Empirical Evaluation for Cricket CommentaryDecoder

SAHIL BODKHE
ssbodkhe@uwaterloo.ca

University of Waterloo

Nandhini Rajasekaran
University of Waterloo

Chinmay Nagesh
University of Waterloo

Research Article

Keywords: Sense Disambiguation, LSTMs, GRUs, SVM, Logistic Classifiers, LightGBM

Posted Date: December 27th, 2024

DOI: https://doi.org/10.21203/rs.3.rs-5712957/v1

Additional Declarations: No competing interests reported.

Empirical Evaluation for Cricket Commentary Decoder

Sahil Bodkhe, Nandhini Rajasekaran, Chinmay Nagesh
Systems Design Department
University of Waterloo, Canada
{ssbodkhe, n2rajasekaran, cbnagesh}@uwaterloo

***Abstract*—Cricket is not just a sport but a global phenomenon, attracting billions of fans and generating immense revenue through broadcasting rights, streaming platforms, and fantasy sports applications. With the advent of professional leagues like the Indian Premier League (IPL) and Big Bash League (BBL), cricket has become a data-rich and analytics-driven game. Real-time event classification from live commentary is crucial for powering these ecosystems, from enhancing viewer experiences to enabling fantasy leagues and predictive analytics. This project aims to develop a real-time cricket commentary decoder to classify events accurately from unstructured and informal textual commentary. Cricket commentary often includes colloquial language, ambiguous phrases, and context-dependent expressions, which traditional rule-based Natural Language Processing (NLP) techniques struggle to handle effectively. The solution incorporates innovative strategies to address sense disambiguation by intentionally removing clear event-indicative keywords, thereby simulating real-world complexity. By tackling the limitations of rule-based systems such as their over reliance on explicit keywords and inability to handle ambiguous contexts, the system achieves robust event classification for runs, outs, extras, and dots game actions. Designed to seamlessly integrate with real-time sports tracking applications, the decoder enables dynamic updates and API calls. By balancing computational efficiency with classification accuracy, this project not only addresses the technical challenges of decoding live commentary but also supports the growing role of data and analysis in the modern cricket ecosystem. To overcome these challenges, the proposed system employs advanced machine learning methods, including LSTM, GRU, SVM, Logistic classifiers, and boosting techniques like LightGBM.**

***Index Terms*—Sense Disambiguation, LSTMs, GRUs, SVM, Logistic Classifiers, LightGBM.**

I. Introduction

The topic is real-time event classification from unstructured textual cricket commentary. The focus is on developing machine learning techniques to interpret informal, ambiguous commentary and classify key cricketing events like runs, outs, and extras. Cricket is a global watched sport with over billions of fans, supported by advanced streaming companies, broadcasting, and fantasy sports applications. With the involvement of top global investors and giant broadcasters offering the second highest broadcasting rights the rise of franchise cricket in the shorter formats have made the game more advance in terms of competition thereby boosting new advancements in the game through data-driven approaches be it the IPL or International stage cricket is increasingly reliant on real-time analytics to enhance viewer engagement, provide actionable insights, and support sports management. Right from decision review systems, hotspot technology, stump mics with approximately 50 cameras covering every angle of the game and audience both have came a long way and its for sure that cricket is no more a spectator sport as fans nowadays are more analytical, practical and statistics oriented so majority of efforts currently even apart from strategic team planning process are spent on appeasing the fans with quality analytical content via different data visualizations presenting the data. The most important thing which clicks here is availing the data at right time for which speed, accuracy and resources required should be scalable, reliable especially during live matches so anything automating this with the required properties will prove the best optimal solution. On this basis we laid the foundation of our project to build a commentary decoder which can utilize live commentary feed to be transcribed with Google API to our model for predicting the event and update the score and statistics for high availability of data for various stakeholders like Media, Commentary box, Broadcast streams Production room, Team rooms, Cric-info apps for different purposes. However, extracting structured data from unstructured commentary remains a significant challenge due to its informal and context-dependent nature. Accurate classification of cricket events can drive innovations in real-time sports tracking and predictive analytics. Currently most of the research work has been around using visual approach to tracks each ball event which faces lower accuracy as well high latency and load due to image related processing and output generation and data collection costs. So precisely aiming for solutions which can get the job done in less time we found the audio to text as the best possible way due to high availability of reliable quality API for transcribing and then modeling textual data to generate output quickly as it just predicts events rest the software part deals with triggering of appropriate API based on the predicted event for updates.

