Norwegian University
of Life Sciences

Master's Thesis 2024 30 ECTS
Faculty of Science and Technology

Automatic Detection of Soccer Events using Game Audio and Large Language Models

Joel Yacob Teklemariam
Data Science

Automatic Detection of Soccer Events using Game Audio and Large Language Models

Joel Yacob Teklemariam

June 5, 2024

Abstract

This thesis tackles the inefficiencies associated with manual annotation in soccer event detection, a process that is time-consuming, expensive, and difficult to scale during major tournaments. By developing an automated audio-based event detection system, this research aims to bypass the extensive resource requirements of traditional video action detection, offering a more efficient and balanced alternative. This research uses the Automatic Speech Recognition (ASR) transcriptions from SoccerNet-Echoes to contribute with two supervised datasets: the 15-Second Standard Deviated Dataset (15-SSDD) and 30-Second Standard Deviated Dataset (30-SSDD). These datasets incorporate a 15-second and 30-second standard deviated window of soccer event context to train models for recognising key events like Goals, Fouls, and Corners. They were evaluated on several Large Language Models (LLMs), including DistilBERT, BERT BASE, BERT LARGE, and all-MiniLM-L6-v2. The findings show that longer contextual samples significantly enhance the model's classification accuracy, underscoring the importance of context within events in soccer. The all-MiniLM-L6-v2 model is noted for its high accuracy and computational efficiency, making it ideal for real-world applications that demand rapid and precise event detection. It performs robustly across various metrics such as F1-score, precision, and recall. It operates efficiently on both datasets with fewer computational resources, underscoring its suitability for efficient and accurate applications. Challenges such as class imbalance impact the overall effectiveness of the detection system. The thesis proposes future enhancements, including audio implementation and exploring class balancing strategies like word embedding oversampling and cost-sensitive learning to refine the system's robustness and effectiveness. This research advances the field of sports analytics by proposing an efficient audio-based event detection system. It also sets the stage for future innovations that could transform the monitoring and analysis of sports events, enhancing viewer experiences and providing sports professionals with critical insights in real-time.

Preface

This research follows the guidelines set by the Norwegian University of Life Sciences (NMBU), specifically adhering to the regulations provided for the use of artificial intelligence by the Faculty of Science and Technology (REALTEK). This outlines that AI has been used responsibly, following these guidelines to ensure academic integrity and the quality of the research conducted. The internal supervisor, Habib Ullah, provided essential guidance on AI's ethical and constructive use, ensuring its application adhered to NMBU's academic standards.

Acknowledgement

The author acknowledges the Orion High-Performance Computing Center (OHPCC) at the Norwegian University of Life Sciences (NMBU) for providing computational resources that have contributed to the research results reported within this paper.

I want to thank all my internal supervisors at NMBU and external supervisors at SimulaMet: Cise Midoglu, Pål Halvorsen, and Habib Ullah for their invaluable support, guidance, and encouragement throughout this thesis.

Finally, I want to thank my family for their constant motivational support.

Contents

List of Figures

List of Tables

Acronyms

15-SSDD 15-Second Standard Deviated Dataset. 30, 31, 35, 38, 50–54, 56–66, 68–71, 73, 75, 83–92, I, VII–IX

30-SSDD 30-Second Standard Deviated Dataset. 30, 35, 38, 39, 50–54, 56–66, 68, 69, 73, 75, 83–92, I, VII–IX

ACC accuracy. 45, 46

AI Artificial Intelligence. 16

AmAP Average mean Average Precision. 21–24, 72, IX

ANN artificial neural network. 7, 8

ASR Automatic Speech Recognition. 2–4, 12, 15–18, 24, 26, 29–31, 35, 67, 69, 72, 73, 75, I

CNN Convolutional Neural Network. 22, 23

CSN Channel-Separated Convolutional Network. 21

DL deep learning. 7

DNN deep neural network. 8, 10, 11, 17

ERR error. 45, 46

GMM Gaussian Mixture Model. 17

HMM Hidden Markov Model. 16, 17

LLM Large Language Model. 2–4, 36, 50, 63, 64, 66–70, 73–75, 85, 87, 92, I, VIII, IX

mAP mean Average Precision. 1, 21, 23, 24, IX

ML machine learning. 3, 6, 7

MLM Masked Language Modeling. 13

MLN multilayer neural network. 8, 9

NLG Natural Language Generation. 11, 12

NLP Natural Language Processing. 3, 11–15, 24, 26, 38, 40, 41, 48

NLU Natural Language Understanding. 11, 12

NSP Next Sentence Prediction. 13

ReLU Rectified Linear Unit. 10, 11

STCD Soccer Text Classification Dataset. 29, 30, 32, 34–36, 68, IX

t-AmAP tight-Average mean Average Precision. 23, 24

WER Word Error Rate. 17, 18

Chapter 1

Introduction

1.1 Motivation

In modern sports analysis and entertainment, soccer stands as a global phenomenon that captures the hearts and minds of millions. FIFA World Cup Qatar in 2022 had more than 5 billion people watching and 3.4 million attending matches [52]. Soccer has a big influence on the whole world. Therefore, it's important to advance the tools that improve the holistic experience for everybody. In this context, accurate and timely detection of critical events during soccer matches is paramount. These events, ranging from goals and penalties to corner kicks and free kicks, not only define the course of a game but also shape team strategies and individual player performances.

Traditional soccer event detection and analysis methods have predominantly relied on manual observation, human expertise, and often subjective judgment. However, with the advent of cutting-edge technologies, including Artificial Intelligence and audio analysis, a new era of automated event detection is dawning. Vanderplaetse et al. [81] was the first multi-modal study using audio and video features. They implemented this on the SoccerNet V1 dataset [35], improving the mAP score of 4.19% for action spotting. Implementing these technologies would revolutionise how soccer matches are perceived, analysed, and ultimately experienced by fans, analysts, coaches, and players alike.

Multi-modal approaches in event detection typically rely on advanced methods demanding significant computational resources. This complexity can hinder real-time implementation. Ideally, an event detection system should be efficient and lightweight yet robust enough to operate effectively in complex environments. Unlike video, which utilises considerable resources, audio requires substantially less and is thus a more viable option for such systems. To achieve this, systems could leverage metadata and audio, where metadata includes textual information about the data being analysed. A more streamlined alternative could involve constructing a dataset from game audio and metadata for event detection through text classification. This strategic combination of audio and metadata reduces system complexity and enhances the ability to correlate relevant game audio with the correct event quickly.

Manual annotation remains the primary method for soccer event identification, creating a significant bottleneck in the process. This procedure is both expensive and time-consuming. It becomes increasingly challenging as the number of games increases. In some cases, manual annotation may not be feasible when working on multiple games within a constricted schedule. Automatic event detection systems offer an alternative solution by independently operating on large amounts of data, potentially saving significant time and cost. Furthermore, the insights gained from this research could facilitate real-time decision-making for coaches, broadcasters, and fans, leading to a more immersive and engaging soccer experience.

1.2 Problem Statement

Accurate and precise temporal detection of critical events in soccer remains a significant challenge. Existing approaches often rely on single-modality methods, which may fall short of capturing the complexity and richness of real-world match scenarios. These limitations can lead to missed detections and inaccuracies in event identification. This research addresses this challenge by investigating the application of supervised learning techniques that leverage the combined power of multiple modalities, including game audio and metadata. By harnessing these diverse data sources, this study seeks to unlock new dimensions of insight and accuracy in event classification. This will contribute to a more comprehensive understanding of soccer matches and potentially revolutionise how these events are analysed. To address these issues, this thesis seeks to answer the following research question:

How can an automatic soccer event detection system be developed by integrating game metadata, game audio with ASR, and LLMs?

The research question is followed up by two objectives:

Objective 1

Construct two distinct datasets using automatic speech recognition to transcribe game audio and metadata, with both datasets tailored to the classes: Goal, Foul, and Corner.

Objective 2

Analyse and compare the effectiveness of LLMs in event detection across the distinct datasets.

1.3 Scope

This research primarily focuses on analysing Goals, Corners, and Fouls in soccer. While the scope is concentrated on these specific events, the foundational methodologies and insights gained could apply to similar events within soccer and comparable sports. The framework established here may also offer valuable perspectives for exploring alternative approaches in related contexts.

This research utilises the SoccerNet V2 dataset, the largest publicly accessible dataset tailored for event detection in soccer [1]. It includes approximately 300,000 temporally anchored annotations across 764 hours of video from 500 matches, equivalent to about one month of continuous footage. Building upon the foundational work of the SoccerNet V1 dataset [35], SoccerNet V2 expands the volume and variety of available data. This comprehensive dataset covers 17 distinct event classes, including Goal, Foul, Corner, Substitution, and other events.

The precise and well-defined nature of its annotations makes SoccerNet V2 ideally suited for the focus of this thesis, which involves creating datasets from game audio and metadata for event detection. This research utilises all 500 games from the SoccerNet V2 dataset. The game audio is processed through an automatic speech recognition model that transcribes the audio into textual information. This text is then synchronised with the events and game time annotations from the SoccerNet V2 dataset. Subsequent steps include data exploration and preprocessing of the provided text.

1.4 Ethical Considerations

This research utilises data from the SoccerNet V2 dataset, which comprises video and audio from soccer broadcasts. Players, coaches, and commentators have consented to the public use of their appearances and voices through contractual agreements. The dataset is publicly available but strictly limited to non-commercial research use in compliance with the Non-Disclosure Agreement (NDA) terms with SoccerNet. This agreement also prohibits the redistribution of copyrighted materials. SoccerNet contains copyrighted content from various European leagues, emphasising that creating a business around soccer video analysis would necessitate acquiring separate video rights [69].

1.5 Research Methods

The research methodology employed in this thesis adheres to the paradigms defined by the Association for Computing Machinery (ACM), specifically focusing on theory, abstraction, and design. These three foundational pillars establish a comprehensive framework for structuring research within the computing field [29].

- **Theory:** This mathematical-based approach involves defining objects of study, hypothesising relationships, proving these hypotheses, and interpreting the results, often iterating to refine the theory.
- **Abstraction:** Drawing from the scientific method, this paradigm focuses on forming hypotheses, building models for prediction, conducting experiments, and analysing outcomes to refine the models.
- **Design:** Rooted in engineering, this practical paradigm includes defining requirements, specifying and implementing solutions, and testing the systems created.

This thesis aims to develop an automated event detection system that leverages game audio and metadata with the help of LLMs to classify key events in soccer matches. Engaging in theoretical exploration by hypothesising the effectiveness of ASR technology for transcribing soccer game audio. This requires a deep understanding of concepts in Natural Language Processing (NLP), machine learning (ML), ASR, and action detection. Abstraction played a critical role in this research by creating predictive models based on LLMs, where the theoretical concepts were transformed into practical algorithms that could simulate and predict real-world soccer events. Finally, the design paradigm was manifested by engineering a practical pipeline that integrates ASR with audio data and utilises LLMs to detect events. This includes defining system requirements, designing software architecture, implementing solutions, and conducting tests to ensure efficiency and accuracy. Each of these paradigms contributed distinctly and aligned with ACM standards to ensure the research is conducted with a disciplined approach widely recognised and respected in the computing community.

1.6 Main Contributions

This thesis advances the field of sports analytics, specifically soccer event detection, by leveraging LLM to extract and interpret textual information from game audio. Through in-depth research in ASR technology and constructing two distinct datasets, this work paves the way for future exploration and enhancement in audio-based event detection. The thesis describes a comprehensive pipeline from dataset creation to evaluation using various LLM models, offering a structured approach to understanding and processing soccer match audio for event detection. This approach could be the foundation for developing lightweight, audio-based automatic event detection systems.

The detailed contributions from this thesis:

- **Developed and utilised LLM architectures** for text classification to identify and classify critical events in soccer, enhancing the understanding of audio-derived textual data.
- **Established a methodological pipeline** for creating and utilising two supervised soccer text datasets. This pipeline includes data collection, preprocessing, model training, and evaluation.
- **Constructed two tailored datasets** that serve as valuable resources for the broader research community. These datasets facilitate the continued improvement of ASR applications in sports analytics.
- **Demonstrated the efficacy of audio-based event detection**, providing insights into the performance of various LLM models and establishing comparison on the different models for future research.

These contributions address a significant gap in the current soccer research community, which typically relies primarily on video modality for event detection. The proposed method from this research is computationally more efficient and requires fewer resources. The outcomes enrich the academic and practical understanding of LLM in sports contexts and enhance the capabilities of event detection systems to operate more effectively with audio inputs.

1.7 Thesis Outline

The thesis is organised into the following chapters:

Chapter 2: Background and Theory. This chapter establishes a solid foundation by reviewing theoretical concepts crucial for understanding the research presented later. It covers key areas such as action spotting, machine learning fundamentals, natural language processing, and automatic speech recognition. Additionally, this chapter includes a comprehensive literature review that explores existing research on event detection and datasets related to soccer.

Chapter 3: Dataset Construction and Exploration. This chapter details the process of constructing supervised datasets and explores the use of the SoccerNet V2 dataset in this research. The steps include data acquisition, dataset construction, and implementing specific window sizes and event classes.

Chapter 4: Methodology. This chapter describes the proposed approach to achieving the results of this research. It outlines the sequential steps taken and specifies the models used. Additionally, this chapter provides an overview of the pipeline, including data preprocessing, the use of the Hugging Face platform, Optuna for hyperparameter optimisation, the hardware utilised, and the proposed evaluation metrics.

Chapter 5: Results. This chapter presents the research findings, providing the results for each model. The models are first evaluated internally with the proposed evaluation metrics and then compared against each other on the various metrics. These evaluations are composed of the different dataset window sizes.

Chapter 6: Discussion. This chapter revisits the research question and objectives. It details the contributions made in this thesis and explores future directions for research, including SoccerNet V2, class imbalance, and data preprocessing.

Chapter 7: Conclusion. This chapter concludes the thesis by summarising the key research findings.

Chapter 2

Background and Related Work

This chapter establishes a robust theoretical framework directly relevant to the research questions explored in this thesis. It aims to give readers a comprehensive introduction to fundamental concepts and principles across essential areas such as action spotting, machine learning, automatic speech recognition, natural language processing, and evaluation metrics. Understanding these theories prepares the reader for deeper exploration and engagement with the following research endeavours.

2.1 Machine Learning

ML is a dynamic and rapidly evolving field at the intersection of computer science and statistics, aimed at developing algorithms that enable computers to learn from and make predictions or decisions based on data. Depending on the labels associated with the data, two main approaches can be categorised: supervised learning and unsupervised learning [48].

2.1.1 Supervised Learning

Supervised learning is the most popular branch of ML [50]. This method involves mapping the input data (X) to the labelled output data (Y). The algorithm attempts to identify patterns between the input data and the ground-truth labelled data. After the training phase, the model is presented with unseen input data to predict the output labels [80]. A primary challenge with supervised learning is the annotation process, which can be time-consuming and costly.

2.1.2 Classification

Classification is a subcategory of supervised learning aimed at categorising input data into distinct labels. This process is divided into two principal approaches:

- **Binary classification**: The dataset is two-dimensional, offering only two possible outcomes for the class label. For example: ' yes' or 'no', 'spam', or 'not spam' [71].
- **Multi-class classification**: Consists of data that encompasses more than two categories. This method often employs probability-based calculations to determine each data point's most likely

class label. It is suitable for scenarios where the input data can belong to multiple categories, such as identifying types of fruits or categorising articles into different topics [71].

2.1.3 Dataset

The Oxford Dictionary explains a dataset as: "A collection of data". For this research, the focus is on supervised learning. It consists of data features and corresponding labels from the data as output. Datasets are often split into three different subsets: training set, testing set, and validation set. Divining the dataset into these parts is crucial to evade biased predictions [79].

- **training set**: samples of randomly retrieved data from the original dataset used to train the machine learning algorithm [78].
- **testing set**: used at the end of the process to assess the performance of the algorithm on unseen data [78].
- **validation set**: utilised to reduce overfitting and have a more generalised and versatile algorithm [78].

2.1.4 Deep Learning

Deep learning (DL) is a subfield of ML characterised by Artificial neural networks (ANNs) with multiple hidden layers [13]. These layers allow DL models to learn complex patterns and relationships from large amounts of data, enabling them to achieve state-of-the-art performance in various tasks. Standard ANNs is composed of layers of interconnected nodes or neurons, each contributing to the ability of a system to process and learn from vast amounts of data. The simplest form of a neural network includes an input layer, one hidden layer, and an output layer [71].

Perceptron

Before further exploring the intricate world of DL, it is essential to establish a strong foundation by understanding the core principles of single-layer neural networks. This section presents the perceptron, introduced by Frank Rosenblatt in the 1950s, as the simplest and most fundamental form of an ANN [45, 68]. The diagram depicted in Figure 2.1 demonstrates the perceptron model, functioning as a basic unit of a single-layer neural network.

The input to the perceptron is denoted by $x_1, x_2, \ldots, x_m$, each multiplied by their corresponding weights $w_1, w_2, \ldots, w_m$. These weighted inputs and a bias weight w_0 are then summed to calculate the net input. This net input is fed into a threshold function, known as the unit step function, which determines the output of the perceptron. The output is either -1 or $+1$, symbolising the predicted class label for the given example. Any discrepancy between the predicted output and the actual class label is considered an error throughout the learning process. This error is fed back into the system to adjust the weights, optimising the perceptron's performance in classifying future examples [71].

It is important to recognise that the perceptron, despite its foundational role, has its limitations. The perceptron can only learn linear decision boundaries, meaning it can classify only linearly separable data. This restricts its utility in scenarios involving more complex patterns that cannot be separated linearly, such as circular or spiral distributions. Due to its simple architecture, the perceptron has

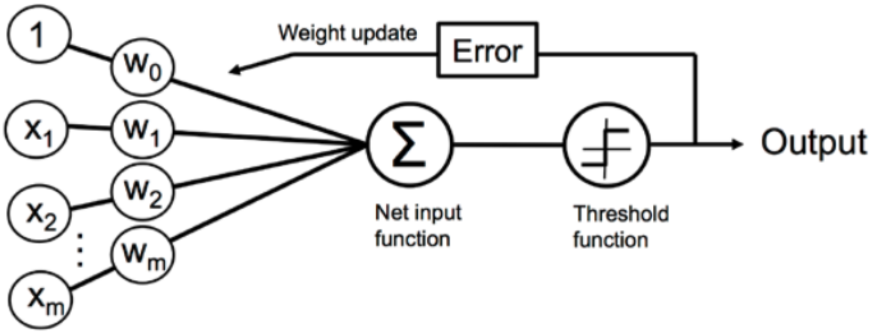


Figure 2.1: Illustration over a perceptron architecture [71].

limited expressive power, which hinders its ability to learn complex relationships within data. The weight update process in perceptrons can lead to longer convergence times because it updates weights in large discrete steps to -1 or +1, with no intermediate values [71].

The significance of the perceptron lies in its ability to provide a crucial foundation for understanding the core principles of ANNs. Studying the perceptron reveals valuable insights into how weights influence the output, the role of bias in shifting the decision boundary, and the importance of activation functions for introducing non-linearity. The perceptron is a stepping stone for more sophisticated deep learning architectures that overcome its limitations and achieve superior performance on a wider range of tasks. Understanding the perceptron equips us with the necessary knowledge to explore the powerful world of deep learning [71].

Multilayer neural network

Multilayer neural networks (MLNs) draw inspiration from the brain's hierarchical structure. They employ multiple hidden layers, each populated with vector-valued units comparable to individual neurons. These units function in parallel, transforming and extracting intricate features from data through non-linear activation functions [57].

Deep neural networks (DNNs), characterised by their "chain-like" structure, combine multiple functions sequentially. This sequential composition allows the network to learn increasingly complex representations of input data through hidden layers. The training minimises the difference between the network's output and the desired target function [57].

The first layer receives raw input data, denoted by (in) in the figure 2.2. The number of neurons in this layer corresponds to the dimensionality of the input data. This intermediate layer, denoted by (h) in the figure, is crucial in extracting features from the input data. Each neuron within this layer receives weighted inputs from all neurons in the previous layer, performing a non-linear transformation using an activation function and transmitting the output to the subsequent layer. This final layer, denoted by (out) in the figure, generates the desired output based on the processed information from the hidden layer. The number of neurons in this layer aligns with the dimensionality of the network's output,

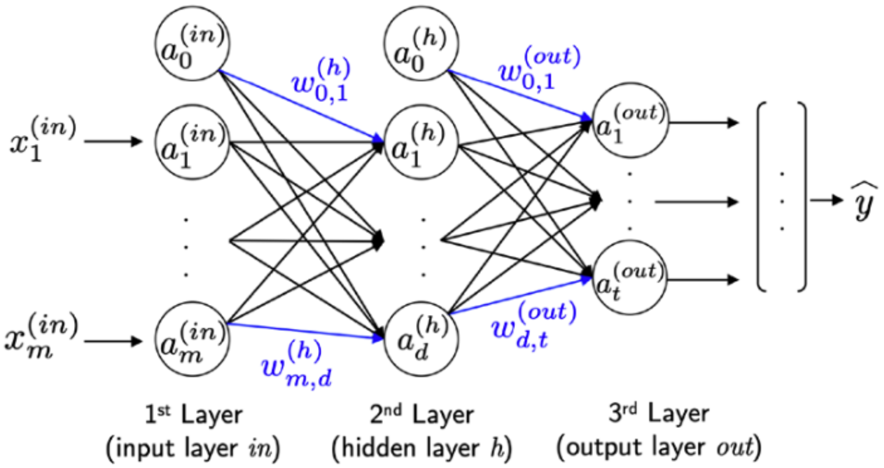


Figure 2.2: Structure of a MLN [71]

determining the number of distinct output categories or continuous values the model can predict.

Forward propagation in MLN involves calculating the activation of each neuron, layer by layer, starting from the input layer and progressing towards the output layer. Each neuron in the hidden layer calculates the net input by taking the weighted sum of the activations from all neurons in the input layer. Mathematically, this is represented as [71]:

$$z_1^{(h)} = a_0^{(l)} w_{0,1}^{(h)} + a_1^{(l)} w_{1,1}^{(h)} + \cdots + a_m^{(l)} w_{m,1}^{(h)} \tag{2.1}$$

Once the net input is calculated, a non-linear activation function ϕ is applied to the network. This helps the network to capture advanced relationships between the input and output data. A common choice for this function is the sigmoid (logistic) function, depicted as [71]:

$$a_1^{(h)} = \phi(z_1^{(h)}) \tag{2.2}$$

Activation function

The concept of artificial neurons forms the foundation of MLNs. In binary classification tasks, perceptrons aim to distinguish between two classes, typically labelled as 1 (positive) and -1 (negative). To achieve this classification, perceptrons employ a decision function. This function takes a linear combination of input values, weighted by corresponding weights, and produces a single output value known as the net input. The weights represent the strength of the connections between the input and the perceptron. The unit step function transforms the net input into a final output, determining the predicted class from the threshold θ. This function operates as follows [71]:

$$\phi(z) = \begin{cases} 1 & \text{if } z \geq \theta, \\ -1 & \text{otherwise.} \end{cases} \tag{2.3}$$

Many real-world applications require probability to determine the specific class of a sample. This necessitates an activation function that can output continuous values between 0 and 1. The logistic sigmoid function, also known as the sigmoid function due to its S-shaped curve, fulfils this requirement and addresses the limitations of the unit step function. Unlike the unit step function's sharp transition, the sigmoid function produces a smooth output. The logistic sigmoid function is mathematically represented by this formula [71]:

$$\phi(z) = \frac{1}{1 + e^{-z}} \tag{2.4}$$

The sigmoid function can suffer from a phenomenon called saturation when the net input becomes incredibly positive or negative. In such cases (z approaching positive or negative infinity), the function's output becomes insensitive to slight changes in the input, potentially hindering the network's learning ability for complex problems [57].

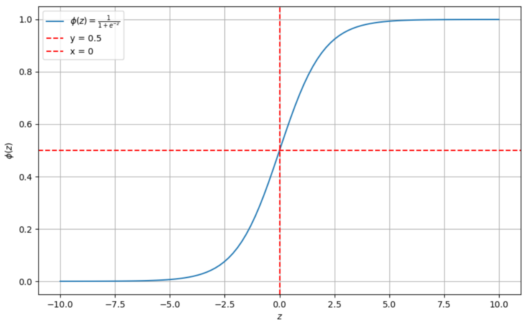


Figure 2.3: Illustration over the logistic sigmoid function with reference lines.

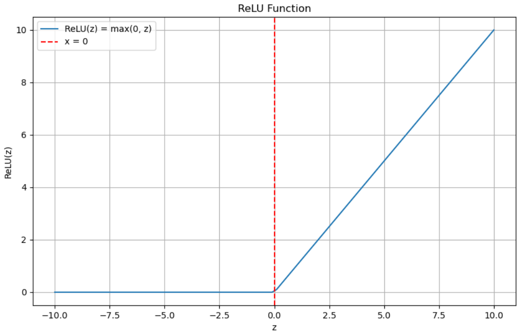


Figure 2.4: Illustration of the Rectified Linear Unit (ReLU) function with reference lines.

Figure 2.3 illustrates the relationship between the logistic sigmoid function $\phi(z)$ and its net input z, as defined in Equation 2.4. As z approaches positive infinity, $\phi(z)$ approaches 1 due to the diminishing effect of e^{-z} in the numerator. Similarly, as z becomes increasingly negative, $\phi(z)$ approaches 0 due to the rapidly increasing denominator, effectively damping the function's output [71].

ReLU is a popular activation function widely used in DNNs. Before delving into ReLU, it is crucial to understand the vanishing gradient problem that can arise with sigmoid. During training, the network adjusts its weights based on the errors it makes. This adjustment relies on gradients, which tell us how much a change in each weight will affect the overall error. The problem arises when these gradients become extremely small or zero as the data flows through the network's layers [71].

Consider a high net input value. Applying the sigmoid function to this value might result in an output close to 1. Imagine a slight increase in the original net input. Due to the asymptotic behaviour of the sigmoid function, calculating the activation with this new input might yield a value even closer to 1. This highlights that even though the net input increased, the corresponding change in the activation output is minimal. This minimal change translates to exceedingly small or vanishing gradients when backpropagating the error during training. Weight updates in earlier layers become slow or negligible, hindering the network's learning ability in deeper layers. ReLU addresses the vanishing gradient problem by introducing a linear relationship for positive net input values as seen in Figure 2.4 and in Equation 2.5 [71].

$$\phi(z) = \max(0, z) \tag{2.5}$$

ReLU is the default activation function for many feedforwards neural networks. It does not strictly create a completely non-linear output. Instead, it introduces a piecewise linearity. The ReLU function can be visualised as a graph with two straight lines. The function acts like a straight line with a slope of 1 for positive net input values. When the net input is zero or negative, the function outputs zero, forming a horizontal line. Due to its piecewise linearity, ReLU helps retain favourable properties from linear models. The non-zero slope for positive inputs allows gradients to flow back during training, unlike functions that saturate at extremes like the sigmoid function. This characteristic aids in efficient network optimisation using gradient-based methods. ReLUs partial linearity is believed to contribute to the good generalisation capabilities observed in DNNs using ReLU activation's [57]. In Transformer models, ReLU is specifically used in the feedforward networks within the encoder and decoder stacks, handling complex data patterns without sequential processing limitations [7].

2.2 Natural Language Processing

NLP is a field at the intersection of computer science, artificial intelligence, and linguistics. It strives to empower computers to understand, interpret, and produce human language in a significant and beneficial way [49]. NLP encompasses two distinct areas: Natural Language Understanding (NLU) and Natural Language Generation (NLG). These components advance the tasks of comprehending and producing text, forming the core of NLPs capacity to interpret and create human language [9].

NLU enables machines to interpret human language by extracting key information. It plays a crucial role in customer service, comprehending customer inquiries and issues regardless of whether they are expressed verbally or in text [49]. NLU encompasses a sophisticated framework that integrates various linguistic components, from phonology to pragmatics, to decode the complexities of human language. It begins with phonology, the study of sound systems within languages, establishing the foundation for understanding how sounds contribute to meaning. Morphology further dissects words into their smallest units of meaning, morphemes, revealing the structural nuances of language construction [9].

At the linguistic level, NLU processes the meaning of individual words and their parts of speech, employing techniques such as stemming and lemmatisation to refine text analysis. This lexical analysis is crucial for syntactic processing, where words are organised into phrases and sentences, highlighting the grammatical relationships that underpin sentence structure [9].

Semantic analysis then delves deeper into the meaning of sentences, utilising context to resolve ambiguities and ascertain the intended message. Discourse analysis extends this understanding to multiple sentences, ensuring text coherence by analysing how sentences relate to each other [9].

Finally, pragmatics considers the inferred meanings and implications beyond the literal text, incorporating real-world context and knowledge. Collectively, these elements of NLU allow for a comprehensive understanding of human language, enabling machines to interpret and generate text with a nuanced awareness of its structure, meaning, and context [9].

NLG is a sophisticated process of transforming structured data into meaningful phrases, sentences, and paragraphs. This process contrasts sharply with NLU, effectively representing the other side of the NLP coin. At its core, NLG requires a source to initiate the generation process and a generator to articulate the source's intentions into contextually relevant language. The process culminates in generating meaningful sentences and paragraphs, turning data into an understandable narrative. This mechanism involves a speaker or application to initiate and a generator to execute, translating structured inputs into contextually relevant language, streamlining complex information into accessible communication [9].

