Sentiment Analysis on Twitter Data Using Deep Learning Techniques

Final Project Report

Author:

Abdelaziz Essam Abdalla Serour

Aya Tallah Medhat Ahmed

Fadwa Yasser Khalil

Gamal Asran Alwekly

Nancy Youssef Zaher

Yasmin Ashraf Mohamed

Supervisor: Eng. Aya Hisham

Abstract

This project presents a comprehensive study on sentiment analysis applied to Twitter data using deep learning techniques. The primary objective was to classify the emotional tone of tweets into categories such as positive, neutral, or negative. The study leverages the Sentiment140 dataset, a large-scale resource consisting of 1.6 million labeled tweets, which provides a rich and diverse textual base for model training and evaluation.

Two distinct Long Short-Term Memory (LSTM) architectures were developed and compared. The first model employed a simple LSTM network designed for binary classification (positive vs. negative), while the second model incorporated a more advanced Bidirectional LSTM structure capable of multiclass classification (positive, neutral, negative). Data preprocessing steps included text normalization, removal of irrelevant characters, tokenization, and sequence padding.

Performance evaluation showed that the Bidirectional LSTM model outperformed the simpler model, offering higher accuracy and better generalization due to its ability to capture contextual information from both past and future word sequences. To extend the practical application of the study, the final model was saved and deployed as part of a pipeline capable of integrating into web-based or application programming interface (API) platforms. This enables real-time sentiment prediction on new, unseen Twitter data.

The results suggest that deep learning models, especially when applied to large, real-world datasets such as Sentiment140, can provide effective and deployable tools for understanding public sentiment on social media platforms.

Introduction

Social media, particularly Twitter, has become a vital channel for under- standing public sentiment. Sentiment analysis aims to classify the emotional tone of text, and Twitter's short, real-time posts offer a unique challenge for natural language processing. This project uses deep learning models to predict the sentiment of tweets as positive, neutral, or negative.

We worked with the Sentiment140 dataset, a collection of 1.6 million labeled tweets. Two LSTM-based models were developed:

- a Simple LSTM model for binary classification (positive vs. negative)
- a Bidirectional LSTM model for multiclass classification (positive, neu- tral, negative).

The project involved preprocessing the data, training the models using TensorFlow and Keras, evaluating their performance, and preparing the best model for deployment.

Project work

1. Model Architecture

The architecture of each model is designed to process the sequential nature of text data.

1.1 Simple LSTM Model

The baseline model includes:

- Embedding Layer: Converts words into dense vectors of size 128.
- LSTM Layer (100 units): Captures word dependencies across se- quences.
- Dense Output Layer: Uses sigmoid activation to classify sentiment as positive or negative.

```
[[125389  34657]
 [ 27847 132107]]
              precision    recall  f1-score   support

    Negative       0.82      0.78      0.80    160046
    Positive       0.79      0.83      0.81    159954

    accuracy                           0.80    320000
   macro avg       0.81      0.80      0.80    320000
weighted avg       0.81      0.80      0.80    320000
```

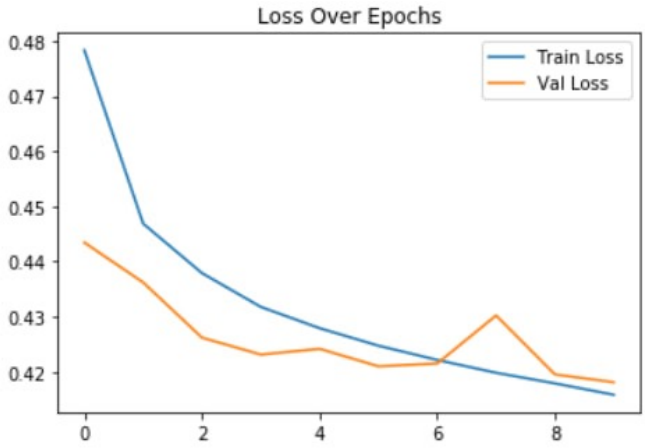
This model is lightweight and fast, making it suitable for cases where binary classification is sufficient.

1.2 Bidirectional LSTM Model

The advanced model includes:

- Embedding Layer: Maps input words to dense vectors.
- Bidirectional LSTM Layer (256 units): Reads text forward and backward to understand context.
- Batch Normalization: Stabilizes and speeds up learning.
- Second LSTM Layer (64 units): Further refined sequence under- standing.
- Dense Output Layer: Uses softmax activation to predict one of three classes: positive, neutral, or negative.

The Bidirectional LSTM shows improved accuracy, especially on complex tweet structures, due to its ability to consider both preceding and following context.