Although it is more simpler than visual approach the catch here lies in sense ambiguity which highlights why it is difficult for rule based classifiers or regular expressions to fail. As in our experiment almost 50% of the data got misclassified which made us arrive at a decision to mask off or eliminate the keywords as the context is at stake. The involvement of keyword marks an event but the context or choice of words around actually decides whether the event happened or failed to happen or was just a possibility been mentioned or may

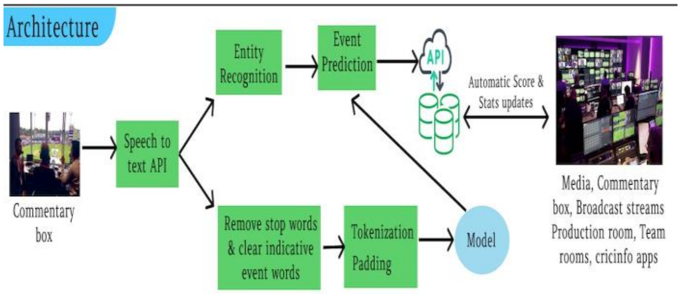

Fig. 1. Proposed solution. High level Architecture of the proposed solution.

be have completely different context. For example "bowled wide down the leg" in cricketing sense implies it as nominal type of delivery called wide but "bowled wide outside off" implies wide as an adjective for the ball delivered which is a fair delivery as completely changes the updates to the database. So by eliminating key event indicator words we started experimenting well renowned methods discussed in the background section as well as evaluated few extra ones too for studying their limitation on this problem approach.

So based on this we evaluated diverse techniques and their results after a lot of experimentation and majorly identified the same ones which we discovered through our literature review in some different sports or similar context but different purpose. The novelty and strength of our work is focused around scalable optimized low latency solution and the most important thing is we are modeling categories like extras and dots which is rare and quit effectively seems complex to disambiguate. So the next section will discuss the prerequisite knowledge which was gathered to plan our approach for different stages and techniques while dealing with this problem. Rest the solutions analyzed will further in depth drive the empirical evaluation, new findings and loopholes discovered, giving a precise conclusion and better knowledge for further research work.

II. Background

It is fascinating to see the progress of sports technology especially in terms of the analytics in cricket as a sport but its also worth noting that most of the progress has been the need of the recent hour. It is definitely rapidly growing but the amount of quality work produced in different areas of cricket are way too little then one can imagine. There are endless possibilities to try hands on for experimentation but getting success to achieve a level of acceptance has been pretty difficult in this sport. As there are so many rules, decisions, considerations formats and style of game play that its an ocean full of opportunities to discover new things but the level of validity varies on lot of factors as this is a century old game but at the same time its expanding it's wings as newly eye catching sport while growing rapidly around all major continents having their stakes and hosting different International events be it recent ICC T20 world Cup in USA and Caribbean Islands or the inclusion of cricket in Los Angeles Olympics 2028.

