The Ensemble of Machine and Deep Learning Models for Sub-event Detection in Twitter Streams

Team LLY:
Ziyi LIU[*], Ling LIU[†], Yixing YANG[‡]

[*†‡]These authors contributed equally to this work.

December 12, 2024

Contents

[*]lzy_ecust@outlook.com
[†]ling.liu@ip-paris.fr
[‡]yixing.yang@example.com

1 Introduction

The widespread use of social media during major sporting events offers unparalleled opportunities for real-time event analysis. This project focuses on detecting sub-events in Twitter streams from the 2010 and 2014 FIFA World Cups by developing a machine learning model to classify specific sub-events within one-minute tweet intervals.

The goal is to build a binary classification model that identifies the presence or absence of key football-related sub-events, such as Full-Time, Goal, Half-Time, Kick-off, Own Goal, Penalty, Red Card, and Yellow Card. The dataset comprises tweets from World Cup matches, with each tweet labeled as eventful or non-eventful based on sub-event references.

Our approach includes preprocessing tweet text[2.1], embedding the processed text into fixed-dimensional vectors[2.2], and training various classification models[2.3] to predict sub-event labels.

2 Methodology

2.1 Data Preprocessing

The dataset comprised tweets from World Cup matches, with the key features described in Table[2.1].

Column	Description	DType
ID	Unique identifier combining Match and Period IDs	string
MatchID	Identifier for each football match	int
PeriodID	1-minute time period within the match	int
EventType	Binary label (0/1) indicating sub-event presence	int
Timestamp	Unix timestamp of the tweet	int
Tweet	Text content of the tweet	string

Table 1: Dataset Column Description

Preprocessing is a critical step to prepare raw tweets for machine learning, ensuring data consistency, reducing noise, and standardizing formats. The pipeline includes sequential transformations such as detecting and translating non-English tweets, normalizing text, and removing irrelevant components like URLs or mentions. These steps enhance the data's semantic coherence and simplify subsequent processing.

Figure[1] illustrates the preprocessing pipeline, which systematically refines tweets into clean, tokenized data. This cleaned text is then ready for embedding and modeling.

2.2 Embedding

After preprocessing, tweets were converted into numerical representations using GloVe Twitter embeddings (200-dimensional). Each tweet vector was computed as the average of the embeddings for its tokens. This step captures semantic relationships between words, resulting in a compact and meaningful fixed-length representation that facilitates input into various machine learning models.

2.3 Model Architectures

We experimented with a variety of classification models to identify the most effective architecture for sub-event detection. These models included both traditional machine learning

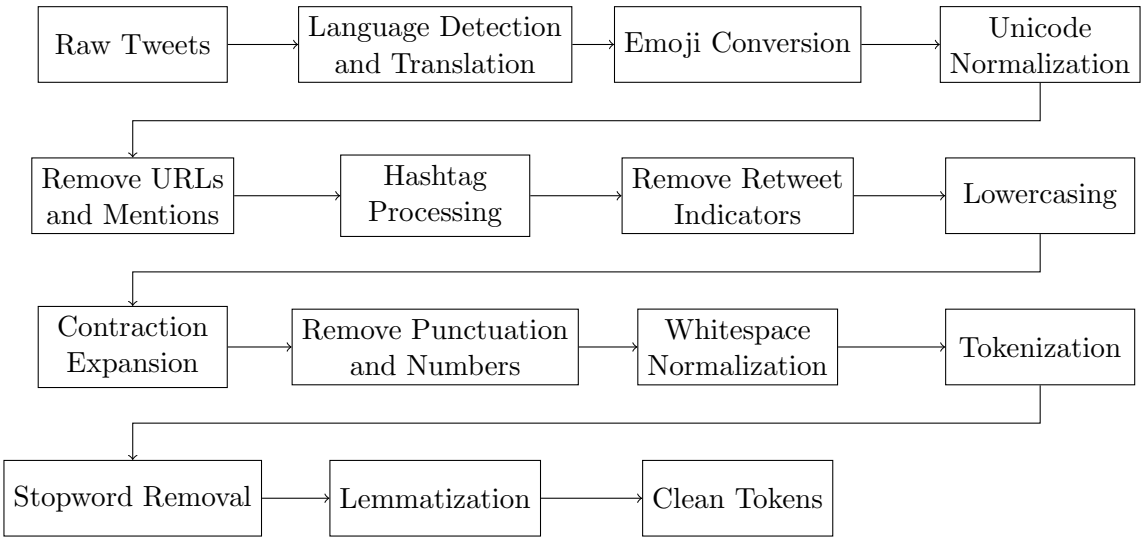

Figure 1: Tweet Preprocessing Pipeline: Sequential steps to transform raw tweets into clean, tokenized data.

approaches, such as Logistic Regression, Multi-Layer Perceptron (MLP), Random Forest, Support Vector Classifier (SVC), and Bagging Classifier, as well as deep learning methods, including Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM). By evaluating these models, we aimed to determine the optimal approach for the task.