2.2.1 Text Classification

Text classification is a fundamental area of NLP, focusing on categorising text into predefined categories. Similar to other classification tasks where a model, described by a function $f : \mathbb{R}^n \to \{1, \dots, k\}$, assigns an input to one of k categories, text classification involves analysing text to determine its category based on content. In soccer event detection, models classify ASR text to identify relevant events. This process involves several critical steps: acquiring a dataset, preprocessing the text, tokenisation, and training a classification model, offering significant insights into the analysed content. These steps are not limited to those mentioned but could include many other preprocessing techniques such as stemming, lemmatisation, and more [57, 71].

The initial step involves collecting a dataset that accurately reflects the diversity of the text categories to be classified. This dataset forms the basis for training and evaluating the classification model. After acquiring the dataset, it is important to preprocess the data. That can entail cleaning the data for unwanted characteristics, ranging from HTML markup to other non-letter characters. HTML markup does not contain useful semantics, but punctuation can be useful depending on the classification task. Word capitalisation often lacks semantic relevant information, depending on the acquired dataset. Stop-words are frequently used words that hold little to no meaning in their context of sentiment analysis. While stop-words might carry some semantic value in general language, they often contribute minimally to text classification [71].

Tokenization is a fundamental step in transforming raw text data into a format suitable for analysis by machine learning models. It acts as a bridge between human language and the numerical world in which models operate. Tokenisation segments the text into individual words. This allows the model to process the text one unit at a time, identifying patterns and relationships between these units. By creating a vocabulary of individual tokens, the model can learn the relationships between words and their potential influence on sentiment. This vocabulary is the foundation for the model to understand the context [71].

2.2.2 Transformers

Transformers are network architecture that gained significant attention in sequence transduction models. Unlike traditional models that rely on recurrent or convolutional neural networks, Transformers are based solely on attention mechanisms, eliminating the need for recurrence and convolutions [7].

The Transformer architecture consists of an encoder and a decoder. The encoder takes an input sequence of symbol representations and maps them to a sequence of continuous representations. These continuous representations, denoted as $z = (z_1, ..., z_n)$, are then used by the decoder to generate an output sequence of symbols $(y_1, ..., y_m)$ one element at a time. The auto-regressive model consumes previously generated symbols as additional input when generating the next symbol [7].

The key component of Transformers is self-attention. A multi-head self-attention mechanism is employed in each layer of the encoder and decoder stacks. This mechanism allows the model to focus on different parts of the input sequence when generating the output sequence. By attending to relevant input parts, Transformers can capture long-range dependencies and improve the quality of the generated output [7].

One of the advantages of Transformers is their parallelizability. Unlike recurrent models that process input sequentially, Transformers can process the input in parallel, making them more efficient and reducing training time. Additionally, Transformers have shown superior performance in terms of quality compared to traditional models. Transformers have been successfully applied to various tasks, including machine translation. They achieved state-of-the-art results on benchmark datasets, surpassing even ensemble models [7]. This approach has accelerated further development in models for language understanding and generation tasks, such as BERT, DistilBERT, and all-MiniLM-L6-v2.

2.2.3 BERT

BERT is an acronym for Bidirectional Encoder Representations from Transformers. It emerged as a groundbreaking language representation model in the NLP. Devlin et al. [15] comprehensively explores BERT's architecture and its impact on language understanding tasks. Unlike traditional models that rely on unidirectional context, BERT leverages the power of bidirectional transformers to capture contextual information from both left and right contexts. This unique approach enables BERT to develop a deep understanding of language semantics and syntactic structures.

The pre-training process of BERT involves training the model on a large corpus of unlabelled text, utilising two primary tasks: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). MLM involves randomly masking certain tokens in a sentence and training the model to predict the masked tokens based on the surrounding context. NSP, on the other hand, focuses on predicting whether two sentences appear consecutively in the original text. By pre-training BERT on these tasks, it learns to generate high-quality contextualised representations that capture the intricacies of language [15].

One of the remarkable aspects of BERT is its ability to be fine-tuned for specific downstream tasks with minimal modifications. Adding a task-specific output layer and fine-tuning the pre-trained BERT model on labelled data achieves state-of-the-art performance on a wide range of NLP tasks, including text classification, named entity recognition, and question answering. The versatility and effectiveness of BERT have been demonstrated through its exceptional performance on various benchmark datasets, surpassing previous models and setting new standards in the field [15].

2.2.4 DistilBERT

DistilBERT was developed in 2020 by Hugging Face. The distilled version of BERT offers several advantages over its larger counterpart. One of the key benefits of DistilBERT is its reduced size. It has 40% fewer parameters than BERT BASE, making it more lightweight and easier to deploy in resource-constrained environments [38].

In addition to its smaller size, DistilBERT offers improved inference speed. It is 60% faster than BERT BASE, allowing for quicker processing of natural language tasks. This speed-up is particularly beneficial for on-the-edge applications requiring real-time or near-real-time processing. Despite its smaller size and faster speed, DistilBERT impressively retains 97% of the performance of BERT BASE. This is achieved through knowledge distillation, implemented during the pre-training phase [38].

Knowledge distillation is a technique where a model is trained to reproduce the behaviour of a larger model. Distillation refers to transferring knowledge from a larger, more complex model like BERT BASE (the teacher) to a smaller, more efficient model like DistilBERT (the student). This knowledge transfer aims to retain the performance of the teacher model while reducing the computational resources required by the student model [38].

The student architecture in DistilBERT is designed to have a similar structure to the original BERT model. However, a few modifications have been made to make it more compact. The token-type embeddings and the pooler are removed, and the number of layers is reduced by a factor of 2. By reducing the number of layers, the student model becomes more lightweight and computationally efficient [38].

The student model in DistilBERT is initialised using a general-purpose pre-training distillation rather than a task-specific distillation. This means the student model is first trained using a distillation signal from the teacher model, which helps transfer the knowledge from the teacher to the student. This initialisation process helps the student model start with some knowledge from the teacher model, which can be further fine-tuned for specific tasks [38].

BERT BASE and BERT LARGE are two variants of the BERT model that differ in model sizes and parameters. BERT BASE has a model size of L=12 (number of layers), H=768 (hidden units), and A=12 (attention heads), with a total of 110 million parameters. On the other hand, BERT LARGE has a larger model size of L=24, H=1024, A=16, with a total of 340 million parameters [15]. With only 66 million parameters, DistilBERT presents a leaner alternative by cutting down on parameter count [38].

2.2.5 all-MiniLM-L6-v2

The all-MiniLM-L6-v2 model is an optimised variant of the MiniLM architecture, explicitly designed to balance performance with reduced model complexity. MiniLM is a streamlined version of larger transformer models developed to bring state-of-the-art NLP capabilities to environments where computational resources are limited. Through the application of knowledge distillation techniques, the all-MiniLM-L6-v2 distils self-attention knowledge from the last Transformer layer of a BERT BASE teacher model, which is particularly effective in capturing complex semantic features necessary for high-performance NLP tasks [39].

This distilled model comprises only 22 million parameters, representing a significant reduction compared to its predecessors. Refining the self-attention mechanism allows the all-MiniLM-L6-v2 to focus effectively on the most relevant parts of text data, thus facilitating an improved understanding and generation of language. This efficiency makes the all-MiniLM-L6-v2 particularly suitable for real-time soccer event detection applications [39].

With only six layers, the model dramatically cuts down on the number of parameters and operational overhead, enabling faster computations and reduced memory requirements. The combination of reduced size and retained efficacy in NLP tasks makes this model a compelling choice for advanced linguistic processing in resource-limited settings [39].

2.3 Automatic Speech Recognition

Speech Recognition has undergone transformative changes with deep learning technologies. ASR systems are designed to transcribe human speech into readable text, serving as the cornerstone for various applications, from voice-activated assistants to transcription services. The evolution of ASR can be primarily attributed to significant advancements in deep learning approaches, which have markedly improved the accuracy and efficiency of these systems. These systems can be divided into three individual approaches [46]:

2.3.1 Acoustic Phonetics Approach

Acoustic phonetics is one of the earliest foundational approaches to Speech Recognition, focusing on analysing physical properties in speech signals [46]. This discipline is instrumental in understanding how speech sounds are produced, transmitted, and perceived, laying the groundwork for pattern recognition techniques in ASR systems. By examining the frequency, amplitude, and duration of speech sounds, acoustic phonetics enables the identification and classification of phonetic units within the speech signal. This analysis is crucial for developing effective feature extraction methods, pivotal for the subsequent stages of ASR. This includes both traditional pattern recognition and modern deep learning approaches. In the context of ASR, acoustic phonetics provides the empirical basis for designing algorithms that can accurately transcribe spoken language into text by recognising and interpreting the nuanced variations in speech sounds [74]. Speech production in humans is based on two mechanisms: the vocal cords and the vocal tract system. The vocal tract is conceptualised as a tube with varying cross-sectional areas extending from the vocal cords to the mouth opening. This configuration effectively acts as an acoustic transmission system, channelling sounds generated within the vocal tract [70].

2.3.2 Pattern Recognition Approach

The pattern recognition approach in speech recognition elaborately unfolds across two principal phases: training and comparison. During the training phase, speech patterns are intricately modelled through

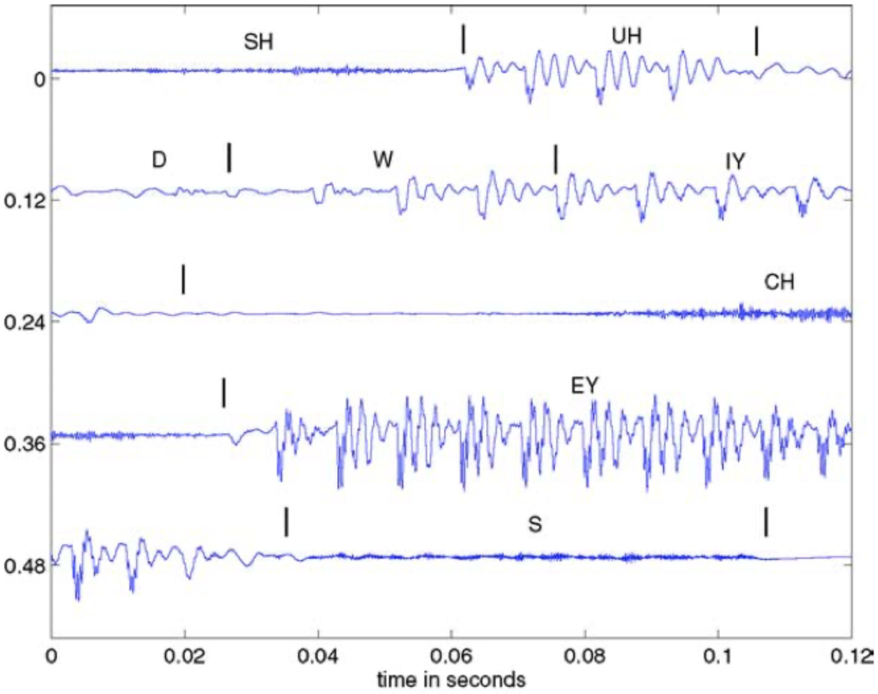


Figure 2.5: Phonetic transcription of the phrase ”Should we chase” depicted as a waveform with time in seconds on the x-axis. The y-axis in the figure represents the amplitude of the sound wave at different points in time. Each track represents a different phonetic segment corresponding to the sounds in the phrase. [70].

direct speech templates or sophisticated statistical frameworks like Hidden Markov Model (HMM), meticulously derived from a corpus of labelled samples. This foundation enables the subsequent comparison phase, where unknown speech inputs are evaluated against these predefined models to ascertain their identity based on the closest match as presented in Figure 2.5 [73].

Evolving over six decades, this methodology integrates feature measurement, rigorous pattern training, nuanced classification, and decisive logic, leveraging both template-based and stochastic models such as HMM to adeptly navigate speech variability and contextual nuances, marking a comprehensive strategy in speech recognition technology [46].

2.3.3 Artificial Intelligence Approach

The Artificial Intelligence (AI) approach in ASR integrates acoustic-phonetic and pattern recognition methods, harnessing both to enhance speech recognition systems [46]. This hybrid approach leverages acoustic-phonetic knowledge for rule-based classification and pattern recognition for dynamic analysis, utilising techniques like Dynamic Time Warping (DTW) for deterministic pattern matching and HMM for stochastic pattern matching. This fusion aims to improve speech recognition by addressing pronunciation and speaker variability, offering a more sophisticated understanding of human speech processing despite challenges in quantifying expert knowledge and integrating diverse linguistic levels [73].

Gaussian Mixture Models (GMMs) plays a critical role in statistical pattern recognition, especially in speech recognition, by providing a probabilistic framework for representing the distribution of speech features. Traditional ASR systems relied on HMM combined with GMM to represent the acoustic signal's statistical properties. They combine multiple Gaussian distributions, each representing a different underlying process or component in the data, to model complex, multimodal distributions. This flexibility allows GMMs to capture a wide range of data variability, making them highly effective for tasks requiring nuanced differentiation of speech patterns, such as distinguishing between different phonemes or speakers. Their mathematical tractability and the ability to approximate any continuous distribution with sufficient components have cemented GMMs as a staple in the speech recognition domain [82]. While the HMM-GMM approach has become a cornerstone in ASR for its simplicity and ease of training, it struggles with data on non-linear manifolds, limiting its ability to model complex speech signals. This reveals a critical balance between its utility in capturing speech's temporal aspects and the need for enhanced models to address intricate, non-linear relationships within the speech signal [73].

Transitioning from traditional ASR systems that leverage GMMs for their probabilistic framework and ability to model complex speech patterns, the field is now embracing deep learning as a pivotal advancement. Its capacity for unsupervised feature learning has begun to replace GMMs in speech recognition, introducing a shift towards architectures capable of capturing high-order data correlations and joint statistical distributions. This evolution marks a significant change from modelling speech's temporal and acoustic properties to understanding its underlying statistical structures, offering enhanced capabilities for addressing non-linear speech signal complexities [73].

It surpasses traditional methods by leveraging generative and discriminative architectures alongside hybrid models to capture intricate data patterns and improve feature coding. This evolution signifies a shift towards utilising DNNs, including auto-encoders and Deep Belief Networks, to refine the recognition process, blending generative insights with discriminative accuracy to understand speech [73] comprehensively.

2.3.4 ASR Models

ASR models are affected by numerous variables such as the number of users, isolated clear speech, vocabulary size, and spectral bandwidth. For this thesis, the ASR model will be in the category of human-machine communication [46]. One of the earliest applications of automatic speech recognition was Q-MED [19]. This method was developed for continuous speech recognition to obtain information about a patient's symptoms and functional dialogue between the patient and the program. Over two decades later, Deep Speech 2 used an end-to-end deep learning approach that outperformed crowd-sourced humans on 3 out of 4 test sets. Zhang et al. achieved state-of-the-art on the LibriSpeech test-clean dataset using a large pre-trained conformer model. They observed a 73% reduction in the Word Error Rate (WER) compared to Deep Speech 2, accomplishing a score of 1.4% [42].

The current state-of-the-art in the field of ASR is a robust speech recognition model called Whisper, which is used with large-scale weak supervision. They focus training on an extensive and diverse supervised dataset emphasising zero-shot transfer [2].

2.3.5 Whisper

Whisper is an advanced ASR model designed to accurately transcribe and understand audio from diverse sources and in multiple languages. Developed through large-scale weakly supervised learning, it utilises 680,000 hours of multilingual and multitask audio data. This extensive training allows Whisper to achieve high-quality speech recognition across various languages and tasks without the need for dataset-specific fine-tuning [2].

Whisper's architecture is optimised for processing speech in various conditions, making it highly adaptable and effective for various applications, from transcription services to voice-controlled systems. Its performance has been noted for approaching human-level accuracy, showcasing the potential of leveraging an extensive amount of varied datasets for training ASR models [2].

The Whisper structure employs a straightforward end-to-end methodology, utilising an encoder-decoder Transformer model as detailed in Figure 2.6. It processes input audio by dividing it into segments of 30 seconds each, transforming them into log-Mel spectrograms. These spectrograms are then fed into an encoder. Subsequently, a decoder generates the associated text captions and incorporates special tokens. These tokens enable the singular model to execute various functions, including identifying the language, providing phrase-level timestamps, transcribing speech in multiple languages, and translating non-English speech into English [2].

The results demonstrate that the Whisper models achieve impressive performance in zero-shot transfer. Despite being primarily trained on an English-heavy dataset, the models show competitive WER when evaluated on other datasets in different languages [2]. Zero-shot transfer refers to the ability of the models to perform well on speech recognition tasks in languages they were not explicitly trained in, without any additional fine-tuning [57].

Table 2.1 presents the various sizes of Whisper models. The smallest zero-shot Whisper model contains only 39 million parameters and achieves a WER of 6.7 on the LibriSpeech test-clean dataset, roughly comparable to the performance of the best supervised models on LibriSpeech. This indicates that even without fine-tuning or specific training on a particular language, the Whisper models can perform well in recognising and transcribing speech in various languages [2].

Model	**Layers**	**Width**	**Heads**	**Parameters**
Tiny	4	384	6	39M
Base	6	512	8	74M
Small	12	768	12	244M
Medium	24	1024	16	769M
Large	32	1280	20	1550M

Table 2.1: Architecture details of the Whisper model family [2].

When compared to human performance, the zero-shot Whisper model (Whisper Large V2) demon-

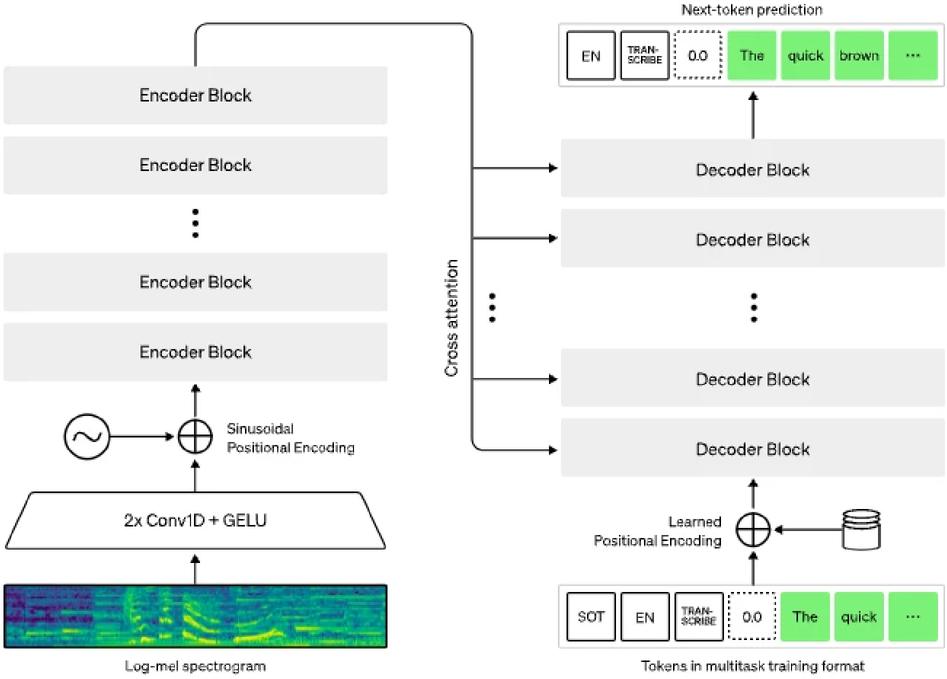


Figure 2.6: An overview of the architecture of the Whisper. Presenting the pipeline from start to end [67].

strates a level of accuracy and robustness that closely matches human capabilities. This suggests that the models could approach human behaviour in speech recognition tasks, primarily when evaluated in zero-shot settings. It's important to note that the paper emphasises the need for further research and improvement, particularly in increasing the amount of training data for rarer languages. By addressing the biases in the training dataset and incorporating more data from diverse languages, the performance of the Whisper models in zero-shot transfer scenarios could be further enhanced [2].

2.3.6 Challenges in ASR

Understanding human speech poses a unique set of challenges. Humans leverage contextual, grammatical, and speaker-specific knowledge in conversation, predicting words and sentences with nuanced comprehension.

People use knowledge about the subject and the speaker while simultaneously interacting in a conversation. Sentences are constructed with consideration of context and grammatical structure. People use world- and speaker knowledge to predict words and sentences accurately. While there is a lack of comprehension between humans and statistical computer models, the requirements for information to measure up to humans are still unknown. Issues also emerge when considering that spoken language is not equal to written language. People use body gestures, facial impressions, less consistency, emotions, etc., when speaking. Background noise is very common when listening to speech. The noises could significantly impact speech recognition depending on the intensity. Speaking style and speaker variation could be challenging when receiving new data [46]. These are just some of the issues occurring when transcribing raw audio.

2.4 Action Detection

Before delving into the intricacies of terminology, it's essential to establish a foundational understanding by clarifying the basics: What constitutes an event? The prestigious Oxford Dictionary describes an event as "Something that happens or takes place, esp. something significant or noteworthy". Defining the precise start, end times and duration of events in soccer can be subjective, making it difficult to achieve consistent and objective agreement.

Action detection, also known as action spotting or event detection, involves identifying the temporal interval and recognising the action within that time frame. Successful event detection entails accurately classifying the action and determining the temporal interval when the action occurred [58]. Action spotting is a critical area of computer vision research with applications in sports analysis, especially soccer. It aims to identify and localise specific events within soccer video sequences automatically. This review investigates the existing literature on action spotting considering SoccerNet V1 and V2, exploring various techniques and remaining challenges.

2.4.1 Datasets

This section provides an overview of the relevant research and developments that form the foundation of this thesis, tracing the progression from early datasets to the current state-of-the-art. It delves into advancements across various modalities, such as video and audio, focusing on the domains of action detection. This review aims to highlight the major strides made in this field, laying the groundwork for the unique contributions of this thesis.

In the field of sports analysis, various datasets have been made available for research purposes. Large and high-quality datasets with detailed annotations are critical for advancements in action detection. Early datasets such as Hollywood2 [62], HMDB [12], and UCF101 [59] laid the groundwork for event detection by providing extensive datasets. These datasets, introduced over a decade ago, started with a baseline accuracy of 44.5% for UCF101 and 23% for HMDB. They have since achieved state-of-the-art accuracies of 99.6% and 88.1% [26], respectively. Later on, datasets like Sports-1M [3], Youtube-8M [32], and Moments in Time [27] have been released, offering even larger volumes of information and facilitating further research in this domain. Moment in Time features one million human-annotated three-second videos showcasing dynamic events. This dataset contains three modalities: spatial, auditory, and temporal. Sports-1M corresponds to 1 million YouTube videos and 487 classes, which only entail sports-related resources. In 2016, YouTube-8M was released with 8.2 million YouTube videos, leveraging metadata for their automatic annotations.

Tran et al. [10] propose Channel-Separated Convolutional Networks (CSNs) for efficient video classification on the Sports-1M dataset. CSNs achieved superior performance while being 2-3 times more efficient. They achieve state-of-the-art scores of 75.5% and 92.8% for video@1 and video@5, respectively.

The Kinetics series started in 2017 with the Kinetics-400 [40], then the Kinetics-600 [17] came with a 50% increase in classes and a 60% increase in videos. Two years after HACS [14] was released with 1.5 million manual annotated clips. This dataset outperformed Kinetics-600, Moments in Time, and Sports1M as a pre-training dataset for action recognition. In 2022, Kinetics-700 [18] was released containing 650k YouTube videos with a 10-second time frame. The latest in this series is AVA-Kinetics [4] originating from the AVA dataset [8] and Kinetics-700 by providing AVA-style human action and spatio-temporal annotations.

Introduced in 2018, SoccerNet V1 [35] is a comprehensive dataset encompassing 500 soccer matches from six different leagues, spanning the years 2014 to 2017. Events ranging from goals, cards, and substitution over 764 hours of content. The manual annotations are defined within a one-second resolution, with an average of one event every 6.9 minutes. SoccerDB [43] increases the class count to 11 with 668.6 hours. They reveal tasks like object detection, action recognition, temporal action localisation, and highlight detection. In 2021, SoccerNet V2 [1] was released, expanding the original SoccerNet V1 dataset by adding 300,000 annotations to the 500 games. This update included 16 times more timestamps and 14 additional classes. Introducing new tasks like camera shot segmentation boundary detection and replay grounding. The summary of the significant datasets can be seen in Figure 2.2. SoccerNet V3 was published in 2022, presenting an upgrade to the SoccerNet V2 dataset with extensive multi-view spatial annotations that advance sports video analysis. This dataset captures actions with different camera viewpoints, resulting in 1,324,732 annotations on 33,986 soccer images. This dataset is one of the largest in sports video analysis and is also easily accessible through the open-source platform GitHub.

2.4.2 SoccerNet V1

Giancola et al. [35] provides two models for this research: a classifier for temporal segments and an event spotter. The approach relied solely on video data and utilised video chunk classification and action spotting techniques with several pooling methods. After a comprehensive analysis, the research used NetVLAD [30]. The temporal segment classifier operates with an impressive mAP of 67.8%, which is significant for classifying one-minute video chunks into Goals, Cards, and Substitutions categories. The event spotter model, crucial for the action spotting task, demonstrates a baseline performance with an AmAP of 49.7% across various tolerances δ ranging from 5 to 60 seconds. This performance

Datasets	Context	Classes	Duration (hrs)	Actions
Ava-Kinetics [4]	Movies	487	-	-
HACS [14]	Human	200	2,166	139,000
Sports-1M [3]	Sports	487	-	-
SoccerNet V1 [35]	Soccer	3	764	6,637
SoccerDB [43]	Soccer	11	668.6	37,715
SoccerNet V2 [1]	Soccer	17	764	110,458

Table 2.2: Comparative summary of relevant datasets across domains.

metric is particularly noteworthy as it illustrates the model's proficiency in pinpointing the exact occurrence of an action within the specified tolerance levels.

In 2020, Vats et al. [24] introduced their multi-tower temporal convolutional network by utilising parallel 1D convolutional branches with varying receptive fields. These branches process pre-extracted features from a 2D Convolutional Neural Network (CNN) at different temporal resolutions, allowing the network to capture temporal information across various durations. This is crucial for pinpointing event boundaries with limited annotation precision. Evaluated on the SoccerNet V1 dataset, the method achieves a state-of-the-art AmAP of 60.1%, demonstrating its effectiveness in handling imprecise annotations.

The same year, Cioppa et al. [5] created a context-aware loss function for action spotting in soccer videos. This approach integrates with a dual-module segmentation and action detection framework inspired by YOLO [21]. The model leverages time-shift encoding to capture temporal context around actions. This approach improved AmAP to 62.5%, highlighting the effectiveness of incorporating context for precise action localisation.

Later that year, Vanderplaetse et al. [81] proposed a multimodal approach for action spotting and classification in soccer videos. They trained the pre-trained ResNet [23] model on ImageNet [16] and VGG [76] model on the AudioSet [20] data to extract features from video and audio data, respectively. These features were then input into various deep learning architectures designed to merge and process multimodal information effectively. They hypothesised that audio cues from the crowd and other in-game sounds could provide significant contextual information that is beneficial for action recognition. Experiments explored different fusion techniques, with the best performance achieved by merging audio and visual features before the final fully connected layers. This multimodal approach yielded a 7.43% absolute AmAP improvement for classification and a 4.19% improvement for spotting, particularly benefiting goal detection tasks where audio cues significantly enhanced the ability to distinguish actual goals from attempts due to crowd reactions.

Tomei et al. [28] introduce RMS-Net, a model that combines regression and masking within a single framework. It takes short video clips as input and predicts both the event type and its precise timing within the clip. Unlike prior methods, RMS-Net avoids relying on a central frame for event localisation, leading to more accurate results. This lightweight network consists of two branches: classification and regression. RMS-Net utilises a 2D backbone for feature extraction from video frames, followed by 1D convolutions to integrate temporal information. Evaluated on SoccerNet, it achieved a 65.5%

AmAP improvement over the state-of-the-art using the same features. Fine-tuning with a more robust backbone increased AmAP to 75.1%.

2.4.3 SoccerNet V2

In 2021 Giancola et al. [56] propose NetVLAD++, a temporally-aware feature pooling method for action spotting. It improves upon NetVLAD by considering the temporal context. The video is split before and after actions, allowing the model to learn distinct vocabularies for these segments, capturing the unique precursors and aftermaths of different events. This approach achieves a strong mAP of 53.4% on the SoccerNet V2 test set. However, the stricter tight-Average mean Average Precision (t-AmAP) metric for the challenge set uses a tolerance δ window of 1 to 5 seconds for correct localisation [33], resulting in a t-AmAP of 9.91%. This highlights the challenge of precise temporal localisation in action spotting tasks.

The SoccerNet challenges were introduced later that year, featuring competition in action spotting and replay grounding. Xin et al. [41] proposed a two-stage method for action spotting. Stage 1 extracts high-level features by fine-tuning multiple action recognition models. Stage 2 utilises a transformer-based module to process these features and predict event probabilities throughout the video. This method achieved a state-of-the-art AmAP of 73.77% and a t-AmAP of 49.56% on SoccerNet V2, highlighting the effectiveness of combining feature extraction with advanced temporal modelling for action spotting in sports videos.

SoccerNet challenges gained more traction in 2022, this year focused on retrieving action timestamps in long untrimmed videos considering action spotting [33]. Soares et al. [77] propose a novel action spotting method for soccer videos using dense detection anchors for precise temporal localisation. Their model utilises a 1D U-Net and Transformer encoder to capture large temporal context and fine-grained features. Advanced training techniques address limited data and low inductive bias. On SoccerNet V2, their method achieves state-of-the-art t-AmAP of 60.7% test and 67.81% [34] on the challenge set, with temporal displacements improving precision.

