EnsLM : Ensemble Language Models for Event Detenction

Shashwat Sharma
Institut Polytechnique de Paris
shashwat.sharma@ip-paris.fr

Abstract

Our approach combines time series features from tweet volumes with text features extracted through ensemble knowledge distillation of multiple language models. The architecture uses GloVe embeddings, RoBERTa, Longformer, and Gemini models to generate prediction probabilities through 5-fold cross-validation. These features feed into different meta-classifiers. Our results show CatBoost achieved 87% accuracy on validation dataset with balanced precision and recall, while TabNet with PCA reduction reached 84% accuracy. The LSTM model showed limited performance with 77.8% accuracy and poor generalization.

1 Introduction

In this study, we address the binary classification challenge of identifying notable events in football matches from tweets. We refine tweet data through preprocessing to ensure relevance for analysis, drawing on established literature. The problem we are called to solve is a binary classification problem. Specifically, the goal is to develop a predictive model that identifies whether a notable event occurred within a given one-minute interval of a football match.

2 Data

2.1 Data Description

This dataset contains information about tweets related to football matches. Each match is divided into periods, and each row in the dataset includes: the text of the tweet, the match it refers to (MatchID), the period of the match during which it was posted (PeriodID), and the specific time it was tweeted (Timestamp) measured to the second. Lastly, the label of each row indicates whether a sub-event occurred during the period in which the tweet was posted.

2.2 Data Preprocessing

To ensure the integrity and relevance of the dataset derived from Twitter, several preprocessing steps were implemented to reduce noise and focus on meaningful content. Retweets, constituting approximately 65% of the data, were removed along with mention-based replies, which are typically irrelevant to the match period being analyzed. Tweets containing hyperlinks were filtered out to reduce external content influence and spam, as suggested in previous literature (**?**, 9) Duplicate tweets were eliminated to ensure analysis is based on unique textual content. Data was confined to PeriodID values up to 129, as the evaluation set does not include matches with a PeriodID beyond this number. Using PeriodID as a feature integrates time-series elements into the classifier while preserving the dataset's chronological structure. Team names were removed using a regular expression to minimize team-specific bias and ensure the data is team-agnostic. This allows models to focus on subevent-indicative words rather than the teams involved. The text was further processed converting characters

Preprint. Under review.

to lowercase, stripping punctuation, and eliminating numbers. Tokenization was performed to break text into individual words, followed by standard natural language processing techniques, including stopword removal and lemmatization. Lemmatized words were filtered to retain only significant terms that are not stopwords. The finalized preprocessed text was consolidated into clean strings with extra spaces removed, ensuring the data is ready for subsequent analytical processes.

3 Feature Engineering

3.1 Time Series Features

First, We extracted both temporal and statistical features from the tweet-level dataset to capture the dynamics of social media activity during matches. We computed basic volumetric features including raw tweet counts and relative frequencies normalized within each match period. To capture temporal patterns, we implemented rolling statistics with a 3-period window, calculating moving averages and standard deviations of tweet volume. We also quantified the momentum of social media activity through period-over-period percentage changes and incorporated relative volume rankings via period percentiles. These engineered features were designed to capture both absolute and relative fluctuations in tweet activity, aligning with established methodologies in social media event detection literature.

3.2 Text Features

To extract text features, we tried ensemble knowledge distillation by generating prediction probabilities through 5-fold cross-validation across four distinct language models. For each model architecture (GloVe with attention, Twitter-RoBERTa, Longformer, and Gemini), we obtained five sets of prediction probabilities, resulting in 20 meta-features per instance. The cross validation is applied across distinct Matches, mitigating overfitting and thus proving more generalised predicition probability to detect subevent. The diversity in model architectures - from Twitter-specific word embeddings to long-range contextual understanding - allows each fold to capture different aspects of the underlying text patterns, creating a rich set of meta-features that encapsulate various perspectives on the data. These prediction probabilities serve as high-level semantic features that s used by the final Meta Model for classification, effectively distilling the knowledge from multiple specialized language models into a compact representation.

4 Methodology

We use a multi-staged approach to detect subevents by combinging temporal tweet patterns with language model features, where the base language models generate prediction probabilities creating a rich set of meta-features. These features, along with time series characteristics, are then fed into a Meta-classifiers, effectively creating an ensemble that uses both the temporal tweet activity and the semantic understanding of tweet content from the base model detection performance.