3 Experiments

In this section, we present both local test results and official submission outcomes for various models and ensemble configurations.

3.1 Local Test Results

Before submitting predictions to the official evaluation platform, we conducted experiments on a held-out local test set. Table[2] shows a subset of our local evaluations, offering initial insights into model strengths and weaknesses.

Model	**Accuracy (%)**
BaggingClassifier (RandomForest)	79.47
RandomForestClassifier	78.50
MLPClassifier	78.19
CNNBinaryClassifier	74.29
LogisticRegression	72.89
DecisionTreeClassifier	66.97

Table 2: Model Performance on Local Test Set

The local results suggest that the MLP Classifier and Bagging Classifier performed relatively well compared to simpler baselines and certain neural architectures. Models like CNN and LSTM showed potential but did not outperform strong baselines without careful hyperparameter tuning. These preliminary experiments guided our subsequent selection and combination of models.

3.2 Submission Results

After refining hyperparameters and feature representations, we produced multiple submissions. A key finding was that *ensembles consistently outperformed individual models*. By leveraging the complementary strengths of diverse classifiers, ensemble methods can reduce variance and bias, thus enhancing overall predictive accuracy.

Specifically, consider a set of m models $f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})$, each producing a prediction for an input $\mathbf{x}$. In a simple voting scheme, the final prediction $\hat{y}$ is given by:

$$\hat{y} = \text{sign}\left(\sum_{i=1}^{m} f_i(\mathbf{x})\right),$$

where $f_i(\mathbf{x}) \in \{-1, +1\}$ for binary classification. More nuanced confidence-weighted voting can incorporate model-specific weights w_i, where:

$$\hat{y} = \text{sign}\left(\sum_{i=1}^{m} w_i f_i(\mathbf{x})\right),$$

with $w_i > 0$ and $\sum_{i=1}^{m} w_i = 1$. This weighted approach amplifies the influence of more reliable models.

Table 3 provides a sample of our submission results. Notably, our best performing submission employed a voting strategy that combined several base learners, such as MLP, SVC, Logistic Regression, and Random Forest. This ensemble capitalized on the distinct decision boundaries and error patterns of each model, leading to a higher collective accuracy.

Description (Models / Parameters)	**Accuracy (%)**
Ensemble (MLP, SVC, LR, RF)	74.61
Ensemble (MLP, Bagging, CNN, LR, DT)	73.04
Ensemble (MLP, Bagging, CNN, LR, LSTM)	71.48
Ensemble (MLP, Bagging, CNN, LR)	71.48
Ensemble (MLP, Bagging)	72.26
MLP	72.26
Bagging on MLPs	71.88
LSTM	69.92
Bagging on RFs	65.23
CNN	64.84

Table 3: Selected Submission Results on Official Test Set. LR: Logistic Regression, RF: Random Forest, DT: Decision Tree.

3.3 Discussion: Utility of Voting Ensembles

The improved performance of the voting ensemble arises from its ability to integrate the strengths of heterogeneous classifiers, thereby mitigating each model's individual weaknesses. Each component model f_i captures different aspects of the data. By pooling their predictions, the ensemble reduces systematic bias and variance. If one model overfits or struggles with particular patterns, others can offset these weaknesses, improving the final prediction stability.

- **MLP:** Captures complex nonlinearities but may overfit, benefiting from regularization via ensemble voting.
- **RandomForest:** Identifies general patterns and is resilient to overfitting, complementing more specialized models.

- **SVC:** Crafts intricate decision boundaries; combining with simpler models (e.g., Logistic Regression) ensures balanced performance.
- **LogisticRegression:** Offers a stable, linear baseline that tempers the flexibility of more complex models.
- **CNN:** Excels at extracting spatial relationships in feature embeddings.
- **LSTM:** Integrates temporal dependencies, adding sequence-based understanding to the ensemble.

On challenging examples, this collective intelligence ensures that the majority vote aligns with the correct label. Unlike standalone methods, the ensemble harmonizes varied decision boundaries and error distributions, resulting in a more robust and stable predictor. This approach secured our top accuracy of 74.61%.

4 Conclusion

This project demonstrated the feasibility of detecting sub-events in Twitter streams using machine learning techniques. Key findings include:

- Ensemble methods significantly improved classification performance.
- Deep learning models showed promise but were not consistently superior to simpler models.
- Text preprocessing and embedding techniques were crucial for effective classification.

Future work could explore more advanced embeddings, improved ensemble strategies, and the incorporation of temporal features to further enhance sub-event detection capabilities.