The latest SoccerNet challenge presented even higher mAP values for action spotting. The winner of the 2023 challenges was MEDet, which explores CNNs and Transformers for local and global features, proposing three encoders: Conv-based, Transformer-based, and a hybrid. To adapt to diverse actions, MEDet assigns actions to encoder groups. It also improves feature extraction with a multi-scale feature pyramid network. The decoder uses a CNN head for classification and a Trident head for regression, predicting labels and boundaries. At inference, results are merged and filtered via Non-Maximum Suppression (NMS), acting like an ensemble model. They achieved a score of 71.31% on the t-AmAP and 78.56% on the AmAP [6].

COMEDIAN [22] is the current state-of-the-art for action spotting, achieving an impressive t-AmAP of 73.1% and a AmAP of 77.6% on SoccerNet V2. This innovative method employs a three-stage pipeline. First, a spatial transformer pre-trains on unlabelled video data using self-supervised learning (SSL) to capture local features in short snippets. Next, a temporal transformer builds upon this by learning global context through knowledge distillation from pre-computed features. Finally, both transformers are fine-tuned for action spotting. This approach leverages the strengths of SSL and knowledge distillation for superior temporal localisation and classification of soccer actions.

Dataset	**Model**	**Tight-AmAP**		**Average-mAP**
		Challenge	**Test**	
SoccerNet V1	Vanderplaetse et al. [81]	-	-	56.0
	Vats et al. [24]	-	-	60.1
	Cioppa et al. [5]	-	-	62.5
	Tomei et al. [28]	-	28.8	**75.1**
SoccerNet V2	Giancola et al. [56]	9.91	11.5	53.4
	Xin et al. [41]	49.56	47.05	73.77
	Soares et al. [77]	67.81	65.1	-
	MEDet [6]	71.31	-	78.56
	COMEDIAN [22]	**68.38**	**73.1**	**77.6**

Table 2.3: AmAP and mAP on the SoccerNet V1 and V2 test sets, comparing action spotting performance of various approaches.

Despite significant progress, action spotting in soccer videos remains an active area of research with several challenges. Recognising rare events, handling cluttered scenes, and dealing with viewpoint variations. These are persistent issues from the computer vision domain. Most approaches are unimodal, relying solely on video features for event detection. Some approaches are multimodal, incorporating audio to enhance detection accuracy.

Considering an alternative approach, action detection could benefit from leveraging preprocessed text derived from an ASR model. This shift to using text as the primary input could mitigate some of the inherent difficulties computer vision techniques face and be less computationally demanding. Refining the input data to be more concrete and directly related to the game audio and metadata makes it possible to reduce ambiguity and improve detection accuracy. Employing such a method can potentially increase the t-AmAP score beyond the current state-of-the-art of 73.1%

2.5 Chapter Summary

This chapter provides a foundational overview for understanding the research explored further in this thesis, encompassing key areas like machine learning, NLP, ASR, action spotting, and a literature review.

The section begins with an exploration of machine learning, focusing particularly on supervised methods which involve training a model on labelled data. It delves into classification techniques within machine learning, explaining binary and multi-class classification methods and the role of datasets in training and testing these models.

In natural language processing, the chapter introduces text classification and the use of Transformer architectures. Automatic speech recognition is discussed with a focus on its development from basic acoustic models to sophisticated neural networks like the Whisper model, which can process speech in various languages and perform with near-human accuracy.

Lastly, the chapter discusses action detection. This concept aims to identify and localise specific events within a temporal interval, exploring various datasets and techniques and highlighting advancements and challenges in effectively capturing and analysing these events. This chapter aims to provide a deeper understanding of the crucial theoretical aspect necessary in this thesis.

Chapter 3

Dataset Construction and Exploration

The field of sports analytics thrives on exploring diverse data sources to gain deeper insights into annotating soccer games. This section presents a novel dataset for supervised text classification tasks within soccer. This dataset originates from the well-established Soccernet V2 collection of raw soccer video game footage.

The key innovation lies in transforming audio data into a structured textual format suitable for analysis. This is achieved by leveraging OpenAI's Whisper Large V1 ASR technology. Applying Whisper to the soccer video game footage bridges the gap between visual and textual data, enabling the extraction of valuable commentary and game metadata for further analysis.

The resulting datasets take the form of a well-organised `DataFrames`, tailored explicitly for supervised text classification tasks. This procedure uses the transcription from SoccerNet-Echoes, which included the 550 games from the SoccerNet dataset, an addition of 50 games from the 2019/2020 La Liga season [54]. Further developments include classification framework design, unlocking the significant potential for research advancements in sports analytics and NLP. By investigating these unique datasets, researchers can explore a new avenue for analysing soccer video games through the lens of text classification, possibly leading to novel insights and a deeper understanding of the sport.

3.1 SoccerNet Datasets

SoccerNet V2 is a comprehensive dataset designed for soccer-related video understanding. It is considered one of the largest datasets in terms of overall size and the number of soccer videos it encompasses. The dataset contains approximately 300,000 manually annotated timestamps, which are temporally anchored in the 764 hours of 500 soccer games from the original SoccerNet V1 dataset [1, 35].

To enhance the quality of the annotations, 33 annotators who are frequent observers of soccer were employed for the annotation process. The annotations are divided into actions, camera shots, and replays. These annotations were validated by observing a high level of consensus among the annotators, ensuring their accuracy and reliability [1].

SoccerNet V2 stands out due to its dense annotations compared to other soccer datasets, and it even rivals the largest fine-grained generic datasets in terms of density and size. This density of annotations enables deep supervised learning at scale, making it an ideal resource for various tasks such as action retrieval, highlight production, and replay grounding [1].

By releasing SoccerNet V2, the aim was to push the boundaries in understanding holistic broadcast soccer videos. The dataset extends the tasks of action spotting, camera shot segmentation, and boundary detection and introduces the novel task of replay grounding. Soccernet provides codes to reproduce experiments to facilitate further research and benchmarking. It also features public leaderboards and hosts challenges open to the research community [6, 33].

League	14/15	15/16	16/17	Total
EN - EPL	6	49	40	95
ES - LaLiga	18	36	63	117
FR - Ligue 1	1	3	34	38
DE - BundesLiga	8	18	27	53
IT - Serie A	11	9	76	96
EU - UEFA CL	37	45	19	101
Total	81	160	259	500

Table 3.1: Overview over all the leagues and games from the SoccerNet V2 datasets [1].

The dataset is composed of soccer games from six main European leagues, as detailed in Table 3.1. These leagues include the English Premier League (EPL), Spanish La Liga, French Ligue 1, German Bundesliga, Italian Serie A, and UEFA Champions League tournament. The dataset covers three seasons from 2014 to 2017 [1, 35]. Including games from multiple leagues and seasons makes the dataset diverse and representative of professional soccer matches.

The SoccerNet V1 dataset was published in 2018 with a rich collection of annotations, with 6,637 temporal annotations automatically parsed from online match reports. These annotations are categorised into three main classes of events: Goal, Yellow/Red Card, and Substitution. These annotations are manually adjusted to a one-second resolution to guarantee precision by anchoring them at specific timestamps according to established soccer rules [35]. The SoccerNet V2 updated the amount of temporal annotated actions to 110,458, which indicates a 17x increase. A total of 17 different classes of actions have been identified and annotated. These classes represent the most important actions that occur in soccer matches [1].

Classes from SoccerNet V2

Goal: It is the game's ultimate objective, where a player successfully puts the ball into the opponent's net, resulting in a point for their team.

Shot on target: This action occurs when a player attempts to score by kicking or striking the ball towards the opponent's net.

Corner: A corner kick is awarded to a team when the ball goes out of play over the goal line, last touched by the defending team. The attacking team then takes a kick from the corner arc, aiming to create a scoring opportunity.

Foul: A foul occurs when a player violates the game's rules, such as tripping, pushing, or using excessive force against an opponent.

Yellow Card: The referee shows A yellow card to caution a player for unsporting behaviour, time-wasting, foul, or other rule violations. It serves as a warning to the player.

Red Card: The referee shows A red card to dismiss a player from the game due to serious misconduct, violent conduct, or accumulating two yellow cards.

Offside: Offside occurs when an attacking player is nearer to the opponent's goal line than both the ball and the second-to-last defender when the ball is played to them.

Substitution: This action occurs when a player is replaced by another player. Substitutions are made during the game to replace tired or injured players or to introduce fresh tactics and strategies.

Penalty: A penalty kick is given to the attacking team when a defending player commits a foul inside their penalty area. It provides a direct scoring opportunity from a designated spot.

Shot off target: This action indicates a missed shot or opportunity to score. It occurs when a player fails to successfully put the ball into the opponent's net despite having a clear chance to score.

Clearance: It refers to a defensive action where a player clears the ball from their half. Clearances are made to prevent the opposing team from creating scoring opportunities by kicking or heading the ball away from their own goal.

Ball out of play: This occurs when the ball leaves the field of play entirely, either by crossing the goal line or the touchline.

Kick-off: This action starts or restarts the game at the beginning of each half and after every goal scored. The ball is placed at the center spot, and a player from one team taps it to initiate play.

Indirect free-kick: Awarded after certain infractions that involve non-dangerous play or technical violations by a player. The ball must touch another player before a goal can be scored.

Direct free-kick: Given following a foul or infringement that prevents a clear opportunity or involves physical contact. The ball can be shot directly into the opponent's goal without touching another player first.

Yellow to red card: Issued to a player who has received two yellow cards in the exact match, resulting in a red card and ejection from the game.

Throw-in: Awarded when the ball completely crosses the touchline. It is taken from the point where it crosses the line, and the player must throw the ball using both hands from behind and over the head while keeping both feet on the ground.

3.2 Whisper Application

This subsection details the origins and implementation of SoccerNet-Echoes utilised in this study, derived from the work of Gautam et al. [54]. The construction of SoccerNet-Echoes commenced with selecting 550 raw soccer matches containing ten different languages from SoccerNet V1. Each game was processed to extract spoken commentary and generate accurate text transcripts by leveraging the Whisper Large V1 model. All the games were translated into English with the help of Google Translate. Whisper's advanced algorithms guaranteed high transcription accuracy, adeptly capturing nuanced commentary with sport-specific terminologies and dynamic play-by-play descriptions. These transcribed ASR texts established a foundational layer for in-depth analysis, facilitating a comprehensive examination of match events in the context of verbal narratives.

This work not only bridged the gap between raw video data and structured analytical datasets but also set a precedent for future sports analysis, leveraging speech recognition technology to enhance the understanding of soccer dynamics. In the presented Figure 3.1, there is a provided layout displaying the structure of the SoccerNet-Echoes dataset.

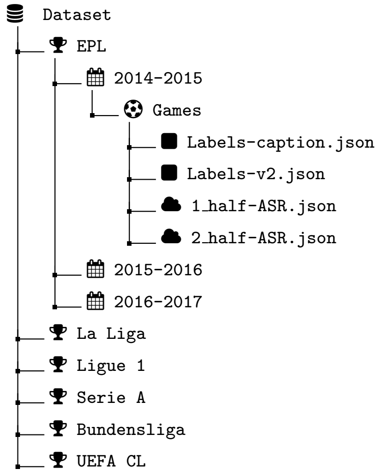


Figure 3.1: SoccerNet-Echoes dataset structure [54].

3.3 Soccer Text Classification Dataset (STCD)

This section outlines the development of a project-specific Pandas `DataFrame`, leveraging the SoccerNet-Echoes ASR dataset to structure data for in-depth analysis [54]. This systematically organises match events and correlates them with ASR commentary. This approach becomes a significant asset for the academic community, encouraging independent and innovative sports analysis research. It provides a detailed view of match events and commentary, fostering the discovery of new insights. The following paragraphs describe the efficient process of compiling this STCD on the SoccerNet V2 dataset, illustrating the synergy between thorough data collection and careful data implementation.

3.3.1 Step 1. Syncronising Files

The process began with a comprehensive search through the SoccerNet-Echoes dataset to locate files related to soccer matches, presented in Figure 3.1. This search aimed to identify files based on their naming conventions, which indicate whether they pertain to the first or second halves of the matches or if they contain relevant game metadata. A function automated this step, as manually handling the search for 500 games would be impractical. This thesis uses only the 500 original games from the SoccerNet V1 dataset with the labels from the SoccerNet V2 [1, 35] from the SoccerNet-Echoes. Consisting of 500 files for the first half, 500 for the second half, and 500 JSON files with game metadata, totalling approximately 1500 files to be synchronised. Such an approach in the initial phase ensured that all subsequent analyses relied on accurately identified and gathered data.

3.3.2 Step 2. From JSON to DataFrame

After identifying the relevant files, the following step involved generating `DataFrame` from the Labeled JSON files, using targeted filtering for specific categories such as Corners, Fouls, and Goals. This targeted filtering was essential, enabling the segregation of match segments that offered significant value for analysis. Subsequently, these segments were refined further to abstract the key elements of match commentary and events into organised lists. This method effectively captured the evolving nature and sequence of soccer matches in textual representation, aligning with academic research standards.

The heart of the dataset construction lies in the creation of a `DataFrame` from these extracted ASR texts. A comprehensive temporal and textual mapping was achieved by detailing each event's start and end times alongside the ASR text. This mapping not only facilitated a granular analysis of match events but also enabled the alignment of textual data with specific moments within the match, enhancing the analytical depth of the dataset.

To ensure consistency and utility across the dataset, game times noted in various formats were standardised into total seconds, providing a uniform temporal reference. This standardisation was crucial for the next step: associating ASR texts with specific timestamps. By filtering and combining ASR texts based on their temporal markers, each text snippet was carefully aligned with corresponding moments in the game relative to crucial events identified through the game metadata. This approach enabled the creation of datasets based on a specific time standard deviation, considering the ASR text. Utilising this method, two distinct datasets were generated: A 15-SSDD and 30-SSDD.

Figure 3.2 illustrates an example of classification applied to a single ASR text sample. This sample was retrieved from the soccer match between Paris Saint-Germain (PSG) and Toulouse on November 7, 2015. The figure displays the first annotated action from this game, implemented by SoccerNet V2. The sample is classified into three categories, as described in STCD. Red dotted arrows indicate the incorrect class, while the green line denotes the correct class for the sample. This text sample shows the raw commentary of a Foul incident.

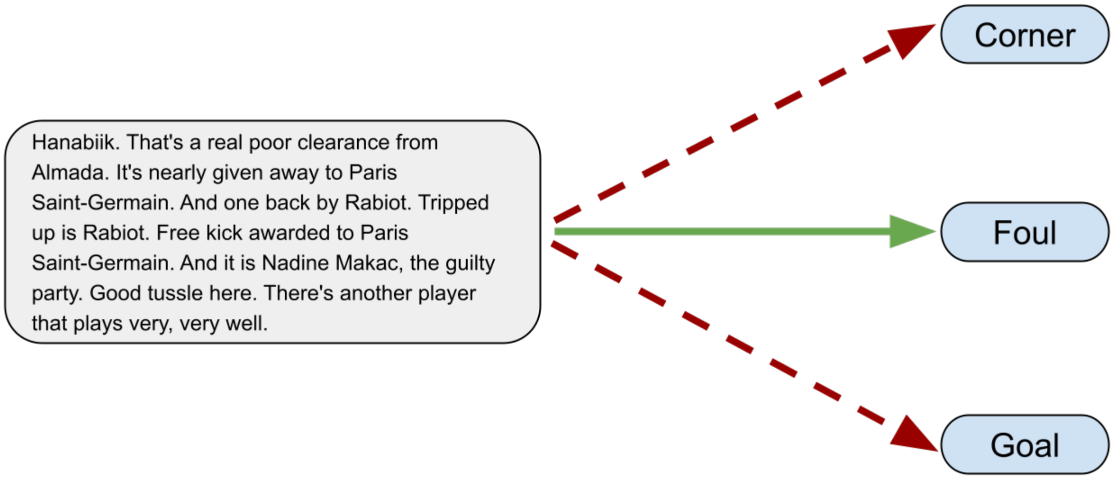

Hanabiik. That's a real poor clearance from Almada. It's nearly given away to Paris Saint-Germain. And one back by Rabiot. Tripped up is Rabiot. Free kick awarded to Paris Saint-Germain. And it is Nadine Makac, the guilty party. Good tussle here. There's another player that plays very, very well.
Corner
Foul
Goal

Figure 3.2: One ASR text sample for the Foul class considering the 15-SSDD.

3.3.3 Step 3. Integration

The culmination of this process used a function to combine all of the processed data into a unified `DataFrame`. This final product integrates the rich textual narratives from ASR texts with the structured timeline of soccer matches. It provides a detailed canvas for exploring the intricate relationships between match events and their verbal annotations. Through this systematic and detailed approach, the construction of the dataset facilitates advanced sports analytics but also stands as a testament to the power of structured data in uncovering new insights within the dynamic realm of soccer matches. All steps were conducted using `Python` [72], utilising various functions and coding procedures.

Label	**Count**
Ball out of play	29568
Throw-in	17625
Foul	10822
Indirect free-kick	9692
Clearance	7321
Shots on target	5383
Shots off target	4908
Corner	4505
Substitution	2634
Kick-off	2369
Direct free-kick	2078
Offside	1940
Yellow card	1884
Goal	1573
Penalty	154
Red card	50
Yellow to red card	38
Total	**102 544**

Table 3.2: The number of events for all the classes in the STCD retrieved from the SoccerNet V2 [1].

Table 3.2 provides a breakdown of all 17 classes from the SoccerNet dataset. Originally, the dataset was expected to include 110,458 actions [1]. The SoccerNet-Echoes dataset had occasionally failed to capture clear audio inputs with the whisper model, resulting in empty values or inadequate text for some significant actions. Consequently, the total number of actions resulted in precisely 102,544 for the STCD on the whole SoccerNet V2 dataset.

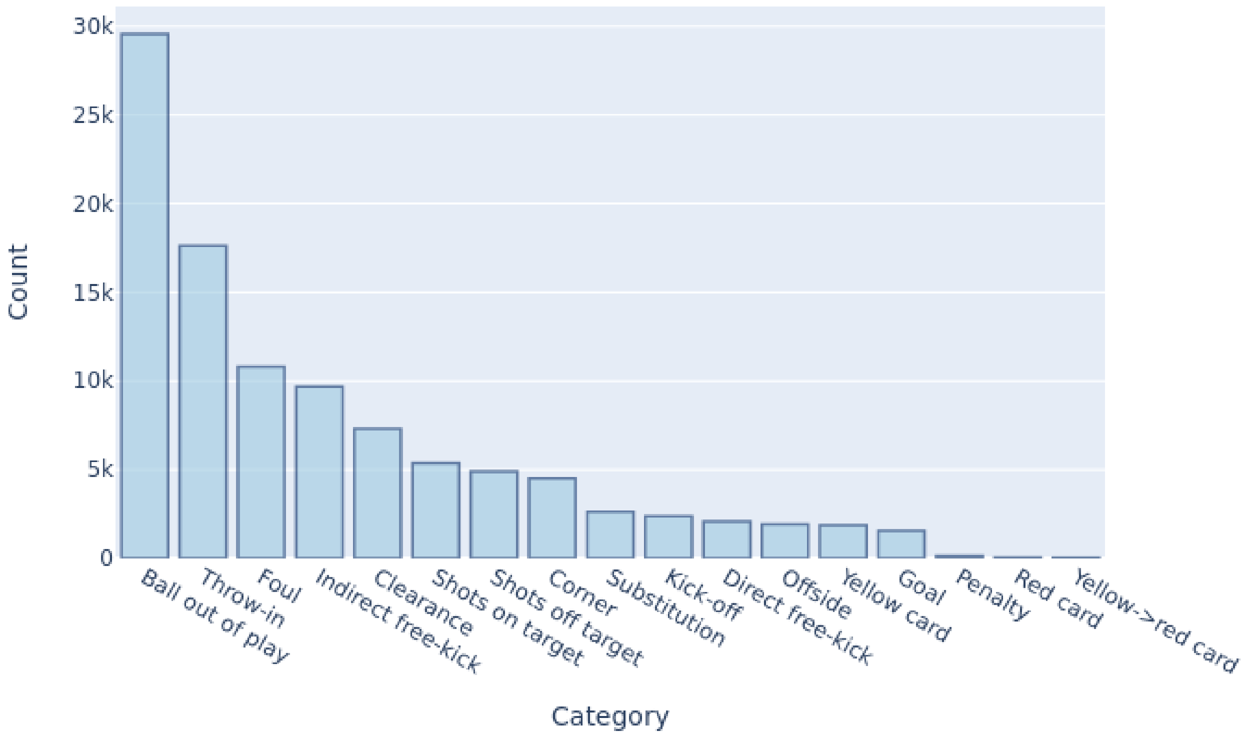


Figure 3.3: Bar plot representation of the 17 classes in SoccerNet V2.

Figure 3.3 illustrates the dataset structure across different classes. This representation reveals a class imbalance in the `DataFrame`, where the largest class is almost 800 times larger than the smallest. This substantial disparity indicates that certain events or actions within soccer matches are much more common or are captured more frequently than others, which can skew the analysis if not properly addressed. Given this imbalance, it is crucial to use appropriate metrics when evaluating the data. Traditional metrics like accuracy may not provide a true reflection of model performance in this context.

In addition to the bar plot that details the numerical distribution, Figure 3.4 presents a pie chart with a percentage layout for the STCD. This pie chart offers an intuitive understanding of each class's proportion within the total dataset.

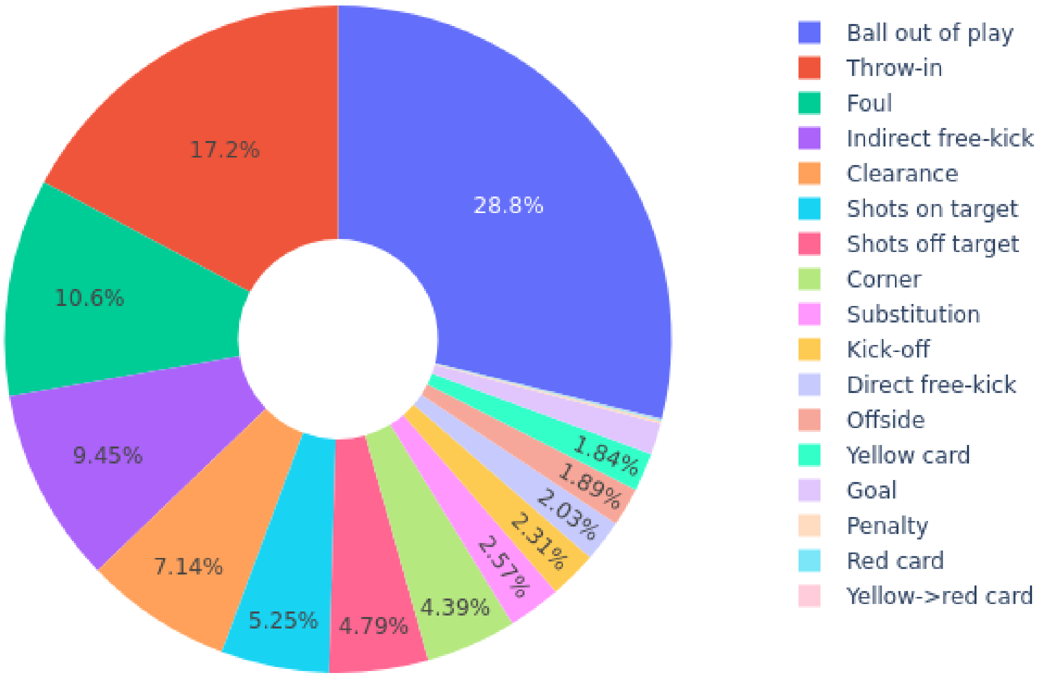


Figure 3.4: Percentage pie chart representation of the 17 classes in SoccerNet V2.

In the presented Figures 3.5 and 3.6, bar plot visualisations are provided for the three classes extracted from the STCD. Since more ASR information is provided in the 30-SSDD, fewer instances of missing text in the samples result in more actions within the dataset.

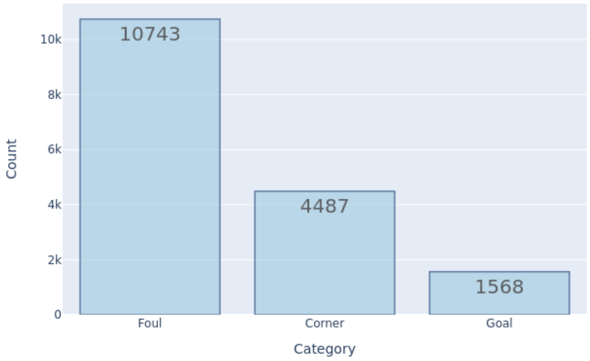


Figure 3.5: Bar plot distribution of the classes for 15-SSDD.

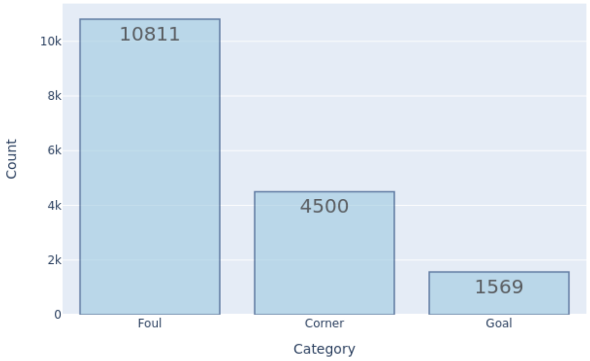


Figure 3.6: Bar plot distribution of the classes for 30-SSDD.

3.4 Chapter Summary

This chapter details the construction and exploration of three distinct datasets and investigates the SoccerNet datasets used for this implementation. The processes are divided into a Whisper application and a STCD. The SoccerNet-Echoes Whisper application is structured into multiple ASR text files accompanied by game metadata files.

The STCD is described in detailed steps and includes nearly all the actions derived from the SoccerNet V2 dataset. The description consists of all steps from the initial setup to final integration and provides an example of how a dataset sample is classified.

The 15-SSDD and 30-SSDD are both derived from the procedures outlined in the STCD and feature a reduced number of classes, including only Foul, Goal, and Corner. The following chapter will elaborate on the methodology used to derive the results later in this thesis.

Chapter 4

Methodology and Implementation

This chapter's primary objective is to explore the methods developed to address the research question. Building on the previous chapter, this section presents the detailed steps from the STCD to the final result through a thorough procedure. This includes an overview of the proposed methodology, data preprocessing, model selection, hyperparameter search, and addressing hardware specifications. The aim is to enable readers to reproduce the results outlined in the subsequent chapter.

4.1 Overview and Proposed Pipeline

Within the methodology of this thesis, a comprehensive pipeline describes the approach from data acquisition to model evaluation, facilitating a rigorous examination of audio data within the soccer domain. The pipeline's starting point is marked by audio input, representing raw data captured from soccer matches. The initial stage utilises SoccerNet V2 to segment the audio stream, followed by Whisper to transcribe the segmented audio. This transcription process converts the auditory information into a textual format, preparing it for analytical treatment. After transcription, the text data is structured into `DataFrames` as seen in Chapter 3. This sets the stage for preprocessing, where the data is carefully cleaned to ensure quality and unbiased input for the machine learning models. These steps can be seen in the Figure 4.1.

With a focus on exploring the nuances of LLM, a selection of state-of-the-art models is employed: DistilBERT, BERT BASE, BERT LARGE, and all-MiniLM-L6-v2, each specified in Table 4.1. DistilBERT and all-MiniLM-L6-v2 have fewer layers and lower parameter counts, offering a leaner approach to processing, while BERT BASE and BERT LARGE provide deeper and more complex architectures, suitable for a comprehensive understanding of the language structures within. The pipeline's final stage is model evaluation, using various metrics to measure performance against the challenges of an imbalanced dataset typical in sports data, where some events are rarer than others.

