Final Project Report: Sub-Event Detection

Yann Baglin-Bunod & Violette Gontran
CSC 554: Machine and Deep Learning
kaggle: ybaglinbunod@gmail.com, violettegontran02@gmail.com

December 2024

1 Introduction

The objective of this project is to detect significant events during football matches by analyzing tweets sent during the football match. The task ahead involved determining relevant information to classify subevents such as goals, penalties, red cards, yellow cards, and own goals. Additionally, we aimed to refine the placement of half-time and full-time periods, which varied slightly across matches.

2 Dataset Description

The dataset consists of tweets annotated with binary labels: 1 for notable events and 0 otherwise.

From the evaluation, words like 'goal' and 'win' appeared with similar frequency in EventType 0 and 1 (appendix 6). To refine this, we identified around 50 words where the frequency in EventType 1 was at least 70% greater than in EventType 0 (appendix 8). Additionally, tweets in EventType 1 were generally more frequent than in EventType 0, except for one match in the training data. However, these differences varied across matches, making them insufficient standalone predictors.

Specific patterns emerged for subevents: periods 57-62 aligned with half-time, while the final 8 periods before match-end (around period 129) often indicated full-time, even during overtime (appendix 7). These trends correspond to half-time, kick-off, and full-time events but require further exploration to distinguish subevents: 'goal', 'own-goal', 'penalty', 'red card', and 'yellow card'.

To further analyse tweet characteristics, we adopted the methodology outlined by Dimosthenis Antypas and Asahi Ushio in their research on Twitter topic classification [2]. The Ratio Normalized is defined as the ratio of uppercase to lowercase letters, and the Measure of Textual Lexical Diversity (MTLD) provides insights into the vocabulary richness of each EventType. MTLD, as introduced by McCarthy and Jarvis (2010)[1], is a robust indicator of lexical diversity, capturing the variability and sophistication of vocabulary in tweets. Investigating these characteristics to our dataset across the two EventTypes revealed no significant differences, underscoring the need for a deep learning model to effectively capture the distinctions between the two types (Figure 1).

Class	Percentage	length	#	Ratio_Normalized	@	emoji	punctuation	MTLD	Tweet Count
0.0	43.53	87.96	1.28	75.07	0.84	0.11	6.77	108.14	2200981.0
1.0	56.47	86.73	1.57	104.96	0.78	0.1	6.9	98.91	2855069.0

Figure 1: Summary of tweet characteristics by EventType. Ratio_Normalized is the ratio of upper to lower case letters. MTLD is the Measure Textual Lexical Diversity. All values (except percentage and tweet count) are normalized over the number of tweets.

3 Data Preprocessing and Feature Extraction

To prepare tweets for classification, we implemented a preprocessing pipeline. Text was normalized by converting it to lowercase, removing punctuation and numbers, and splitting it into words. Stopwords were removed using NLTK's English stopword list, and words were lemmatized to their base forms to ensure consistency.

For feature extraction, we used the GloVe embedding model, pre-trained on Twitter data, to capture semantic and syntactic relationships. A short attempt to use the Each word in the tweet was mapped to its embedding, with key terms such as 'goal' and 'penalty' receiving additional weight by multiplying their embeddings by a factor of 5. The final tweet representation was obtained by averaging the embeddings of its words. If no words were present in the GloVe vocabulary, a zero vector was used as a fallback. This process emphasized important keywords while maintaining a robust numerical representation for all tweets.

4 Model Selection and Tuning

Baseline models included a Dummy Classifier and Logistic Regression, which served as references for comparison.

4.1 Proposed Models

1. Keyword Frequency and Tweet Volume: Predefined keywords (e.g., 'goal', 'penalty') and tweet volume trends during notable events were used as features. It was almost a coin flip output. Output accuracy: **55%**.

2. Normalized Averages for Event Periods: Features such as tweet counts, top 50 keyword counts, and average tweet lengths were grouped by `MatchID` and `PeriodID`, and normalized by dividing each value by the average for the correspond-

ing match. This normalization allowed comparisons across different matches and EventTypes by accounting for variations in match-specific tweet volumes. Output accuracy: **58%**.

3. Support Vector Machine: SVM built using metrics. The extracted features include PeriodID, TotalTweets per period, TotalWords words per period, Event1WordCount (count of EventType 1-related words), AvgEmbedding (avg embeddings on GloVe). Finally, RelChangePrev and RelChangeNext measured relative changes in features compared to previous or next periods (1, 5, and 10 periods). This method was quickly discarded. Output accuracy: **46%**

4. k-Nearest Neighbors (k-NN): Tweets were represented by average word embeddings and classified using k-NN. Initial accuracy (using average embeddings) : **42%**, improved to **62.5%** (using individual embeddings). Issues with this method: very long evaluation time.