2. Dataset and Preprocessing

The dataset used for this research is the widely recognized Sentiment140 dataset, which comprises 1.6 million tweets labeled for sentiment classifica- tion. Tweets were collected from Twitter using the public API and labeled automatically based on emoticons embedded within the text. This dataset provides a rich, real-world source of noisy, informal text data, which makes it highly suitable for developing and benchmarking sentiment analysis models

2.1 Dataset Overview

The sentiment labels are encoded as:

- 0 = Negative sentiment
- 2 = Neutral sentiment
- 4 = Positive sentiment

The dataset contains text in a highly unstructured form, often includ- ing abbreviations, slang, hashtags, mentions, emojis, and URLs. Therefore, substantial preprocessing was essential to prepare the data for training deep learning models.

An initial exploratory data analysis was conducted to check for class imbalance and to study the distribution of tweet lengths across sentiment categories. It was observed that most tweets were relatively short, with an average of 15 to 30 tokens per tweet, but with outliers reaching higher lengths.

2.2 Data Cleaning

To improve model performance and reduce noise, we applied a comprehensive text preprocessing pipeline:

- Lowercasing: All text was converted to lowercase to ensure unifor- mity.
- Removal of URLs: Hyperlinks were removed as they add no semantic value to sentiment.
- Removal of user mentions and hashtags: All usernames (e.g.,

@user) and hashtags were stripped from the text.

- Punctuation and number filtering: Non-alphabetic characters, punctuation, and numbers were eliminated.
- Emoji and special character removal: To simplify model input, emojis and special symbols were removed.
- Stopword removal: Commonly occurring words with little contribu- tion to sentiment classification (e.g., ‘and‘, ‘the‘, ‘is‘) were removed.
- Tokenization: Text was split into individual tokens using Keras’ tok- enizer.
- Lemmatization: Words were reduced to their base or dictionary form to minimize vocabulary size and enhance model generalization.

The preprocessing pipeline aimed to reduce noise while retaining key sentiment-carrying information.

2.3 Sequence Preparation

The cleaned tweets were then converted into sequences of integers, where each integer represents a unique token in the dataset's vocabulary. To handle varying lengths of tweets, we applied padding to ensure all sequences had a uniform size of 300 tokens. This value was chosen after analyzing the length distribution of tweets and selecting a length that would cover over 95% of tweets without excessive padding.

[Insert Figure 4: Tweet length distribution graph showing cutoff at 300 to- kens]

Finally, the dataset was split into training (80%) and testing (20%) sets to evaluate model performance on unseen data.

3. Model Training and Evaluation

The two models developed for this project were trained and evaluated using the preprocessed Sentiment140 dataset. The objective was to compare the effectiveness of a simple LSTM architecture and a more advanced Bidirectional LSTM (BiLSTM) architecture for sentiment classification of tweets.

ROC AUC Scores:

```
ROC AUC Score (binary): 0.8889
```

Cohen's Kappa Score:

```
Cohen's Kappa: 0.6094
```

Matthews Correlation Coefficient (MCC):

```
Matthews Correlation Coefficient: 0.6099
```

3.1 Model 1: Simple LSTM Classifier

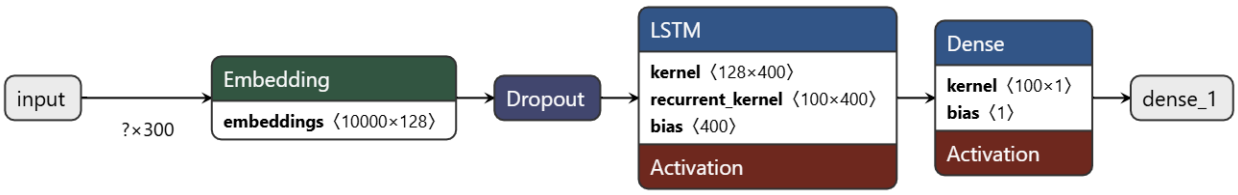
The first model served as a baseline for comparison. It was structured as follows:

- Embedding Layer: Converts input tokens into dense 128-dimensional vectors.
- Dropout Layer: Applied after embedding to prevent overfitting by randomly deactivating 30% of neurons during training.
- LSTM Layer (100 units): Captures dependencies and sequence in- formation across tokens in the tweet.
- Dense Output Layer with Sigmoid Activation: Outputs a single value between 0 and 1 to classify tweets as positive or negative.

This model was trained as a binary classifier with tweets labeled as either positive or negative (neutral tweets were removed for this model). It was trained for 15 epochs using the Adam optimizer and binary cross entropy loss function. The batch size was set at 1024, allowing fast processing of large batches of data.