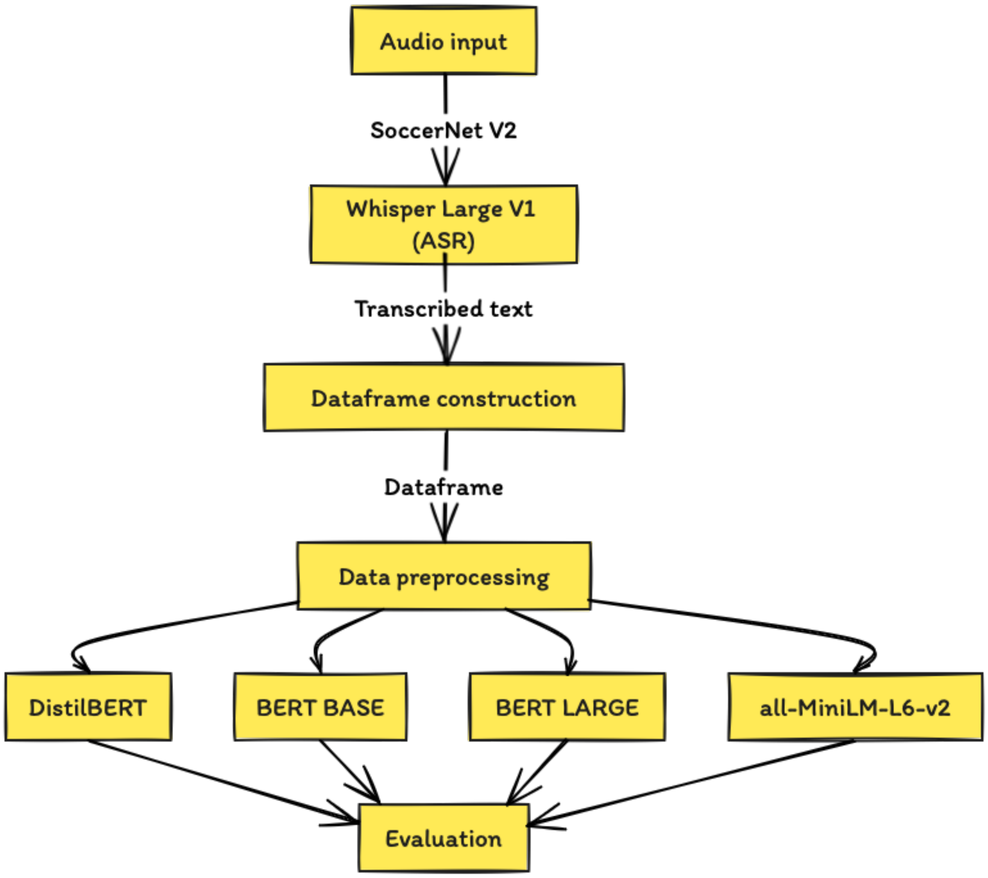


Figure 4.1: End-to-end pipeline structure.

Table 4.1: Detailed Model Specifications.

Model	**Specifications**			**Training Data**
	Layers	**Hidden Size**	**Parameters**	
BERT LARGE [15]	24	1024	340M	BooksCorpus [44] and English Wikipedia
BERT BASE [15]	12	768	110M	BooksCorpus [44] and English Wikipedia
DistilBERT [38]	6	768	66M	Distillation of BERT BASE
all-MiniLM-L6-v2 [39]	6	384	22M	Distillation of BERT BASE

4.1.1 Dataset Preprocessing

This section presents the steps to preprocess the data, an important step before text classification. A detailed explanation is provided on removing missing values, eliminating stopwords, tokenising, and splitting the dataset into train, test and validation sets.

Removing Missing Values

The first step in preprocessing the 15-SSDD and 30-SSDD from Chapter 3 involves checking for missing values across all classes in the dataset. Several options are available to address this situation if a class is missing values. The most fundamental approach would be to create a `Python` function that loops through the data and removes the missing values. Another implementation could be to impute values for the missing data. However, this may not be necessary as only a few missing samples exist in these datasets.

Eliminating Stopwords

Eliminating stopwords is a crucial next step in the data preprocessing sequence. Stopwords are frequently occurring words in any language that typically do not add significant semantic value to the text and can be removed without altering the intrinsic meaning of the text. Examples include common words such as ”the”, ”is”, and ”at”. For NLP tasks like text classification, removing these stopwords can dramatically reduce the dataset's dimensionality and improve model performance [71]. In Figure 4.2, a bar plot over the most frequent words in the class Foul is presented for 30-SSDD. The presence of many stopwords is evident for this class, and it's similar to the other classes. The word ”the” is mentioned a staggering 70,000 times for this class. Considering this Figure, the emphasis on removing stopwords is imminent.

To execute the removal of stopwords, the data is processed through a series of transformative steps facilitated by the Natural Language Toolkit (`NLTK`), a powerful `Python` module widely used in text processing and linguistic analysis. The initial phase involves text normalisation, where all characters are converted to lowercase to ensure uniformity. Such standardisation is essential, enabling the model to recognise and treat the same words consistently, irrespective of case variations. Tokenisation is then performed on the text, decomposing it into individual words (tokens). This step utilises the `word_tokenize` function from the `NLTK` module. Each word is then scrutinised against a comprehensive list of stopwords obtained from the set within the `NLTK` corpus [47, 61].

All punctuation, apostrophes, hyphens, and numerals, which can be considered as another form of ’noise’ in the context of linguistic data, are removed. This is achieved by creating a set comprising these values, thereby further purifying the text data to contain only meaningful lexical items. The dataset is effectively preprocessed through these detailed steps facilitated by the functionalities of the `NLTK` module.

Removing these elements streamlines the text, enhancing the focus and effectiveness of subsequent analysis. This process purifies the data and tailors it for precise interpretation in sports commentary texts. Through these preprocessing steps, the dataset is thoroughly prepared for the comprehensive

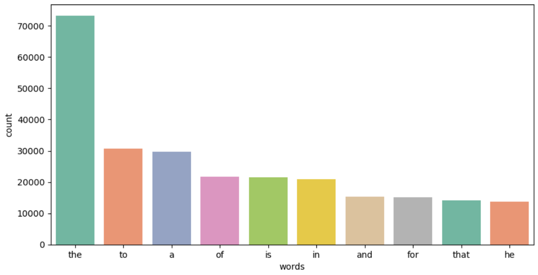


Figure 4.2: Unfiltered word count for class Foul considering 30-SSDD.

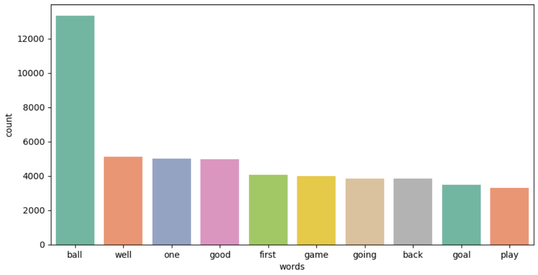


Figure 4.3: Filtered word count for class Foul considering 30-SSDD.

analysis central to the text classification goals of this thesis. Figure 4.3 displays a filtered representation of the most frequent words for the class Foul. The words in this Figure hold more importance than those in the unfiltered Figure displayed above.

Tokenization

Tokenisation is a foundational step for all four models in Table 4.1. The input text is segmented into individual tokens representing words or meaningful subword parts, capturing the lexical semantics within the processing layer. Acts as a bridge between human language and the numerical world in which models operate. All the models employ WordPiece embedding with a 30,000 token vocabulary, which enables the models to manage a diverse language set effectively. Each input sequence begins with a distinct classification token [CLS], the final hidden state acting as the aggregate sequence representation for classification tasks. When processing sentence pairs, they are concatenated into a single sequence and distinguished by a unique token [SEP], along with learned embeddings that indicate the sentence association, sentence A or B. The tokenisation process encompasses three types of embeddings: Token embeddings, segment embeddings, and position embeddings [15, 39].

- **Token embeddings** play a pivotal role in capturing the semantic meaning of each token within the input text. These embeddings are derived by aggregating the corresponding token, segment, and position embeddings [15].
- **Segment embeddings** distinguish between different sentences or segments within the input. By incorporating segment embeddings, these models gain the ability to comprehend the interrelationships among various segments of the text [15].
- **Position embeddings** encode the positional information of each token within the input sequence. These embeddings furnish the model with valuable insights into the sequential order of the tokens. By integrating these three types of embeddings, models effectively represent the input text and capture the contextual nuances associated with each token [15].

A visual depiction of how these embeddings interact within the model's framework is provided in Figure 4.4. This illustration shows how these models integrate various linguistic features to understand

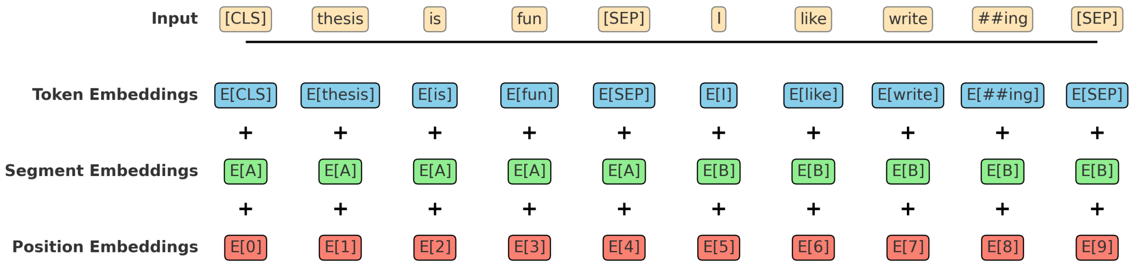


Figure 4.4: Illustrating the tokenization process for all the models presented in Table 4.1, this Figure is inspired by Devlin et al. [15]. The input "thesis is fun. I like writing." is the sum of all three embedding steps.

and interpret the input text fully. The tokenisation process, critical for preparing the text for model training and analysis, utilises the `AutoTokenizer` from the Huggingface `transformers` library [37]. This method is employed to tokenise the text according to the WordPiece embedding for all the models.

Train, Test, Validation Split

The dataset was partitioned into training, validation, and test sets to facilitate effective training of models and evaluation of their performance. This division is crucial for mitigating the risk of overfitting and ensuring the models' generalisation ability on unseen data.

The dataset was split into three distinct subsets using the stratified sampling method to ensure that each subset is representative of the overall dataset. Stratified sampling is essential when dealing with imbalanced datasets, as it helps maintain the proportion of each class across the different sets. The splitting process was implemented using the `train_test_split` function from the module `Scikit-learn` [11]. This function facilitates random shuffling of the data before the split, which is crucial for removing any bias that could be introduced by the order of the data.

This split was conducted using a random seed to ensure reproducibility of the results. This method ensures that the training, validation, and test sets contain approximately 60%, 20%, and 20% of the data, respectively. This division provides the training set with sufficient data to learn the intricacies of soccer events. This ensures enough data is available to validate and test on unseen data.

4.1.2 Model Selection: Huggingface Platform

In selecting the appropriate tools for the NLP tasks, the choice of Huggingface as a platform stands out for its distinct advantages. Huggingface is a pioneering hub in the AI community, specifically tailored for modern NLP. This platform offers several significant benefits, making it a preferred choice for this research project.

Huggingface hosts an extensive repository of pre-trained models, including the latest advancements in transformer technologies such as BERT BASE, BERT LARGE, DistilBERT, and all-MiniLM-L6-v2. These models represent the forefront of NLP research, having been trained on diverse and expansive datasets. The repository allows researchers to access a variety of models optimised for different NLP tasks, ensuring that the most suitable tools are readily available without the need for extensive computational resources to train from scratch [37]. This thesis used the Huggingface platform to develop the framework for training and evaluating these models. This included everything from data preprocessing using `Autotokenizer` to the design of the text classification models with `PyTorch`.

One of the primary advantages of using Huggingface is its vibrant community. The platform is supported by a network of researchers, developers, and data scientists who continuously contribute to improving models and documentation. This collaborative environment not only enhances the reliability and effectiveness of the models but also ensures that users can quickly find solutions and suggestions through forums and discussions. This aspect is particularly beneficial for academic research, where peer insights and shared experiences can significantly impact the success of a project [37].

Huggingface's `Transformers` library is designed with user-friendliness in mind. It provides a straightforward API that allows researchers to implement, fine-tune, and deploy state-of-the-art models easily. This ease of use accelerates the research process, enabling researchers to focus more on experimental design and less on technical implementation. The library also supports seamless integration with popular machine learning frameworks like `PyTorch`, making it highly adaptable to various research needs. Huggingface provides tools, extensive documentation, and learning resources crucial for researchers. These resources include detailed guides on implementing models, fine-tuning processes, and applying models to specific NLP tasks [37].

The platform is continuously updated with the latest NLP innovations, which ensures that the tools remain at the cutting edge of technology. This dynamism is crucial in the fast-evolving field of AI, where staying updated with the latest advancements can significantly influence research outcomes.

4.1.3 Optuna: Hyperparameter Search

Hyperparameter search is a crucial and often challenging task in machine learning projects. As the complexity of deep learning methods continues to grow, there is an increasing demand for efficient automatic hyperparameter tuning frameworks. Optuna presents an open-source optimisation framework that embodies a considerable advancement in the domain of hyperparameter search [36].

Optuna is designed with the define-by-run principle, making it the first of its kind in the realm of optimisation software. This principle allows users to dynamically construct the search space for hyperparameters, providing flexibility and adaptability during the optimisation process. By utilising Optuna's define-by-run API, researchers can easily define and modify hyperparameters, tailoring them to the specific requirements of their models. It also incorporates efficient sampling and pruning algorithms, which play a crucial role in the optimisation process. These algorithms enable the software to effectively explore the hyperparameter search space and discard unpromising trials, leading to faster convergence and improved optimisation results [36].

Optuna offers an easy-to-setup and versatile architecture that can be deployed for various tasks. It supports lightweight experiments conducted through interactive interfaces and heavy-weight distributed computations for large-scale experiments. This versatility allows researchers to adapt Optuna to their specific experimental setups and computational resources, making it a valuable tool for a wide range of machine learning projects [36].

The hyperparameters selected for optimisation and their respective ranges are detailed in Table 4.2. The ranges were inspired by the recommendations from Devlin et al. [15].

Table 4.2: Hyperparameter Ranges for Model Optimization

Hyperparameter	**Range**
Learning Rate	range$(1e-5, 1e-3)$ (log scale)
Per Device Train Batch Size	$\{8, 16, 32\}$
Weight Decay	range$(0, 0.3)$
Number of Training Epochs	$\{2, 3, 4, 5\}$
Warmup Steps	range$(0, 500)$

Several metrics were used to quantify the model's performance across trials, including accuracy, macro F1-score, precision, recall, and weighted F1-score. These metrics provide a comprehensive overview of the model's behaviour under various configurations, which is essential for evaluating the models. During the optimisation process with Optuna, the objective function was designed to maximise the macro F1-score. This metric was explicitly chosen to improve classification performance across minority classes. Given the dataset imbalance, the macro F1-score is the preferred metric for achieving balanced results across all classes. In contrast, the weighted F1-score might not adequately address class imbalances, particularly for minority classes such as Corner and Goals. The weighted F1-score would enhance the overall score but with emphasis on the majority class. This optimisation process was conducted over 32 trials for all the models listed in Table 4.1.

Best Hyperparameters from Optuna Search

The results of the Optuna hyperparameter search for DistilBERT are presented in Table 4.3. The two different datasets yielded dissimilar hyperparameter values. This indicates that even minor changes in the dataset can result in completely different hyperparameters. Despite the differences in the datasets for DistilBERT, the all-MiniLM-L6-v2 exhibits some similar hyperparameters, specifically in the per device train batch size and the number of training epochs. Table 4.4 shows that the weight decay varies significantly between the datasets, with minor variations also observed in the learning rate and warm-up steps.

Table 4.5 displays even more contrast between the hyperparameter values for different datasets, especially regarding weight decay where the dissimilarity is drastic. BERT BASE was the only model that preferred a set of 5 epochs.

Table 4.3: Hyperparameter values used for DistilBERT considering both 15-second- and 30-second datasets

DistilBERT		
Hyperparameter	**15-second dataset**	**30-second dataset**
learning rate	2.892e-05	4.378e-05
per device train batch size	8	16
weight decay	0.04096	0.03418
num train epochs	2	3
warmup steps	461	152

Table 4.4: Hyperparameter values used for all-MiniLM-L6-v2 considering both 15-second- and 30-second datasets

all-MiniLM-L6-v2		
Hyperparameter	**15-second dataset**	**30-second dataset**
learning rate	5.612e-05	3.996e-05
per device train batch size	8	8
weight decay	0.0468	0.2213
num train epochs	4	4
warmup steps	354	415

While BERT LARGE is the only model that uses the per device train batch size of 32. It is also the model with the least value for the warm-up steps. This is shown in the Table 4.6.

Table 4.5: Hyperparameter values used for BERT BASE considering both 15-second and 30-second datasets

BERT BASE		
Hyperparameter	**15-second dataset**	**30-second dataset**
learning rate	4.872e-05	3.798e-05
per device train batch size	8	16
weight decay	0.0030	0.1900
num train epochs	4	5
warmup steps	358	270

Table 4.6: Hyperparameter values used for BERT LARGE considering both 15-second and 30-second datasets

BERT LARGE		
Hyperparameter	**15-second dataset**	**30-second dataset**
learning rate	3.581e-05	3.658e-05
per device train batch size	16	32
weight decay	0.2336	0.0502
num train epochs	3	3
warmup steps	368	51

4.1.4 Hardware

This section presents the hardware utilised to address the research question. It outlines the specific components of the computer used in this thesis and describes the resources provided by the Orion High-Performance Computing Center at the Norwegian University of Life Sciences (NMBU).

Table 4.7: Hardware specifications of the MacBook M1 used for this thesis

Component	**Limitation**
Processor	Apple M1, 8-core CPU (4 performance cores, 4 efficiency cores)
Graphics	Integrated 8-core GPU
Memory	8 GB unified memory
Storage	256 GB SSD, 3.4 GB/s sequential read
Operating System	macOS Ventura 13.3.1

Table 4.7 outlines the specifications of the MacBook M1. Equipped with an 8-core CPU comprising four performance cores and four efficiency cores. It also features an integrated 8-core GPU that supports graphics-intensive applications. The machine includes 8 GB of unified memory, which ensures

smooth multitasking and quick access to files and programs. Its 256 GB Solid State Drive (SSD), with a sequential read speed of 3.4 GB/s, provides ample storage and fast data retrieval. The system operates on macOS Ventura 13.3.1, which offers a stable and secure platform for all computing needs.

Table 4.8 presents the specifications of the computing resources at Orion. The table lists key hardware components such as the operating system, GPU, memory interface, system memory, and system interface, which collectively enable handling large-scale computational tasks requiring high memory capacity.

Table 4.8: Hardware Specifications of the Orion High Performance Computing Center [66]

Component	**Specification**
Operating System	CentOS Linux release 7.9.2009 (Core)
GPU	NVIDIA Quadro RTX 8000
Memory interface	384-bit
Memory	48 GB GDDR6
System Interface	PCI Express 3.0 x 16

The inclusion of a high-performance NVIDIA Quadro RTX 8000 GPU with a 384-bit memory interface and 48 GB of GDDR6 memory was particularly pivotal. This setup provided the necessary computational power and memory bandwidth to efficiently process the large language models presented in Table 4.1.

4.2 Proposed Evaluation

While evaluating machine learning models, metrics are crucial in assessing performance and uncovering potential issues like class imbalance. This is vital for ensuring model fairness and alignment with project goals. Different metrics shed light on diverse aspects of model behaviour, guiding optimisation and selection strategies to prevent biased outcomes. The model's ability to handle imbalanced datasets is a critical factor in achieving reliable and unbiased predictions by carefully choosing the appropriate metrics.

4.2.1 Accuracy

Metrics like prediction Error (ERR) and Accuracy (ACC) offer insights into a model's effectiveness in classifying data. These metrics represent the proportion of misclassified and correctly classified examples, which are instrumental for evaluating and refining models [71].

TP: True positive **FP:** False positive

TN: True negative **FN:** False negative

Prediction error and accuracy serve as fundamental metrics to assess the classification performance of a model. ERR quantifies the proportion of false predictions across the dataset, while ACC measures the proportion of correctly identified instances. ERR is calculated by dividing the sum of all incorrect predictions by the total number of predictions, as presented in Equation 4.1. ACC is determined by dividing the sum of accurate predictions by the total prediction count, as seen in Equation 4.2. Prediction accuracy can be directly inferred from the error rate, providing a comprehensive understanding of the model's predictive capability [71]. This metric can also be balanced to take class imbalance into account. Balanced accuracy calculates the average recall for each class. When comparing these metrics, it's easier to understand the class distribution of the dataset [63].

$$ERR = \frac{FP + FN}{FP + FN + TP + TN} \tag{4.1}$$

$$ACC = \frac{TP + TN}{FP + FN + TP + TN} = 1 - ERR \tag{4.2}$$

Accuracy and prediction error are inherently complementary metrics. A high accuracy directly translates to a low prediction error, and vice versa. Recognising the limitations of using accuracy as the sole metric is important, especially in datasets with class imbalances. When one class dominates the dataset, solely relying on accuracy can be misleading. Models may show high accuracy by predominantly predicting the majority class, neglecting accurate identification of less represented classes [71].

4.2.2 Precision

Precision is a critical evaluation metric that indicates a model's accuracy in predicting true positives while minimising false positives, as shown in Equation 4.3. It is especially relevant in scenarios demanding high correctness, such as medical diagnoses, where the cost of false negatives could be significant. A model with high precision demonstrates its capability to correctly identify instances of a particular class without incorrectly labelling instances from other classes. This metric is crucial for models where the accurate detection of specific outcomes is prioritised over the comprehensive identification of all potential positives [71].

$$PRE = \frac{TP}{TP + FP} \tag{4.3}$$

A high precision value indicates a model's effectiveness in accurately identifying true positive instances. This translates to a lower likelihood of misclassifying negative sentiment samples as positive.

4.2.3 Recall

Recall, or True Positive Rate (TPR), is a critical metric in evaluating performance, particularly for imbalanced datasets. It measures the proportion of actual positives a model correctly identifies, emphasising the model's sensitivity to detecting relevant instances [71], as provided in Equation 4.4.

$$REC = TPR = \frac{TP}{P} = \frac{TP}{FN + TP} \tag{4.4}$$

In critical domains like medical diagnosis, optimising for recall minimises the risk of overlooking crucial positive cases. A high recall ensures the model identifies most malignant tumours, even if it leads to a higher false positive rate. This highlights the inherent trade-off between recall and precision. Increasing one can often decrease the other, underlining the necessity of choosing the right evaluation metric based on specific project needs [71].

4.2.4 F1-score

The F1-score is a popular evaluation metric used in machine learning classification tasks. It provides a way to measure a model's performance by considering precision and recall.

This formula is presented in Equation 4.5 and can be interpreted as a weighted average between precision and recall, where both metrics contribute equally to the final score. The harmonic mean is an average that penalises cases where either precision or recall is exceptionally low. A high precision might be crucial in some scenarios, while a high recall might be preferred in others. F1-score helps to balance the trade-off between these two objectives [63].

$$\text{F1-score} = 2 \cdot \left(\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \right) \tag{4.5}$$

When considering the binary classification of the F1-score, only the positive class is used when evaluating. The true negative class is insignificant in the results. In multiclass classification, it is important to consider all the classes individually. Since each class is reviewed, the precision and recall need to be implemented for multiclass approach [63].

Macro F1-score

In multiclass classification problems, the F1-score can be calculated using different averaging methods. One of the approaches is the macro F1-score. This method focuses on treating all classes equally, regardless of size. This is achieved by first calculating macro-precision and macro-recall.

Macro-precision and macro-recall are computed as the arithmetic means of the metrics for single classes. This metric is the average precision across all classes. It is calculated by summing the precision of each class and dividing it by the total number of classes [63]:

$$\text{macroAveragePrecision (MAP)} = \frac{\sum_{i=1}^{n} \text{Precision}_i}{n} \tag{4.6}$$

Similarly, macro-recall is the average recall across all classes. It is calculated by summing the recall of each class and dividing it by the total number of classes [63]:

$$\text{macroAverageRecall (MAR)} = \frac{\sum_{i=1}^{n} \text{Recall}_i}{n} \tag{4.7}$$

Macro F1-Score is a weighted harmonic mean of macro-precision and macro-recall, as formulated in Equation 4.8. High macro F1-scores indicate that the algorithm performs well on all or most classes. Low macro F1-scores suggest that the model might struggle with some or all the classes [63].

$$\text{macro F1-Score} = 2 \cdot \left(\frac{\text{MAP} \cdot \text{MAR}}{\text{MAP}^{-1} + \text{MAR}^{-1}} \right) = \frac{\sum_{i=1}^{n} \text{F1-score}_i}{n} \tag{4.8}$$

Weighted F1-score

In multiclass classification problems, the weighted F1-score addresses class imbalance by taking the average F1-scores for each class. These averages are weighted by the number of true samples in each class, giving more influence to classes with more data. This approach primarily focuses on the model's performance within the majority class [55, 75].

$$\text{weighted F1-Score} = \sum_{i=1}^{n} w_i \cdot \text{F1-score}_i \tag{4.9}$$

Equation 4.9 shows the mathematic representation of weighted F1-score. w_i is the weight for the i-th class, which is the number of true instances of the class divided by the total number of instances. The weighted F1-score is calculated by applying the weights to the F1-score of each class. This gives an average score that best reflects the model's performance across all classes, taking their distribution into account [55].

High weighted F1-score suggests that the model performs well across the majority classes, which might be critical in some imbalanced datasets. However, a low value for weighted F1-score might indicate poor performance in the minority classes, which might be equally important depending on the application [55, 65].

4.3 Chapter Summary

This chapter outlines the methods developed to address the research question, covering the entire process from data acquisition to model evaluation. It introduces several NLP models like DistilBERT, BERT BASE, BERT LARGE, and all-MiniLM-L6-v2, detailing their specifications and uses in the thesis.

The methodology begins with data preprocessing, where audio from soccer matches is captured and processed. This step includes removing missing values, eliminating stopwords, tokenising the text, and splitting the data into training, testing, and validation sets. These steps are crucial for maintaining data quality and ensuring unbiased inputs for modelling.

The thesis recommends the Huggingface platform as a key resource for NLP tasks due to its extensive repository of pre-trained models. Huggingface enhances usability with a user-friendly API and seamless integration with major machine learning frameworks like `PyTorch`, making it invaluable for staying at the forefront of NLP technology.

The model training process is described in detail, including hyperparameter tuning using Optuna, a framework that dynamically optimises hyperparameters. This section highlights the importance of selecting appropriate hyperparameters to enhance model accuracy and efficiency. Lastly, the chapter details the hardware specifications used for computations, emphasising the high-performance computing resources at Orion that support large-scale model training. The following chapter uses this methodology to conduct research and obtain results.

Chapter 5

Results

This chapter presents the results and offers a comprehensive comparison between various LLMs using 15-SSDD and 30-SSDD. These structured datasets detailed in Chapter 3 are analysed using the methodologies outlined in Chapter 4. The outcomes of these analyses are evaluated based on key metrics, including precision, recall, and F1-score. Standard metrics provide a foundation, but a comprehensive assessment requires examining additional measurements. These include loss, runtime efficiency, and computational throughput. The results are discussed further in the Chapter 6. This analysis showcases the current performance of various large language models and establishes a crucial benchmark for future research efforts.

5.1 Model Evaluation

5.1.1 DistilBERT

This section explores the performance of DistilBERT, one of the prominent models assessed in this thesis. This analysis is pivotal as it reveals how well DistilBERT maintains its efficacy when applied to the 15-SSDD and 30-SSDD described in Chapter 3. The evaluation focuses on the model's performance metrics emphasised in the introduction across two different dataset durations: 15-second and 30-second. The metrics chosen are critical for understanding the model's ability to correctly identify and classify instances of imbalanced datasets.

From Table 5.1, it is evident that DistilBERT performs well on most of the considered classes, with notable variations between the two dataset durations. For the Corner class, the model achieves higher precision with the 15-second samples (82.72%) compared to the 30-second samples (79.58%), indicating a reliable prediction when a Corner event is identified in shorter clips. However, recall improves significantly from 65.66% in 15-second samples to 71.00% in 30-second samples, suggesting that longer samples capture more true Corner events.

For the class Foul, the model exhibits a robust performance. It shows a very high precision and recall across both datasets. Precision improves from 84.01% in the 15-second samples to 87.12% in the 30-second samples, and although recall slightly decreases, it remains impressively high (94.14% to

Table 5.1: **Classification Metrics** for **DistilBERT** with 15-SSDD and 30-SSDD.

	15-SSDD			**30-SSDD**		
Class	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
Corner	82.72%	65.66%	73.21%	79.58%	71.00%	75.04%
Foul	84.01%	94.14%	88.79%	87.12%	91.99%	89.49%
Goal	75.83%	57.96%	65.70%	71.72%	66.24%	68.87%
Macro Avg	**80.86%**	**72.59%**	**75.90%**	**79.47%**	**76.41%**	**77.80%**
Weighted Avg	**82.90%**	**83.15%**	**82.47%**	**83.68%**	**84.00%**	**83.72%**

91.99%). The F1-scores are correspondingly high, reflecting excellent model accuracy in identifying fouls. This is more clearly presented in Figure 5.1, where the classes are depicted concerning the datasets.

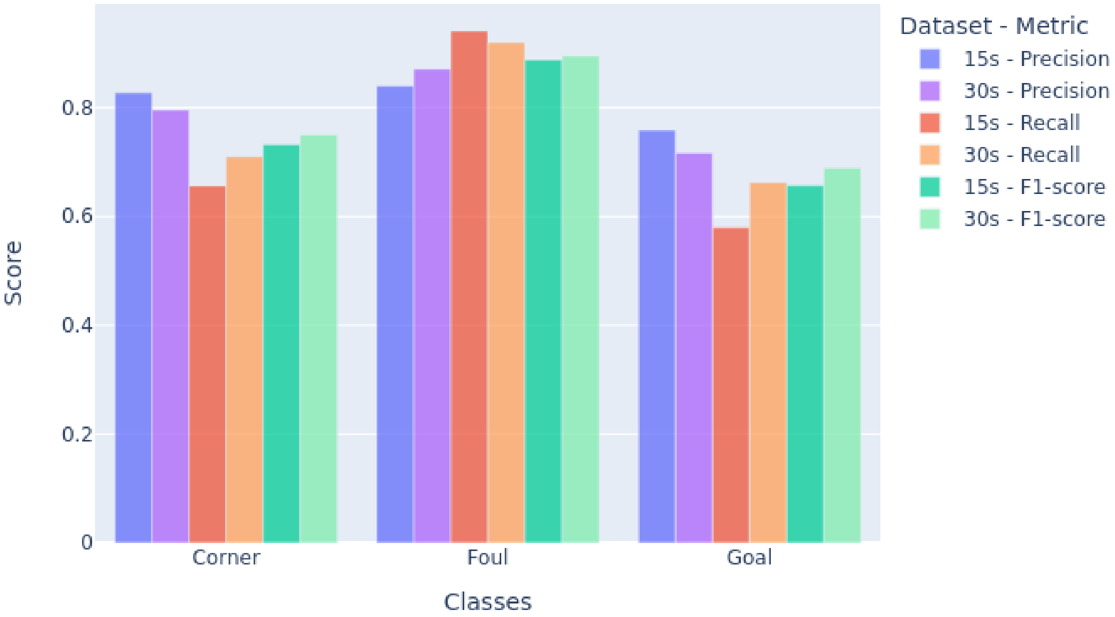


Figure 5.1: Representing the classes with consideration to the respective datasets for DistilBERT.