5. Convolutional Neural Networks (CNN):

1. A 1-D input CNN (1 x 200) was built using GloVe embeddings padded or truncated to fixed lengths, with weighted embeddings for EventType 1 words. Output accuracy: **64%**.

2. Two 2-D input CNN (16 x 200) was built using GloVe embeddings, inspired by the structure described in Kim's work on Convolutional Neural Networks for Sentence Classification [3].

   1. First rendition: taking each *individual* tweet, converting them into 16x200 matrices and training a CNN. A problem of time was encountered because for converting the 5 million training tweets. Output accuracy using 1 million tweets from training and all tweets from evaluation.
      Output accuracy: **58%**

   2. Second rendition: taking *averaged* embeddings per period, creating 16x200 matrices and training a CNN.
      Output accuracy: **67%**

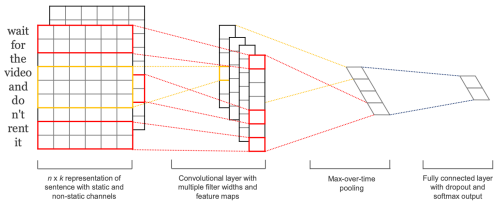

Figure 2: CNN architecture for sentence classification. Reproduced from [3].

6. Long Short Term Memory neural network (LSTM): Tweets were represented by average word embeddings and classified using a LSTM. Initial accuracy: **68%**, improved to **69.5%** after tuning.
The idea of using a LSTM neural network comes from the article published by the department of Information Technology of Ghent University. Time is an important aspect of the problem and the other models we used did not take this in account. This is why this approach seems promising at first sight. We tried different architectures (Figure 3) and number of epochs of training and the best results which enables us to obtain **69.5%** of accuracy was the following:

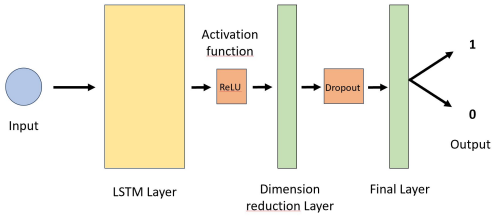

Figure 3: Architecture of the RNN with the best output accuracy

The hidden parameters which gave the best results were: hidden dim = 128, dropout = 0.5,lr = 0.001, batch size = 16.
We also tried to use a Bidirectionnal LSTM to take in account both events from the past and the future. Event if this idea seems richer and promising we did not manage to get an output accuracy as high as with the simple LSTM. The best submission we got with this method had an accuracy of **67.6%**.

7. Transformer: Tweets were represented by average word embeddings and classified using a Transformer. Output accuracy: **72.7%**.
The first transformer we used had the following architecture and gave us the best accuracy (**72.7%**). After this promising result we tried to optimize the hyperparameters (number of layers, number of heads, hidden dimension, learning rate and dropout) by using the library Optuna. After this optimization we did not manage to get a better accuracy than before even with the optimized values of hypermarameters 4.

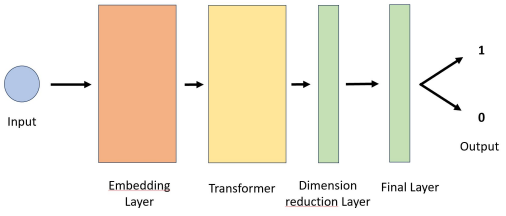

Figure 4: Architecture of the Transformer with the best output accuracy

We tried to understand this phenomena by using cross validation method (Figure 5) either for the best accuracy parameters and the optimized ones. The results showed us a trend: the accuracy on test subset was very variable in the case of the first transformer we tried, it was not the case for optimized parameters for which the accuracy on test set were almost always higher and whith much less

variation. We can conclude that the model with optimized parameters is less depending of the training and testing set. It is more stable. This crossfolding test is described on the following figure:

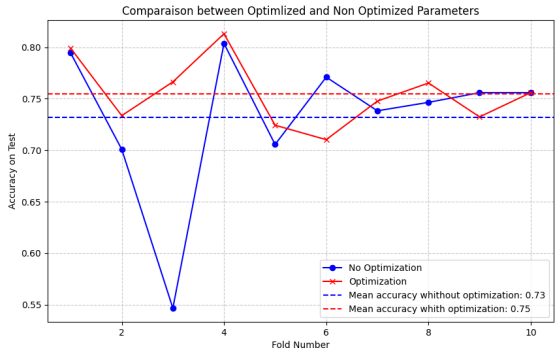

Figure 5: Comparison of cross-folding results for different models.

Finally, we tried to make a prediction on the evaluation test with all the models trained during the cross-folding step. At the end we took the EventType which appears the most in all predictions. We obtain an accuracy of **68.4%**. By only taking the model which the best accuracy on the testing subset, we got an accuracy of only **64.5%**. This is why aggregating relsults from different models seems to be efficient. Here we only trained 10 models but it would be interesting to try which more.