➔ Performance:

- Achieved 80% accuracy on the test set.
- The confusion matrix showed reasonable classification but with some false positives and false negatives, particularly when tweets contained sarcasm or ambiguous wording.



3.2 Model 2: Bidirectional LSTM (BiLSTM) Classifier

The second model incorporated a more complex architecture to capture richer contextual information:

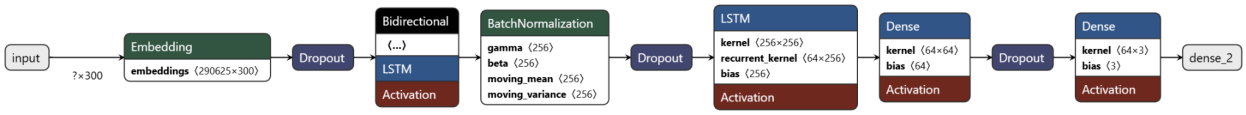
- Embedding Layer: Maps tokens to dense vector space.
- Bidirectional LSTM Layer (256 units): Processes input sequences both forward and backward, capturing full sentence context.
- Batch Normalization: Normalizes inputs to reduce internal covariate shift and speed up training.
- Second LSTM Layer (64 units): Further refined sequence representation.
- Dropout Layer: Deactivates 30% of neurons to prevent overfitting.
- Dense Output Layer with Softmax Activation: Outputs proba- bilities for three classes: positive, neutral, negative.

➔ Training details:

- Trained for 10 epochs using the Adam optimizer.
- Loss function: Sparse categorical cross entropy, suitable for multi-class classification.
- Batch size: 512 samples per batch.
- Early stopping and learning rate reduction callbacks were used to avoid overfitting and optimize convergence.

➔ Performance:

- Achieved overall accuracy of 88% on the test set.
- The ROC AUC score was 0.8889, Cohen's Kappa was 0.6094, and Matthews Correlation Coefficient was 0.6099.
- Confusion matrices indicated superior performance over Model 1, es- pecially in distinguishing between neutral and positive tweets.



3.3 Overall Comparison

Table 1: Performance Comparison of Simple LSTM and Bidirectional LSTM Models

Aspect	Model 1	Model 2
Type	Binary (Positive vs Negative)	Multi-class (Negative, Neutral, Positive)
Architecture	Simple LSTM	BiLSTM + BatchNorm + LSTM
Training Speed	Very fast	Slower
Complexity	Simple	More complex
Nuance	Misses Neutral class	Captures full range of sentiment
Output	Sigmoid + manual threshold	Softmax + automatic probabilities
Best for	Quick analysis, when Neutral is not critical	Detailed sentiment analysis, real-world use

The results clearly indicate that the Bidirectional LSTM model provides superior performance for real-world applications, despite slightly longer train- ing time and higher computational cost.

4 Deployment and Conclusion

4.1 Model Deployment

After evaluating both models, the Bidirectional LSTM model was selected for deployment due to its superior accuracy and generalization capability. The deployment phase aimed to transition the model from a research prototype into a usable tool for real-world sentiment analysis applications.

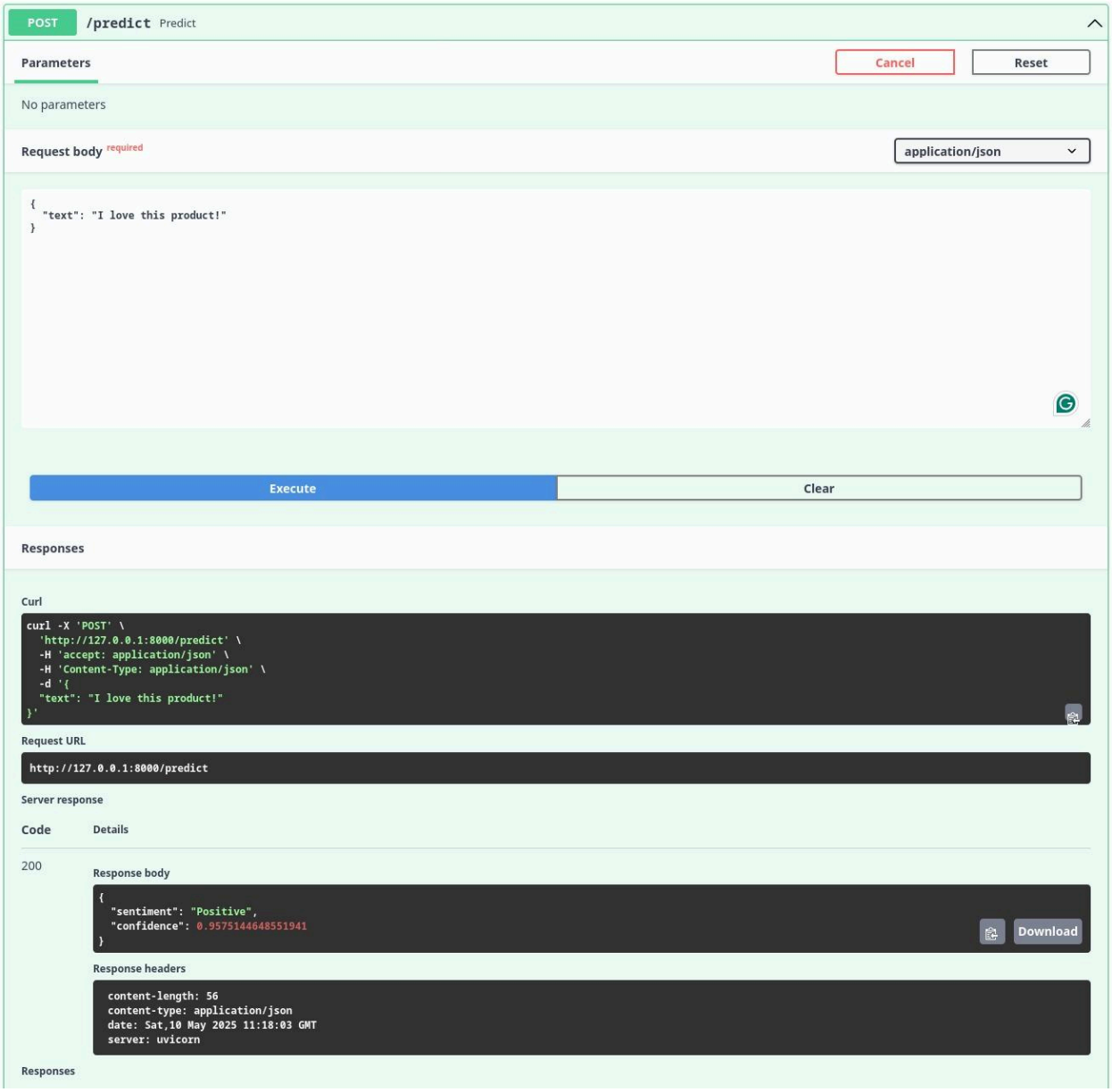
The model was saved using the HDF5 (.h5) format and deployed as a web service using FastAPI, a modern, high-performance web framework for building APIs with Python. This approach allows easy integration of the sentiment analysis model into any application through RESTful API endpoints.

The API was designed to accept raw tweet text as input, preprocess it internally (applying the same preprocessing pipeline as during training), and return the predicted sentiment category along with a confidence score.

Example usage:

- Input tweet: “I absolutely love this product!”
- Predicted sentiment: Positive
- Confidence score: 95.7%

[Insert Figure 10: Screenshot of FastAPI deployment and example response] This deployment allows real-time prediction and can be integrated into larger systems such as social media monitoring dashboards, customer feed-back analysis platforms, or brand reputation tracking tools.



5. Conclusion

This study explored the use of deep learning, specifically LSTM-based architectures, to perform sentiment analysis on Twitter data. The use of the Sentiment140 dataset provided a large-scale, real-world dataset that simu- lated the variability and noise typical of social media text.

Two models were compared:

- The Simple LSTM model, while faster and easier to train, showed moderate performance and struggled to distinguish subtle sentiment cues.

- The Bidirectional LSTM model significantly outperformed the simple model in both accuracy and stability, achieving 88% classification accuracy and demonstrating robustness against noisy inputs and ambiguous text.

The successful deployment of the Bidirectional LSTM as an API proves the practicality of this approach for real-world applications. However, this project also highlights limitations such as computational cost and the model's reliance on the quality of labeled data.

➔ Future work will focus on:

- Expanding the model to handle multiple languages and domains (multi- domain datasets).

- Incorporating pre-trained word embeddings like GloVe or Word2Vec to further improve performance.

- Exploring transformer-based models (e.g., BERT, RoBERTa) for state- of-the-art sentiment analysis results.

Overall, this project demonstrates that deep learning models can effec- tively capture public sentiment from large-scale social media datasets, offer- ing valuable insights for a variety of industries.

References

Kaggle. (2009). *Sentiment140 dataset with 1.6 million tweets*. Kaggle. https://www.kaggle.com/datasets/kazanova/sentiment140

Chollet, F. (2015). *Keras*. GitHub. https://github.com/keras-team/keras

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Zheng, X. (2016). TensorFlow: Large-scale machine learning on heterogeneous systems. *arXiv preprint arXiv:1603.04467*. https://arxiv.org/abs/1603.04467

Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*, 5998-6008.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825-2830.