The class Goal presents a challenge, especially with the 15-second samples, where both precision (75.83%) and recall (57.96%) are lower compared to other categories. While there is an improvement in recall (to 66.24%) and F1-score (to 68.87%) with the 30-second samples, these figures suggest that detecting goals accurately and consistently may require further model tuning or data enhancement. The reason behind the alternating results across different classes is class imbalance. In Chapter 3, the constructed datasets were heavily imbalanced, favouring the class Foul and least favouring the class Goal.

The macro-averaged metrics, which treat all classes equally, show an overall better performance with the 30-second samples (F1-score of 77.80% compared to 75.90% in the 15-second samples). This indicates a general improvement in model performance when processing longer textual standard deviated samples. Reflecting the distribution of class instances, the weighted-averages also favour the 30-second samples across all metrics, with an F1-score of 83.72% compared to 82.47% for the 15-second samples. This improvement underscores the model's enhanced capability to handle more extensive and complex data.

Table 5.2: Additional Performance Metrics for DistilBERT with 15-SSDD and 30-SSDD.

Metric	**15-SSDD**	**30-SSDD**
Loss	0.4596	0.4755
Accuracy	83.15%	84.00%
Balanced Accuracy	72.59%	76.41%
Runtime (s)	3.3904	2.4627
Samples per Second	991.024	1370.858
Steps per Second	123.878	85.679

Table 5.2 provides additional information between these two datasets. These additional metrics offer a comprehensive view of the DistilBERT model's performance, suggesting that increased sample duration leads to statistically significant improvements in classification accuracy and computational efficiency. The loss metric indicates a trade-off, possibly due to the complexity introduced with more extended data. Increased accuracy, balanced accuracy, and runtime efficiency with longer samples justify their use in practical applications. Proficiency with extended data suggests suitability for scenarios with longer context, often encountered in real-world applications. This opens doors for further optimisation, focusing on loss reduction while maintaining or improving other metrics.

Further examination of the different datasets reveals the distribution between classes, as shown in Figure 5.2 for the 15-second dataset and in Figure 5.3 for the 30-second dataset. Increasing the standard deviation of textual samples enables the model to differentiate between the respective classes. The model's enhanced ability to accurately distinguish between Corner and Goal events in longer samples suggests that these event types benefit from additional context or features that become more prevalent or distinguishable with extended durations.

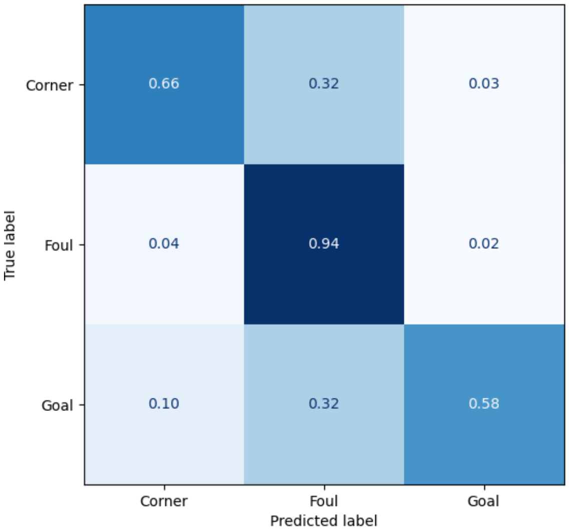


Figure 5.2: Confusion matrix for DistilBERT considering 15-SSDD.

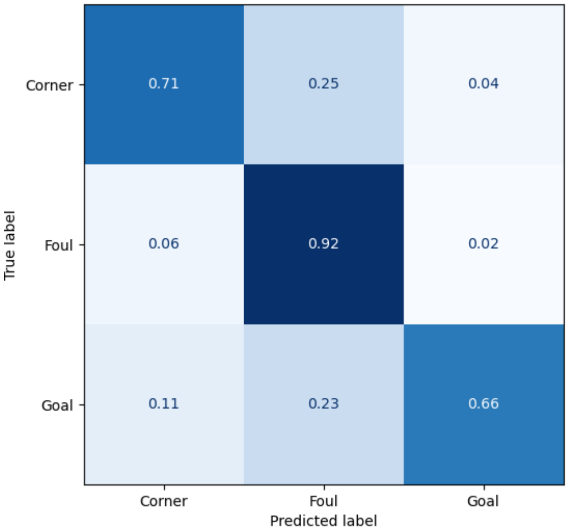


Figure 5.3: Confusion matrix for DistilBERT considering 30-SSDD.

5.1.2 all-MiniLM-L6-v2

This section explores the performance of the all-MiniLM-L6-v2 model, a distilled version of the larger BERT BASE model specifically optimised for greater speed and efficiency while retaining considerable high-quality results. In the context of this research, all-MiniLM-L6-v2 was evaluated using the same structured datasets and methodologies as its larger counterparts to assess its performance in a directly comparable framework. This analysis provides insights into how well the model handles various linguistic tasks under the constraints of limited computational resources, which is crucial for deploying systems in environments where processing power and memory are limited.

Table 5.3 presents metrics for the 15-second and 30-second datasets. The precision measure at 74.94% indicates that when the model predicts an event as a Corner, it is correct about three-quarters of the time. However, the recall of 70.01% suggests that it fails to identify around 30% of actual Corner events within the shorter textual samples. The resulting F1-score of 72.39% indicates a reasonable level of performance but with room for improvement, particularly in capturing more true positive Corner

Table 5.3: **Classification Metrics** for **all-MiniLM-L6-v2** with 15-SSDD and 30-SSDD.

	15-SSDD			**30-SSDD**		
Class	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
Corner	74.94%	70.01%	72.39%	75.17%	74.00%	74.58%
Foul	86.21%	89.90%	88.02%	87.79%	90.15%	88.95%
Goal	71.17%	63.69%	67.23%	72.22%	62.10%	66.78%
Macro Avg	**77.44%**	**74.54%**	**75.88%**	**78.39%**	**75.42%**	**76.77%**
Weighted Avg	**81.80%**	**82.14%**	**81.90%**	**82.98%**	**83.23%**	**83.06%**

events. With the longer samples, both precision and recall improve slightly, with precision increasing to 75.17% and recall to 74.00%. This indicates that the model is not only correctly identifying Corner events more often but also missing fewer actual Corner events. The enhanced F1-score of 74.58% indicates this improved balance, which can also be seen in Figures 5.5 and 5.6. This growth can be attributed to the additional context in the longer samples, which may include more distinctive cues or patterns that the model can learn to associate with Corner events. The difference between the metrics is presented more clearly for the class Corner in Figure 5.4.

The model excels for the class Foul with high precision (86.21%) and recall (89.90%), suggesting that it is adept at correctly predicting Foul events and capturing most of the actual Foul occurrences. The high F1-score of 88.02% reflects strong performance, showing that Foul events have distinctive features that the model can recognise even in shorter samples. There is a further improvement when the sample duration is increased, with precision edging up to 87.79% and recall to 90.15%. The corresponding F1-score of 88.95% demonstrates that the model's ability to detect Foul events is maintained and slightly enhanced with longer samples. It suggests that Foul events have features that become even more distinguishable to the model in a broader context. This can be seen in Figure 5.4.

The class Goal presents the greatest challenge for the model in the shorter dataset, as shown in both Table 5.3 and Figure 5.4. With a precision of 71.17% and a recall of 63.69%, it indicates that while the model can reasonably identify Goal events, it does so with less certainty than Foul and Corner events. The relatively lower F1-score of 67.23% signifies that the model tends to miss true Goal events and misclassify other events as Goals. This can be seen in Figures 5.5 and 5.6. Increasing the duration to 30 seconds does not uniformly improve the model's performance for Goal events. Precision sees a slight improvement to 72.22%, yet recall slightly drops to 62.10%. The F1-score decreases marginally to 66.78%, suggesting that while the model is slightly more precise in its predictions of Goal events with the longer samples, it does not capture a greater proportion of actual Goal events. This could reflect an inherent complexity within the Goal event, possibly due to less distinct or more varied patterns than other event types or overlapping characteristics with other classes that confuse the model, particularly in longer sample contexts.

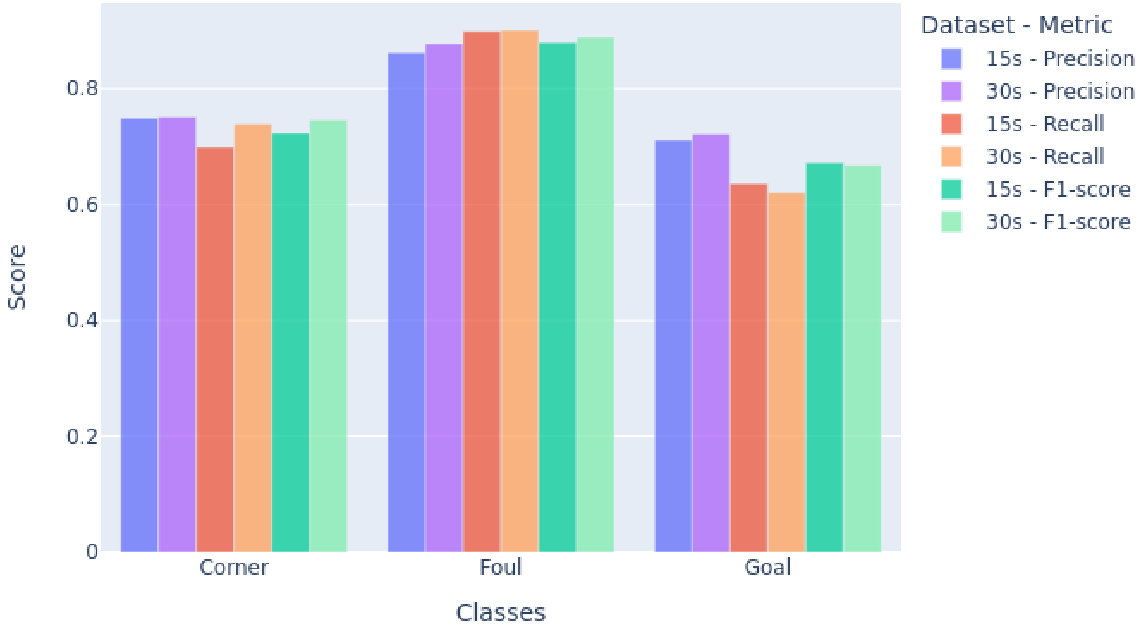


Figure 5.4: Representing the classes with consideration to the respective datasets for all-MiniLM-L6-v2.

The macro-averaged precision for the model across the 15-second samples is reported in Table 5.3 at 77.44%, with a recall of 74.54% and an F1-score of 75.88%. This indicates a balanced capacity of the model to accurately identify and label events across all classes. With the 30-second samples, these macro-averaged metrics display a slight growth, with precision increasing to 78.39%, recall exhibiting a negligible enhancement to 75.42%, and the F1-score rising to 76.77%. The modest improvement in precision and F1-score, along with stable recall, suggests that while longer samples enhance the model's classification accuracy, they do not significantly improve its ability to identify all true instances of each class.

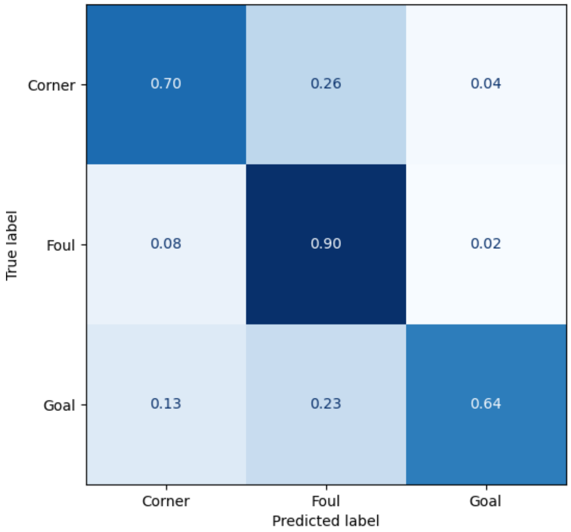


Figure 5.5: Confusion matrix for all-MiniLM-L6-v2 considering15-SSDD.

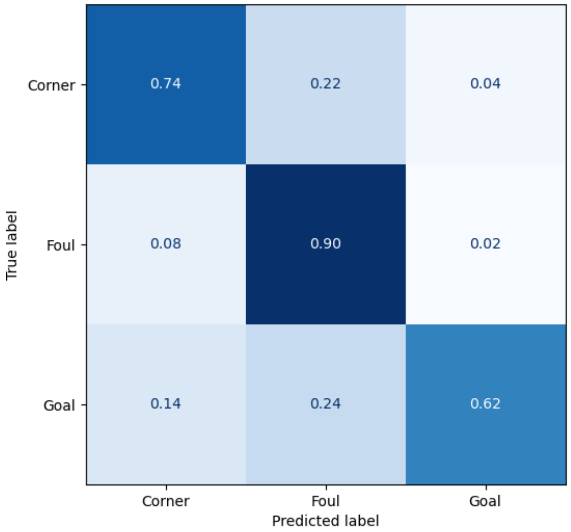


Figure 5.6: Confusion matrix for all-MiniLM-L6-v2 considering30-SSDD.

When considering the weighted-average metrics, a more pronounced improvement is observed. For the 15-SSDD, the weighted-average precision is at 81.80%, recall at 82.14%, and F1-score at 81.90%. These metrics increase with the 30-SSDD to a precision of 82.98%, recall of 83.23%, and an F1-score of 83.06%. This suggests the model performs better with the classes when given more context from longer samples. The assumption is confirmed by comparing Figures 5.5 and 5.6, where the class Foul performs 4% better for longer samples.

Table 5.4: Additional Performance Metrics for all-MiniLM-L6-v2 with 15-SSDD and 30-SSDD.

Metric	**15-SSDD**	**30-SSDD**
Loss	0.5556	0.5071
Accuracy	82.14%	83.23%
Balanced Accuracy	74.54%	75.42%
Runtime (s)	3.387	3.6593
Samples per Second	992.018	922.576
Steps per Second	124.002	115.322

Table 5.4 provides additional information between the two datasets. The improvement in accuracy for the 30-second dataset indicates increased proficiency in handling accuracy-critical tasks but comes at the cost of lower computational throughput and longer runtime. Specifically, the model's loss decreased from 0.5556 to 0.5071, and accuracy improved slightly from 82.14% to 83.23% with longer

samples. Balanced accuracy also improved slightly, suggesting more consistent performance across different classes. However, these benefits are offset by a slight increase in runtime from 3.387 seconds to 3.6593 seconds and reduced throughput, highlighting the computational trade-offs in processing larger samples.

5.1.3 BERT BASE

BERT BASE is the second-largest model presented in this research framework, as detailed in Table 4.1. The model is evaluated on the structured datasets described in Chapter 3, applying the methodologies established in Chapter 4. This analysis is essential to determine how effectively the model manages the complexities of the two datasets and to assess its performance across various metrics, including precision, recall, and F1-score. This controlled experiment highlights BERT BASE's strengths, limitations, and potential for improvement.

Table 5.5: **Classification Metrics** for **BERT BASE** with 15-SSDD and 30-SSDD.

	15-SSDD			**30-SSDD**		
Class	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
Corner	74.82%	69.90%	72.28%	78.00%	70.56%	74.10%
Foul	86.48%	89.90%	88.16%	87.15%	92.23%	89.62%
Goal	73.61%	67.52%	70.43%	75.55%	65.92%	70.41%
Macro Avg	**78.30%**	**75.77%**	**76.96%**	**80.24%**	**76.24%**	**78.04%**
Weighted Avg	**82.17%**	**82.47%**	**82.26%**	**83.63%**	**84.00%**	**83.69%**

For the 15-second dataset, the BERT BASE model achieved a precision of 74.82%. This means the model correctly predicts an event as a Corner roughly three-quarters of the time, demonstrating similar performance to the all-MiniLM-L6-v2 model shown in Table 5.3. The recall rate of 69.90% indicates that the model missed about 30% of actual Corner events. The resulting F1-score of 72.28% shows in Figure 5.7 and Table 5.5 that there is a balance between precision and recall, but with considerable room for improvement. When evaluated on the 30-second dataset, the model's precision increased to 78.00% and recall to 70.56%. The model's improved precision with longer samples suggests increased reliability in Corner predictions. This aligns with the rise in recall, indicating better detection of Corner events when provided with more context. The combined effect is a more balanced and higher F1-score of 74.10%.

The model demonstrated a high precision of 86.48% and a recall of 89.90%, with an F1-score of 88.16% for the 15-second samples, indicating the effective classification of Foul events. With the 30-second samples, the model's precision saw a marginal increment to 87.15% and recall to 92.23%, raising the F1-score to 89.62%. These metrics suggest a robust performance that marginally improves with longer samples. This can be clearly seen in Figure 5.7 and Table 5.5.

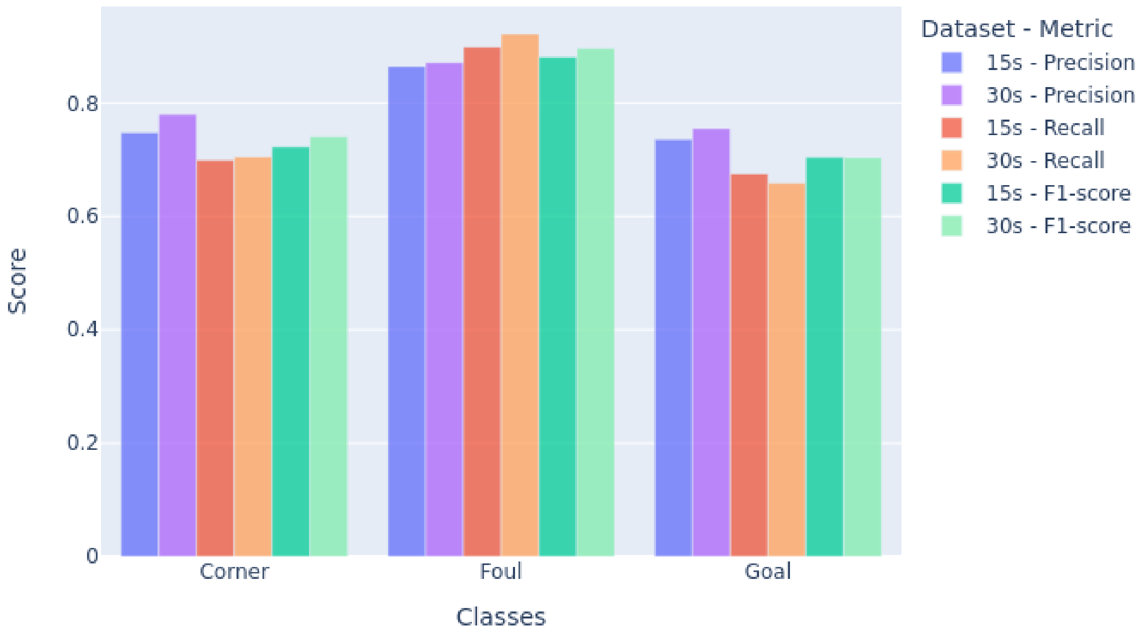


Figure 5.7: Representing the classes with consideration to the respective datasets for BERT BASE.

The model's performance on Goal events showed slight variations in precision and recall between the 15-SSDD and 30-SSDD, suggesting no significant difference overall in the F1-score. With the shorter samples, the model achieved a precision of 73.61% and a recall of 67.52%, leading to the highest F1-score for Goal events at 70.43%. The precision increases significantly to 75.55%, but recall decreases slightly to 65.92%, with the F1-score essentially remaining stable at 70.41%. This suggests that while the model is more precise in predicting Goal events with longer samples, its ability to recognise all true Goal events is not correspondingly improved. This can be seen the Figure 5.7 and Table 5.5.

Loss metric indicates how well the model's prediction matches the actual labels. Table 5.6 shows a lower loss (0.7545 to 0.4369) and higher accuracy (82.47% to 84.09%) with 30-second samples indicating the model benefits from longer context. Even balanced accuracy shows a modest improvement (75.77% to 76.18%), suggesting better handling of imbalanced classes.

Table 5.6: Performance Metrics for BERT BASE with 15-SSDD and 30-SSDD.

Metric	**15-SSDD**	**30-SSDD**
Loss	0.7545	0.4369
Accuracy	82.47%	84.09%
Balanced Accuracy	75.77%	76.18%
Runtime (s)	5.3708	4.7378
Samples per Second	625.606	712.569
Steps per Second	78.201	22.373

The runtime is shorter for the 30-second dataset (4.74 seconds) compared to the 15-second (5.37 seconds) despite processing more data. This suggests that despite processing larger samples, the model is optimised to perform its computations more efficiently on longer sequences. The model processes more samples per second (712.57 vs. 625.61) and takes fewer steps per second (22.37 vs. 78.20) when dealing with 30-SSDD compared to 15-SSDD. The higher samples-per-second rate with the 30-second dataset indicates improved computational efficiency. Meanwhile, the reduced steps per second might reflect a more streamlined or effective training process, where each step possibly encompasses more data and hence conveys more information.

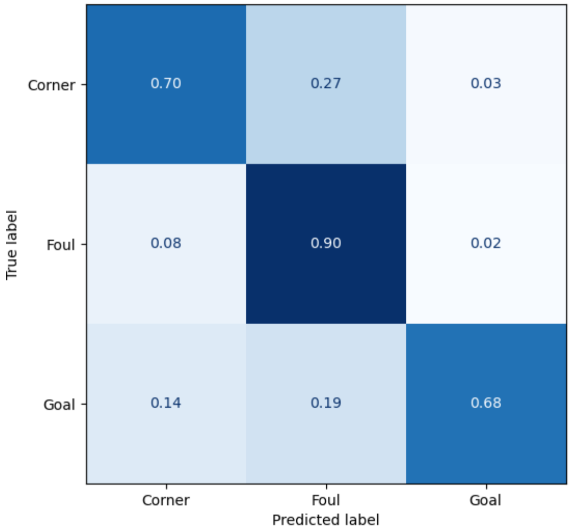


Figure 5.8: Confusion matrix for BERT BASE considering 15-SSDD.

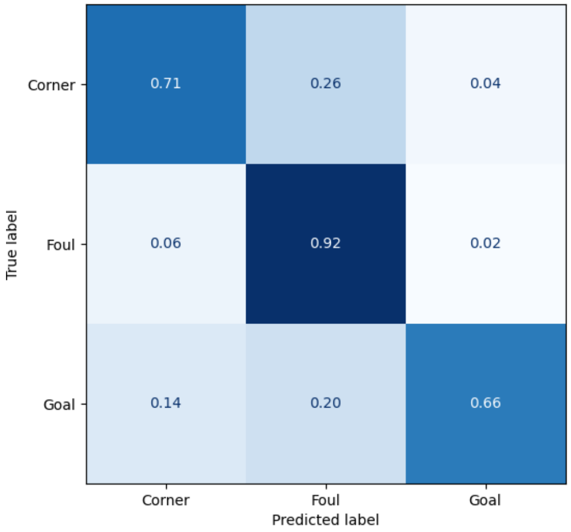


Figure 5.9: Confusion matrix for BERT BASE considering 30-SSDD.

The model's performance varies across event types. Corner events see a slight improvement in accuracy (70% to 71%) and reduced misclassification as Fouls (27% to 26%) with the 30-second duration. This suggests the model performs better when analysing events within a longer window. Foul events excel with high accuracy (90% to 92%) and lower Corner misclassification (6%) with longer samples. Goal events are correctly identified with a probability of 68%, with misclassifications as both Fouls (19%) and Corners (14%). While the change in performance is minimal (66% with 30-second data), there's a slight increase in misclassification as Fouls (20%). This suggests that additional duration may not significantly improve Goal event identification.

5.1.4 BERT LARGE

This section evaluates BERT LARGE, the largest and most robust model within the BERT family, detailed in Table 4.1. It features an extensive architecture with more layers and parameters and aims to capture deeper linguistic nuances. The methodologies outlined in Chapter 4 are used to assess its performance on the structured datasets described in Chapter 3. Key evaluation metrics, including precision, recall, and F1-score, are employed to provide a comprehensive analysis. This examination highlights the strengths and potential limitations of BERT LARGE.

Table 5.7: **Classification Metrics** for **BERT LARGE** with 15-SSDD and 30-SSDD.

	15-SSDD			**30-SSDD**		
Class	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
Corner	80.61%	70.90%	75.44%	78.55%	72.44%	75.38%
Foul	86.62%	92.79%	89.60%	87.95%	91.12%	89.50%
Goal	78.81%	67.52%	72.73%	71.57%	69.75%	70.65%
Macro Avg	**82.01%**	**77.07%**	**79.26%**	**79.36%**	**77.77%**	**78.51%**
Weighted Avg	**84.29%**	**84.58%**	**84.24%**	**83.92%**	**84.15%**	**83.98%**

From Table 5.7, the model exhibits high precision (80.61%) but lower recall (70.90%) for Corner events, resulting in an F1-score of 75.44%. This indicates good accuracy when predicting Corners, but it misses some true occurrences. While precision dips slightly to 78.55% with longer samples, possibly due to increased complexity, recall improves to 72.44%. The model seems to identify more true Corners with additional information. The F1-score remains consistent at 75.38%, suggesting the model's overall effectiveness in classifying Corners is maintained across both sample lengths. This is also demonstrated in Figure 5.10.

Foul events showcase the model's strengths, as seen in Table 5.7. It achieves a high precision (86.62%) and an exceptional recall (92.79%), leading to a strong F1-score (89.60%). This indicates excellent ability to capture Foul event characteristics in shorter samples. Even with longer durations, the performance remains robust: precision increases slightly (87.95%), while recall dips marginally (91.12%), resulting in a near-identical F1-score (89.50%). These minimal changes suggest the model effectively recognises Foul events regardless of sample length. This is visualised more clearly in Figure 5.10.

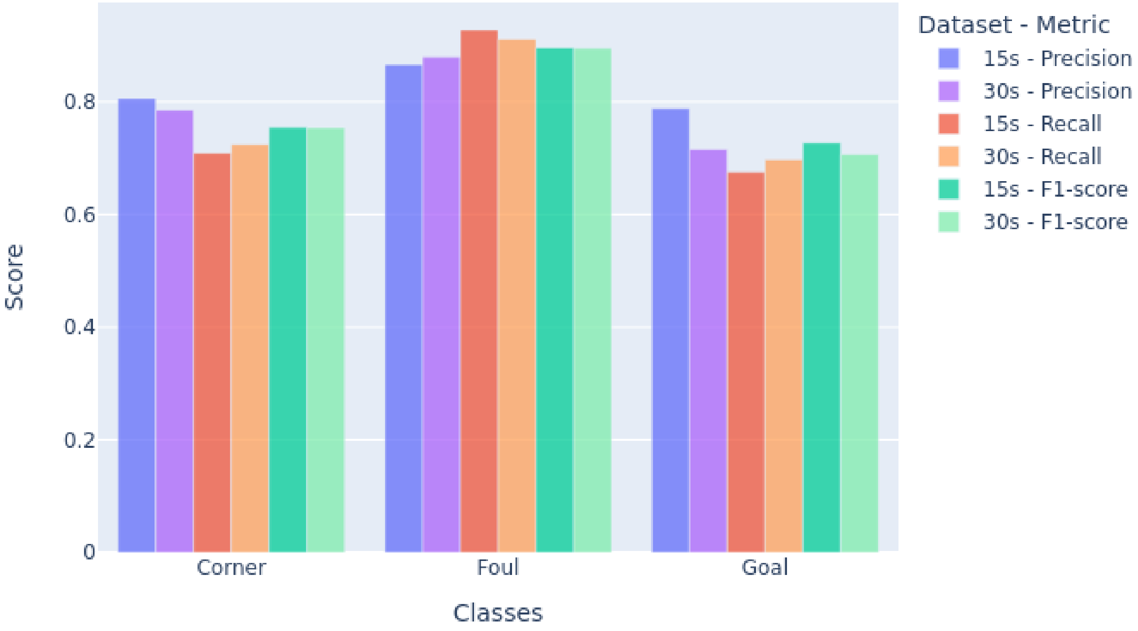


Figure 5.10: Representing the classes with consideration to the respective datasets for BERT LARGE.

Goal events present the greatest challenge among all the classes, as shown in Table 5.7. Despite high precision (78.81%), indicating accurate Goal predictions, the model misses a significant portion of actual events (recall of 67.52%). This results in a moderate F1-score (72.73%). With longer samples, precision dips (71.57%), suggesting some loss of accuracy. However, recall improves (69.75%), indicating better identification of true Goals. While the F1-score dips slightly to 70.65%, this may indicate a trade-off. The model might capture more actual Goal events but also potentially make more false identifications.

