efficiency, and classification accuracy. Traditional models like VADER provide quick and interpretable results, while deep learning models like BERT and LLaMA enhance accuracy and contextual understanding. The final selection prioritizes models that maximize performance while maintaining efficiency for real-world applications.

5 Model Development and Training

5.1 Architecture and Configuration

For this project, we utilized two large language models (LLMs):

- BERT (Bidirectional Encoder Representations from Transformers)
- LLaMA (Large Language Model Meta AI)

Each model leverages Transformer architecture but differs in implementation and usage.

BERT Model Architecture

- Transformer-based model that utilizes self-attention mechanisms.
- Pretrained model used: bert-base-uncased from Hugging Face.
- Number of layers: 12 Transformer layers.
- Hidden size: 768 dimensions per token embedding.
- Self-Attention Heads: 12 heads.
- Maximum sequence length: 512 tokens.
- Fine-tuning strategy:
  - The pretrained BERT model was fine-tuned for classification.
  - A classification head (fully connected layer) was added to predict sentiment clusters.

LLaMA Model Architecture

- Transformer-based decoder model optimized for efficiency.
- Pretrained model used: ”meta-llama/Llama-2-7b-hf” (7-billion parameters).
- Number of layers: 32 Transformer layers.
- Hidden size: 4096 dimensions per token embedding.
- Self-Attention Heads: 32 heads.
- Pretraining vs. Fine-tuning:
  - LLaMA-2 was pretrained on a vast dataset.
  - Fine-tuning was performed on a task-specific dataset using supervised learning.

5.2 Training Process

To prepare the dataset for training, we applied text preprocessing steps such as lowercasing, removal of special characters and numbers, stopword removal, lemmatization, and tokenization were. The dataset consisted of three main textual features: Title, Body, and Comments, which were concatenated into a single text representation for both models. We split the dataset into 80% training and 20% testing sets using train_test_split from Scikit-learn, ensuring a sufficient number of examples for both training and evaluation.

Since the dataset was originally scrapped from Reddit, the labels for the same were not initialized during the web scrapping process. We used the clusters learned from the unsupervised learning K-means implementation to assign labels before training the Transformer-based models.

For BERT, tokenization was performed using BERT Tokenizer (bert-base-uncased), which converted input text into input IDs, attention masks, and token type IDs. Fine-tuning was carried out using the AdamW optimizer with a learning rate of 2e-5, a batch size of 8, and a total of 3 training epochs. Cross-entropy loss was used as the objective function to optimize classification performance. To prevent overfitting, dropout layers and weight decay regularization were applied during fine-tuning.

For LLaMA, the text was tokenized using LLaMA Tokenizer (meta-llama/Llama-2-7b-hf), converting input text into input IDs and attention masks. Unlike BERT, LLaMA does not require token type IDs. Training was conducted using the Hugging Face Trainer API, incorporating gradient checkpointing and mixed precision (fp16) to reduce memory usage. The batch size was reduced to 2 to accommodate the 7B parameter model within a 14GB GPU, and Adam optimizer with weight decay was used to fine-tune the model.

6 Evaluation and Comparison

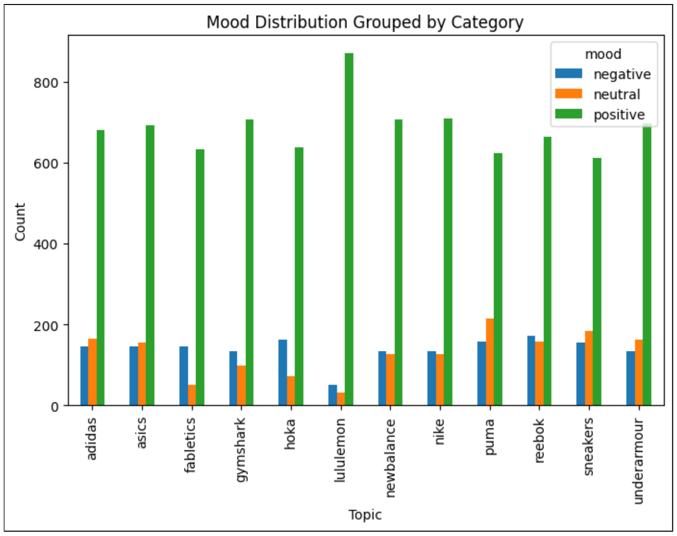

Figure 2: Sentiment Distribution

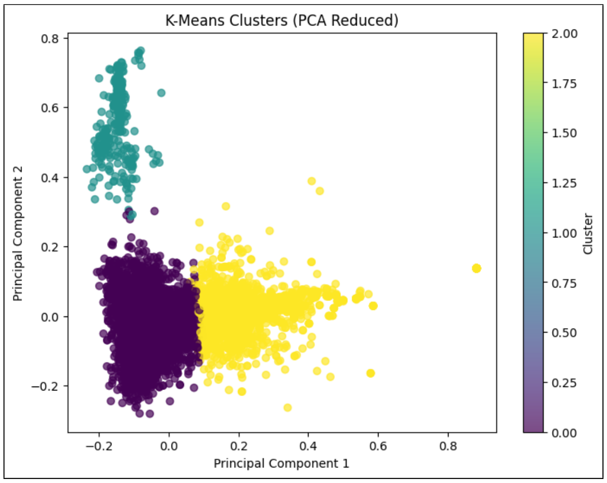

Figure 3: K-Means Clusters

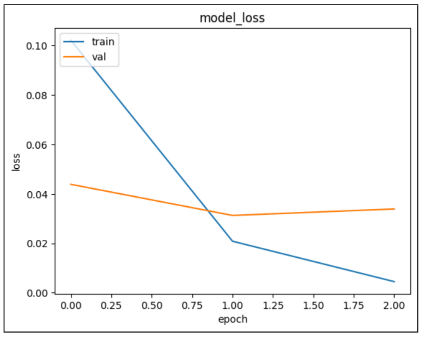

Figure 4: Transformer Based Model Loss Plot

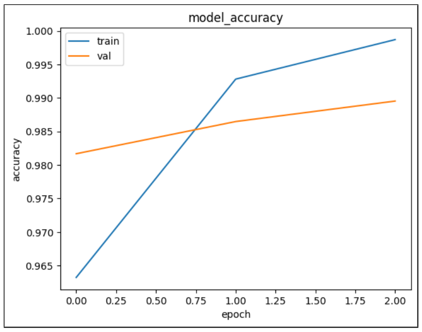

Figure 5: Transformer Based Model Accuracy Plot

We were able to validate the K-Means clusters with the sentiments classified by the VADER model. These clusters were used to label the unlabelled dataset. Both models demonstrated high classification accuracy, with BERT achieving 85-90% accuracy.