4.1 Base Language Models

GloVe Embeddings on Words with Attention We utilized GloVe's 200-dimensional Twitter embeddings as a foundational representation of tweet text. These embeddings are pre-trained on a large corpus of Twitter data, capturing domain-specific semantics such as informal language, slang, and hashtags. To enhance the importance of relevant terms within each tweet, a word-level attention mechanism was applied, weighting words based on their contribution to the task. This approach leverages the compact, domain-relevant nature of GloVe vectors while allowing the model to dynamically adjust its focus to critical parts of the input text.

RoBERTa-Based Features on Sub-Periods Tweets were processed using CardiffNLP's Twitter-RoBERTa base model, a transformer model fine-tuned on extensive Twitter datasets. This model provides contextual embeddings that capture complex dependencies within and across tokens in 512-token chunks. Its fine-tuning of Twitter data ensures the effective handling of informal language, hashtags, and emojis, making it highly suitable for tweet classification tasks and a nuanced understanding of short texts in this domain.

Longformer Features on Periods The Longformer model from AllenAI was applied to process extended sequences of up to 4096 tokens, accommodating entire periods of aggregated tweets. Its architecture, based on efficient attention mechanisms, enables the model to capture global and local contextual information effectively. By handling longer input spans, Longformer facilitates the integration of broader temporal and thematic context, which could provide a better representation for detecting a subevent across the whole period rather than just a single tweet.

Gemini Embedding Features on Periods The Gemini language model was employed to generate dense embeddings for tweets, which were further refined using a word-level attention mechanism. Gemini's advanced language representation capabilities, trained on diverse datasets, allow it to model complex semantic and syntactic relationships. The combination of Gemini embeddings and attention enables the system to prioritize meaningful parts of the text while leveraging the rich representational power of the model for downstream tasks.

4.2 Training Strategy

The training strategy for every base model is similar, using a 5-fold cross-validation approach, where the data is split based on unique MatchIDs to prevent any potential data leakage between related tweets. For each fold, the model is trained for upto 50 epochs with a batch size of 8 or 16, using the Adam optimizer and binary cross-entropy loss. To combat overfitting, the training process incorporates early stopping with a patience of 10 epochs and learning rate reduction with a patience of 3 epochs. The model's performance is monitored on a validation set, and training stops when no improvement is observed, we finally select the best model on the eval accuracy metric.

4.3 Meta Classifiers

CatBoost The CatBoost classifier emerged as the top performer among meta-classifiers, achieving 87% accuracy on the validation set with notably balanced precision and recall metrics (precision: 0.92/0.83, recall: 0.79/0.94 for classes 0/1 respectively). The final model utilized a moderate learning rate of 0.1, tree depth of 6, and L2 regularization of 3, with early stopping after 20 rounds of no improvement to prevent overfitting. These parameters, combined with CatBoost's gradient boosting approach and built-in handling of categorical features, effectively captured the complex relationships between model predictions and temporal features, resulting in robust F1-scores of 0.85 and 0.88 for the respective classes.

XGBoost The XGBoost classifier demonstrated strong performance with an 83% accuracy on the validation set, showing balanced precision and recall metrics (precision: 0.83/0.82, recall: 0.79/0.86 for classes 0/1 respectively). The final model employed a moderate tree depth of 6, learning rate of 0.1, and 200 estimators, along with careful regularization through subsample and 'colsamplebytree' parameters both set at 0.8 to prevent overfitting. XGBoost's implementation of gradient boosting with these parameters effectively handled the feature interactions, achieving consistent F1-scores of 0.81 and 0.84 for the respective classes. The model's balanced performance across both classes, coupled with early stopping after 20 rounds, suggests it successfully captured the underlying patterns while maintaining generalization capability.