Table 5.7 summarises the average performance across event types using macro-averages and weighted-average. Macro performance metrics across classes reveal a slight decrease in precision (from 82.01% to 79.36%) and a small increase in recall (from 77.07% to 77.77%) when comparing the 15-SSDD to the 30-SSDD. The F1-score also exhibits a minor decrease (from 79.26% to 78.51%). The weighted-average performance metrics showcase the model's strong overall capability. It achieves a high precision (84.29%), recall (84.58%), and F1-score (84.24%), indicating good alignment with class distribution and consistent performance across categories. While there are slight decreases in all metrics with the

30-SSDD, the overall performance remains high. This suggests that using longer samples, although potentially beneficial for individual classes due to class imbalance, may not significantly improve weighted performance.

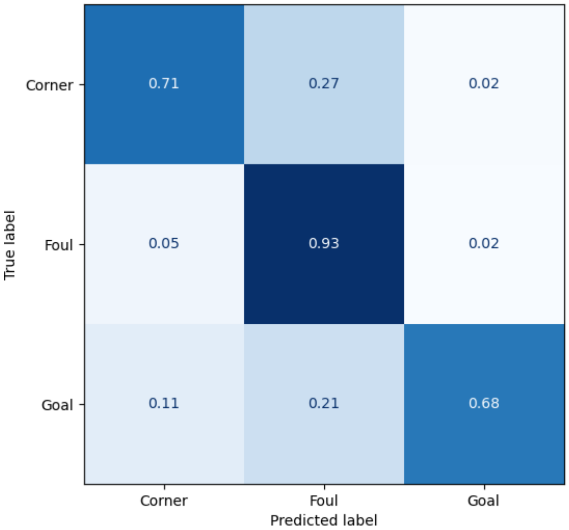


Figure 5.11: Confusion matrix for BERT LARGE considering 15-SSDD.

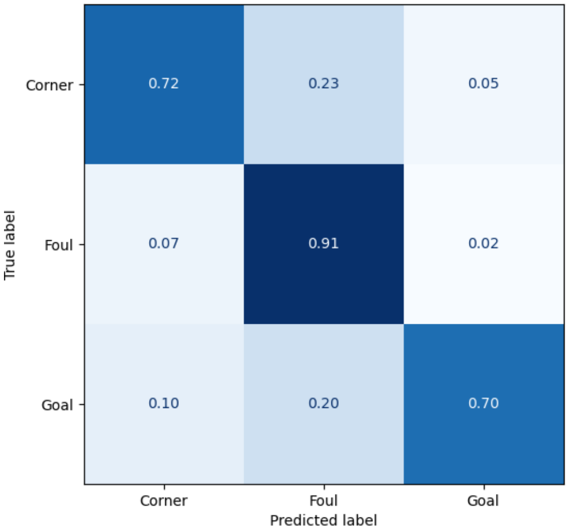


Figure 5.12: Confusion matrix for BERT LARGE considering 30-SSDD.

Based on the Figures 5.11 and 5.12 the model excels at identifying Corners (71%), but some confusion exists with Fouls (27% misclassification). There's a slight improvement in accuracy (72%) with longer samples, but further improvement is needed to differentiate these events. The model demonstrates exceptional performance for Fouls (93%) with shorter samples, effectively capturing their defining features. This strong performance is maintained (91%) with longer samples, with minimal change in misclassification. Goal identification presents a challenge. While the model achieves reasonable accuracy (68%), it struggles to distinguish Goals from both Corners (11% misclassification) and Fouls (21%). There's a slight improvement in accuracy (70%) and reduced misclassification (20% as Fouls) with longer samples, suggesting some benefit from additional context.

The confusion matrices reinforce the earlier observations. BERT LARGE shows a slight improvement with longer samples, especially for Goal events. However, persistent challenges remain in differentiating between event types, particularly Goal versus Foul. This suggests that future model refinements should focus on improving feature extraction or classification algorithms to distinguish these classes better.

Table 5.8: Performance Metrics for BERT LARGE with 15-SSDD and 30-SSDD.

Metric	**15-SSDD**	**30-SSDD**
Loss	0.4558	0.4660
Accuracy	84.58%	84.15%
Balanced Accuracy	77.07%	77.77%
Runtime (s)	5.8159	10.7575
Samples per Second	577.727	313.827
Steps per Second	36.108	9.854

Analysing the model's performance with longer samples reveals a few key points. There's a minimal increase in training loss (0.4558 to 0.4660), suggesting the model adapts well to both durations. Overall accuracy dips slightly (84.58% to 84.15%), but balanced accuracy improves (77.07% to 77.77%), indicating better handling of all event types with more context. However, this benefit comes at a cost. Processing time nearly doubles, increasing from 5.82 seconds to 10.76 seconds, and throughput, measured by samples and steps processed per second, significantly decreases. This translates to the model taking more time per sample and making fewer predictions with longer samples. In essence, while longer samples offer a slight accuracy improvement, it comes with a trade-off in computational efficiency.

5.2 Analysis and Comparison of Models

This section summarises the evaluation of the four LLMs: DistilBERT, BERT BASE, BERT LARGE, and all-MiniLM-L6-v2. The analysis examines their performance on 15-SSDD and 30-SSDD, as detailed in Chapter 3. Using precision, recall, and F1-score metrics are used to provide a comprehensive benchmark. The analysis extends beyond individual class performance by incorporating both macro-average and weighted-averages. Evaluating these metrics reveals important trends and performance variations across the models, offering valuable insights into how they handle various event types across sample lengths, considering class imbalances.

Table 5.9 demonstrates the resilience of all-MiniLM-L6-v2 regarding precision and recall across classes, particularly for Corner events. With 30-SSDD, it achieved a precision improvement (74.94% to 75.17%) and a substantial recall gain (70.01% to 74.00%), reflected in the increased F1-score (74.58%). This surpasses DistilBERT performance for Corner events, which, despite starting with higher initial precision (82.72% at 15 seconds), showed a decrease with longer samples.

all-MiniLM-L6-v2 excelled at Foul event detection, maintaining high precision (86% and 88%) and exceptional recall (89% and 90%) across both sample durations. F1-scores consistently above 88% demonstrate its robustness in capturing Foul events even with extended samples. all-MiniLM-L6-v2 outperforms BERT BASE and BERT LARGE, whose performance improvement with longer samples is less pronounced. Despite the overall challenge of Goal event detection, all-MiniLM-L6-v2 showed improvement with longer samples. Its precision increased from 71.17% to 72.22% for 30-SSDD, suggesting a better ability to identify Goals in richer contexts, even with a slight recall dip.

Table 5.9: Evaluation Metrics for LLMs with 15-SSDD and 30-SSDD. The green cells indicate the best score for the specific class and metric, while the blue cells indicate the second-best score.

Model	**Class**	**15-SSDD**			**30-SSDD**		
		Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
DistilBERT	Corner	**82.72%**	65.66%	73.21%	**79.58%**	71.00%	75.04%
	Foul	84.01%	**94.14%**	88.79%	87.12%	91.99%	89.49%
	Goal	75.83%	57.96%	65.70%	71.72%	66.24%	68.87%
all-MiniLM-L6-v2	Corner	74.94%	70.01%	72.39%	75.17%	**74.00%**	74.58%
	Foul	86.21%	89.90%	88.02%	87.79%	90.15%	88.95%
	Goal	71.17%	63.69%	67.23%	72.22%	62.10%	66.78%
BERT BASE	Corner	74.82%	69.90%	72.28%	78.00%	70.56%	74.10%
	Foul	86.48%	89.90%	88.16%	87.15%	**92.23%**	**89.62%**
	Goal	73.61%	67.52%	70.43%	**75.55%**	65.92%	70.41%
BERT LARGE	Corner	80.61%	**70.90%**	**75.44%**	78.55%	72.44%	**75.38%**
	Foul	**86.62%**	92.79%	**89.60%**	**87.95%**	91.12%	89.50%
	Goal	**78.81%**	67.52%	**72.73%**	71.57%	**69.75%**	**70.65%**

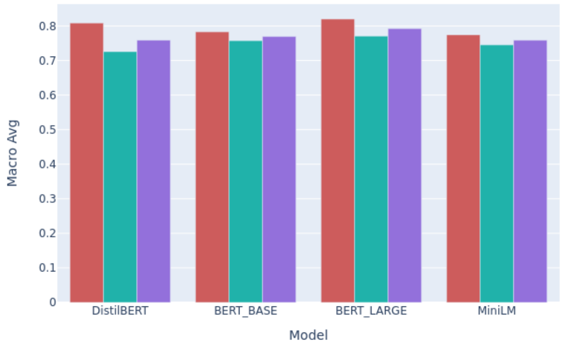


Figure 5.13: Overview of the Macro-average for the 15-SSDD.

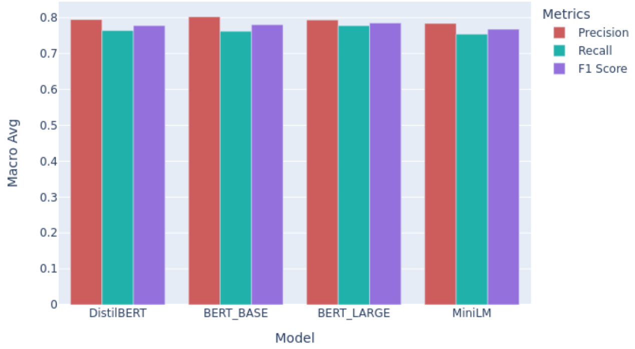


Figure 5.14: Overview of the Macro-average for the 30-SSDD.

Analysis of macro-averages in Figures 5.13 and 5.14 reveals a clear trend for the F1-score metric. while all models benefit from the extended temporal context provided by the 30-SSDD, the degree of improvement varies. The macro-average indicates a general trend of enhanced model performance, with the longer samples demonstrating higher F1-scores across all classes, suggesting that the models are better able to leverage the additional context to refine their predictions across all classes.

When evaluating the macro-average, all-MiniLM-L6-v2 exhibits a clear enhancement with the transition to longer samples. The macro-averaged F1-score rose from 75.88% to 76.77%, and while BERT LARGE showed a slightly higher macro-averaged F1-score, it did not maintain the same level of throughput efficiency. In the realm of computational efficiency, all-MiniLM-L6-v2 demonstrated a slight increase in runtime from 3.387 seconds to 3.6593 seconds when handling the longer samples, a modest trade-off compared to the significant increase for BERT LARGE, from 5.82 seconds to 10.76 seconds. This significant difference in processing times is critical in operational settings where time and resource constraints are pivotal factors.

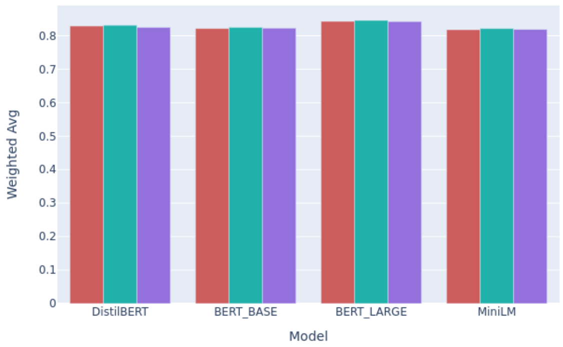


Figure 5.15: Overview of the Weighted-average for 15-SSDD.

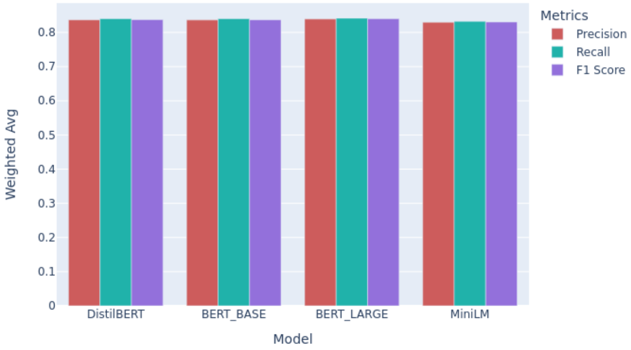


Figure 5.16: Overview of the Weighted-average for 30-SSDD.

The weighted-average is presented in Figures 5.15 and 5.16. Further substantiating this trend, showing a pronounced improvement in model performance with the longer samples. This improvement is especially noteworthy given the inherent class imbalances in the datasets, affirming that the models are not only improving their overall predictive accuracy but also doing so in a manner that more evenly addresses all classes. Both macro-average and weighted-averages favour the 30-SSDD across all classes, suggesting that longer samples benefit model performance despite class imbalances in the data.

The all-MiniLM-L6-v2 emerges as the best-performing model based on its balanced metrics and computational efficiency. It provides a favourable trade-off between precision, recall, and processing demands. While it may not achieve the highest scores in minority classes such as Corner or Goal, it offers commendable performance across the board. If computational efficiency is a lesser concern and the focus is primarily on detecting minority classes with higher precision, then BERT LARGE becomes the model of choice. Its deeper and more complex architecture yields better results for such specific applications despite requiring more substantial computational resources.

5.3 Chapter Summary

This chapter conducts a detailed analysis of various LLMs like DistilBERT, BERT BASE, BERT LARGE, and all-MiniLM-L6-v2, assessing their performance on 15-SSDD and 30-SSDD. The evaluation focuses on precision, recall, and F1-score metrics to gauge each model's effectiveness in identifying events within imbalanced datasets, with additional insights from loss, runtime efficiency, and computational throughput metrics.

This chapter begins with an individual evaluation of the LLMs, emphasising performance differences across various classes and sample durations. All models demonstrate improved performance with longer samples, indicating a preference for more contextual data in soccer event detection. The models particularly struggle with classifying Goal events, which exhibit the highest misclassification rates across all classes and datasets. Corner events are the second-best performing class, with BERT BASE achieving the lowest F1-score of 72.28% among the models and BERT LARGE reaching the highest at 75.44%. Meanwhile, the majority class Foul achieved high results across all the models.

Lastly, the chapter concludes with a comparative analysis of all models. It uses both macro and weighted averages to provide a comprehensive view of each model's effectiveness across different event types and sample lengths while considering the impact of class imbalances. This benchmarking highlights significant performance variations and trends across the models. The analysis reveals that all-MiniLM-L6-v2 maintains robust performance across metrics and sample sizes while utilising fewer computational resources than the other models. This suggests its suitability for applications requiring high efficiency and accuracy. Insights and findings derived from these evaluations are further discussed in the following chapter.

Chapter 6

Discussion

The previous chapters have detailed empirical findings on the performance of four LLMs: DistilBERT, BERT BASE, BERT LARGE, and all-MiniLM-L6-v2. This chapter builds on those analyses by revisiting the objectives and problem statement outlined in Chapter 1. It provides a clear overview of the contributions made in this thesis. Additionally, this chapter discusses the limitations observed throughout the research, highlighting the challenges encountered and suggesting methods for future mitigation. It further explores opportunities for enhancing the datasets through improvements in class imbalances, data preprocessing, and audio. The discussion concludes by considering potential use cases and applications.

6.1 Contributions

This thesis has contributed to advancing the field of sports analytics, with a particular focus on soccer event detection, through innovative dataset creation and the use of LLMs. By extracting and interpreting textual information from SoccerNet-Echoes, this research has not only filled a critical gap in current methodologies but has also established a basis for further development in audio-based event detection.

- **Application of LLM models:** The application of LLM architectures have enhanced the precision in classifying key soccer events from audio transcripts. This has broadened the understanding of how textual data derived from audio can be effectively used in sports contexts.

- **Methodological Advancements:** The establishment of a methodological pipeline for generating and analysing supervised soccer text datasets represents a significant stride towards systematising the process of audio event detection in sports. This pipeline covers everything from data collection and preprocessing to model training and final evaluations, ensuring a comprehensive approach to the research.

- **Dataset Creation:** The construction of two specialised datasets has not only facilitated this specific research but also provided a foundation for ongoing and future academic work. These resources are pivotal for continued enhancements in the field of ASR considering sports analytics.

- **Efficacy of Audio-Based Detection:** Demonstrating the efficacy of audio-based event detection systems through this research has provided new insights into the comparative performance of various LLM models. This contribution is crucial for setting benchmarks for future research in this area.
- **Software Contribution:** The software developed for this thesis is publicly available on GitHub (`https://github.com/simula/forzify`). Its accessibility promotes further research and collaboration, extending the impact of this work beyond the academic community into broader practical applications.

These contributions collectively enhance the capabilities of event detection systems to operate more effectively and with fewer resources compared to traditional video-based methods. By addressing these significant gaps, this thesis enriches the theoretical and practical landscapes of sports analytics. It paves the way for future innovations that could transform the monitoring and analysis of sports events worldwide.

6.2 Revisiting the Problem Statement

This section aims to reexamine the problem statement initially outlined in Chapter 1, providing a comprehensive review of how each research objective has been addressed throughout this thesis. The purpose is to systematically summarise the research findings, offering detailed insights into the methodologies employed, the results obtained, and the extent to which each objective was fulfilled. This evaluation will also explore how the objectives enhanced the understanding of the research question.

Objective 1

Construct two distinct datasets using automatic speech recognition to transcribe game audio and metadata, with both datasets tailored to the classes: Goal, Foul, and Corner.

This objective was conducted in Chapter 3, detailing the development of two supervised datasets specifically designed for text classification in the context of soccer event detection. These datasets leverage Automatic Speech Recognition technology to transcribe game audio data from SoccerNet V2. This transcription was provided by the SoccerNet-Echoes Whisper Large V1 application. The transcribed text and the metadata consisted of 1500 JSON files, which were then synchronised to construct the datasets. Preprocessing of the dataset commenced after the construction of the 15-SSDD and 30-SSDD detailed in Chapter 3. The STCD SoccerNet V2 dataset contained over 100,000 samples. It was then narrowed down to focus on three specific classes: Fouls, Goals, and Corners. This resulted in a reduction to approximately 17,000 samples. Before defining these classes to the datasets, window sizing had to be considered. Given the fast-paced nature of soccer, where events occur frequently, two window sizes were chosen to capture these occurrences effectively: 15-second and 30-second standard deviated time intervals. These windows are designed to record the 15 and 30 seconds before and after an event, encompassing 30 seconds and one minute of game time surrounding each event. This approach ensures sufficient context for accurately classifying each soccer event within the matches.

Chapter 4 presents the methodologies employed, providing a detailed explanation alongside a visual pipeline representation in Figure 4.1. This figure outlines the sequential steps involved in dataset development, from initial audio transcription to evaluation. It effectively highlights the workflow and integration of various data processing stages. The results from the research presented in Chapter 5 reveal that the LLMs benefited from increased contextual information for event classification. The 30-second standard deviated dataset achieved the best overall performance. Based on these findings, larger contextual windows are recommended to enhance event identification accuracy.

Objective 2

Analyse and compare the effectiveness of LLMs in event detection across the distinct datasets.

This objective was addressed in Chapter 5, focusing on a comparative evaluation of various LLMs for their effectiveness in event detection. The models were evaluated by comparing performance across two different dataset durations (15-SSDD and 30-SSDD). The initial step involved internally assessing each model's performance on the datasets using metrics such as precision, recall, and F1-score. The following evaluations compared the performance across these datasets within each model, revealing that all models performed better with longer samples. This suggests the benefit of providing greater contextual information for accurate event classification. Lastly, the external evaluation presented a benchmark analysis that compares the performance across the different LLM architectures, also considering their computational efficiency.

From this external benchmark among the models, the all-MiniLM-L6-v2 was identified as the best-performing model overall. This evaluation was based on the results for each class while also considering additional factors such as loss, runtime efficiency, size, and computational throughput. Originating as a distilled version from the BERT BASE model, all-MiniLM-L6-v2 was the smallest model in this research framework, as detailed in Table 4.1. Despite its size, this model achieved high accuracy in event classification while maintaining efficient runtime. This combination of accuracy and efficiency makes all-MiniLM-L6-v2 a strong candidate for real-world soccer event detection applications. Smaller models like the all-MiniLM-L6-v2 excel in balanced performance but lack precision for minority classes. In contrast, larger models like BERT LARGE prioritise rare events more accurately, resulting in better classification for minority classes but requiring substantial computational resources.

Problem Statement

The objective outlined in Chapter 1 serves as a guide to answer the overall research question mentioned in the problem statement. Upon addressing all the objectives, this thesis concludes by answering the main question:

How can an automatic soccer event detection system be developed by integrating game metadata, game audio with ASR, and LLMs?

Based on the research conducted and the objectives achieved, this thesis has demonstrated the feasibility of building an automatic event detection system for classifying events in soccer. The detailed methodologies provided in Chapters 3 and 4 outline the steps used to tackle this research question. This approach could be extended to encompass all 17 event classes in the SoccerNet V2 dataset and include samples with varying time intervals for a more comprehensive analysis. There is also potential for experimenting with a range of models, from smaller than the all-MiniLM-L6-v2 to larger than

BERT LARGE. Further development based on this thesis could lead to the creation of a real-time system deployable during games, enhancing the utility and impact of the research. The possibilities for expanding upon this research are substantial, offering significant opportunities for future advancements in sports analytics.

6.3 Limitations and Future Work

This section explores potential avenues for further development to enhance the soccer event detection approach presented in this thesis. It will explore opportunities for improvement in addressing class imbalances, expanding and diversifying the datasets, and optimising data preprocessing techniques. These areas are critical for advancing the accuracy and efficiency of event detection systems and will contribute to more robust and scalable solutions in soccer event detection.

6.3.1 SoccerNet V2

SoccerNet V2 is the largest publicly accessible dataset for soccer event detection, consisting of 300,000 annotations temporally anchored to 764 hours of raw video footage [1]. The untrimmed nature of the dataset makes it difficult to pinpoint the exact temporal location of event occurrences. This is particularly critical when evaluating goals within the context of full matches. Consequently, deploying an automatic event detection system for real-time applications in live games becomes challenging.

The thesis addresses the hardware limitations that can arise during this research. The SoccerNet V2 dataset [1] is notably large, necessitating advanced hardware for efficient processing. Employing LLMs necessitates advanced computational resources, primarily since these models may comprise millions of parameters, with the most extensive reaching up to 340 million parameters. The automatic speech recognition model also demands significant computational power to accurately translate and transcribe audio from soccer games. To address these technological constraints, this study recommends leveraging GPUs for their ability to accelerate computations.

6.3.2 Class Imbalance

Chapter 3 offers a detailed analysis of the class distribution for the constructed datasets. It illustrates the broad range of event frequencies in soccer matches, highlighting the challenge of imbalanced class distribution. The datasets feature two distinct durations, each containing the classes Foul, Corner, and Goal. The distribution across these classes varies: Foul is the majority class, representing 64% of events, followed by Corner at 26.7%, and Goal as the minority class at 9.33%. This distribution is visualised in Figure 6.1, which shows the class representation for the 15-SSDD. Both durations exhibit approximately the same representation across these classes, underscoring the consistent challenge of addressing the class imbalance in event detection.

To address this issue, this thesis has utilised metrics that consider the challenges of class imbalance. The Optuna hyperparameter search was conducted with 32 trials, focusing on optimising the Macro F1-score. This metric evaluates the predictions by treating all classes equally, regardless of their frequency. This approach optimises the hyperparameter search to perform well across all classes, with a particular focus on the minority classes such as Corner and Goal.

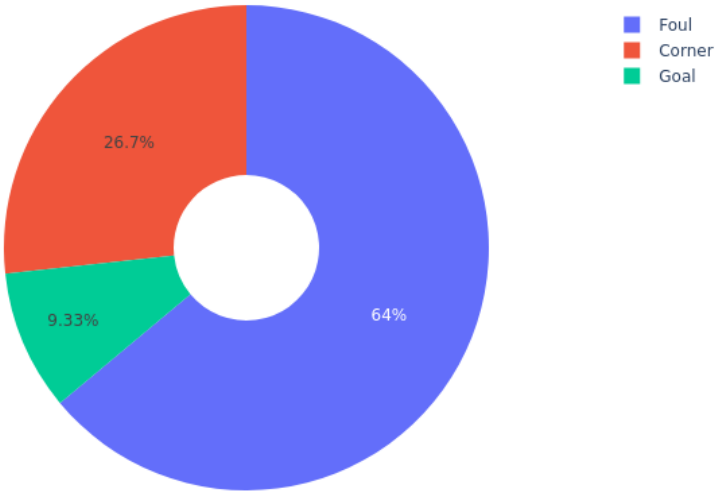


Figure 6.1: This per cent representation shows how the class imbalanced the 15-SSDD is considering the classes Foul, Corner, and Goal.

Further development could facilitate oversampling, undersampling, or cost-sensitive learning to focus on identifying the different classes more clearly. Downsampling or undersampling is a method where the majority class or classes are reduced to match the other classes, making the representation of the classes the same. The dataset becomes more balanced, and the training and evaluation steps become more efficient. The main issue is that the undersampling removes information that might have been valuable [60].

Oversampling is a technique used to balance imbalanced datasets by increasing the number of samples in the minority class. While it avoids information loss, it can introduce other issues. There are two main approaches to oversampling: creating new samples or replicating existing minority class samples. However, replicating existing samples can lead to overfitting, especially if the generated data is too similar to the existing data [25, 64].

Another approach to addressing class imbalance is cost-sensitive learning. Unlike traditional classification algorithms that treat all misclassifications equally, cost-sensitive learning assigns different penalties to the class misclassification errors. This allows the model to prioritise correctly classifying the minority class, which is often more crucial in imbalanced datasets [51].

The last method, which might be the most preferred approach, is word embedding oversampling. Ruifeng Xu et al. [31] tackle class imbalance in sentiment and emotion classification of text data with a comprehensive strategy. They develop word embeddings by training a continuous skip-gram model on a large corpus to capture both syntactic and semantic features of words. These embeddings are then used to construct sentence vectors through a recursive neural tensor network (RNTN), integrating the semantic context of entire sentences.

To ensure fairness and balance in the training dataset, the Synthetic Minority Over-sampling Technique (SMOTE) is used to generate synthetic samples for underrepresented classes [25]. This method effectively addresses the model's bias toward majority classes by ensuring that minority classes are adequately represented. The combination of advanced word embedding, semantic sentence vector construction, and strategic oversampling forms the Word Embedding Compositionality with Minority Oversampling Technique (WEC-MOTE). This approach not only remedies class imbalance but also enhances the model's ability to discern and accurately classify complex sentiment nuances in text, leading to improved classification outcomes [31].

6.3.3 Data Preprocessing

The data preprocessing steps outlined in Chapter 4 included removing missing values, eliminating stopwords, tokenisation, and splitting the data into training, testing, and validation sets. Future improvements in preprocessing should focus on the transcription process of the datasets. These datasets are derived from ASR transcriptions of soccer commentaries. A recurring issue with direct transcription is repetition, which occurs more frequently than might be expected, as commentators often repeat themselves. Implementing a method to reduce this repetition could diminish the prevalence of low-sentiment information, potentially enhancing the model's classification accuracy. Additionally, it is crucial to analyse and remove events that lack sufficient textual information. Since the Whisper model does not always accurately capture audio content, establishing a backup system to correct or remove flawed transcriptions could improve data quality. Further preprocessing techniques can be used as stemming or lemmatisation. Additional preprocessing techniques like stemming or lemmatisation can be employed to standardise text data, potentially enhancing text processing efficiency by reducing the variability in word forms [71].

6.3.4 Audio Features

The integration of audio feature extraction could facilitate increased accuracy through a multimodal approach that incorporates textual information, metadata, and audio features. As discussed in the literature review in Chapter 2, Vanderplaetse et al. [81] utilised audio in conjunction with video streams to enhance the AmAP. This study emphasises the significance of audio cues alongside visual data to improve the accuracy and effectiveness of action detection in soccer videos. Specifically, it highlights how audio can significantly contribute to distinguishing key moments like goals, where crowd reactions provide valuable contextual information. Implementations like this could provide models with additional context that enhances their ability to accurately classify critical events, particularly in identifying goal occurrences.

6.4 Use Cases and Applications

In the world of online sports content, videos are rich in data but heavy to process. The presented pipeline addresses this challenge by leveraging game audio rather than visual cues to detect key events in soccer matches. This approach significantly reduces computational demands and makes the processing faster and more efficient.

For applications like soccer analytics, this method can revolutionise how events are monitored and analysed. Platforms such as Forzify from Forzasys exemplify how such a pipeline could be integrated into existing platforms to enhance sports analytics. Forzify is an extensive live and on-demand video management system that supports Over-The-Top (OTT) streaming. It functions across multiple platforms with backend infrastructure, including metadata management, web servers, and search engines. This system can be hosted either in the cloud or on local servers, enabling Forzify to serve not only as a streaming platform but also as a searchable archive, enriching user engagement with sports content [53].

By implementing this research's audio-based event detection technology, Forzify could offer real-time alerts and insights during soccer matches. This capability would enable platforms like Forzify to deliver instant notifications about significant events such as Goals, Fouls, Corners, and other key moments directly to viewers. Such features could dramatically enhance the fan experience during live broadcasts by automatically providing interactive content that engages viewers with timely and relevant information. This approach would improve the viewer experience and offer coaches and analysts a powerful tool to access specific game events for tactical and performance analysis.

The application of this technology could extend beyond real-time event notification. For example, it could enable enhanced content discoverability within Forzify's archival system, allowing users to search for and locate specific events in recorded matches. Fans could easily find and rewatch their favourite moments, coaches could analyse game tactics, and analysts could gather detailed game insights.

