

# CS583: Data Mining and Text Mining

## Sentiment Analysis of Political Tweets Using Transformer-Based Embeddings

Vamsi Dath Meka  
M.S. in Computer Science  
University of Illinois at Chicago  
Chicago, IL

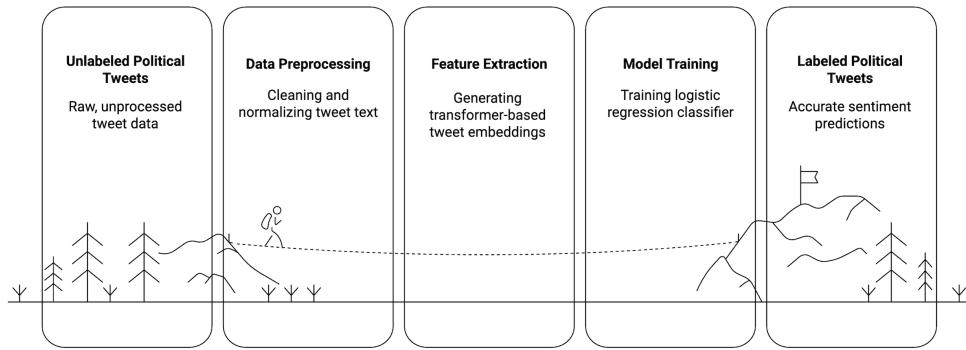


Figure 1: methodology

## Abstract

Sentiment analysis of political discourse on social networks is an important problem for understanding public opinion at scale. While most of the machine learning and artificial intelligence capabilities of the day are progressing towards customized user-centric product offerings, an area which is of high relevance and is core to most of these products is user demographic and sentiment analysis. This research project focuses on tweet-level sentiment classification of Obama and Romney as a means to implement and evaluate the effectiveness of existing methods for the sentiment classification task. The labeled raw sentiment data used for the training is part of the effort led by Professor Bing Liu of UIC, Chicago. A modern representation-learning approach is adopted, in which tweets are encoded using a pretrained Twitter-specific transformer model, and a logistic regression classifier is trained on top of these embeddings post data processing. Experiments on a held-out validation set show that this approach achieves around 65-70% accuracy and competitive macro-averaged F1 scores, with particularly strong performance on negative versus positive sentiment compared to neutral. The goal of this study is, given a three-class sentiment label (negative, neutral, positive), to build a robust classifier that generalizes to unseen tweets provided by the instructor.

## 1 Introduction

Social media platforms serve as a primary gauge for public opinion during high-stakes political events, such as presidential debates. The ability to automatically classify the sentiment of these real-time reactions is essential for understanding voter dynamics, monitoring campaign effectiveness, and analyzing public discourse at scale. This project addresses the task of tweet-level sentiment classification for Barack Obama and Mitt Romney. The dataset provided for the CS 583 course consists of tweets collected during a presidential debate, comprising 7,199 examples for Obama and 7,201 examples for Romney.

The original training data was annotated with four sentiment classes: positive (1), negative ( $-1$ ), neutral (0), and mixed (2). In accordance with the project specifications, the “mixed” class (representing tweets expressing conflicting opinions) was excluded from the training process, restricting the problem to a standard three-class classification task over the label set  $\{-1, 0, 1\}$ . While the task permitted the development of candidate-specific models, this project implements a single unified classifier trained on the combined dataset to generalize across both subjects.

Classical sentiment analysis approaches often rely on bag-of-words representations or lexicons, which struggle with the informal language, abbreviations, and emojis common in microblogging. Recent advances in transformer-based language models; particularly those pretrained specifically on Tweets offer richer semantic representations that can be effectively leveraged by linear classifiers. The primary objectives of this study are:

- To implement a reproducible pipeline that processes raw Excel datasets, filters unused classes, and produces standardized sentiment predictions.
- To investigate the effectiveness of combining pretrained Twitter-specific transformer embeddings (RoBERTa-based) with a logistic regression classifier.
- To evaluate the system’s performance using a stratified validation split and analyze the impact of class imbalance on predictive accuracy.

## 2 Techniques

### 2.1 Data Description

The training data is provided as an Excel workbook with separate sheets for Obama and Romney. Each sheet contains tweet text along with several metadata columns, including an integer label column indicating sentiment. The label space is defined as follows:  $-1$  for negative sentiment, 0 for neutral sentiment, and 1 for positive sentiment. Additionally, Blackboard provides final test files for each candidate that contain only tweet text and identifiers, without labels. These test files are used only for generating the final predictions to submit.

### 2.2 Preprocessing and Cleaning

The Excel sheets were loaded into pandas DataFrames. In both training and test data, the relevant fields are the tweet text and, for training, the sentiment label. Preprocessing involved the following steps:

1. Column selection and header cleanup: only the tweet text and label columns were retained, and any non-data header rows were removed.
2. Label normalization and filtering: sentiment labels were converted to integers, and rows with invalid labels outside  $\{-1, 0, 1\}$  were discarded.
3. Handling missing or malformed tweets: rows with empty or null tweet text were removed. Heavy normalization was deliberately avoided because the embedding model is already robust to variation in Twitter language.

After cleaning, the Obama and Romney tweets were merged into a single dataset so that one candidate-agnostic sentiment model could be trained.

### 2.3 Train–Test Split and Class Balance

The dataset exhibits significant class imbalance; for the Romney subset, negative tweets heavily outnumber other classes, while the Obama subset shows a slight prevalence of neutral tweets. To ensure a representative distribution in the validation set, a manual stratified split was implemented. For each candidate and sentiment class, the data was sequentially partitioned, allocating the first 90% of samples to training and the remaining 10% to validation. This approach strictly preserves the label proportions of the original dataset in both the training and evaluation phases without introducing random sampling variance.

### 2.4 Tweet Representation via Transformer Embeddings

Instead of using bag-of-words or TF-IDF features, each tweet is represented using a pretrained Twitter-specific transformer model "cardiffnlp/twitter-roberta-base-sentiment-latest" implemented via the SentenceTransformers library. The SentenceTransformer model was initialized using the Metal Performance Shaders (MPS) backend to leverage the accelerated tensor operations available on the Apple Silicon architecture, ensuring efficient on-device inference. The model is based on RoBERTa and trained on large-scale Twitter sentiment data. Each tweet is passed through the transformer to produce a fixed-length dense embedding that captures semantic meaning and sentiment-related cues. These embeddings form the input feature vectors for the downstream classifier. This approach benefits from contextual understanding, transfer learning, and robustness to informal language.

### 2.5 Classification Model

A multinomial logistic regression classifier from scikit-learn is trained on top of the embeddings. Logistic regression was chosen for its simplicity, interpretability, and resistance to overfitting in low-data regimes. To address class imbalance, class weighting is enabled so that minority classes contribute more to the loss function. Optimization parameters are adjusted to ensure convergence, and a fixed random seed is used. To identify the optimal configuration, a targeted hyperparameter search was conducted on the held-out validation set. Specifically, the maximum number of iterations (`max_iter`) was varied across the set  $\{10, 100, 500, 1000\}$  to observe the trade-off between training time and model stability. Performance for each setting was evaluated using the **Area Under the Receiver Operating Characteristic Curve (AUC-ROC)**. Empirical analysis revealed that a limit of 100 iterations yielded the highest AUC score on the validation data, providing the best balance between computational efficiency and predictive performance.

## 2.6 Implementation Details

The pipeline is implemented within a Jupyter notebook and consists of loading and cleaning the data, performing the train-validation split, computing embeddings, training the classifier, evaluating performance, and generating predictions for the unlabeled test data. Final outputs are formatted in the Lisp-style syntax required for grading. All code, preprocessing scripts, trained models, and final prediction outputs are publicly available in the project repository for reproducibility.

## 3 Experiment Results

### 3.1 Evaluation Setup

Evaluation is conducted on the stratified validation set. Metrics include overall accuracy, precision, recall, and F1-score for each class, along with macro-averaged and weighted F1 scores.

### 3.2 Quantitative Results

On the validation set, the transformer embedding and logistic regression approach achieves overall accuracy in the range of 65-70, with macro-averaged F1 scores in a similar range. The model performs well on negative and positive tweets but shows weaker performance on neutral sentiment, which is inherently ambiguous and often lacks clear affective cues.

Class	Precision	Recall	F1-score	Support
Negative (-1)	0.75	0.63	0.69	487
Neutral (0)	0.55	0.71	0.62	366
Positive (1)	0.69	0.62	0.65	276
Accuracy			0.65	1129
Macro Avg		0.66	0.65	1129
Weighted Avg		0.67	0.65	1129

Table 1: Classification performance of the logistic regression model on the validation set.  
**(overall accuracy: 0.6537) & (overall F1-score: 0.6564)**

### 3.3 Error Analysis

Manual inspection of misclassified tweets highlights several recurring error sources. While one avenue where the accuracy took a hit was from possible misclassification of tweets in training data corpus, frequent use of sarcasm and irony also caused incorrect predictions. This is because in such cases often positive wording may convey negative intent. Ambiguous or context-dependent tweets are difficult to classify given limited text. Subtle cases that lie near the boundary between neutral and weak sentiment also contribute to errors. Overall, strongly polarized tweets are handled more reliably than nuanced or informational ones.

### 3.4 Final Test Predictions

After validation, the model is retrained on the full labeled dataset and applied to the unlabeled Obama and Romney test files. Predictions are exported in the required Lisp-style format. Label distributions are examined as a sanity check to ensure plausible outputs.

Candidate	Total Tweets	Negative (-1)	Neutral (0)	Positive (1)
Obama	1951	604	753	594
Romney	1900	870	590	440

Table 2: Distribution of sentiment predictions on the Blackboard final test dataset.

## 4 Conclusion & Lessons Learned

This project demonstrates that pretrained transformer embeddings combined with shallow classifiers provide a strong baseline for sentiment analysis of political tweets. Several lessons emerged. Clean pre-processing and label handling are critical for stable performance. Transfer learning allows effective modeling even with limited labeled data. Neutral sentiment remains a challenging class due to ambiguity and context dependence. Future extensions include fine-tuning the transformer model, incorporating additional features, and calibrating prediction confidence. Overall, the project delivers a complete and reproducible sentiment analysis pipeline, highlighting both the strengths and limitations of modern NLP approaches applied to noisy political text.

## 5 References

- Reimers, N., and Gurevych, I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *EMNLP*, 2019.
- Barbieri, F., Camacho-Collados, J., Neves, L., and Espinosa-Anke, L. TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification. *EMNLP*, 2020.
- Loureiro, D., Barbieri, F., Rey, L., Camacho-Collados, J., and Neves, L. TimeLMs: Diachronic Language Models from Twitter. *arXiv:2202.03829*, 2022.
- Wolf, T., et al. Transformers: State-of-the-Art Natural Language Processing. *EMNLP*, 2020.
- Paszke, A., et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *NeurIPS*, 2019.
- Pedregosa, F., et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 2011.
- Liu, B. CS 583 Course Materials. Data Mining and Text Mining, University of Illinois at Chicago, Fall 2025.
- Meka, V. D. CS 583 Research Project GitHub repository: <https://github.com/Vamsi-Dath/CS-583-Fall-2025-Data-Mining-and-Text-Mining>