Based on this here are some of the recent highly relative work which we identified as a good fit for narrowing down possibilities for further detailed exploration. The problem identified by Dixit et al. [1] for cricket ball by ball outcome classification achieved great max results of almost around 80% consisting of three different architectures where the approach focuses on predicting the outcome given the sequence of frames using Convolutional Neural Networks, LSTMs to predict 'Run', 'No Run', 'Boundary', 'Wicket' which are almost similar to our categories except the fact that we are tackling 'Extras' category instead of Boundary as in our case it's just a subset run based on our application aim. The work shows that training a Long-Term Recurrent Convolutional Network using a pre-trained VGG16Net can show great performance on video-based tasks of classifying outcomes. Followed by this a similar attempt by Kumar et al. [2] demonstrated an approach with LSTM followed by CNNs with Adam optimizer achieving accuracy around 70% but stating an important fact that is truly evident in our approach too that temporal dependencies do play an important role in the task of correct outcome classification.
Later the work by Balaji et al. [3] worked on the same previous work categories but worked on the angle of textual commentary and masking of clear indicative event words making it the benchmark work to compare against not completely but to a great extent. In their work they employed techniques like TF-IDF, masked by key words and ultimately using Random forest Classifier achieved an accuracy of 84% and F1 score of 83.6%. The work by Abideen et al. [4] shows an encoder-decoder approach by extracting features using VGG-16 followed by LSTMs to generate commentary based on identified events. Minard et al. [5] demonstrates how soccer commentary can be effectively utilized to classify soccer events, using SVM, achieving an F1 score of 77.25.
Overall, all these approaches truly show what roles they play and it is significantly evident that LSTMs and Random Forest alongside SVMs and other Neural Networks like GRU's as they are similar in nature so these are our main approaches to target.
Miraoui et al. [6] explored the use of TF-IDF embeddings and BERT transformers for sentiment analysis in sports commentary. They developed an SVM model using TF-IDF embeddings, achieving an F1 score of 0.85, outperforming the baseline model [7]. Furthermore, they fine-tuned a BERT transformer model, which achieved an average accuracy of 92%, making it the best-performing model for their task.
Rest for transcribing the audio in text we can use Chiu et al. [8]. Speech-to-Text API developed by Google as training a speech recognition model is not a feasible option and will be better to go with pretrained model as it proves highly reliable, optimal and time saving at the same time. Semwal et al. [9] proposed a solution for recognizing and classifying different types of bat shots in cricket, leveraging state-of-the-art techniques like saliency, optical flow, and Deep Convolutional Neural Networks (DCNN). Their model achieved an accuracy of 83.098% for right-handed shots and 65.186% for left-handed shots. Sports are unique in their respective stochastic

nature, making analysis, and accurate predictions valuable here.By leveraging the diversity of individual models and their collective wisdom, ensemble methods often outperform single models, yielding more accurate and robust predictions. Daniel et al. [10] developed six models and compared, a LightGBM, a XGBoost, a LightGBM (Contrastive Loss), LightGBM (Triplet Loss), a XGBoost (Contrastive Loss), XGBoost (Triplet Loss). It is clear LightGBM is the most effective model in ranking. Jingdi et al. [11] worked on the analysis on the momentum of tennis match and used LightGBM to evaluate the performance of our model and use SHAP feature importance ranking and weight analysis to find the key points that affect the performance of players. Subramaniyaswamy et al. [7] in his work with aimed towards increasing the efficiency of commentary for the end users proposed a solution to provide relevant data to the commentators during a match. A study conducted by ALZAMZAMI et al. [12] proposed to build a general multi-class sentiment classifier using Domain-Free Sentiment Multimedia Dataset (DFSMD). Based on the proven capabilities of Light Gradient Boosting Machine (LGBM) in dealing with high dimensional and imbalance data, trained an LGBM model to recognize one of three sentiments of tweets: positive, negative, or neutral. Essa et al. [13] proposed a novel hybrid fake news detection system that combines a BERT-based (bidirectional encoder representations from transformers) with a light gradient boosting machine (LightGBM) model. The proposed solution solution by Hashmi et al. [14] for the football event classification proposes a Convolutional Autoencoder pipelined with multilayered Extreme Learning Machine giving results of 79.33% accuracy demonstrating dimensionality reduction with fast computation in visual based classification for 4 target events.

III. Method

A. *Dataset Preparation*

The data set used for this project was sourced from Kaggle [15] and originally contained approximately 735,000 samples, each corresponding to ball-by-ball commentary. Our initial approach involved training a multilayer perceptron (MLP) model with ReLU activation. However, the model accuracy was surprisingly very high around 93% and above.

Upon analysis, we hypothesized that the imbalanced distribution of classes in the data set significantly affected the model performance. Although overall accuracy appeared high during initial training, the model performed poorly for classes of lower frequencies. This issue arose because the high-frequency classes dominated the learning process, effectively overshadowing the less-represented categories. Consequently, the model failed to generalize well across all categories.

Although the proportions of the original dataset mirrored real-world scenarios in cricket commentary, this imbalance was not suitable for building a robust model. To address this, we reduce the dataset to 80,000 samples, ensuring a balanced distribution between classes, as shown in Table I. Specifically, categories 3 and 4, which approximately had 21,000 samples each, were used as a baseline. We limited all categories to 20,000 samples each, resulting in a requirement of an evenly distributed data set of 20,000 samples for each category.

This balancing strategy ensured that all categories were adequately represented during training, improving the model's ability to perform consistently across different classes.

TABLE I
DATASET DISTRIBUTION

Label	Event	Number of Samples
0	Runs	20,000
1	Out	20,000
2	Extras	20,000
3	Dot Ball	20,000

B. *Data Transformation and Feature Engineering*

To prepare the dataset for effective model training, additional preprocessing steps were applied to transform the raw data and extract features suitable for machine learning. This subsection outlines the key steps taken:

1) ***Commentary Keyword Removal.****:* The Commentary column contained ball-by-ball text commentary, which required cleaning to reduce noise and improve relevance. A set of predefined cricket-specific keywords, including *wide, no ball, leg bye, bye, four, six, out, no run, single*, and similar terms, was identified and removed using a regular expression-based approach. This step ensured that redundant and potentially misleading terms were excluded from the input data. A verification process confirmed that these keywords were successfully removed, with no instances remaining in the cleaned commentary.

2) ***Balancing the Dataset.****:* The original dataset exhibited class imbalance, with certain categories significantly underrepresented. A subset of the data was extracted to balance the class distribution, ensuring 20,000 samples per category for a total of 80,000 samples. This balancing process enhanced the dataset's suitability for model training by ensuring that all classes were equally represented, which mitigated the risk of the model overfitting to high-frequency categories.

3) ***Tokenization and Sequencing.****:* The cleaned commentary data was transformed into numerical sequences suitable for machine learning:

- A tokenizer was fit on the Commentary text, with a vocabulary size of 5,000. An *Out-of-Vocabulary (OOV)* token was used to handle words not in the vocabulary.
- The text was converted into sequences of integer tokens, where each token represented a word or phrase.
- To maintain uniform input size, the sequences were padded or truncated to a fixed length of 100 tokens. Padding was applied at the end of sequences to ensure consistency.

*4) **Experiment Setup**:* The transformed dataset was split into training and testing subsets:

- The split ratio was set at 80:20, allocating 80% of the data to training and 20% to testing.
- K fold cross Validation with (*3,5,10*) folds was used during the tests to counter over fitting.

These transformations and feature engineering steps were crucial in preparing the dataset for machine learning. By addressing noise, imbalance, and variability in sequence lengths, the preprocessing phase laid the foundation for robust model development and evaluation.

C. Approaches

Using the above setup and the transformed dataset we played around 3 approaches:

- First we tried to build our model using top 100 most non repetitive frequent words across all categories as features to train.
- The next approach was using PCA with different variance coverage step levels from 80% to 100% based on 166 features gained from overall non repetitive but individual top 100 filtered features.
- Finally we utilized the NLP approach of ngrams using TF-IDF vectorizer with sequencing, padding etc.

So iterating through the above three approaches one by one we discovered few things which led us to change our approach gradually till we ended on approach 3.

As in the first approach we experimented with different no of features out of the total 21,597 features but realized the model is not heading anywhere and it is pretty directionless. So then we moved towards the PCA approach were we reduced 21,597 features to 166 features which are most frequent distinguishing unique features through a series of preprocessing steps and then experimented with these 166 at different PCA variance coverage levels from 80 to 100. But as we have already reduced it to 166 so it felt reasonable to go with all 166 features as per the received results instead of 70 features (80% variance) of the new reduced dataset which is giving comparatively good results than approach 1 but failed to cross the mark of 76% test accuracy results mostly due to important data loss.

After the stall of PCA feature based model building we approached to the traditional NLP approach of using N grams alongside tokenization,padding,sequencing requisites to train a model on identified techniques.

The identified techniques for experimentation consisted of diverse options from:

- Linear Classifiers
- Non Linear Classifiers
- Ensemble Approaches

which are mentioned below in the results section.

IV. Results & Analysis

The model was tried to build around many options apart from the main identified ones just for ruling out any chance of missing data specific model fit cases. So it consisted of classifiers like Logistic Regression, K Nearest Neighbors, Kernel based Support Vector Machines, Multi Layer perceptrons, Gated Recurrent Units, Long Short Term Memory and their Bidirectional versions, Random Forest classifier, AdaBoost, XGBoost, CatBoost, LightGBM. The algorithms which failed to touch the mark of 80% test accuracy were discarded and not considered for the results and analysis section.

TABLE II
Results and Analysis of All the Models

Model	**Test Accuracy (%)**
LightGBM	84.93
SVM (RBF Kernel)	84.59
Bi-LSTM	84.62
Bi-GRU	84.12
SVM (Linear Kernel)	84.08
Logistic Regression	83.00
Random Forest	80.00
LSTM	79.24
GRU	79.16

The table II summarizes the overall analyzed research end results for the problem statement. During the model building phase we tried different combinations and values of hyperparameters to reach this best results. In this we went to two different approaches one is without cross validation and regularization and one using them. It came to our notice that in almost all of our cases our validation accuracy just increased during k=10 folds. L2 regularization had comparatively performed better in increasing our model accuracy but the maximum increase in our accuracy was around 1%.

A. Support Vector Machines

Considering the dataset had 4 categories which have distinguishing features apart from clear indicative words we decided to go with kernel based SVM. Also from the previous work by Minard et al. [5] and the further enhancement by Miraoui et al. [6] to take SVM upto 85% accuracy mark in football event classification having more categories inspired us. So based on that it was decided to use SVM where we used radial basis function as well as linear kernel. We analyzed that although SVM couldn't catch up with the sequential memory of LSTMs or their Bidirectional versions initially but it started performing well when we started using approach 3. In most cases the accuracy got uplifted but in terms of SVM the growth was high. The performance in terms of classification went neck to neck with Bi-LSTMs in the end. Thus highlighting the potential reason behind the data being well enough separable to achieve 84.59% mark. In terms of linear kernel the SVM gave an accuracy around 84.08% which highlights the fact that the data seems to be linearly separable to great extent. To experiment with what results we can fetch by adding embeddings to boost the contextual part we tested both radial basis function and linear SVM using Glove 100 as well as 300 dimension but the results failed to touch 80% getting stuck around high 77%. The result was anticipated to perform like this as word2vec and GloVe, both are limited in terms of interpreting context and polysemous words.

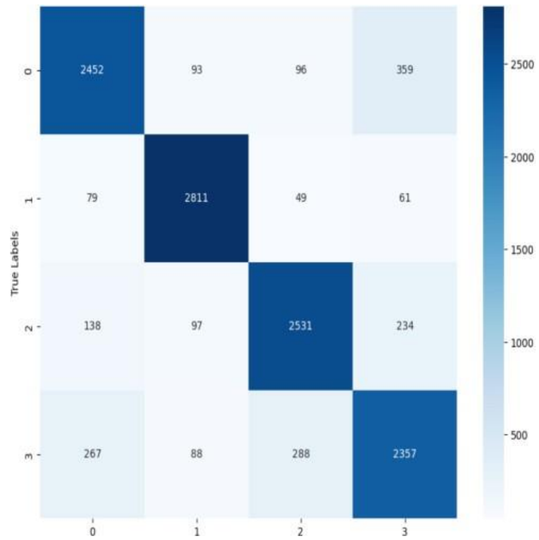

Fig. 2. Confusion matrix for SVM with RBF Kernel

B. *Light Gradient Boosting Machine*

For this classification, we have explored various ensemble methods, including random forest, adaboost, xgboost, and catboost classifiers. It was found that the LightGBM model produced better results with the accuracy of 84.94 than the others. RandomForest classifier provided an accuracy of 80.12%, whereas other models provided less than 80% accuracy. LightGBM is a sophisticated, open-source, tree-based system that was introduced in 2017. It is another tree-based method that is similar to XGBoost but differs in ways that make it computationally more efficient. Where XGBoost and Random Forests are based on branches, LightGBM grows leaf-wise. LightGBM outperfomed due to its advanced boosting mechanism and optimized tree-building strategy. Unlike Random Forest, which builds multiple independent trees and averages their outputs, LightGBM employs gradient boosting, where trees are built sequentially to correct the errors of the previous ones, leading to more refined and accurate predictions. LightGBM's use of **Gradient-Based One-Side Sampling (GOSS)** focuses on the most informative instances during training, improving efficiency and effectiveness. Additionally, its **histogram-based algorithm** speeds up computations and reduces memory usage, allowing it to handle large datasets and complex feature interactions more effectively. These advantages enable LightGBM to capture subtle patterns in the data, leading to improved performance over Random Forest. Here Optuna is used in hyperparameter tuning for its efficient optimization framework that uses techniques like **Tree-structured Parzen Estimators (TPE)** and **pruning strategies** to explore the parameter space intelligently. It stands out for its ability to automatically focus on promising regions of the search space, significantly reducing the time required to find optimal parameters. In your case, Optuna fine-tuned critical LightGBM parameters like `learning_rate`, `num_leaves`, `max_depth`, and regularization terms, achieving a robust configuration tailored to your dataset. The resulting parameters balance model complexity and regularization, improving multi-class classification performance while minimizing overfitting. Class 1(OUT) predictions are better with high precision and recall, leading to confident predictions. Class 3(DOT) shows the most ambiguity, with noticeable mis-classification into classes 0(RUN) and 2(EXTRAS) which affects its overall precision and F1-score. The model performs fairly well for classes 0(RUN) and 2(EXTRAS)

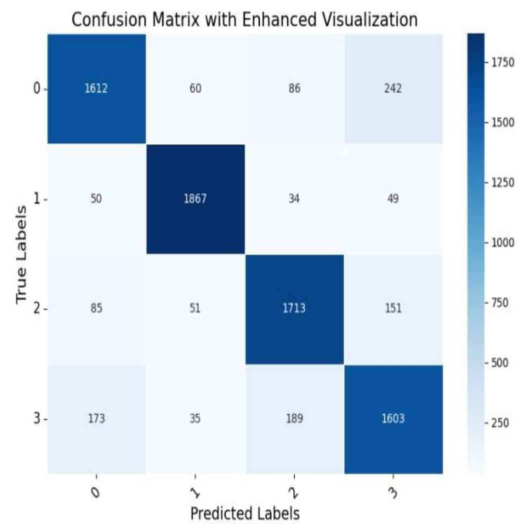

Fig. 3. Confusion matrix of LightGBM

C. *Bidirectional Long Short Term Memory*

While exploring recurrent neural networks, came across two prominent existing approaches Gated Recurrent Units and Long Short Memory. Before LSTMs our initial efforts went on to train and tune GRUs which fetched the best results. Getting confirmation of GRUs reach, moved to LSTMs which fetched a considerably high accuracy growth leading to 79.24% at this point we haven't tested GRU's with regularization and validation but just for experimental sake GRU with cross validation and regularization was tested for any chance of major increase but it just ended up on 79.06.

As the LSTM results went close to 80% the next idea was to lookup for GRU and LSTMs Bidirectional versions to be tested. So without regularization and cross validation it fetched results of higher 82% accuracy with Bi-GRUs and 84.51% with Bi-LSTMs. Later with Regularization, the accuracies increased by 0.4% in case of Bi-GRU resulting in 84.12%. While Bi-LSTM accuracy after regularization and cross validation grew by 0.1% leading to end result of 84.62%.

The above two models were tested with different hyper-parameter combination values and number of epochs ranging from 5-200 to track the model performance with the learning rate being decisive in terms of overshooting for optimal results was slowed down to rate of 0.001. The batch size was one of the important factors as it affected the train of the model significantly as higher the number of diverse samples higher tends to be the coverage of model avoiding overfit. But finding the optimal number was one of the important things as high numbers too can fetch disparity in proportions and lower can tend to do so too. So after experimentation, the optimal size was found to be around 100 for the batch size. Rest in our case the RMSprop with Sparse categorical cross-entropy alongside the combination of tanh and relu activation with 4 hidden layers and softmax at the end have given promising results.

But in case of Bi-LSTM the training accuracy used to stall after 40 epochs and near 80 it used to converge a bit and then became a mere case of overfitting. Thus after considering few checkpoints to do early stopping we ended our search on 7 as it was having high validation and test accuracy and the overtraining did not lead us to any significant increase further too as 7 epochs proved enough to train Bi-LSTM considering the amount of training samples. The model generalizes well, given the minimal gap between training and validation accuracy. Bi-LSTM captures temporal dependencies effectively, which is crucial for text-based predictions. The L2 regularization with 0.001 proved beneificial in increasing the accuracy across LSTMs and several other models.

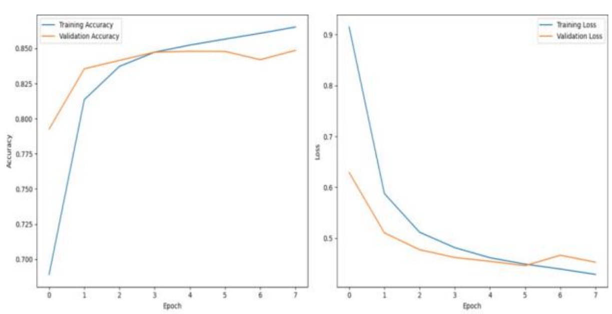

Fig. 4. Training vs Validation Accuracy & Loss graph for BI-LSTM

D. *Pre Trained Generative Models:*

In this experiment we took our original commentary with keywords to test ChatGPT 4.0 to predict on our 80,000 samples which fetched us an accuracy of 38.49%. Next for comparative purposes we decreased our test size to same ratio as our models to 20% were we gained a mere insignificant increase of 0.07% leading to 38.56%. This proves the fact that pretrained models fail terribly to understand the context of the game and need to fine tuned for further specific uses or one should focus building models from scratch without leveraging such Gen AI options as they have high latency as well as less ambiguity solving power. Although it was evident during data preparation in another sports analytics project, we wanted to be assured of the fact that it is not the perfect fit for specific purposes like sports yet.

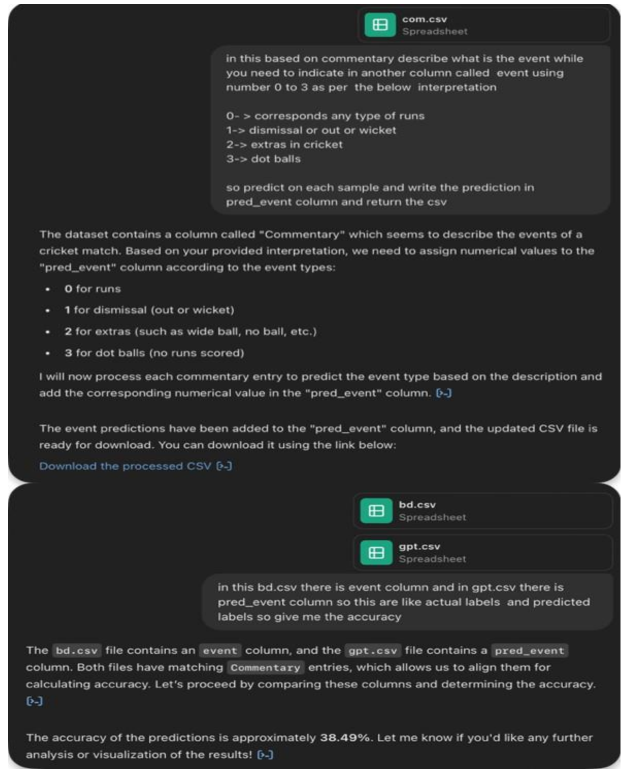

Fig. 5. GPT generating predictions on Commentary

V. Conclusions

Due to sequential nature and clear indicative words removed for sense disambiguation random forest classifier and most others couldn't deal noise sensitivity or kind of overfit. Also the extras category is added in this project which adds more complexity to the task which was missing in previous works as per the literature review. Using N grams approach (1,2 in our case) with TF-IDF vectorizer and embedding helped to achieve a significant positive increase in accuracy. So contextual awareness being the most crucial part of the commentary decoder we decided to move to one of the state of the art language model BERT for which we have initiated our efforts to fetch results and it is anticipated to outperform others based on the results shown by Miraoui et al. [6]. The only answer remains by how much % the accuracy will grow using BERT. In terms of Neural networks recurrent neural networks proved promising with temporal dependency handling abilities. So we tried BERT for classification which is a transformer as it deals best with contextual ambiguity using surrounding words to help computers understand the meaning properly but facing few difficulties to get initial results due to a bit of unfamiliarity and experience using this technique but it is definitely a great promising option to explore further. Also, it is worth researching how much feasible will pretrained generative AI models will work in a real world setup given the above results and to what extent it is able to cover the requirements of ambiguity.

VI. Authors Contributions & Remarks

We conducted extensive research and reviewed various papers in the domain of event classification in sports. However, we realized that there is a noticeable lack of recent, high-quality work in this field, and the reported results often fall short of validation benchmarks. Additionally, we faced the challenge of limited reference projects, as every reference seems to use a diverse set of approaches that did not directly fit our work.

When evaluating this report, we kindly request you to consider that the quality and validity of existing work in this area is at a nascent stage. So, based on that, we tried to refine our covered study by only citing the work which are recent and, moreover, has scope to improve with time and current advancements

References

[1] Kalpit Dixit and Anusha Balakrishnan. Deep learning using cnns for ball-by-ball outcome classification in sports. 2016.

[2] S. D. R. Kumar and J. Barnabas. Outcome classification in cricket using deep learning. *IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)*, 2019.

[3] D. S. V. N. S. S. Anudeep Md. Tayyab A. Siva Balaji, N. Gunadeep Vignan and K. S. Vijaya Lakshmi. Cricket commentary classification. *Intelligent Data Communication Technologies and Internet of Things*, page 825–836, 2022.

[4] Summra Saleem Muhammad Usman Ghani Khan Zain Ul Abideen, Saira Jabeen. Ball-by-ball cricket commentary generation using stateful sequence-to-sequence model. *International Conference on Communication Technologies (ComTech)*, 2021.

[5] Manuela Magnini Bernardo Qwaider Mohammed. Minard, Anne-Lyse Speranza. Semantic interpretation of events in live soccer commentaries. 2016.

[6] Yanis Miraoui. Analyzing sports commentary in order to automatically recognize events and extract insights. *arXiv:2307.10303*, 2023.

[7] V. Raj J. Sharma A. Semwal, D. Mishra and A. Mittal. Cricket shot detection from videos. 2018.

[8] Rohit Prabhavalkar Patrick Nguyen Zhifeng Chen Anjuli Kannan Ron J. Weiss Kanishka Rao Ekaterina Gonina Navdeep Jaitly Bo Li Jan Chorowski Michiel Bacchiani Chung-Cheng Chiu Tara N. Sainath, Yonghui Wu. State-of-the-art speech recognition with sequence-to-sequence models. 2018.

[9] V. Subramaniyaswamy; R. Logesh; V. Indragandhi. Intelligent sports commentary recommendation system for individual cricket players. *Int. J. Adv. Intell. Paradigms*, 10:103–117, 2018.

[10] Terence L. Van Zyl Daniel Yazbek, Jonathan Sandile Sibindi. Deep similarity learning for sports team ranking. *arxiv.org/abs/2103.13736*, 2021.

[11] Yuluan Cao Shiwei Ren Jingdi Lei, Tianqi Kang. Capturing momentum: Tennis match analysis using machine learning and time series theory. *arxiv.org/abs/2404.13300*, 2024.

[12] Mohammad Farukh Hashmi, Tejas Bhat Bellare, Ankith Suresh, and Banoth Thulasya Naik. Football event classification using convolutional autoencoder and multilayer extreme learning machine. *IEEE Sensors Letters*, 6(10):1–4, 2022.

[13] Omar K. Alqahtani Essa, E. Fake news detection based on a hybrid bert and lightgbm models. *IEEE Access*, 9, 2023.

[14] Mohammad Farukh Hashmi, Tejas Bhat Bellare, Ankith Suresh, and Banoth Thulasya Naik. Football event classification using convolutional autoencoder and multilayer extreme learning machine. *IEEE Sensors Letters*, 6(10):1–4, 2022.

[15] T. Raghuvansh. Cricket scorecard and commentary dataset. *Kaggle*, https://www.kaggle.com/datasets/raghuvansht/cricket-scorecard-and-commentary-dataset, 2022.