5 Further Work

Future improvements could and explore new methods to enhance performance. Training the 2D CNN on all 5 million tweets preprocessed into individual 16x200 matrices may capture granular patterns missed by averaged embeddings with higher computational demands. Trying embeddings like BERTweet, TimeLM-19, and TimeLM-21, could provide richer semantic representations tailored to this dataset. Additionally, ensemble learning combining CNNs, LSTMs, and Transformers could leverage the strengths of different models for better accuracy. Lastly, data augmentation techniques such as paraphrasing or back-translation could increase dataset diversity and improve generalization.

6 Conclusion

This project marked our initial attempts to really understand the dataset and iteratively improve our machine learning models for sub-event detection. Starting with heuristic-based algorithms, which achieved a modest accuracy of **58%**, we recognized their limitations in capturing nuanced patterns in the data. To address this, we incorporated GloVe embeddings in subsequent models, ensuring a richer representation of semantic and syntactic relationships. Progressing through k-NN, CNN, and LSTM models, we gradually refined our approach, ultimately achieving the best accuracy of **72.7%** using a Transformer model. This iterative process underscored the importance of experimenting with diverse methodologies to optimize performance and highlighted the potential of advanced deep learning techniques in this context.

References

[1] Philip M. McCarthy and Scott Jarvis. 2010. MTLD, vocd-D, and HD-D: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior Research Methods*, 42(2):381–392.

[2] Dimosthenis Antypas, Asahi Ushio, Jose Camacho-Collados, Leonardo Neves, Vítor Silva, and Francesco Barbieri. Twitter Topic Classification. *Cardiff NLP, School of Computer Science and Informatics, Cardiff University and Snap Inc.*, 2021.

[3] Yoon Kim. Convolutional Neural Networks for Sentence Classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Association for Computational Linguistics, Doha, Qatar, 2014. URL: `https://aclanthology.org/D14-1181`, DOI: 10.3115/v1/D14-1181.

A Appendix A: Additional Figures and tables

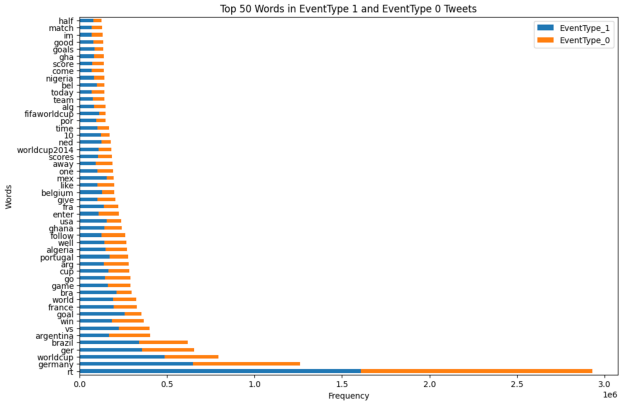

Figure 6: Frequency of the top 50 words in each EventType. Each word's frequency is normalized based on its occurrence across EventTypes, providing insight into distinctive patterns.

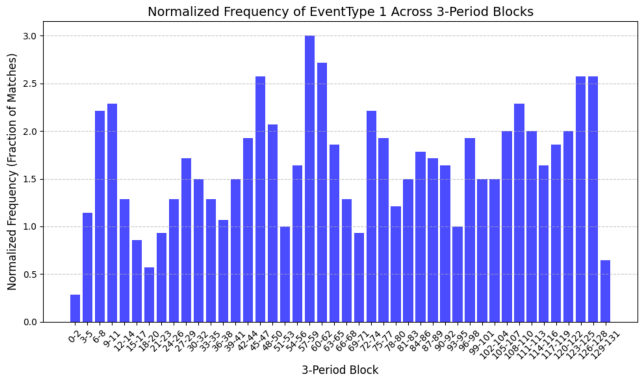

Figure 7: Bar graph showing the occurrence of label 1 (EventType 1) by every three periods, averaged over all training data. This visualization highlights the temporal distribution and trends of EventType 1 across the dataset.

bra	mex	fifaworldcup	bel	joinin
cmr	chi	au	esp	group
lead	round	cro	crc	ht
col	mexico	ft	top	villa
eliminated	finish	porgha	sui	boye
qualified	congratulation	thank	foxsoccer	ausned
john	finally	cmrbra	tear	honduras
neymarjr	congrats	dutch	champion	thanks
easportsfifa	geniusfootball	cromex	bench	kor
uru	strike	scorer	svn	gd

Figure 8: Table displaying the top 50 words with a frequency of 70% or higher in EventType 1. This table highlights the most prominent terms that are highly representative of EventType 1.