The use of an audio-based event detection system for soccer not only streamlines the processing of sports events but also opens up new possibilities for interactive fan engagement and detailed game analysis, all in a more resource-efficient manner.

6.5 Chapter Summary

This chapter builds upon the empirical findings from all of the previous chapters. It revisits the problem statement set out at the beginning of Chapter 1, assesses the contributions made from this thesis, discusses encountered limitations, and suggests potential improvements and applications.

The thesis achieved its first objective by creating 15-SSDD and 30-SSDD using ASR transcription from SoccerNet-Echoes, focusing on the classes Goal, Foul, and Corner. These datasets were designed to capture the dynamics of soccer events within 15-second and 30-second windows, providing adequate context for effective classification.

The second objective involved a thorough evaluation of the LLMs across varying data durations. The analysis confirmed that models benefit from larger contextual windows, improving their ability to classify events accurately. The all-MiniLM-L6-v2 model emerged as particularly effective due to its balance of high accuracy and computational efficiency.

The chapter outlines key limitations such as class imbalances and transcription accuracy. It proposes future enhancements, including refined data preprocessing, audio implementation, and exploring advanced class balancing techniques like word embedding oversampling, undersampling, or cost-sensitive learning to improve model training and prediction accuracy.

This chapter also delineates the significant contributions of the research, including the development of a structured event detection pipeline, utilisation of LLM architectures and a comparative benchmark, construction of two supervised datasets, and open-source software (GitHub repository). It reflects on how this thesis advances the field of sports analytics by leveraging audio data to efficiently detect and analyse soccer events, proposing a scalable approach for broader sports applications.

The discussion concludes with practical applications, emphasising how the audio-based event detection system can be integrated into existing platforms like Forzify, an OTT streaming and video management system. Such integration would allow real-time alerts during soccer matches, enhancing fan engagement and providing valuable analytics for coaching and tactical analysis.

Chapter 7

Conclusion

The motivation for this thesis stems from the significant challenges associated with manual annotations in soccer event detection, which are labour-intensive, costly, and not scalable, especially during tournaments with numerous matches. To address these inefficiencies, this research developed an automated audio-based event detection system that is designed to bypass the extensive resource requirements of traditional video analysis methods. This system enhances speed and requires fewer computational resources. It could also provide real-time actionable insights that improve strategic decisions and fan engagement during live broadcasts.

This thesis presents an advancement in sports analytics with SoccerNet-Echoes implementation of Automatic Speech Recognition (ASR) technology, specifically the Whisper Large V1 model, to detect soccer events through game audio. The transcribed ASR text combined with metadata resulted in two distinct supervised datasets (15-SSDD and 30-SSDD). These datasets were used for training models to recognise soccer events such as Goals, Fouls, and Corners. The findings reveal that longer text samples significantly enhance the model's ability to classify events accurately, underscoring the importance of contextual information.

The methodology starts with preprocessing the datasets, removing missing values, eliminating stopwords, tokenising the text, and splitting the data into training, testing, and validation sets. This ensures quality inputs for the LLM models. It highlights the use of the Huggingface platform for its comprehensive library of pre-trained models and its compatibility with major machine learning frameworks like `PyTorch`. The training process involves rigorous hyperparameter tuning with the Optuna framework to optimise model performance, supported by high-performance computing resources from Orion for large-scale training. This structured approach is a vital process in research for obtaining quality results.

A comprehensive analysis across several large language models (LLMs), including DistilBERT, BERT BASE, BERT LARGE, and all-MiniLM-L6-v2, demonstrated that models equipped with longer sample durations performed better. Among these, all-MiniLM-L6-v2 was notable for its balance of high accuracy and computational efficiency, making it particularly suitable for real-world applications where rapid and precise event detection is crucial.

The research also encountered challenges such as class imbalance and transcription accuracy, which could impact the overall effectiveness of the detection system. To refine this, future enhancements include advanced data preprocessing techniques and audio implementation. Exploring class balancing strategies like word embedding oversampling and cost-sensitive learning is anticipated to improve the robustness and effectiveness of the system.

Future research could expand by integrating multimodal data sources that combine game audio, metadata, and visual cues to enrich the event detection process. Addressing class imbalance with techniques such as the Synthetic Minority Over-sampling Technique (SMOTE) combined with word embedding and exploring applications in other sports could significantly widen the impact of this research.

In conclusion, this thesis advances the field of sports analytics by introducing a diligent audio-based event detection system. It lays the groundwork for future innovations that could transform how sports events are monitored and analysed. The potential for real-time, efficient, and accurate event detection holds significant promise for enhancing viewer experiences and providing sports professionals with valuable insights.

Bibliography

[1] Adrien Deliège et al. "SoccerNet-v2: A Dataset and Benchmarks for Holistic Understanding of Broadcast Soccer Videos". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. IEEE, June 2021, pp. 4503–4514. DOI: 10.1109/CVPRW53098.2021.00508.

[2] Alec Radford et al. "Robust Speech Recognition via Large-Scale Weak Supervision". In: *Proceedings of the 40th International Conference on Machine Learning (ICML)*. Ed. by Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett. Vol. 202. Proceedings of Machine Learning Research. PMLR, July 23–29, 2023, pp. 28492–28518. URL: https://proceedings.mlr.press/v202/radford23a.html.

[3] Andrej Karpathy et al. "Large-Scale Video Classification with Convolutional Neural Networks". In: *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014, pp. 1725–1732. DOI: 10.1109/CVPR.2014.223.

[4] Ang Li et al. "The AVA-Kinetics Localized Human Actions Video Dataset". In: *arXiv:2005.00214 [cs.CV]* abs/2005.00214 (2020). arXiv: 2005.00214. URL: https://arxiv.org/abs/2005.00214.

[5] Anthony Cioppa et al. "A Context-Aware Loss Function for Action Spotting in Soccer Videos". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2020, pp. 13126–13136.

[6] Anthony Cioppa et al. "SoccerNet 2023 Challenges Results". In: *arXiv preprint arXiv:2309.06006* abs/2309.06006 (2023). arXiv: 2309.06006. URL: https://arxiv.org/abs/2309.06006.

[7] Ashish Vaswani et al. "Attention is All You Need". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS)*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 6000–6010. ISBN: 978-1-5108-6096-4.

[8] Chunhui Gu et al. "AVA: A Video Dataset of Spatio-Temporally Localized Atomic Visual Actions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2018.

[9] Diksha Khurana et al. "Natural Language Processing: State of the Art, Current Trends and Challenges". In: *Multimedia Tools and Applications* 82 (2023), pp. 3713–3744. URL: https://doi.org/10.1007/s11042-022-13428-4.

[10] Du Tran et al. "Video Classification with Channel-Separated Convolutional Networks". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV).* IEEE, 2019, pp. 5552–5561.

[11] Fabian Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *The Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[12] H. Kuehne et al. "HMDB: A large video database for human motion recognition". In: *Proceedings of the 2011 International Conference on Computer Vision (ICCV).* IEEE, 2011, pp. 2556–2563. DOI: `10.1109/ICCV.2011.6126543`.

[13] H. Riaz et al. "Anomalous Human Action Detection Using a Cascade of Deep Learning Models". In: *Proceedings of the 2021 9th European Workshop on Visual Information Processing (EUVIP).* IEEE, 2021, pp. 1–5.

[14] Hang Zhao et al. "HACS: Human Action Clips and Segments Dataset for Recognition and Temporal Localization". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV).* IEEE, Oct. 2019.

[15] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *arXiv preprint arXiv:1810.04805* abs/1810.04805 (2018). arXiv: `1810.04805`. URL: `https://arxiv.org/abs/1810.04805`.

[16] Jia Deng et al. "ImageNet: A Large-Scale Hierarchical Image Database". In: *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE, 2009, pp. 248–255.

[17] João Carreira et al. "A Short Note about Kinetics-600". In: *arXiv:1808.01340 [cs.CV]* abs/1808.01340 (2018). arXiv: `1808.01340`. URL: `https://arxiv.org/abs/1808.01340`.

[18] João Carreira et al. "A Short Note on the Kinetics-700 Human Action Dataset". In: *arXiv:1907.06987 [cs.CV]* abs/1907.06987 (2019). arXiv: `1907.06987`. URL: `https://arxiv.org/abs/1907.06987`.

[19] Johnson K et al. "A History-Taking System That Uses Continuous Speech Recognition". In: *Proceedings of the Symposium on Computer Applications in Medical Care (SCAMC).* American Medical Informatics Association, 1992, pp. 757–761.

[20] Jort F. Gemmeke et al. "Audio Set: An Ontology and Human-Labeled Dataset for Audio Events". In: *Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2017, pp. 776–780.

[21] Joseph Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE, 2016, pp. 779–788.

[22] Julien Denize et al. "COMEDIAN: Self-Supervised Learning and Knowledge Distillation for Action Spotting Using Transformers". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops.* IEEE, Jan. 2024, pp. 530–540.

[23] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE, 2016, pp. 770–778.

[24] Kanav Vats et al. "Event Detection in Coarsely Annotated Sports Videos via Parallel Multi-Receptive Field 1D Convolutions". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPR).* IEEE, 2020, pp. 882–883.

[25] Kevin W. Bowyer et al. “SMOTE: Synthetic Minority Over-sampling Technique”. In: *arXiv preprint arXiv:1106.1813* abs/1106.1813 (2011). arXiv: `1106.1813`. URL: `https://arxiv.org/abs/1106.1813`.

[26] Limin Wang et al. “VideoMAE V2: Scaling Video Masked Autoencoders with Dual Masking”. In: *Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 14549–14560. DOI: `10.1109/CVPR52729.2023.01398`.

[27] Mathew Monfort et al. “Moments in Time Dataset: One Million Videos for Event Understanding”. In: *CoRR* abs/1801.03150 (2018). arXiv: `1801.03150`. URL: `https://arxiv.org/abs/1801.03150`.

[28] Matteo Tomei et al. “RMS-Net: Regression and Masking for Soccer Event Spotting”. In: *Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 7699–7706.

[29] Peter J. Denning et al. “Computing as a Discipline”. In: *Computer* 22.2 (1989), pp. 63–70.

[30] Relja Arandjelovic et al. “NetVLAD: CNN Architecture for Weakly Supervised Place Recognition”. In: *arXiv preprint arXiv:1511.07247* abs/1511.07247 (2015). arXiv: `1511.07247`. URL: `https://arxiv.org/abs/1511.07247`.

[31] Ruifeng Xu et al. “Word Embedding Composition for Data Imbalances in Sentiment and Emotion Classification”. In: *Cognitive Computation* 7 (2015), pp. 226–240.

[32] Sami Abu-El-Haija et al. “YouTube-8M: A Large-Scale Video Classification Benchmark”. In: *arXiv:1609.08675 [cs.CV]* abs/1609.08675 (2016). arXiv: `1609.08675`. URL: `https://arxiv.org/abs/1609.08675`.

[33] Silvio Giancola et al. “SoccerNet 2022 Challenges Results”. In: *Proceedings of the 5th International ACM Workshop on Multimedia Content Analysis in Sports (MMsports)*. MM ’22. ACM, Oct. 2022, pp. 882–883. DOI: `10.1145/3552437.3558545`. URL: `http://dx.doi.org/10.1145/3552437.3558545`.

[34] Silvio Giancola et al. “SoccerNet 2022 Challenges Results”. In: *Proceedings of the 5th International ACM Workshop on Multimedia Content Analysis in Sports (MMsports)*. MMSports ’22. Lisboa, Portugal: Association for Computing Machinery, 2022, pp. 75–86. ISBN: 9781450394888. DOI: `10.1145/3552437.3558545`. URL: `https://doi.org/10.1145/3552437.3558545`.

[35] Silvio Giancola et al. “SoccerNet: A Scalable Dataset for Action Spotting in Soccer Videos”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2018. DOI: `10.1109/CVPRW.2018.00223`. URL: `https://doi.ieeecomputersociety.org/10.1109/CVPRW.2018.00223`.

[36] Takuya Akiba et al. “Optuna: A Next-generation Hyperparameter Optimization Framework”. In: *arXiv preprint arXiv:1907.10902* abs/1907.10902 (2019). arXiv: `1907.10902`. URL: `https://arxiv.org/abs/1907.10902`.

[37] Thomas Wolf et al. “Transformers: State-of-the-Art Natural Language Processing”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP)*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. URL: `https://www.aclweb.org/anthology/2020.emnlp-demos.6`.

[38] Victor Sanh et al. “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter”. In: *arXiv preprint arXiv:1910.01108* abs/1910.01108 (2019). arXiv: `1910.01108`. URL: `https://arxiv.org/abs/1910.01108`.

[39] Wenhui Wang et al. "MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers". In: *arXiv preprint arXiv:2002.10957* abs/2002.10957 (2020). arXiv: 2002.10957. URL: https://arxiv.org/abs/2002.10957.

[40] Will Kay et al. "The Kinetics Human Action Video Dataset". In: *arXiv:1705.06950 [cs.CV]* abs/1705.06950 (2017). arXiv: 1705.06950. URL: https://arxiv.org/abs/1705.06950.

[41] Xin Zhou et al. "Feature Combination Meets Attention: Baidu Soccer Embeddings and Transformer Based Temporal Detection". In: *arXiv preprint arXiv:2106.14447* abs/2106.14447 (2021). arXiv: 2106.14447. URL: https://arxiv.org/abs/2106.14447.

[42] Yu Zhang et al. "BigSSL: Exploring the Frontier of Large-Scale Semi-Supervised Learning for Automatic Speech Recognition". In: *IEEE Journal of Selected Topics in Signal Processing* 16.6 (2022), pp. 1519–1532. DOI: 10.1109/JSTSP.2022.3182537.

[43] Yudong Jiang et al. "SoccerDB: A Large-Scale Database for Comprehensive Video Understanding". In: *Proceedings of the 3rd International Workshop on Multimedia Content Analysis in Sports (MMsports)*. Association for Computing Machinery, 2020, pp. 1–8.

[44] Yukun Zhu et al. "Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2015, pp. 19–27.

[45] A. S. Aljaloud and H. Ullah. "IA-SSLM: Irregularity-Aware Semi-Supervised Deep Learning Model for Analyzing Unusual Events in Crowds". In: *IEEE Access* 9 (2021), pp. 73327–73334.

[46] Shipra Arora and Rishi Singh. "Automatic Speech Recognition: A Review". In: *International Journal of Computer Applications* 60 (2012), pp. 34–44. DOI: 10.5120/9722-4190.

[47] Steven Bird. "NLTK: The Natural Language Toolkit". In: *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions (COLING/ACL)*. Ed. by James Curran. Sydney, Australia: Association for Computational Linguistics, July 2006, pp. 69–72. DOI: 10.3115/1225403.1225421. URL: https://aclanthology.org/P06-4018.

[48] Giuseppe Bonaccorso. "Machine Learning Algorithms". In: Packt Publishing, 2017. Chap. 1. ISBN: 978-1-78588-962-2.

[49] K. R. Chowdhary. "Natural Language Processing". In: *Fundamentals of Artificial Intelligence*. New Delhi: Springer India, 2020, pp. 603–649. ISBN: 978-81-322-3972-7. DOI: 10.1007/978-81-322-3972-7_19. URL: https://doi.org/10.1007/978-81-322-3972-7_19.

[50] Matthieu Cord and Pádraig Cunningham. "Machine Learning Techniques for Multimedia". In: Springer, 2008. Chap. 2. ISBN: 978-3-540-75171-7. DOI: 10.1007/978-3-540-75171-7.

[51] Charles Elkan. "The Foundations of Cost-Sensitive Learning". In: *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI)*. Vol. 17. 1. Lawrence Erlbaum Associates Ltd. Lawrence Erlbaum Associates Ltd, 2001, pp. 973–978.

[52] Fédération Internationale de Football Association (FIFA). *FIFA Annual Report 2022: Football Unites the World*. Fédération Internationale de Football Association (FIFA). 2022. URL: https://digitalhub.fifa.com/m/2252cd6dfdadad73/original/FIFA-Annual-Report-2022-Football-Unites-The-World.pdf (visited on 01/14/2023).

[53] Forzasys. *Forzify*. URL: https://www.forzasys.com/Forzify.html (visited on 05/01/2023).

[54] Sushant Gautam et al. "SoccerNet-Echoes: A Soccer Game Audio Commentary Dataset". In: *arXiv* (May 12, 2024). DOI: 10.48550/arXiv.2405.07354. arXiv: 2405.07354. URL: https://doi.org/10.48550/arXiv.2405.07354 (visited on 05/14/2024).

[55] GeeksforGeeks. *F1 Score in Machine Learning*. Dec. 27, 2023. URL: https://www.geeksforgeeks.org/f1-score-in-machine-learning/ (visited on 02/17/2023).

[56] Silvio Giancola and Bernard Ghanem. "Temporally-Aware Feature Pooling for Action Spotting in Soccer Broadcasts". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2021, pp. 4490–4499.

[57] Yoshua Bengio Ian Goodfellow and Aaron Courville. *Deep Learning*. MIT Press, 2016. ISBN: 978-1-4471-5779-3.

[58] Soo Min Kang and Richard P. Wildes. "Review of Action Recognition and Detection Methods". In: *arXiv:1610.06906 [cs.CV]* abs/1610.06906 (2016). arXiv: 1610.06906.

[59] Amir Roshan Zamir Khurram Soomro and Mubarak Shah. "UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild". In: *CoRR* abs/1212.0402 (2012). arXiv: 1212.0402 [abs/1212.0402]. URL: http://arxiv.org/abs/1212.0402.

[60] Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. "Exploratory Undersampling for Class-Imbalance Learning". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39.2 (2008), pp. 539–550.

[61] Edward Loper and Steven Bird. "NLTK: The Natural Language Toolkit". In: *arXiv preprint arXiv:cs/0205028* cs.CL/0205028 (2002). arXiv: cs/0205028. URL: https://arxiv.org/abs/cs/0205028.

[62] Ivan Laptev Marcin Marszalek and Cordelia Schmid. "Actions in Context". In: *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009, pp. 2929–2936. DOI: 10.1109/CVPR.2009.5206557.

[63] Enrico Bagli Margherita Grandini and Giorgio Visani. "Metrics for Multi-Class Classification: An Overview". In: *arXiv preprint arXiv:2008.05756* abs/2008.05756 (2020). arXiv: 2008.05756. URL: https://arxiv.org/abs/2008.05756.

[64] Roweida Mohammed, Jumanah Rawashdeh, and Malak Abdullah. "Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results". In: *2020 11th International Conference on Information and Communication Systems (ICICS)*. IEEE, 2020, pp. 243–248.

[65] Andreas C. Müller and Sarah Guido. "Introduction to Machine Learning with Python: A Guide for Data Scientists". In: Sebastopol, CA: O'Reilly Media, Inc., 2016. Chap. 5, pp. 296–300.

[66] NVIDIA. *Quadro RTX 8000 Product Literature*. 2019. URL: https://www.nvidia.com/content/dam/en-zz/Solutions/design-visualization/quadro-product-literature/quadro-rtx-8000-us-nvidia-946977-r1-web.pdf (visited on 04/13/2023).

[67] OpenAI. *Whisper*. 2023. URL: https://openai.com/index/whisper/ (visited on 04/23/2023).

[68] J. Orbach. "Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms". In: *Archives of General Psychiatry* 7.3 (Sept. 1962), pp. 218–219. ISSN: 0003-990X. DOI: 10.1001/archpsyc.1962.01720030064010. eprint: https://jamanetwork.com/journals/jamapsychiatry/articlepdf/488205/archpsyc_7_3_010.pdf. URL: https://doi.org/10.1001/archpsyc.1962.01720030064010.

[69] Soccer-Net Organization. *FAQ - Soccer-Net.* URL: https://www.soccer-net.org/faq (visited on 05/10/2023).

[70] Lawrence R. Rabiner and Ronald W. Schafer. "Introduction to Digital Speech Processing". In: Now Publishers Inc, 2007. Chap. 1 and 2, pp. 1–2, 19–21. ISBN: 978-1-60198-070-0.

[71] Sebastian Raschka and Vahid Mirjalili. *Python Machine Learning.* Packt Publishing, 2019. ISBN: 978-1-78995-575-0.

[72] Guido van Rossum and Fred L. Drake Jr. *Python Tutorial.* Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995.

[73] Karpagavalli S. and Evania Chandra. "A Review on Automatic Speech Recognition Architecture and Approaches". In: *International Journal of Signal Processing, Image Processing and Pattern Recognition* 9 (Apr. 2016), pp. 393–404. DOI: 10.14257/ijsip.2016.9.4.34.

[74] Biswajit Dev Sarma and S. R. Mahadeva Prasanna. "Acoustic-Phonetic Analysis for Speech Recognition: A Review". In: *IETE Technical Review* 35.3 (2018), pp. 305–327. DOI: 10.1080/02564602.2017.1293570. URL: https://doi.org/10.1080/02564602.2017.1293570.

[75] scikit-learn developers. *sklearn.metrics.f1_score.* 2024. URL: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html (visited on 03/08/2023).

[76] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *arXiv preprint arXiv:1409.1556* abs/1409.1556 (2014). arXiv: 1409.1556. URL: https://arxiv.org/abs/1409.1556.

[77] Joao V.B. Soares, Avijit Shah, and Topojoy Biswas. "Temporally Precise Action Spotting in Soccer Videos Using Dense Detection Anchors". In: *Proceedings of the 2022 IEEE International Conference on Image Processing (ICIP).* IEEE, 2022, pp. 2796–2800.

[78] Iryna Sydorenko. *Machine Learning and Training Data: What You Need to Know.* Accessed: 2024-02-06. Mar. 17, 2021. URL: https://labelyourdata.com/articles/machine-learning-and-training-data#splitting_your_data_set_training_data_vs_testing_data_in_machine_learning.

[79] Iryna Sydorenko. *What Is a Dataset in Machine Learning: Sources, Features, Analysis.* Accessed: 2024-02-06. Dec. 21, 2023. URL: https://labelyourdata.com/articles/what-is-dataset-in-machine-learning.

[80] Jerome Friedman Trevor Hastie and Robert Tibshirani. "The Elements of Statistical Learning". In: Springer, 2001. Chap. 2. ISBN: 978-0-387-21606-5. DOI: 10.1007/978-0-387-21606-5.

[81] Bastien Vanderplaetse and Stéphane Dupont. "Improved Soccer Action Spotting Using Both Audio and Video Streams". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE, 2020. arXiv: 2011.04258 [cs.CV].

[82] Dong Yu and Li Deng. "Automatic Speech Recognition: A Deep Learning Approach". In: Springer-Verlag London, 2015. Chap. 2, pp. 13–20. ISBN: 978-1-4471-5779-3.

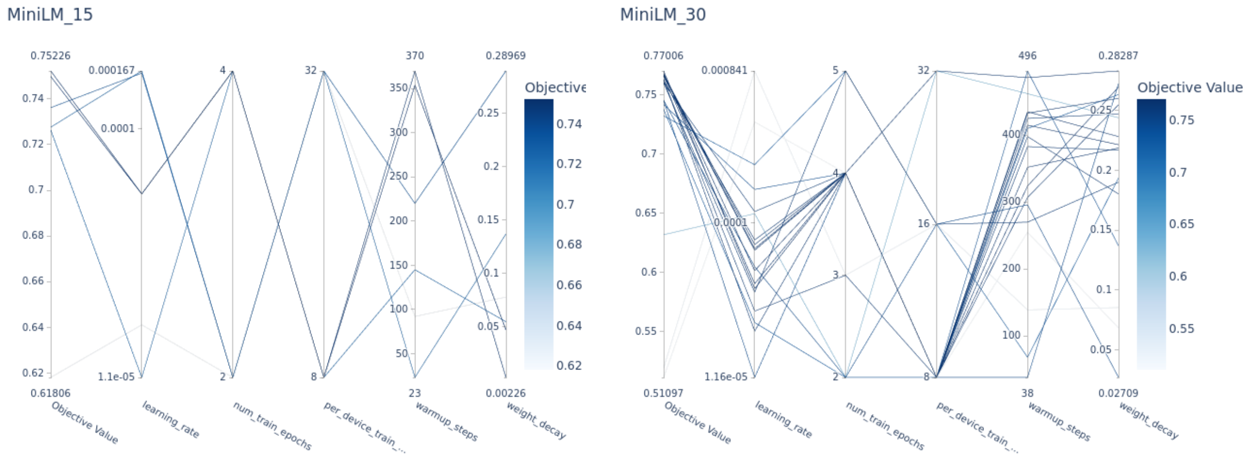
Appendix A

Optuna Hyperparameter Search Visuals

This appendix presents the plots generated from the Optuna hyperparameter search detailed in Chapter 4.

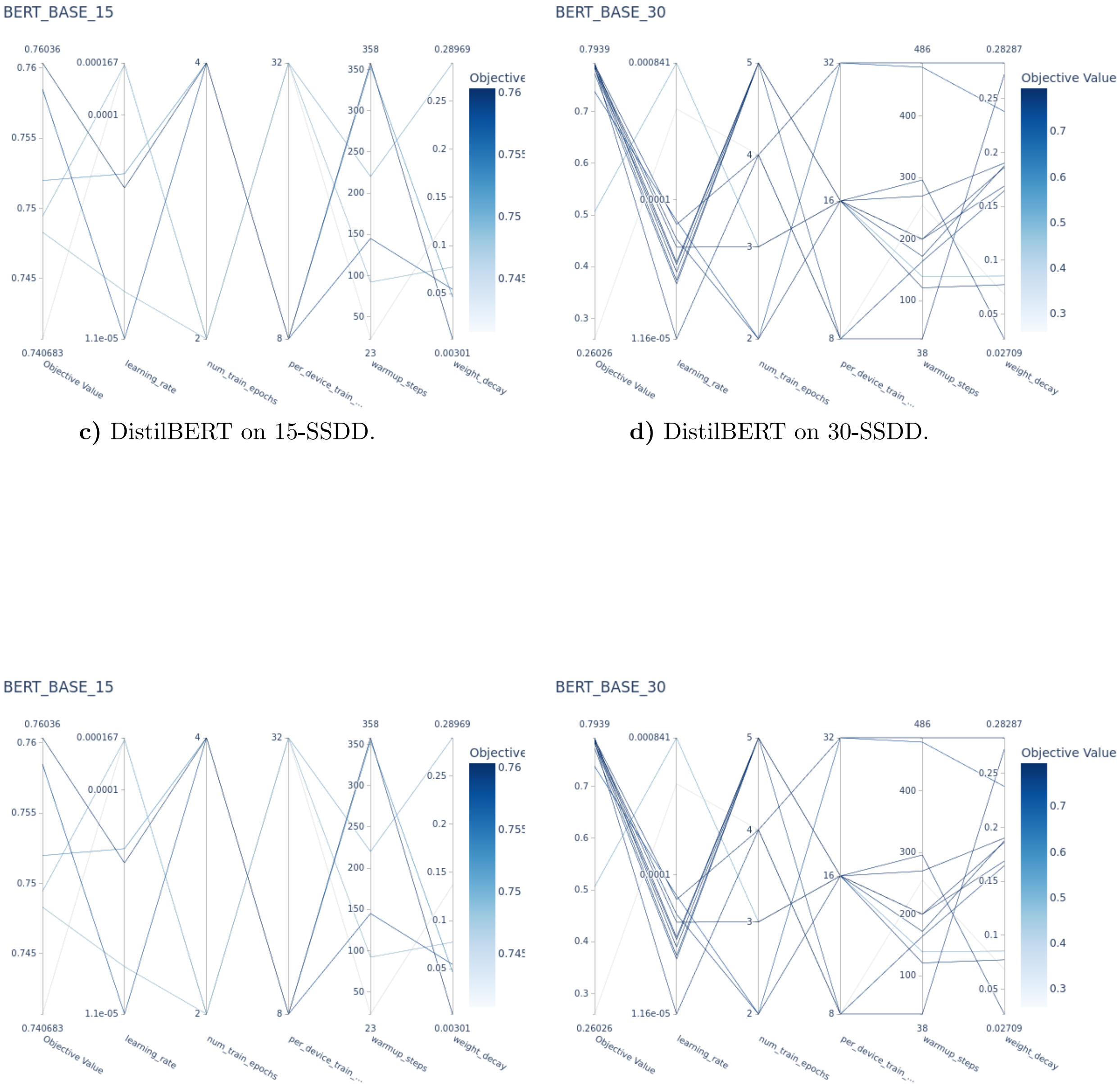
A.1 Parallel Coordinate

These visualisations depict the Macro F1-score achieved for each combination of hyperparameter values explored during the optimisation process. Analysing the distribution of lines across each axis can reveal potential relationships between hyperparameters and their impact on the model's performance. The plots can identify which hyperparameter combinations led to the highest and lowest Macro F1-scores.



a) MiniLM on 15-SSDD.

b) MiniLM on 30-SSDD.