TabNet TabNet's implementation showed interesting results in two configurations. Initially, without dimensionality reduction, the model achieved 80% accuracy on the validation set with balanced metrics (precision: 0.79/0.81, recall: 0.78/0.81 for classes 0/1). However, recognizing TabNet's potential sensitivity to multicollinearity in our feature space which combines multiple model predictions, we applied PCA reduction to 10 components. This modification significantly improved performance, reaching 84% accuracy with enhanced precision and recall (0.84/0.85 for both classes), and achieving early stopping at epoch 16 with an AUC of 0.907. The model architecture used width parameters of 64 for both decision prediction and attention embedding layers when we trained it without PCA, with 5 steps and a feature selection parameter of 1.3. With PCA we reduced the width parameters to 16 and 3 steps. The learning process was optimized using Adam optimizer with an initial learning rate of 2e-2 and ReduceLROnPlateau scheduler for adaptive learning rate adjustment. This improvement after PCA suggests that TabNet's feature selection and attention mechanisms work more effectively with reduced, orthogonal feature spaces.

Table 1: Summary of Classifier Performance

Model	Accuracy (%)	F1-Score (Class 0/1)	Key Notes
CatBoost	87	0.85 / 0.88	Balanced precision/recall
XGBoost	83	0.81 / 0.84	Effective regularization.
TabNet	84	0.84 / 0.85	Improved with PCA.
LSTM	77.8	0.63 / 0.83	Biased to Class 1

LSTM The LSTM model implemented a sequential architecture with three LSTM layers (256, 128, and 64 units) followed by dense layers, utilizing batch normalization and dropout for regularization. The model processes sequences of length 10, created by sliding windows over match periods, with zero-padding for test sequences. While the architecture theoretically should capture temporal dependencies in our feature space effectively, in practice we achieved only 77.8% cross-validation accuracy. More concerning was the model's strong bias toward predicting class 1 in the test set, suggesting poor generalization. Given these suboptimal results compared to our other meta-classifiers, we decided not to pursue further optimization of this approach.

5 Conclusion

The approach combines tweet volume patterns and ensemble knowledge distillation from multiple language models (GloVe, RoBERTa, Longformer, Gemini) to detect notable football match events. Among the meta-classifiers, CatBoost achieved the highest accuracy at 87% with balanced precision and recall, while TabNet with PCA reduction reached 84%. The LSTM's sequential approach proved less effective with 77.8% accuracy and showed poor generalization, suggesting that temporal feature processing might not be as relevant in detecting a subevent within a period interval.

References

[1] Brochier, R., Guille, A., & Velcin, J. (2019). Global Vectors for Node Representations. In *The World Wide Web Conference* (WWW '19), pp. ACM. doi:10.1145/3308558.3313595.

[2] Loureiro, D., Rezaee, K., Riahi, T., Barbieri, F., Neves, L., Anke, L. E., & Camacho-Collados, J. (2023). Tweet Insights: A Visualization Platform to Extract Temporal Insights from Twitter. arXiv preprint, arXiv:2308.02142. Retrieved from https://arxiv.org/abs/2308.02142.

[3] Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The Long-Document Transformer. arXiv preprint, arXiv:2004.05150. Retrieved from https://arxiv.org/abs/2004.05150.

[4] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2019). CatBoost: Unbiased Boosting with Categorical Features. arXiv preprint, arXiv:1706.09516. Retrieved from https://arxiv.org/abs/1706.09516.

[5] Arik, S. O., & Pfister, T. (2020). TabNet: Attentive Interpretable Tabular Learning. arXiv preprint, arXiv:1908.07442. Retrieved from https://arxiv.org/abs/1908.07442.

[6] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD '16), pp. 785-794. ACM. doi:10.1145/2939672.2939785.

[7] Vennerød, C. B., Kjærran, A., & Bugge, E. S. (2021). Long Short-term Memory RNN. arXiv preprint, arXiv:2105.06756. Retrieved from https://arxiv.org/abs/2105.06756.

[8] Meladianos, P., Xypolopoulos, C., Nikolentzos, G., Vazirgiannis, M. (n.d.). An Optimization Approach for Sub-event Detection and Summarization in Twitter. LIX, École Polytechnique, France; Athens University of Economics and Business, Greece.

[9] Bekoulis, G., Deleu, J., Demeester, T., Develder, C. (n.d.). Sub-Event Detection from Twitter Streams as a Sequence Labeling Problem. Ghent University - imec, IDLab, Department of Information Technology. firstname.lastname@ugent.be.