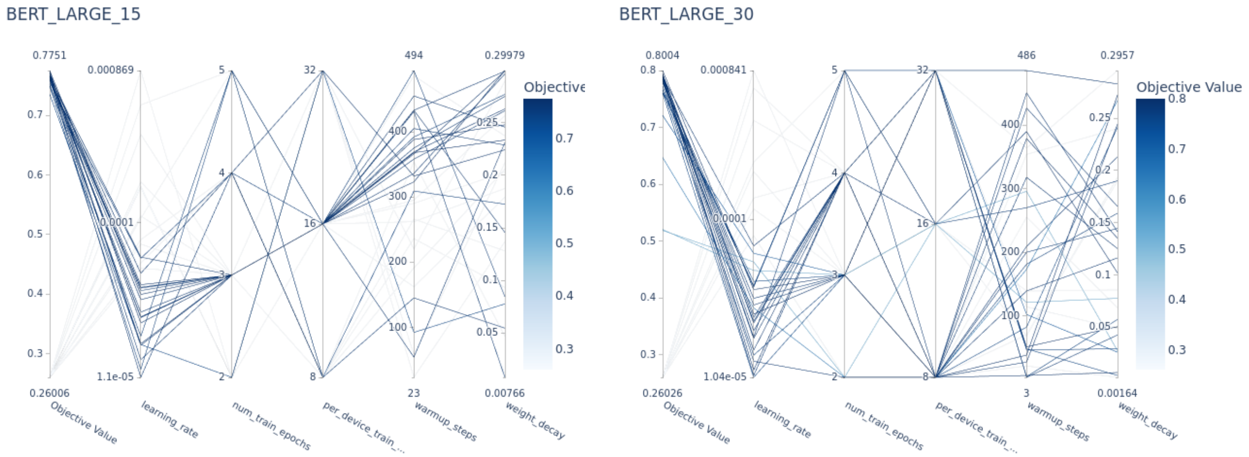


c) DistilBERT on 15-SSDD.

d) DistilBERT on 30-SSDD.

e) BERT BASE on 15-SSDD.

f) BERT BASE on 30-SSDD.



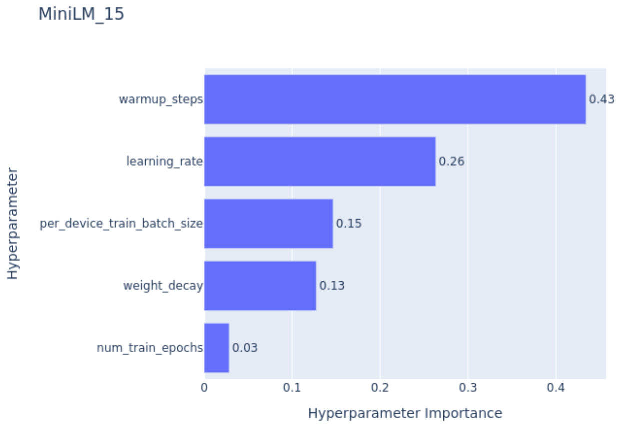
g) BERT LARGE on 15-SSDD.

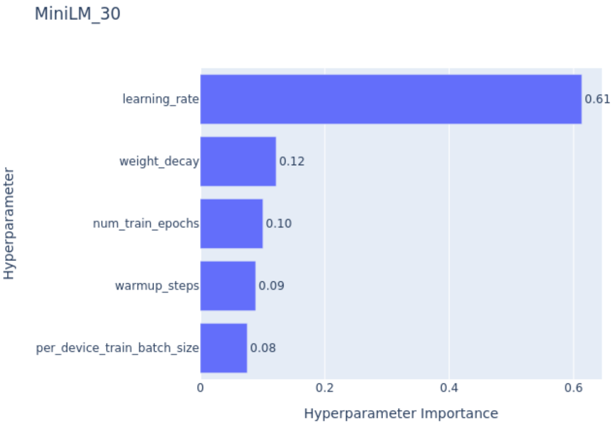
h) BERT LARGE on 30-SSDD.

Figure A.1: Comparative parallel coordinates considering the two datasets and the LLM models.

A.2 Parameter Importance

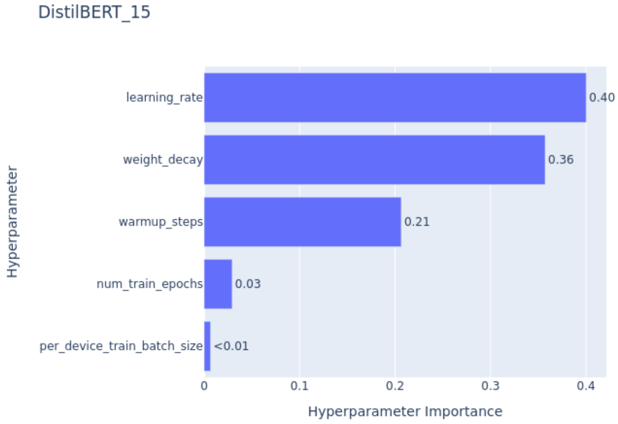
This section presents a bar chart ranking the most influential hyperparameters based on their importance scores.

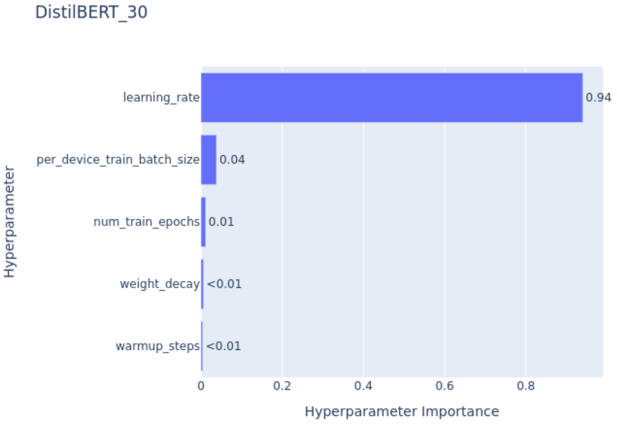




a) MiniLM on 15-SSDD.

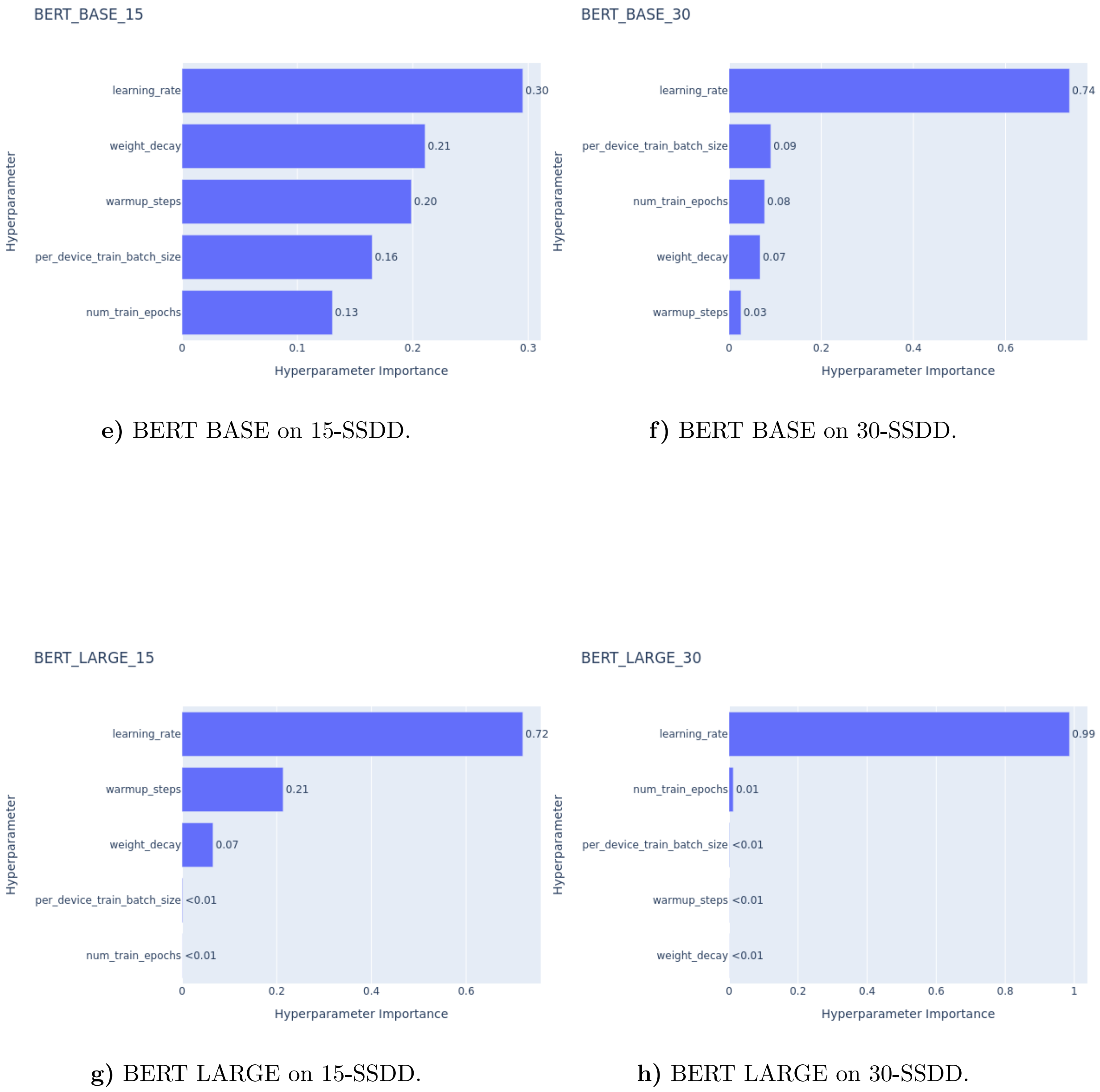
b) MiniLM on 30-SSDD.





c) DistilBERT on 15-SSDD.

d) DistilBERT on 30-SSDD.



e) BERT BASE on 15-SSDD.

f) BERT BASE on 30-SSDD.

g) BERT LARGE on 15-SSDD.

h) BERT LARGE on 30-SSDD.

Figure A.2: Comparative parameter importance considering the two datasets and the LLM models.

A.3 Slice Plots

These plots visualise the objective function value across different hyperparameter combinations. Each plot slices the objective landscape along a single hyperparameter, revealing how the objective function changed regarding that particular hyperparameter while holding others constant.

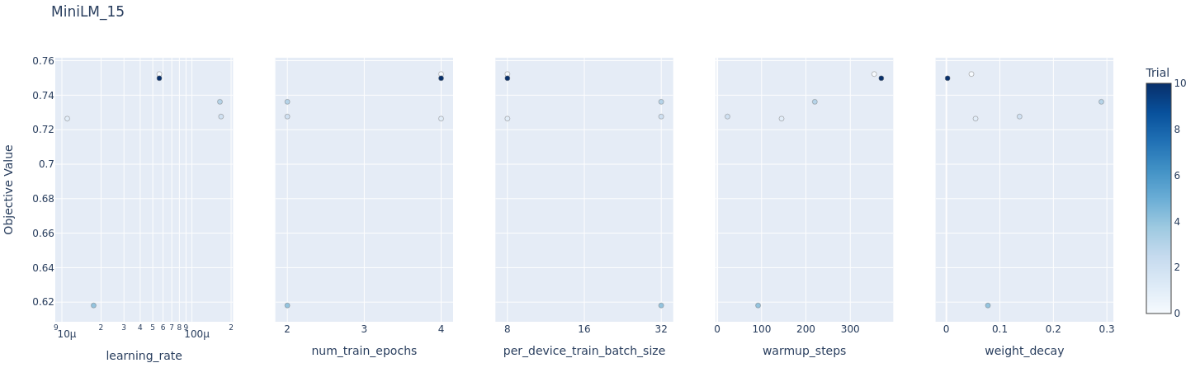


Figure A.3: Slice plot for MiniLM (15-SSDD)

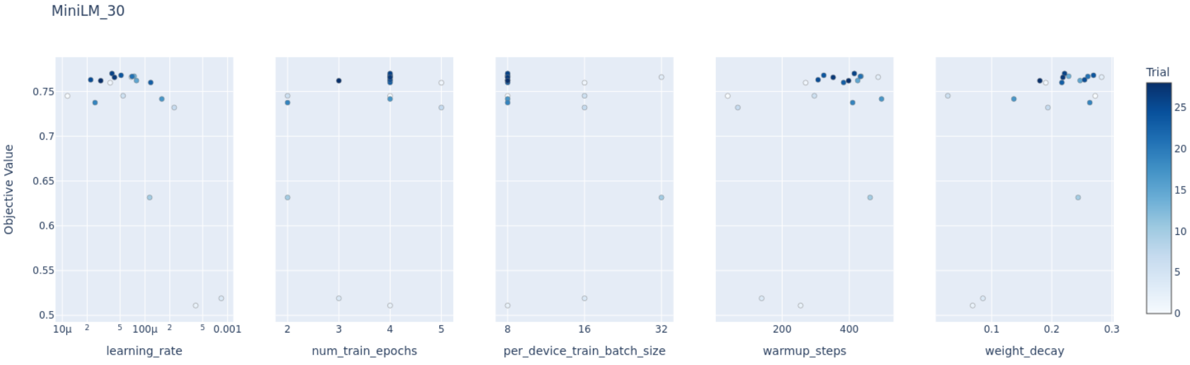


Figure A.4: Slice plot for MiniLM (30-SSDD)

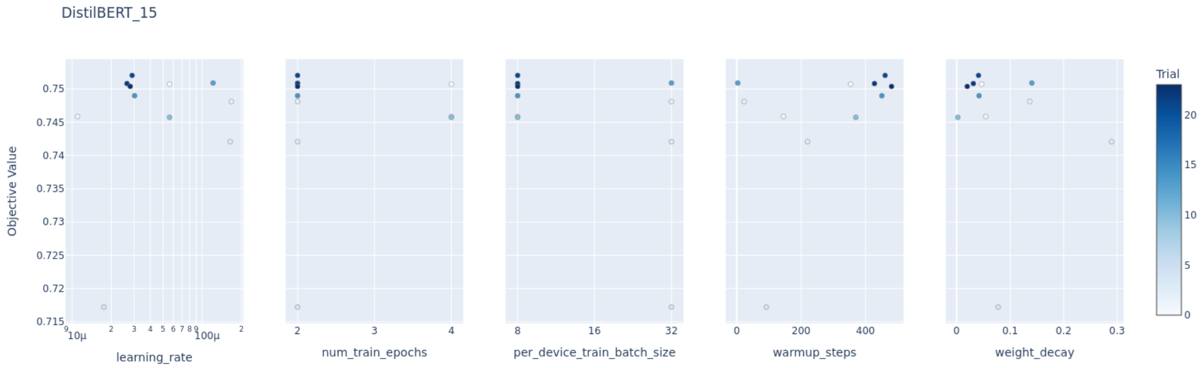


Figure A.5: Slice plot for DistilBERT (15-SSDD)

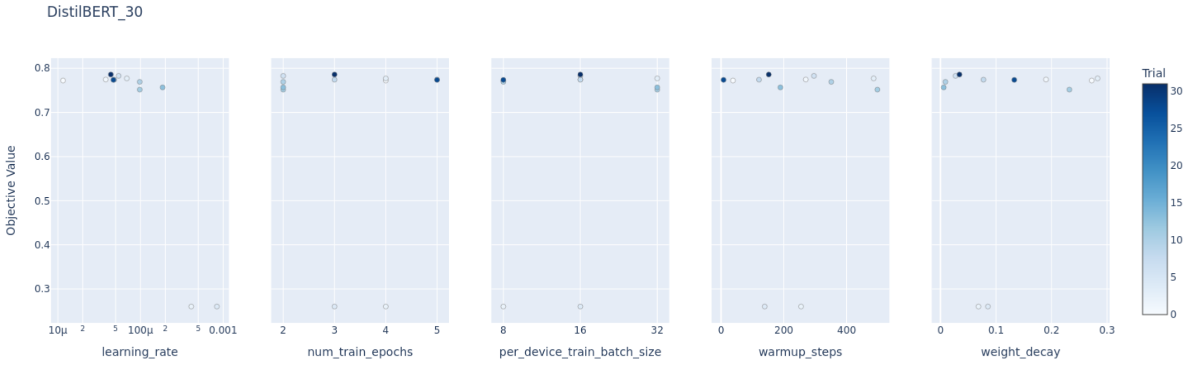


Figure A.6: Slice plot for DistilBERT (30-SSDD)

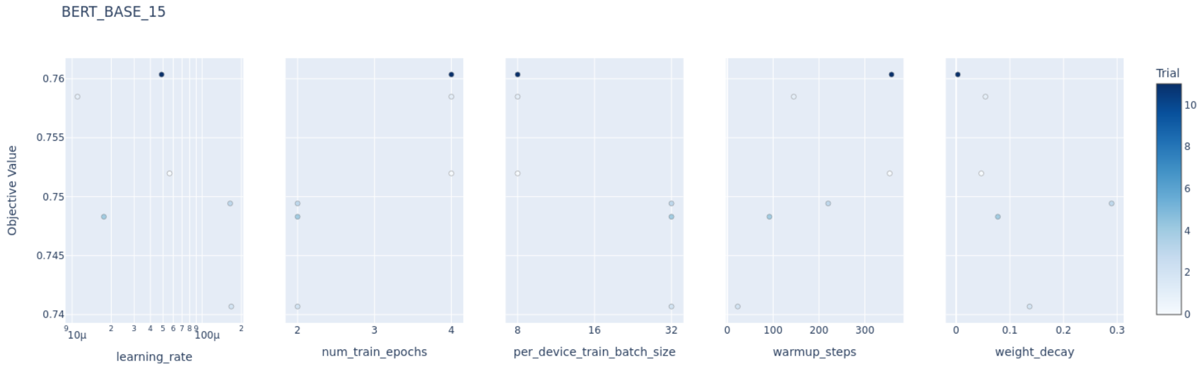


Figure A.7: Slice plot for BERT BASE (15-SSDD)

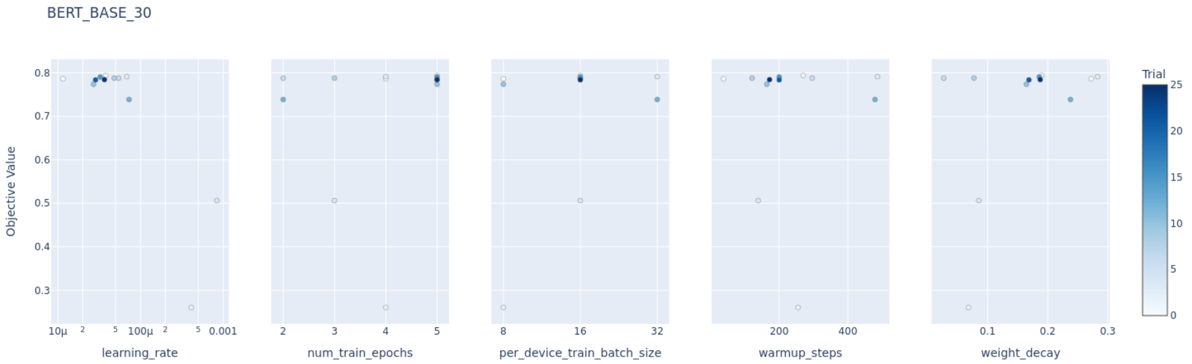


Figure A.8: Slice plot for BERT BASE (30-SSDD)

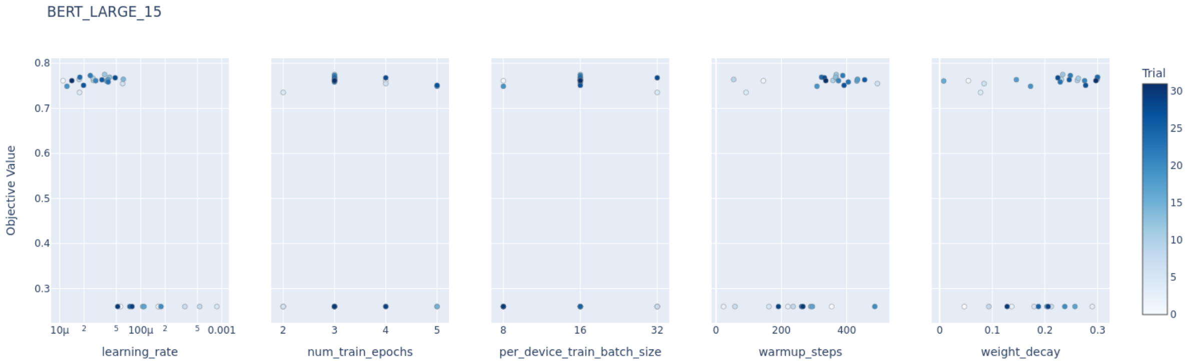


Figure A.9: Slice plot for BERT LARGE (15-SSDD)

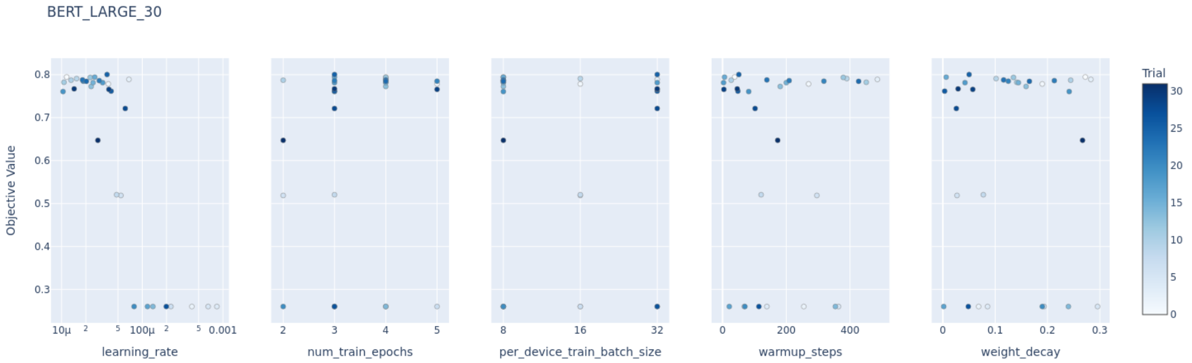
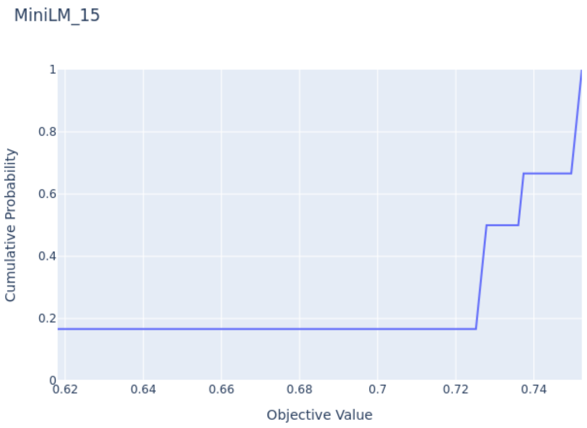


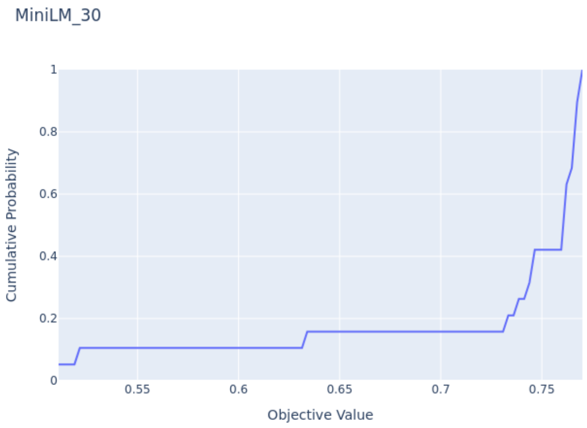
Figure A.10: Slice plot for BERT LARGE (30-SSDD)

A.4 EDF Plots

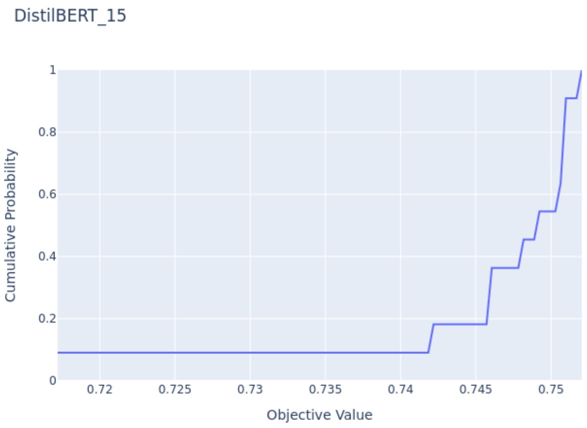
This section explores Empirical Distribution Function (EDF) plots, which reveal the distribution of objective function values achieved during the Optuna search. EDF plots are valuable for understanding how likely different hyperparameter configurations are to yield high performance. These insights aid in identifying optimal hyperparameters and assessing model sensitivity to parameter variations.



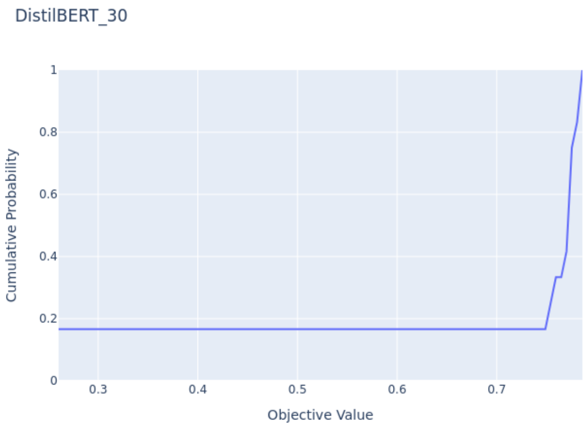
a) MiniLM on 15-SSDD.



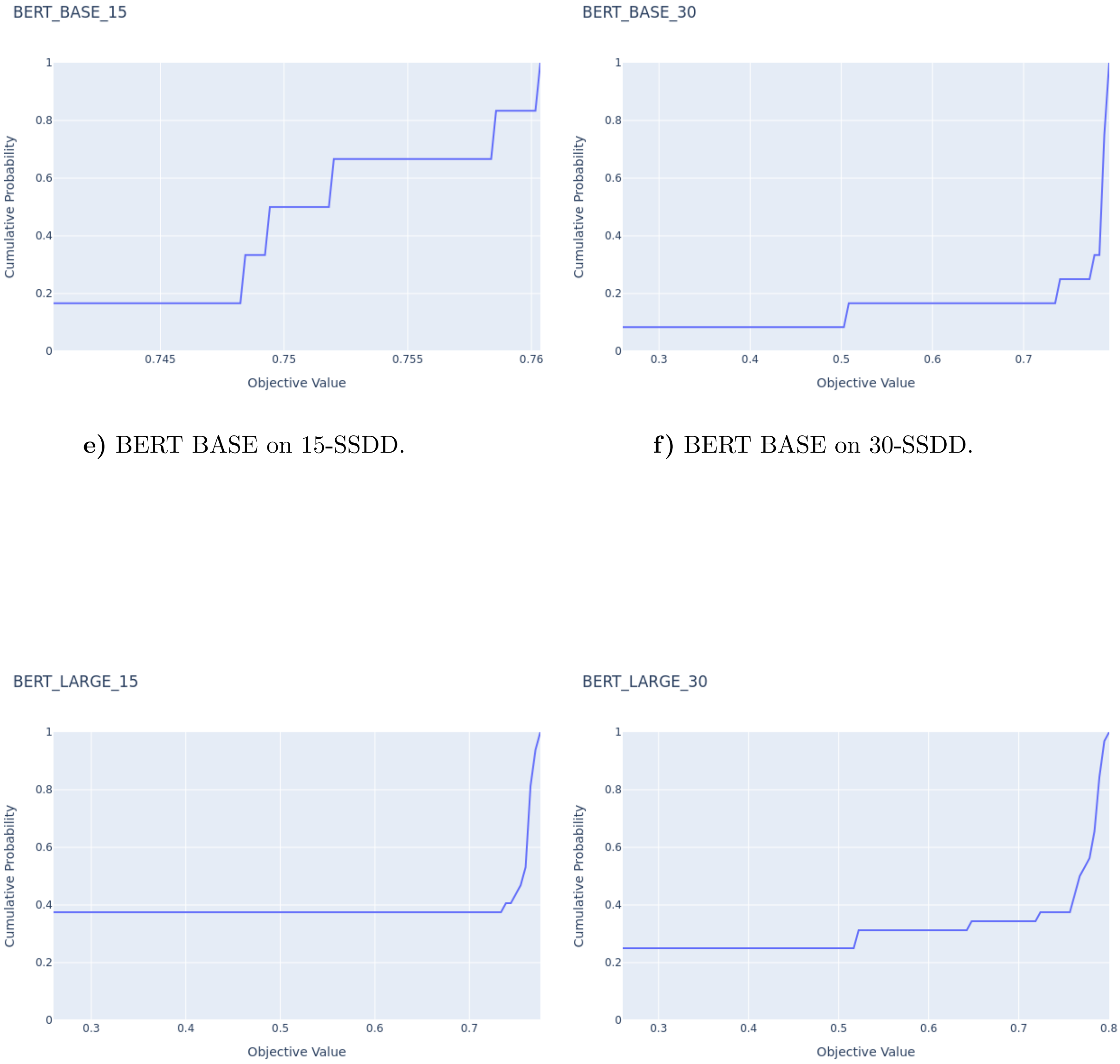
b) MiniLM on 30-SSDD.



c) DistilBERT on 15-SSDD.



d) DistilBERT on 30-SSDD.



e) BERT BASE on 15-SSDD.

f) BERT BASE on 30-SSDD.

g) BERT LARGE on 15-SSDD.

h) BERT LARGE on 30-SSDD.

Figure A.11: Comparative EDF plots considering the two datasets and the LLM models.

Norges miljø- og biovitenskapelige universitet
Noregs miljø- og biovitskapelege universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway