EVENT CLASSIFICATION FROM TEXT BASED COMMENTARIES FOR SPORTS ANALYTICS

Kuruppuge Piumi Kanchana Nanayakkara

(158227L)

Thesis/Dissertation submitted in partial fulfillment of the requirements for the degree
Master of Science

Department of Computer Science and Engineering

University of Moratuwa
Sri Lanka

February 2019

DECLARATION

I declare that this is my own work and this thesis/dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant the University of Moratuwa the non-exclusive right to reproduce and distribute my thesis/dissertation, in whole or in part in print, electronic or any other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature:..Nanayakkara........ **Date:** ...27/05/2019.

Name: K.P.K. Nanayakkara

The above candidate has carried out research for the Masters/MPhil/Ph.D. thesis/dissertation under my supervision.

Signature of the supervisor:.......................... **Date:** ...27/05/19......

Name: Dr. Amal Shehan Perera

Abstract

Despite being relatively a new field, which is still evolving, sports analytics have already proven to provide a competitive advantage to sports teams. Especially with large number of commercial leagues taking place across the globe for every sport, the insights generated through analytics has helped franchise owners in bidding auctions for selecting players, where every decision the owners make, will cost them billions of money. However, most of the research work done in sports analytics such as score predictions, player profile analysis has been involved with mainly number crunching, with score boards being the predominant information source of such analytical work.

Over time research had been focused on exploring other sources of information such as video and audio analysis, social media text analysis, micro blogging analysis etc. But there still exists large amount of untapped data with potential of being used for sports analytics. One such data source is online sports commentaries where the web sites give live updates on games, in the form of a temporal series such as minute by minute commentary for soccer or ball by ball commentary for Cricket. These publicly available commentaries give not only the scores, but coverage of all events happening during the match, including the ones which do not go into the scoreboards such as "dropping a catch" in a cricket match. If this information can be captured through event extraction and stored as structured data, that could be useful for analytical purposes.

In this research an end to end system is proposed to produce structured data from online sports commentaries, while extracting information from the commentary text during the process.

Keywords: Sports Analytics, Online Sports Commentaries, Text Mining, Event detection, Multi-label Classification

Acknowledgements

I would like to express my profound gratitude to my supervisor, Dr. Amal Shehan Perera, for introducing this research idea to me and for his invaluable support, encouragement, supervision and useful suggestions throughout this research work. His continuous guidance enabled me to complete my work successfully.

I am as ever, indebted to my family and friends for their continued support throughout my work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
BR	Binary Relevance
BOW	Bag of Words
BI	Business Intelligence
CC	Classifier Chains
ECC	Ensembles of Classifier Chains
EPS	Ensembles of Pruned Sets
GNB	Gaussian Naive Bayes
IE	Information Extraction
LP	Label Power Set
LBW	Leg Before Wicket
ML	Machine Learning
NB	Naive Bayes
NER	Named Entity Recognition
ODI	One Day International
POS	Part of Speech
RNN	Recurrent Neural Network
RE	Regular Expressions
SVC	Support Vector Classifier
SVM	Support Vector Machine
TF-IDF	Term Frequency – Inverse Document Frequency

CHAPTER 1: INTRODUCTION

1.1 Background

The objective of a team (with one or more players) playing a game in any sport is to win. Success of a team in winning a game will depend not only on individual player performances and recent form, but also on key decisions taken both on and off the field. On field decisions could include a player substitution in Soccer, deciding batting lineup in Cricket etc., whereas off fields decisions include deciding optimal team composition for a game and formulating game strategies considering weather, pitch/ground, strengths and weaknesses of own player(s) as well as opponent(s).

In early days these decisions were taken based on expertise and game awareness of human personnel involved (coach, captain team manager etc.), thus, always had the possibility of being subjective. Breaking this ground Sports Analytics comes in to the table, emphasizing the need of providing a basis for each decision taken, mainly through historical stat so the subjectivity of the decisions could be eliminated to some extent. It also presents teams with an opportunity to explore minor details of the game which could be overseen humans as well as dive in to complex calculations which could reveal vital information.

Traditional Sports Analytics is focused mainly on number crunching in terms of analyzing play/team stats, predicting scores etc. Main source of data had been score boards and match reports where information was derived by applying statistical models on them. Also watching and analyzing video footages of past games is a common practice in doing an analysis.

Today, with increasing competitiveness of professional sports, combined with their unpredictable nature, the role played by Sports Analytics in making a team win is no longer trivial. In addition, with the number of commercial leagues and bidding auctions taking place across the world with huge price tags attached to players, the margin of error in decision making need to be kept as low as possible. This increasing demand has moved the focus of analytics beyond traditional score card based number crunching to derive information from other sources.

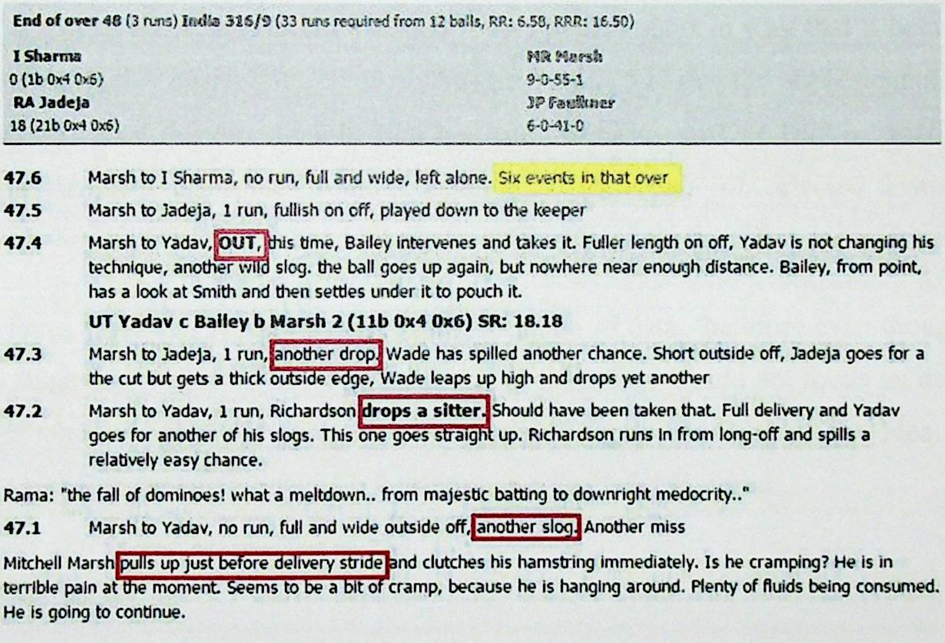
With increasing volume of unstructured and semi structured data in the form of emails, web pages, blog posts and social media content, most organizations fail to grab the maximum available information at their hands using traditional Business Intelligence (BI) tools. No longer could traditional BI and limited text analysis practices by applying keyword search or pattern recognition alone provide the insights quickly and accurately in fast faced business environment, thus the need for Artificial Intelligence (AI) and Machine Learning (ML) driven analytics increased.

Similar to how business organizations could use AI and ML to improve their service offerings and business processes, sports leagues could make use of those insights for better sports trading, targeted training and player assessments. Only the teams which are, flexible in adopting this technologies, make use of all available information and incorporate the insights into their decision making process would be able to surface to the top of the game.

1.2 Problem Statement

Even though the focus of sports analysis is now moving beyond traditional number crunching and manual analysis of video records, towards other sources such as multimedia analysis [1] and social media text analysis [2], [3], there is still vast amount of information relating to sports domain untapped for analytical purposes. One such data source is online sports commentaries where the web sites give live updates on games in the form of a temporal series: e.g. minute-by-minute commentaries for soccer, ball-by-ball commentaries for baseball or cricket and move-by-move commentaries for chess. These commentaries are easily accessible and can be considered a publicly available large text source. Very thin amount of research has been done based on these commentaries and their potential of contributing to sports analytics is yet to be explored.

The sites like Planet Rugby[1] and Espn Cricinfo[2] is viewed live by millions of sport fans all over the world to track the match scores. These commentaries however give not only the scores, but a complete analysis on what is happening out there in the play field in every single moment during the match. It covers all the events of the match including the incidents that do not go in to score board as shown in Figure 1, thus not used in performing analytics later. In many sports, a player is defined by figures and statistics but there could be many events which define the character of a player, but these events do not go in to those calculations. For an example in Soccer, the number of goals a player scores is highly spoken of but, role played by a player in creating a goal or forming a strong defense go unnoticed.



End of over 48 (3 runs) **India 316/9** (33 runs required from 12 balls, RR: 6.58, RRR: 16.50)

I Sharma	**MR Marsh**
0 (1b 0x4 0x6)	9-0-55-1
RA Jadeja	**JP Faulkner**
18 (21b 0x4 0x6)	6-0-41-0

47.6 Marsh to I Sharma, no run, full and wide, left alone. Six events in that over

47.5 Marsh to Jadeja, 1 run, fullish on off, played down to the keeper

47.4 Marsh to Yadav, **OUT,** this time, Bailey intervenes and takes it. Fuller length on off, Yadav is not changing his technique, another wild slog. the ball goes up again, but nowhere near enough distance. Bailey, from point, has a look at Smith and then settles under it to pouch it.

UT Yadav c Bailey b Marsh 2 (11b 0x4 0x6) SR: 18.18

47.3 Marsh to Jadeja, 1 run, another drop. Wade has spilled another chance. Short outside off, Jadeja goes for a the cut but gets a thick outside edge, Wade leaps up high and drops yet another

47.2 Marsh to Yadav, 1 run, Richardson **drops a sitter.** Should have been taken that. Full delivery and Yadav goes for another of his slogs. This one goes straight up. Richardson runs in from long-off and spills a relatively easy chance.

Rama: "the fall of dominoes! what a meltdown.. from majestic batting to downright medocrity.."

47.1 Marsh to Yadav, no run, full and wide outside off, another slog. Another miss

Mitchell Marsh pulls up just before delivery stride and clutches his hamstring immediately. Is he cramping? He is in terrible pain at the moment. Seems to be a bit of cramp, because he is hanging around. Plenty of fluids being consumed. He is going to continue.

Figure 1 Events mentioned in commentary for Cricket

Source: http://www.espncricinfo.com/

[1] http://www.planetrugby.com/

[2] http://www.espncricinfo.com/

The challenge with this kind of information is that though large, they are unstructured or semi structured in nature when compared to match reports and score cards. However if this data can be transformed to a structured format covering all the events happened in the field for a particular match, where it can be easily queried extract information from, it could be proven really useful for the sports analysts.

1.3 Research Objective

The Objective of this research is to come up with a mechanism to convert online sports commentaries in to a structured format such that, querying and deriving information could be made easier than searching free text, while capturing maximum amount of information contained in the text commentaries to the structured schema.

1.4 Scope

The approach taken is to build a model for a selected sport in way that it could be extended to other sports with minimal changes. In this research the implementation is done on Cricket domain using ball by ball commentaries for One Day International (ODI) matches. Database would contain structured data of selected historical matches and at this stage this would not be implemented for real time commentaries.

The focus would be on step by step transformation of data, therefore even though at final stage data would be saved in a structured format we would not focus on details of the model to ensure it cover all the aspects of a game of Cricket. The idea is to make sure maximum amount of information is collected and stored.

1.5 Challenges

Extraction of information from commentaries is not trivial. With these being written by a human being in real time the descriptions of events are not long and not descriptive as in a match report. Also the same commentary could be interpreted in more than one way even by a human being. A machine learning algorithm can bridge the semantic gap between machines and humans, but not the ambiguity present in the language. Therefore main challenge faced in this work was absence of ground truth.

Though it can be seen on matches that there are dedicated personnel to take down notes throughout the game time of a match, as per our knowledge there is no publicly available data set recording actual events happened at each instance of a game e.g. at every minute in a game of Soccer or at each delivery in a game of Cricket. Therefore in this research, in order to validate the accuracy of the information extraction process, it was required to build a tagged data set first. This is a very time consuming task which needs to be done manually, by someone who possess the knowledge of the game. Still, there is a chance of some subjectivity being associated with the tags due to ambiguity present in some commentaries.

Further with very few existing literature based on sports commentary, comparing and benchmarking results was another challenge faced. In existing literature which uses similar text mining and event extraction mechanisms as in this work, the results are analyzed by comparing different algorithms used [4], [5], but not with another published results set. There are no classification results published for Cricket domain as per our knowledge, thus in Chapter 4 an attempt was made to compare results with research done outside Cricket, using the same event extraction technique as ours. However there could be domain specific challenges affecting this comparison.

1.6 Overview of rest of the Chapters

Rest of the Chapters of this thesis is organized as follows. Chapter 2 contains a Literature Survey on related work both in sports and text analytics in general. Chapter 3 describes the proposed solution and implementation details while Chapter 4 publishes the results in comparison to related work. Finally Chapter 5 concludes the work done on this research while discussing the possible further improvements.

CHAPTER 2: LITERATURE REVIEW

As discussed in previous chapter, while there is a rise in sports analytics than ever in last few years, there are many areas yet to be explored in this domain. In addition, with cricket being played only in limited number of countries as compared to other sports such as soccer or rugby, portion of work done on sports analytics for Cricket is very low. Therefore, in this literature examples are drawn from other domains where work is thin in sports analytics.

In the Cricket domain, most of the research work been done to date, are based on numerical analysis. When compared with a sport like Soccer where the only major numeric score would be the number of goals scored, which can be even zero for a given match, Cricket, being a game where many numbers are involved in terms of no of runs scored, wickets taken and many derived attributes such as player averages, economy rates, new run rates etc., there are countless number of possibilities of crunching these numbers and derive new insights with complex mathematical models such as D&L method [6] used when resetting targets in interrupted limited over matches (mainly due to weather). Numerical work done on this game runs way back even to year 1945 [7] and in this book chapter [8] author has done extensive analysis of all cricket related research work based on statistics. He has identified no. of applications such as player assessment, evaluating team strengths, finding optimal line up and deducing game plans where numerical analysis can contribute to a teams' success.

In contrast in this research the focus is on deriving information from other approaches than number crunching and making use of barely touched online sports commentaries for analytical purposes. Therefore, in the literature review also the focus is set on text analytics on sports and event extraction mechanisms used to identify events related to a certain domain from text.

This Chapter will be divided into few sections as Section 2.1 Overview of text mining in sports analytics, Section 2.2 Existing work on sports commentary and Section 2.3 Event detection from text, where first an overview of event detection approaches that has been tested is discussed, and then focus is narrowed down on to work done on using Multi Label Classification which we use in this research. Section 2.4 discusses briefly on Attribution of events to actors and finally Section 2.5 provides a summary of all literature analyzed.

2.1 Overview of Text Mining in Sports Analytics

Sports being an industry with great popularity and large financial deals are involved; the importance of analytics in the domain has been highlighted with time. The traditional number crunching has been in place over decades, but now the sports professionals seek much more information than scoreboard and statistical analysis. There are many sources available including multimedia and similar amount of information can be gathered from text analysis as well. There are number of text sources that could be found such as news reports, articles, match reports, live online commentaries etc. Also, there had been some analytical work carried out for interviews[3] and video/webcast text [9], [1] some of which we discuss in later sections.

The authors of this book chapter [10] discuss about sports data mining in general with an initial flavor of what are available sources of sports data mining and some possible general applications that could be built upon them. They start with how data came in to current stage from being locked away in the books of scoreboard keepers. According to them there are official web pages of sport leagues which were created initially and then had been a breakthrough of the amount of sports data flooded in the form of third-party sources which give easy access to data and also has aggregated raw figures and represented in visually appealing manner.

[3] http://harvardsportsanalysis.org/2015/03/sports-interviews-lets-talk-about-who-talks-the-most/

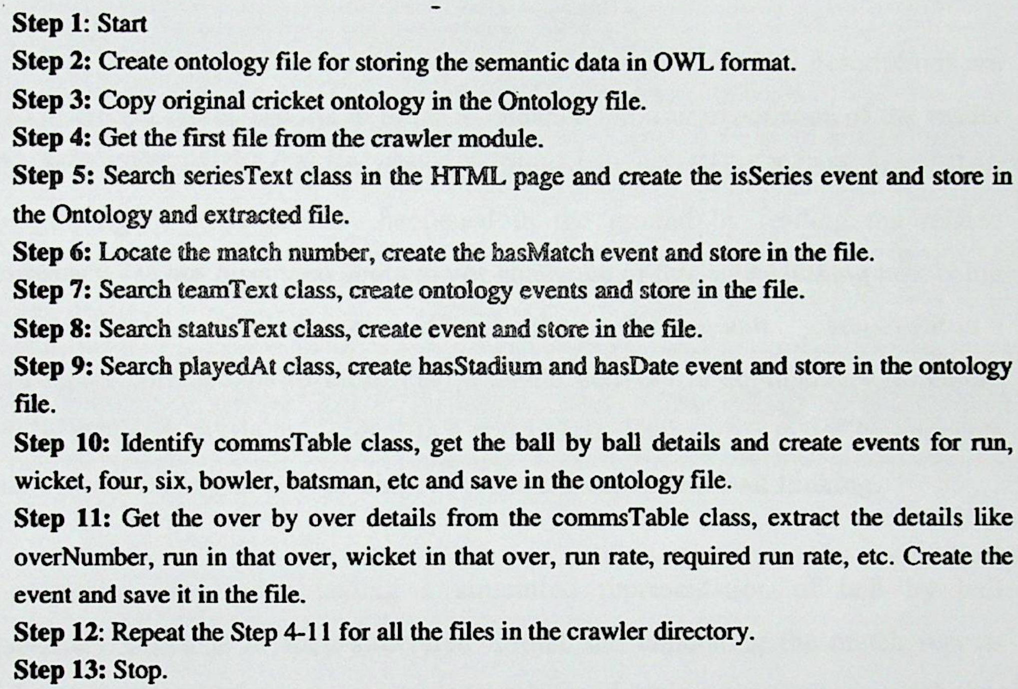
Next, they describe the available web sources of such data. They have analyzed few such web pages for various sports which provides information like strong & weak shots of a batsmen and speed & movement of a pitcher in baseball, statistics at a glance for basketball, real time update in terms of ball by ball commentary for cricket, Game Center which is an interactive graphical description of the game & predictive analysis such as game excitement rating, comeback expectancy for football and even there are pages where abstracts video footage of game events are incorporated into a console game-like environment for soccer.

Next the authors go on to describe the difficulty of using a general program to retrieve data due to lack of homogeneity across websites where each site is having a different coding mechanism and schemas which are expected to change from time to time. However there has been some such applications such as crowd control data applications for security reasons where previous complaints could be analyzed when placing security personnel in future matches and resolving the problem of fast-moving games where some frames are blurred due to the speed by applying transformation matrices to each frame and then calculating movement between them.

2.2 Existing work on sports commentary

Every sport can be viewed as sequence of events and lately there are many platforms reporting these temporal events in real time for many sports e.g. minute by minute commentary for soccer, ball by ball commentary for cricket and move by move commentary for chess. Even though these commentaries can be considered as rich sources of information containing entire sequence of events in a game and captures incidents that does not go in to traditional score cards or reports, utilization of these resources has been relatively low in sports analytics. In this section few such work is analyzed both in cricket and soccer domain.

In this thesis [11] authors describe a system which obtained cricket commentary from web and store the retrieved information in a semantic coherent schema by means of an ontology. Their focus is on doing semantic search using those systems. They have developed an automated crawling module using html parsing and information extraction from crawled pages are done in a template-based method where each html tag is mapped to a xml node. Then they design an ontology for cricket domain and create events for each tagged event in online commentary and save in ontology in OWL format. They perform SPARQL search with semantic indexing on this ontology and results are published in comparison with key word search and traditional free text indexing. They have claimed to gain very high precision values for SPARQL search when comparted with traditional free text search.



Step 1: Start

Step 2: Create ontology file for storing the semantic data in OWL format.

Step 3: Copy original cricket ontology in the Ontology file.

Step 4: Get the first file from the crawler module.

Step 5: Search seriesText class in the HTML page and create the isSeries event and store in the Ontology and extracted file.

Step 6: Locate the match number, create the hasMatch event and store in the file.

Step 7: Search teamText class, create ontology events and store in the file.

Step 8: Search statusText class, create event and store in the file.

Step 9: Search playedAt class, create hasStadium and hasDate event and store in the ontology file.

Step 10: Identify commsTable class, get the ball by ball details and create events for run, wicket, four, six, bowler, batsman, etc and save in the ontology file.

Step 11: Get the over by over details from the commsTable class, extract the details like overNumber, run in that over, wicket in that over, run rate, required run rate, etc. Create the event and save it in the file.

Step 12: Repeat the Step 4-11 for all the files in the crawler directory.

Step 13: Stop.

Figure 2 Algorithm used in [11] for information extraction

Even though this research make use of ball by ball commentaries and transfer them to a structured format, the commentary text describing each delivery goes untouched as only events identified and modeled in the ontology are the ones that comes pre-tagged in web site commentary.

Same authors in this paper [12] present an extension of the previous research by storing ontology in a RDBMS system. The main contribution of this work is that they are trying to make is to come up with a technique to store ontology in a relational database while all the data being stored, so that system can utilize the advantages of relational databases for the cricket domain. The problem in the existing technologies as they have stated is that, "they either deal with single ontology or they do not store complete semantics expressed in OWL ontologies". There are no measurements or figures provided on how successful they have been in achieving the advantages of RDBMS system and amount of information successfully stored in, which has been stated as one of the key objectives of their research.

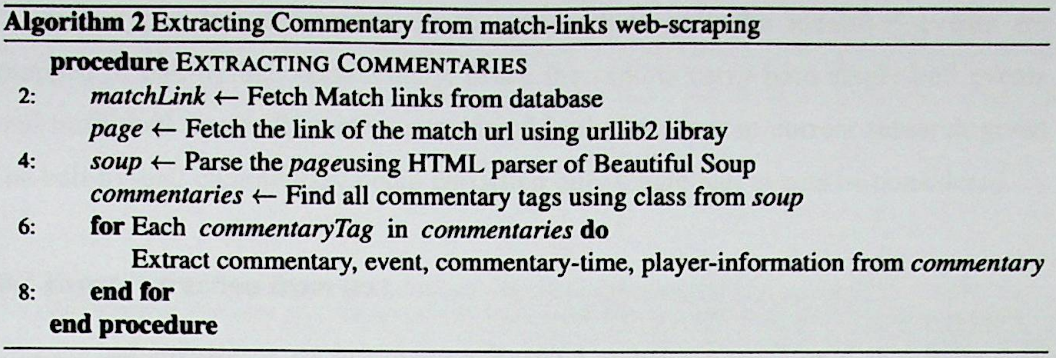
The author of this paper [13] introduces a system where ball by ball descriptions are linked to cricket match reports in order to enhance the user experience of the reader of the report where for a given event mentioned in the report he can get a clear understanding of what actually happened in the ground by reading the related commentary. He has discussed about major challenge of this entity linking task being reporters using versatile language to describe the same thing and a given event in a match report corresponds to more than a single delivery in commentary, therefore range of instances needs to be identified and linked. There are 3 major components of this system namely Pre-processing, Mention Detection and Ball Linking.

Pre-processing involves obtaining a structured representation of ball by ball commentary which is in semi-structured format and annotating the match reports which are in completely unstructured format by performing co-reference resolution at a paragraph level, parse tree construction, and dependency analysis. In mention detection primary task is to classify whether it's linked to a single ball or multiple balls. For this classification has been done based on a feature set comprised of a cricket term dictionary he developed containing shot words, bowl type words etc., entities such as player names, team names, dates and locations and few other features such as similarity between two entities in terms of common word count. He has claimed that best result for accuracy of this initial classification task is achieved when boosted decision trees are used and therefore the same algorithm is used for subclass classification described next. After classifying to single or multi class as next step each identified mention is divided to sub level such as ‘OUT’, ‘SIX’, ‘DROP’ etc. For this sub class classification same type of features used with dictionary being enhanced with words relating to these sub types.

In linking phase, first he has considered temporal clues like: “in the 42nd over”, “during first 10 overs” etc. and linked the text accordingly. In absence of such clues the candidate entities are chosen based on Jaccard similarity between the candidate and the mention with respect to a cricket term dictionary they have created. In order to maintain consistency, the entities for every mention are re-ranked based on the sequential proximity of the linked set of balls. Further if a mention is linked to many number of balls a summary is formed by picking eventful balls while giving the user the option to view all balls if he needs.

This work [5] aims to exploit the potential of soccer commentaries as a rich data resource together with other text sources such as voice to text conversion, newspaper columns, internet blogs for player analytics by identifying key events and using these events to derive player attributes. They speak about challenge of attributing an identified event to a certain player which will be discussed in Section 2.4 in detail.

As it can be seen in Figure 3 they have also used html parsing in web scraping to extract commentaries from match URLs. They have extracted the commentary together with pre-labeled set of events from one website and used another site to obtain player and respective team information.



Algorithm 2 Extracting Commentary from match-links web-scraping

```
procedure EXTRACTING COMMENTARIES
    matchLink ← Fetch Match links from database
    page ← Fetch the link of the match url using urllib2 libray
    soup ← Parse the pageusing HTML parser of Beautiful Soup
    commentaries ← Find all commentary tags using class from soup
    for Each commentaryTag in commentaries do
        Extract commentary, event, commentary-time, player-information from commentary
    end for
end procedure
```

Figure 3 Algorithm used for web scraping using html parser in [5]

They define another set of events in addition to the pre-tagged events extracted and used a crowd source approach to tag them. Next as for preprocessing entity names (identified using NER) and stop words has been removed from text. In feature engineering Count Vectorizer has been used in bag of words model and classification has been done both as a Single label multi – class classification task and a Multi label classification task. They have used 3 variations of Naive Bayes (GaussianNB, BernoulliNB and MultinomialNB) and Logistic Regression classifiers for Single label task where all except GaussianNB has given good results. Approach taken for multi –label classification will be discussed in Section 2.3.2 Multi label classification for event extraction.

In summarizing the discussed literature based on online sports commentaries, a group of researches has worked on storing sports commentaries in a structured format like a database [12] or ontology [11]. In some other work some processing is done on commentaries in order to extract information before using them either to match with some other source [13] or to save for future analytic purposes [5].

Though the primary focus of this paper [13] was on entity linking it can be considered one of the closest works that has been done in line with the initial part of current research in analyzing ball-by-ball commentary and identifying event. It However, in this work it is not the ball-by-ball commentary that is being classified in to event types, but the reports on the games then only the identified events are mapped to ball-by-ball commentary. Since the reports carry both single ball events and multi-ball events they have considered both, however in current research given the ball-by-ball commentary being classified only single ball events be considered.

2.3 Event extraction from text

During last decade or so Information Extraction has emerged as a major field of computer science and natural language processing, with the exponential growth rate of web data in terms of social networks, blogs etc. However, with this rate of growth no longer the human analysts are able to keep on track with the events happening around the globe. Thus, the requirement of Information Extraction (IE) systems which detects the Events from free text aroused. However, there could be more than one way an event is represented in such systems. In this research [14] the authors have identified two such approaches; one being the TimeML model where event is considered as "a word that points to a node in a network of temporal relations" and other being ACE model in which an event is considered as a "complex structure, relating arguments that are themselves complex structures, but with only ancillary temporal information.".

In last few years extensive research work has taken place in terms of event extraction, however it still considered as one of the challenging tasks as it could be domain specific most of the times and also there are always more than way to linguistically represent an event in text. With development of machine learning techniques research work on this area also has been improved from pure knowledge-based hand coded systems to semi-automated systems. Later with Semantic Web being introduced not only isolated pieces of information, but events, entities and relationships have been of major interest of the researchers.

In domain specific applications where the user already knows the possible semantics of the surrounding text, that information could be incorporated therefore relatively small amount of training data would be required. However, in applications which are supposed to be domain independent such as event extraction from news feeds the news item and the surrounding text can belong to literally any domain, therefore it is not possible to incorporate domain knowledge in to the system. In such scenarios a large amount of training data should be analyzed to identify specific patterns/texts surrounding an event with a certain confidence along with the use of lexical grammars.

According to [15] "A lexicon-grammar is a dictionary that provides exhaustive and detailed sub categorization information about the predicates of a natural language such as verbs, predicative nouns and adjectives. Predicates with related syntactic and semantic behavior are grouped together. The lexical, syntactic and semantic features provided by the lexicon-grammars are used for establishing grammatical and dependency rules."

This paper [16] contains a literature survey of text mining techniques that can be used for event extraction. Further it also contains general guidelines on how to choose a particular event extraction technique depending on the user, the available content, and the scenario of use. It has categorized such techniques in to three types as below and for each of them the authors have cited reasonable amount of literature for different domains.

- **Data Driven Event Extraction** – based on quantitative theories such as statistics, probabilistic modeling, Information Theory and linear algebra.
- **Knowledge Driven Event Extraction** – based on linguistic patterns defined or identified using lexical and prior domain knowledge. The patterns can be either lexico-syntactic or lexico-semantic.
- **Hybrid Event Extraction** – combination above two.

2.3.1 Approaches used for event extraction in general

In this section various research work which involves extracting events from text and the approaches taken in those are discussed without any restriction on domain or algorithm used.

The system introduced here [15] is aimed at building an Information Extraction system which has the capacity to handle large no of events and relations with high accuracy and minimum amount of porting effort. It comprises of three specialized pattern-based tagging modules, a high-precision co-reference resolution module, and a configurable template generation module. In the tagging module three components have been defined; NameTagger, NPTagger and EventTagger respectively (where output of one tagger becomes the input of the next), Each of these taggers module relies on the same pattern-based extraction engine, but different sets of patterns.

The EventTagger identifies the events by applying lexicon-driven, syntactically-based generic patterns, where the events are tagged in presence of at least one of the arguments specified in the lexical entry for a predicate. The output xml of the EventTagger is next sent through the co-reference module which resolves definite noun phrases and singular person pronouns. Finally, the template generation module uses declarative rules for merging and generating which as a unique features when compared with most hard coded systems.

The highlight of this system is that it uses a declarative, lexicon driven approach where they use a verb as the event-denoting word. This makes it easily portable as new types can be extracted by simply adding new verb entries to lexicon. Their evaluation results have shown that system it achieves 70% or higher F-Measure for 26 types of events. One drawback of this system is that it is focused only at verb-based events where adding noun-based event extraction capability would be critical in improving performance.

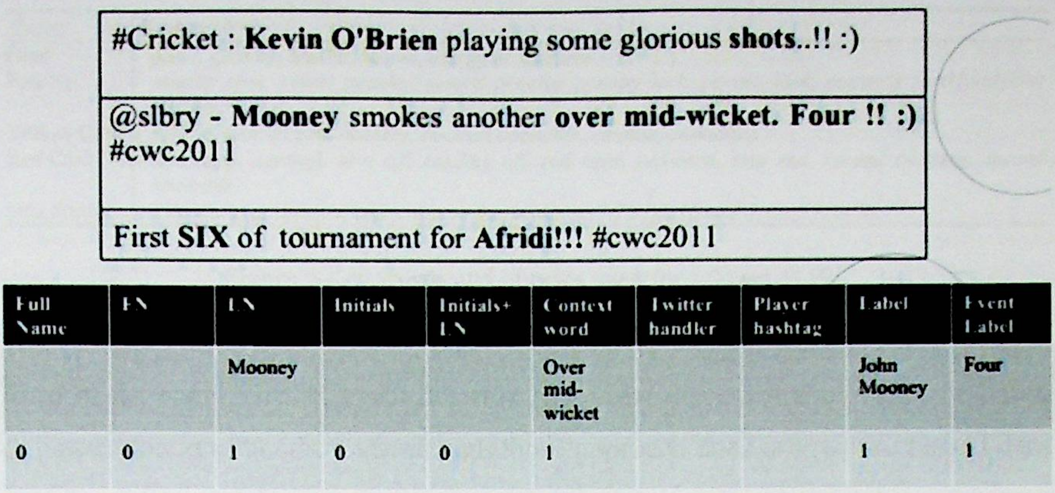
In this paper [17], the authors introduce an event extraction mechanism which using a classifier based on lexico-syntactic features, where they try to "extract any kind of events that have a structure that can answer the question: Who did what, when and where?" This is an end to end system called 'EEQuest' where events are extracted from news articles using above technique and store them in a graph database to create a query system on top of it.

When answering the questions of 'Who did what, when and where', they primarily rely on NER and use the labels with named entities assigned to each token of text to answer the questions. They start with identifying 'Who' assuming that "who or the subject of the article occurs frequently in the whole article and is also a named entity". After applying NER they use few features to classify whether a given entity can be considered as a 'who' in the article or not. Since this research is done for news articles 'who' can be either a person, organization or even a location. Therefore, entity type tagged by NER is taken as a feature. In addition, they have calculated number of occurrences in text, number of occurrences in title and mean position ("it is assumed that more important entities occur at the beginning of the article while the less important ones may feature somewhere later") of a token as other features. Then they use GNB (Gaussian Naive Bayes) classifier to identify the most likely entity to be 'who' or the subject of the article.

In identifying 'what' or the actual event, they start by looking at the 'who' candidates present in the title first occurrence of the highest ranked who in the article text and their subsequent verb phrases. It is stated that "we created a parse tree for each sentence and then matched the 'who' candidates in the parse tree. Since the 'who' candidates are always in the noun phrase part, it was easy to find the subsequent verb phrase". If such a pair exists they consider the verb phrase as event. 'When' and 'Where' questions are also answered in similar manner by focusing on entities tagged as 'DATE' and 'LOCATION' respectively in NER. These identified events and other information then stored in a graph database attributed to manually created network of companies and their associates/competitors. They have provided three pre-defined queries (request events related to 1. a particular organization 2. the competitors and associates of a particular organization 3. organizations working in a particular sector) that can be used to extract data from this storage.

The drawback of this approach for event extraction is that there can be events which does not have a 'who' or a subject and those won't be identified and also similar to [15] they focus on verb-based events only. Further if there are two articles describing the same event which could be a common scenario when retrieving news articles from web, entries will be duplicated and adversely affect the scaling of the system.

Large volume of sports related content could be found on micro blogging site Twitter lately. There has been research work going on in an attempt to tap this information for analytic purposes. However, there are few challenges specific to this domain with the character limit of certain no. of character for a tweet, users to use short forms to convey their message ignoring language rules. Also, these are high use of slang words and abbreviations that should be dealt with. In this paper [3] authors attempt to detect player mentions and event mentions in tweets for cricket domain by defining separate feature sets based on linguistic features, background knowledge and twitter specific features. For an example for the feature set of a player they use background knowledge of players in terms of full name, initials, nick names, player twitter handle etc., and context feature indicating whether the game-related key terms (relating to a wicket, four or six) appear within a window of four words of player name. In Figure 4 below context feature is highlighted in bold for three tweets and in the table in the bottom the feature set is indicated for the middle tweet out of three.



#Cricket : **Kevin O'Brien** playing some glorious **shots**..!! :)

@slbry - **Mooney** smokes another **over mid-wicket. Four** !! :) #cwc2011

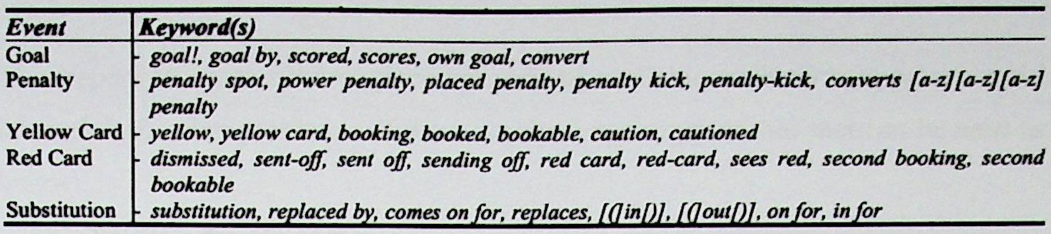
First **SIX** of tournament for **Afridi**!!! #cwc2011

Full Name	FN	LN	Initials	Initials+ LN	Context word	Twitter handler	Player hashtag	Label	Event Label
		Mooney			Over mid-wicket			John Mooney	Four
0	0	1	0	0	1			1	1

Figure 4 Feature vector for a tweet in [3]

For event extraction in addition to key terms, features relating to twitter activity volume in given duration i.e. twitter volume and information diffusion level calculated through no of re-tweets (assuming "users will be busy sharing and communicating the event through their own tweets rather than reading and forwarding others"). This paper [2] also deals with the same task of identifying events from tweets for four other sports (soccer, basketball, tennis and baseball) but they first use volume burst to identify existence of an event and then use simple tf-idf (Term Frequency - Inverse Density Frequency) weighting to identify the event with highest score in detected burst.

In this research [9] the authors use an unsupervised approach based on the belief that "a generic rule-set, if applied within a reduced and localized search space, is sufficient for event detection". They reduce the search space by restricting their work to soccer domain where specific set of events can be pre-defined. They use minute-by-minute commentary for soccer to identify a set of events by a rule-based approach using key words/phrases commonly used in online commentary (Figure 5). Once events identified they use the corresponding time stamps to localize search space in audio and video processing to obtain exact frame of event occurring.



Event	*Keyword(s)*
Goal	- *goal!, goal by, scored, scores, own goal, convert*
Penalty	- *penalty spot, power penalty, placed penalty, penalty kick, penalty-kick, converts [a-z][a-z][a-z] penalty*
Yellow Card	- *yellow, yellow card, booking, booked, bookable, caution, cautioned*
Red Card	- *dismissed, sent-off, sent off, sending off, red card, red-card, sees red, second booking, second bookable*
Substitution	- *substitution, replaced by, comes on for, replaces, [(]in[)], [(]out[)], on for, in for*

Figure 5 Key words and phrases used for rule set of [9]

Most of the approaches of event detection from text discussed above are rule-based or pattern-based methods. Even though those approach does not require tagged data (which are time & labor consuming to produce), there are few drawbacks common to these approaches. First creating a rule set is a time-consuming process which require considerable domain knowledge as there can be many possible ways an event can be described lexically. On the other hand once a strict set of rules are defined there will be a risk of missing events that are described in ways that were not considered when creating the rule set. This will result in low recall values. Also, once rules are defined for a certain domain it is not easily portable to other domains and need to be redefined. All these drawbacks are present in knowledge-based ontology driven even extraction mechanisms too.

2.3.2 Event extraction using multi label classification

As discussed in Chapter 1 the focus of this research is to identify the existence of pre-defined set of events in the online sports commentaries and this is implemented for the cricket domain using ball-by-ball commentary available in web. This is approached as a classification task where commentary for each ball classified in to events defined. When carefully looking at commentary line for a ball it can be seen that commentary for each ball can belong to zero events and also there can be instances where it can contain more than a single event. For an example the batsman can hit a 'FOUR' or 'SIX' to a 'SLOW-BALL'. Therefore, this was modeled as a multi label classification problem thus literature related to this is considered in this section.

This paper [18] does a thorough comparison of available methods to approach multi label classification problem and there are three main approaches that can be used as below:

- Problem transformation

  This approach transforms the multi label classification problem to a set of single label classifications tasks or a regression problem. Main types of this include Binary Relevance (BR) methods which uses an OnevsRest strategy to transform problem to set of binary classification tasks for each label, Label Power set (LP) methods where problem is approached as a single label classification problem by considering each combination of available labels as a single label and pairwise comparison methods where series of classifiers are trained taking two labels each and then combined based on majority vote.

- Algorithm adaptation

  This approach deals with extending an existing algorithm to handle multi label classification problem. Work has been done on Boosting, kNN, Decision trees and many other algorithms to adopt them to multi label classification problem

- Ensemble methods

  These methods are developed on top of the common problem transformation or algorithm adaptation methods with most frequently used ones being RAKEL system, ensembles of pruned sets (EPS) and ensembles of classifier chains (ECC).

This paper [19] uses a multi label classifier in the form of a Support Vector Machine (SVM) to extract events in economic domain from news articles in Dutch. They model this as a multi label classification task because a news article can contain more than one event and use a combination of lexical, syntactic and semantic features for classification to be done at sentence level. As for lexical features they consider

BOW- Bag of Words features for all tokens (unigrams, bigrams…), BOW features for all characters (unigrams, bigrams… inside tokens), features indicating presence of numbers & symbols (such as $, % since these would be common in economic and finance text) and BOW features for all lowercased lemma + PoS – Part of Speech tag pairs. They consider few attributes based on PoS tags as syntactic features (for each PoS tag attributes such as whether tag is present or not, no of occurrences of tag, frequency of tag etc.). Similar set of semantic features (counts, features etc.) are defined based on named entities in the text.

In this paper [4] the authors perform the same task but on English news articles. In addition to the rich feature set based SVM approach used in earlier research, they use a word vector based long short-term memory recurrent neural network (RNN-LSTM). SVM approach is tested both with a linear kernel and a RBF kernel by recasting the problem as a one-vs-rest binary classification task for each class. The neural network classifier is tested both as a multi-label single model classifier and a one-vs-rest set-up. And they claim to have best results using SVM with a linear kernel as one-vs-rest classifier which are discussed in detail in Chapter 4.

In this thesis [5] which is focusing on player analytics from soccer commentaries as discussed in section 2.1, they model the classification of commentaries to pre-defined events both as a single label multi class problem as well as a multi label classification problem. Prior to classifying they remove all entity names from text using NER and use auto tagging template to tag events using syntactic similarity in lexicons. For an example a foul will always be recorded as foul by <player name> in the web site they used to extract data from. For multi-label multi-class classification, they experiment with set ups for Binary Relevance, Chain Classifier and Label Powerset models, together with Multi-Label kNN (MLkNN) and three variations of Naive Bayes (NB) classifiers i.e. Gaussian Naive Bayes, Bernoulli Naïve Bayes, Multinomial NB models. They claim to have best results with MLkNN with all other models except for Gaussian Naive Bayes ones closely following. They explain this by the fact that

“Gaussian Naive Bayes is more suitable for normally distributed and in this scenario, it was random, and the labels were more suitable for Bernoulli Naïve Bayes and Multinomial Naive Bayes where more emphasis is given on word-presence and word-count”.

2.4 Attribution of events

In this thesis [5] done on soccer domain the researches are working on the task of player attribution i.e. connecting identified an event in the field to a certain actor/player. With ambiguity of language being used in commentary this task requires some domain knowledge for any sports. For soccer this would be further difficult when compared to a game like cricket, because all the players of both the teams will be on the field at a given time so unless the knowledge on player positions, team information available, it is not possible to attribute an identified action such as ‘GOAL’ to a player if commentary text contains more than one player name. In contrast in a cricket match if an event like ‘BOWLED’ is considered from domain knowledge it can be understood that only players will be related to this event current batsman and the bowler.

The authors use Named Entity Recognition (NER) to identify the players mentioned in text and to validate them in the player list they have used a free text search mapping since they have obtained commentaries and player info from two different web sites. This means after identifying player from commentary data set they will do a search in player list obtained from another source to make sure the identified name corresponds to an actual player. They have come up with a model to assign events based on player positions i.e. “based on the assumption that mostly strikers or mid-fielders are responsible for creating chances or missing upon any opportunities, defender or midfielders are accountable for tackles/blocks/corners, and goalkeeper for making saves”. They have admitted that this approach is not robust and discussed possible scenarios of it failing.

2.5 Summary of existing Work

When analyzing the related literature for this research it could be seen that even though there are few research work done based on online sports commentaries, according to my knowledge there had not been any work that detects events from live commentaries and use them for analytics purposes in cricket domain. For event extraction wide range of approaches had been tested such as general rule-based algorithms based on lexical patterns, incorporating semantic information through entity extraction and supervised algorithms in multi label classifications.

Analyzing these approaches, for event extraction task of this research a hybrid approach based on both data driven and knowledge driven approaches can be used. As the domain is fixed to cricket and the commentary web site contains specific format for describing several events those linguistic patterns could be applied and for other events which are not easily discoverable by analyzing lexical patterns, statistical information on word tokens and semantic information via entity recognition can be used to train a classifier.

CHAPTER 3: METHODOLOGY

This chapter will discuss the proposed solution of this research of producing structured data from online sports commentaries. Section 3.1 will discuss the overall architecture of the system which is an end to end system implemented as a pipeline of steps and Section 3.2 goes in to details of how each stage of the pipeline is implemented. Section 3.3 discusses the adaptability of the system to other domains and Section 3.4 concludes the chapter with a summary.

3.1 Overall Architecture

This research presents an end to end system of producing structured data from a large corpus of text in the form of online sports commentaries. Implementation is done for cricket domain by scraping ball-by-ball commentaries from web, identifying the events mentioned for each delivery in these commentaries and saving the commentary in a database delivery wise for each match so that this information can be used in future for analytic purposes.

Architecture of the system can be mapped to a pipeline where output of one step becoming the input for the next step. This pipeline consists of most steps of a Machine Learning (ML) application i.e. Data collection, Pre-processing, Model building and also an application utilizing the model built. Figure 6 depicts overall architecture of the system.

The system first scrapes the web collect training data (ball-by-ball commentary for cricket), tag and uses them to build a classification model to extract events from the commentary lines. During this process commentary text is saved in to an intermediate form, as a set of csv files. Prior to saving csv files commentary text is pre-processed and actors are identified using NER which then used as semantic information in classifier building.

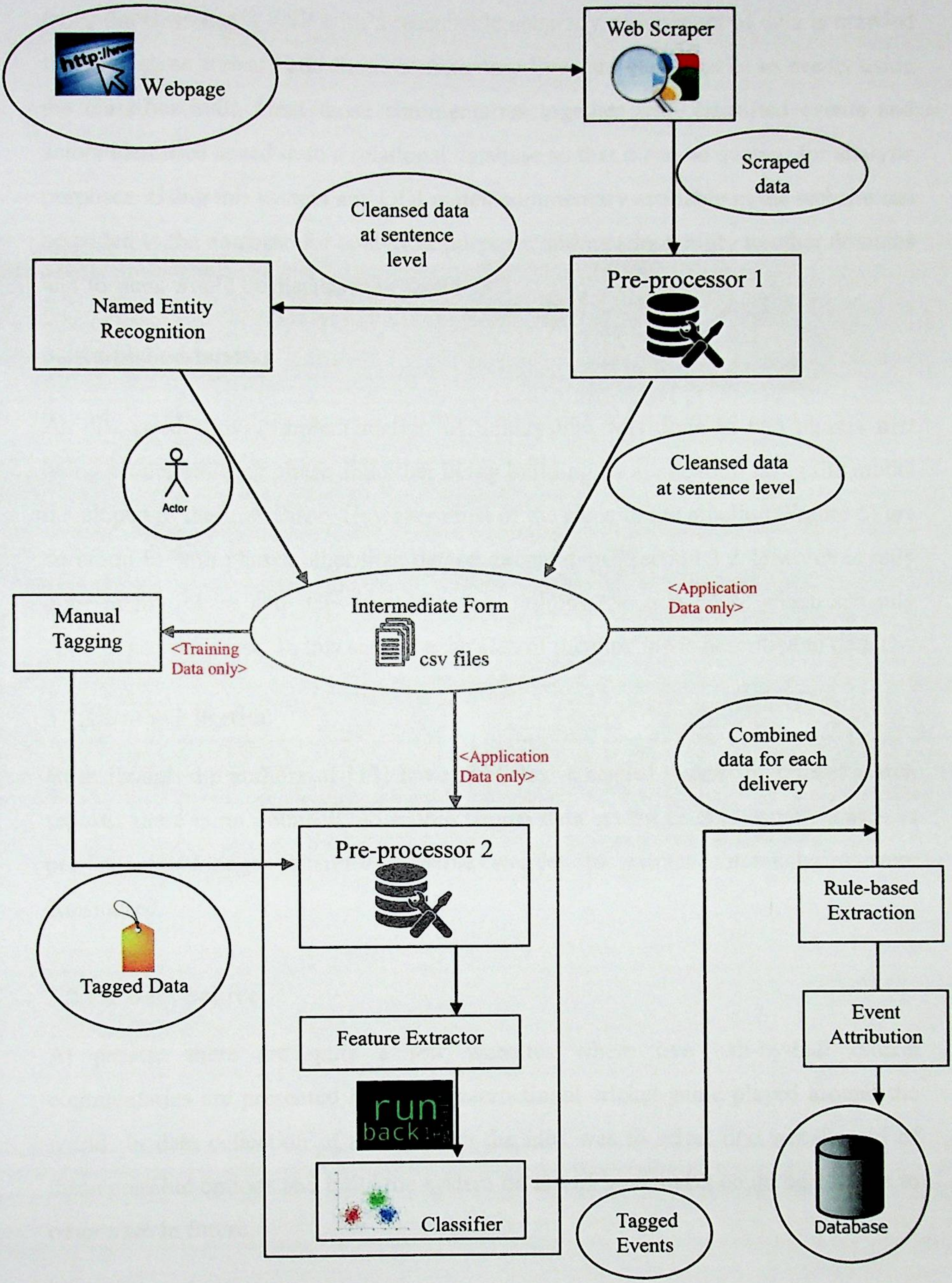


Figure 6 Overall Architecture of the system

Once the classifier is built with a reasonable accuracy, another set of data is crawled from the same website and those commentary lines are classified in to events using the classifier built. Next those commentaries together with classified events and actors identified saved in to a relational database so that it can be queried for analytic purposes. Using this system any ODI match commentary available in the website can be added to the database for analytical purposes and its adoptability to other domains and formats would be discussed in section 3.3.

3.2 Implementation

As discussed above, implementation of this system was done in two phases first being model building phase and other being building an application using the model developed in the first phase. However most of the steps in the pipeline (Figure 5) are common to both phases other than dataset preparation (Section 3.2.4) which is only done in first phase and steps in application building (Section 3.2.6) which are only done in second phase. In this section each step of the pipeline is described in detail.

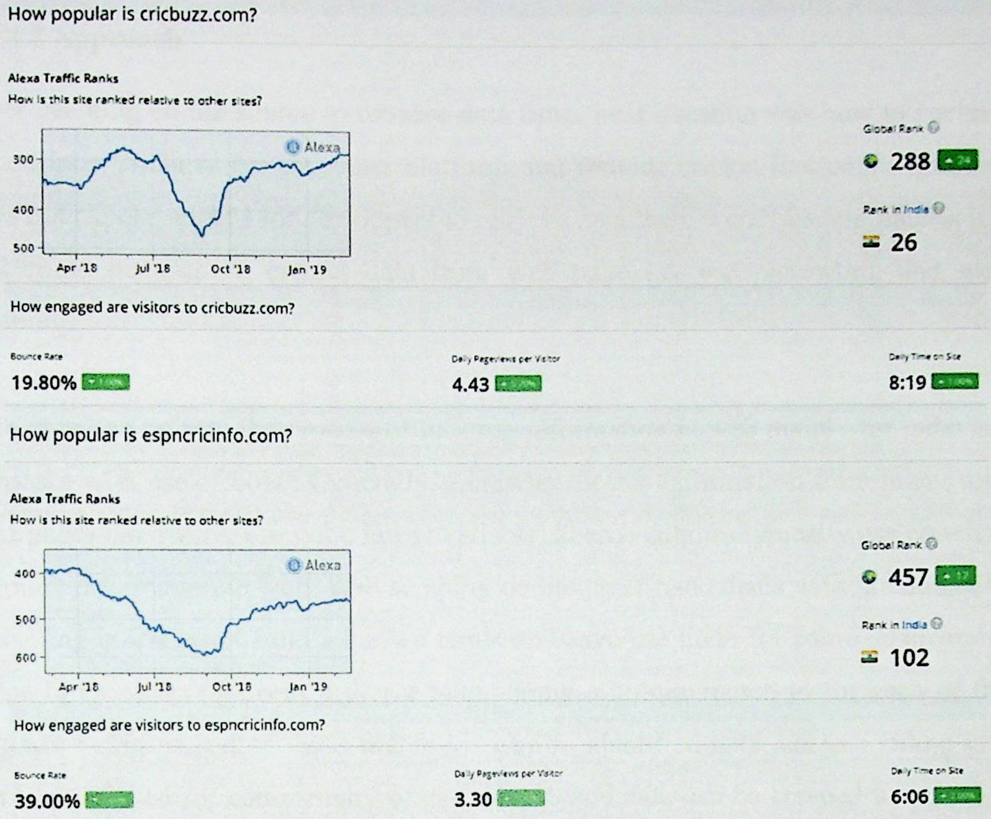
3.2.1 Data Collection

Even though the authors of [13] have published a tagged dataset of cricket match reports, there is no publically available tagged data set for cricket commentaries as per our knowledge. Therefor possible sources to extract commentaries were considered.

3.2.1.1 Data Source

At present there are quite a few websites where live ball-by-ball cricket commentaries are presented for every international cricket game played around the world. In data collection of the research the idea was to select one website out of these possible options and build the system on top of it so that it could be adapted to other sites in future.

With Cricbuzz[4] and Cricinfo[5] being the major two rivalries here, a comparison between the two sites is included in Figure 6 below.



How popular is cricbuzz.com?

Alexa Traffic Ranks

How is this site ranked relative to other sites?

Global Rank

288 ▲ 24

Rank in India

26

How engaged are visitors to cricbuzz.com?

Bounce Rate

19.80%

Daily Pageviews per Visitor

4.43

Daily Time on Site

8:19

How popular is espncricinfo.com?

Alexa Traffic Ranks

How is this site ranked relative to other sites?

Global Rank

457 ▲ 17

Rank in India

102

How engaged are visitors to espncricinfo.com?

Bounce Rate

39.00%

Daily Pageviews per Visitor

3.30

Daily Time on Site

6:06

Figure 7 Comparison between Cricbuzz and Cricinfo.

Source: Amazon Alexa [20]

Stats shown in Figure 7 had last been updated on February 20, 2019 and calculated based on data for last 3 months from that date. According to definition in website [20] Global Rank which is updated daily “is calculated using a combination of average daily visitors to this site and pageviews on this site over the past 3 months. The site with the highest combination of visitors and pageviews is ranked #1”. According to this Cricbuzz is ranked higher than Cricinfo. In addition, the feasibility of using each site was considered and implementation seemed possible in almost the

[4] https://www.cricbuzz.com/

[5] http://www.espncricinfo.com/

same approach using same technologies. Therefor based on above factors it was decided to proceed with Cricbuzz.

3.2.1.2 Approach

After deciding on the source to retrieve data from, next question was how to retrieve data. Since Cricbuzz (or any other platform that provide cricket live commentaries) does not expose an API for developers to retrieve data there were two approaches left that could be used to extract data from web page i.e. web scrawling and web scraping.

Web crawling refer to the process of locating information in web mainly for indexing purposes with use of bots. Generally, a crawler locates information from many web sites/pages navigating using the links available. Search engines typically use crawlers to index information in web. Web scraping on the other hand deals with intentionally extracting information from a known target and save/use them for some application. Since in Cricbuzz platform they are maintaining a unique match id for each of the matches being played, if those match ids can be identified data, corresponding urls can be identified for commentary of each match and data can be scraped using these urls. Therefore for this system web crawling would not be required as it had been stated in [11], [12] and [13] research work which involved in extracting cricket ball-by-ball commentaries discussed in Section 2.2.

3.2.1.3 Technologies used

When it comes to web scraping there are several technologies available. Most common mechanisms will be to use Regular Expressions (RE) and few Python libraries such as BeautifulSoup[6] [21], lxml[7] and HTML Parser[8]. Out of the work discussed in Section 2.2 which were all involved with online sports commentaries, only [5] had used a web scraping approach and Beautiful Soup. However, all these

[6] https://pypi.org/project/beautifulsoup4/

[7] https://pypi.org/project/lxml/

[8] https://docs.python.org/3/library/html.parser.html

libraries are designed to extract data from HTML or XML documents and needs to be used together with Python core libraries urllib[9] or urllib2[10] which are first used to send HTTP requests to the website's server to obtain data.

However, with Cricbuzz website using AJAX with XMLHttpRequest objects which are returning json objects, it's much easier to use Requests[11] library in python which comes with a built-in JSON decoder. This eliminates the need for separate HTML or XML parser. Therefore, it was decided to web scrape Cricbuzz website using Python Requests library in order to collect data.

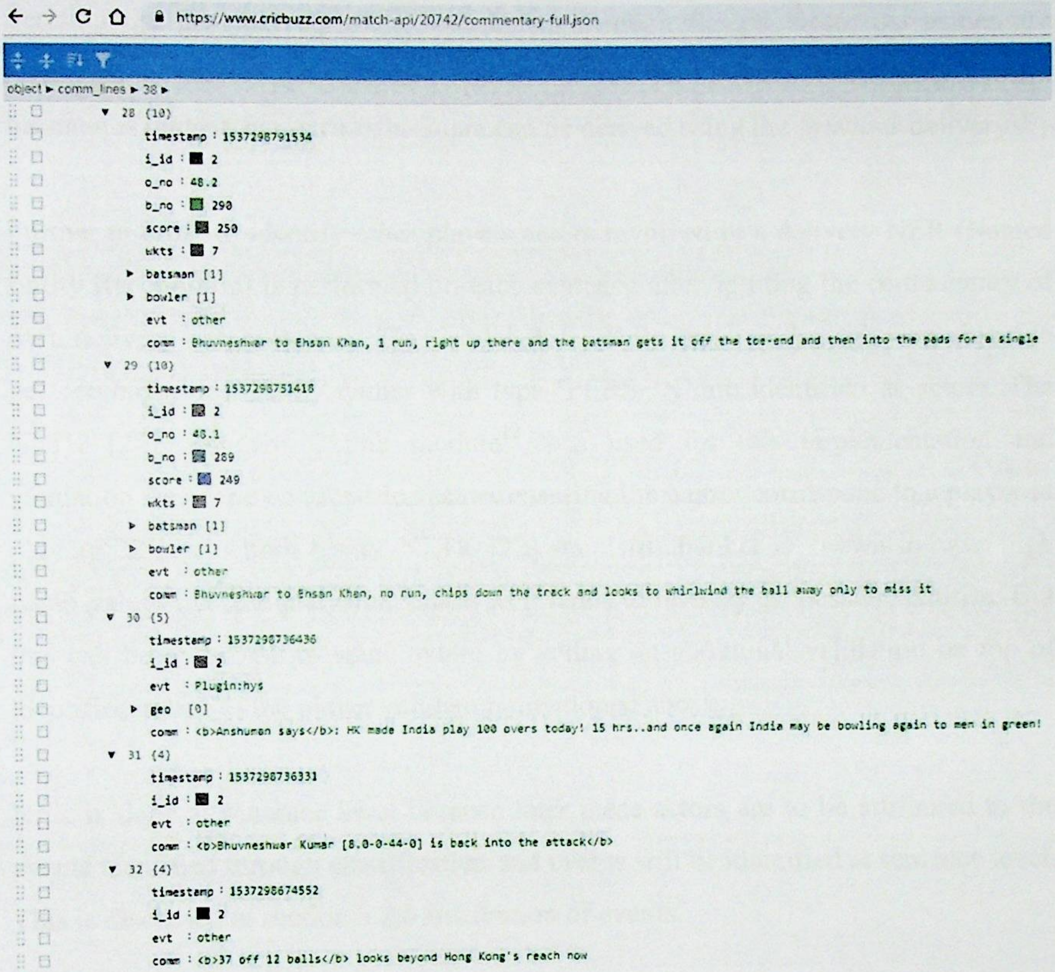
3.2.1.4 Structure of Data

With scraped data of a match in hand in the form of a single json object, information extraction becomes easy after studying the structure of json object. Commentary is already in a semi structured format where commentary text describing deliveries of the match are presented in separate nodes when compared to personal views sent in by followers of the site. In figure 8 below, nodes 28 29 correspond to two deliveries (48.2 and 48.1) of the match whereas node 30 contain a comment send in by a follower. Based on this structure basic details of the match (two teams, toss info, winner info etc.), player info of each team and commentaries describing deliveries could be obtained by iterating through the json object and filtering the name of respective json nodes. For an example tag 'o_no' exists only in nodes which are used to describe deliveries, thus the commentaries can be obtained by filtering json nodes using this tag.

While iterating over the json object to retrieve ball-by-ball commentary for each delivery, the commentary text describing the deliveries are splitted in to sentence level. This is done thinking in line that when a single delivery contains multiple

[9] https://docs.python.org/3/library/urllib.html
[10] https://docs.python.org/2/library/urllib2.html
[11] https://pypi.org/project/requests/

events, if the entire commentary text is used to identify an event during classification results would be less accurate as features used would contain information of other events as well. As a solution it was decided to break down the commentary in to sentence level for classification purpose. In addition, all commentary text were cleaned by removing unnecessary tags such as <br>, <i> used to differentiate font.



```
https://www.cricbuzz.com/match-api/20742/commentary-full.json

object ▸ comm_lines ▸ 38 ▸
    ▼ 28 {10}
        timestamp : 1537298791534
        i_id : ■ 2
        o_no : 48.2
        b_no : ■ 290
        score : ■ 250
        wkts : ■ 7
      ▸ batsman [1]
      ▸ bowler [1]
        evt  : other
        comm : Bhuvneshwar to Ehsan Khan, 1 run, right up there and the batsman gets it off the toe-end and then into the pads for a single
    ▼ 29 {10}
        timestamp : 1537298751418
        i_id : ■ 2
        o_no : 48.1
        b_no : ■ 289
        score : ■ 249
        wkts : ■ 7
      ▸ batsman [1]
      ▸ bowler [1]
        evt  : other
        comm : Bhuvneshwar to Ehsan Khan, no run, chips down the track and looks to whirlwind the ball away only to miss it
    ▼ 30 {5}
        timestamp : 1537298736436
        i_id : ■ 2
        evt  : Plugin:hys
      ▸ geo  [0]
        comm : <b>Anshuman says</b>: HK made India play 100 overs today! 15 hrs..and once again India may be bowling again to men in green!
    ▼ 31 {4}
        timestamp : 1537298736331
        i_id : ■ 2
        evt  : other
        comm : <b>Bhuvneshwar Kumar [8.0-0-44-0] is back into the attack</b>
    ▼ 32 {4}
        timestamp : 1537298674552
        i_id : ■ 2
        evt  : other
        comm : <b>37 off 12 balls</b> looks beyond Hong Kong's reach now
```

Figure 8 Structure of json object

Commentary in Cricbuzz already has a set of events pre-tagged at delivery level. These are tagged under 'evt' tag of json object and include few main events such as wicket, four, six, drop, fifty and century. This information is extracted while iterating through json object. However, since there are many more events that can occur in game of Cricket, a classifier is built in order to extract those from commentary text.

3.2.2 Actor Recognition

Each delivery in Cricbuzz commentary is pre-tagged with current batsman and bowler. In addition, every last ball of the over has non-striker tagged too. So, while iterating the json object, when both batsmen are indicated these two names are identified as current batsman and non-striker for the delivery. When only current batsman is tagged, non-striker batsman can be derived using the previous delivery.

Further in order to identify other players/actors involved in a delivery NER (Named Entity Recognition) is performed on each sentence after splitting the commentary of each delivery into sentences. This is also done while iterating through json object to get commentaries. Entity names with type 'PERSON' are identified as actors. The NLTK [22] Standard Chunk module[12] was used for this implementation and validation was done on identified actors ensuring the names correspond to a player in playing 22 across both teams. NLTK [22] standard chunker is known to have high recall values but low precision values as it tends to identify all possible entities. But this can be overcome to some extent by adding an additional validation on top of identified actors as the player validation mentioned above.

NER is done at sentence level because later these actors are to be attributed to the events identified through classification and events will be identified at sentence level. This is discussed in section 3.2.6 Attribution of events.

3.2.3 Intermediate Form

In the model building phase after all information is extracted from json objects several csv files are created as an intermediate form in data flow. Out of these files the one which contains commentary information for each delivery with splitted

[12] https://www.nltk.org/api/nltk.chunk.html

sentences is used for manual tagging. Below are the csv files created with respective column set.

- MatchInfo.csv – an entry will go in for each match (Mattch_ID,Team1_ID,Team1_Name,Team2_ID,Team2_Name, Toss_By,Win_Team_ID)

- PlayerInfo.csv – an entry will go in for each player (Player_ID,Full_Name, Name,Bat_Style,Bowl_Style)

- MatchPlayers.csv – an entry will go in for each player in the match (Match_ID,Team_ID,Team_Name,Player_ID,Player_Name)

- MatchCommentary.csv – one csv for each match is created (Match_ID,Innings,Timestamp,Batting_Team_ID,Batting_Team_Name,Bowling_Team_ID,Bowling_Team_Name,Nonstriker_ID,Nonstriker_Name,Batsman_ID,Batsman_Name,Bowler_ID,Bowler_Name,Ball,Commentary,Main_Event, Actors)

3.2.4 Dataset Preparation

This step will only be followed in model building phase, to train the classifier. As discussed above ball-by-ball commentaries were scraped from Cricbuzz website by looping through a manually created list of match ids. These match ids correspond to set of One Day International (ODI) matches played between international cricket teams recently (post October 2017). Since in an ODI match each of the two teams have the possibility of batting for maximum of 300 deliveries (50 overs containing 6 deliveries each), matches having more than 550 balls being bowled were considered when creating the list. After scraping and extracting the commentary lines at sentence level total of 14764 sentences were available to train the model.

3.2.4.1 Set of Events

When selecting the events to be identified, care was taken to avoid events that come pre-tagged in web scraping and events that could be extracted using a rule-based approach which are discussed in Section 3.2.6.1. Events are identified by considering point of view of different roles (bowler, batsman etc.) in game of cricket and set of events that could be used to analyze those roles were selected as below.

- Batsman – Type of shots (1. *Pull, 2. Drive, 3. Sweep, 4. Reverse Sweep, 5. Edged*)
- Batsman - Type of dismissals (*6. LBW, 7. Catch, 8. Run-Out, 9. Bowled*)
- Bowler - Type/Variations of balls bowled (*10. Yorker, 11. Full-toss, 12. Bouncer, 13. Slow ball*)
- Fielder (*14. Misfield, Catch - same as 7*)

Brief description of each event is given below based on document ICC Men's One Day International Playing Conditions Effective 30 September 2018 [23], BBC Sports Academy page[13] and Wikipedia page for Cricket[14]. For all shot types and Yorkers the commentary sentence is tagged with the event even if the shot or Yorker is only attempted but not successful because it indicates the intention of batsman/bowler which could be useful in assessing players.

1. Pull – "A cross-batted shot played off the back foot towards the leg side off a short-pitched delivery (ball bouncing around waist height)."
   e.g. "Lakmal was late on the pull an he cops a blow on his thigh pad"
   "Shai Hope gets on top of the bounce and pulls it to deep square leg for a single"

2. Drive – "A drive is a straight-batted shot, played by swinging the bat in a vertical arc through the line of the ball, hitting the ball in front of the batsman along the ground. The straight drive is usually played to an overpitched delivery (not quite on a good length, but not quite a yorker) on or outside off stump."
   e.g. "Rahim drives it past the diving bowler and down to long-off"
   "driven to the left of short cover"

[13] http://news.bbc.co.uk/sport2/hi/academy/default.stm

[14] https://en.wikipedia.org/wiki/Cricket

3. Sweep – "The sweep is a cross-batted shot played to a low bouncing delivery on or around leg stump behind square on the leg side. To keep balance, the batsman usually goes down on one knee on the back leg. Paddle–Sweep and Slog-Sweep which are variations of the same shot are tagged as Sweep."
   e.g. "swept hard to deep backward square leg"
   "de Silva gets down on a knee and slog-sweeps it over backward square leg"

4. Reverse-Sweep – "A reverse sweep is a cross-batted sweep shot played in the opposite direction to the standard sweep, thus instead of sweeping the ball to the leg side, it is swept to the off side, towards a backward point or third man."
   e.g. "reverse swept from off and straight to the man at short third"
   "reverses his stance and smokes the sweep shot past backward point"

5. Edged – This refers to ball hitting on edge of the bat when batsman fails to hit intended shot from the middle of the bat
   e.g. "the nibble off the seam clicks the inside edge and clatters the off bail"
   "Duminy presses forward to block and got an inside edge onto his front leg"

6. LBW – "batsman could be out lbw if the ball would have struck the wicket, but was instead intercepted by any part of the batsman's body. The decision would depend on where the ball pitched, whether the ball hit in line with the wickets, and whether the batsman was attempting to hit the ball.
   e.g. "Mathews has been given out lbw", "D de Silva lbw b Mortaza 0(3)"

7. Catch – "The striker is out Caught if a ball delivered by the bowler, not being a No ball, touches his bat and is subsequently held by a fielder as a fair catch, before it touches the ground."
   e.g. "but Ngidi completed the catch sans any fuss"
   "Mulder runs to his right and then forward to complete a low catch"

8. Run Out - "Either batsman is out Run out, if, at any time while the ball is in play, he is out of his ground and his wicket is fairly put down by the action of a fielder". Stumpings of batsman where "The striker is out Stumped, if a ball is delivered and he is out of his ground, and he has not attempted a run when his wicket is fairly put down by the wicket-keeper" are also tagged as run-outs due to its rarity to be tagged as a separate event. Considering this definition even if the batsman was attempting a run, it would be marked as a run-out anyway.
   e.g. "One of their best batsmen is OUT courtesy a run-out", "out Stumped"

9. Bowled – "The striker is out Bowled if his wicket is put down by a ball delivered by the bowler, even if it first touches the striker's bat or person."
   e.g. "out Bowled"

10. Yorker – "a yorker is a ball bowled (a delivery) which hits the cricket pitch around the batsman's feet."
    e.g. "squeezes out the yorker to deep mid-wicket with a flick"
    "an attempted yorker"

11. Full-toss – "is a type of delivery in the sport of cricket. It describes any delivery that reaches the batsman without bouncing on the pitch first."
    e.g. "low full-toss outside off"
    "targets the stumps and serves a friendsly full toss in the process"

12. Bouncer – "bouncers are usually directed more or less at the line of the batsman's body. Aiming at the batsman is legal provided the ball bounces on the pitch; or upon reaching the batsman, the ball is below the batsman's waist"
    e.g. "bumper fired outside off"
    "slower bouncer and it just rises enough to clear a ducking Mortaza"

13. Slow-ball – "a slower ball is a slower-than-usual delivery from a fast bowler. The

bowler's intention is to deceive the batsman into playing too early so that he either misses the ball completely or hits it high up in the air to offer an easy catch"

e.g. "slower and around off'

"Mortaza bowls this one a touch slow and Shahidi was early into the shot"

14. Misfield – "To field the ball clumsily or ineptly, especially when this results in the batsman scoring another run"

    e.g. "driven to the left of the bowler who tumbles and makes a half-stop"

    "and a diving Chahal fails to collect cleanly"

When identifying the events in addition to above mentioned set, reviewing a decision and a decision being overturned (measuring a captain's ability to effectively use reviews) were considered but had to be ignored later in the process they did not have enough labels in training set due to those events being not that frequent.

3.2.4.2 Tagging and Validation

Tagging has been done manually on the MatchCommentary.csv file produced in Section 3.2.3. Tagging was done based on the boundaries defined for each event as discussed in previous section. Further care was taken to always consider only the sentence that is being tagged and ignore the surrounding sentences even though they could belong to the same delivery of the match. Once manual tagging completed a validation check was done by random checks and also by using a dictionary-based approach similar to one used in [13] to classify events itself. Table 1 contain the dictionary of words used to validate the tags and validation was done by constructing an excel function to auto-tag the sentences and then compare those automatic tags with the manual ones to see if any sentence containing the defined keywords had been missed when tagging. Since tagging was done csv after csv, once tags are validated each ready csv was taken as training data and used in classification model. Based on output of the model training guidelines and the keyword dictionary were refined after analyzing which were tagged incorrectly.

Table 1 Key Word dictionary used to validate manual tags

Event	Keywords
Misfield	misfield, mess, concede, fumble, tumble, sloppy, overthrow
Bouncer	bouncer, bumper
Yorker	Yorker
Full toss	full-toss, full toss
Slow ball	slow, off-paced
Sweep	sweep, swept
Reverse-Sweep	reverse-sweep, reverse sweep, reverse swept
Drive	drive, drove
Pull	Pull
Edged	edge, edging, nick
Run Out	run out, stumped, run-out
LBW	Lbw
Bowled	Bowled
Catch	caught, catch

3.2.5 Event Classification

The objective of building a classifier in this research is to identify the events contained in the commentary text of ball-by-ball commentary for cricket. The tagged data set contains commentary for deliveries splitted in to sentence level and there could be sentences tagged with more than one label as seen in Section 3.2.5.2 Exploratory Analysis.

As discussed at the end of section 2.3.2 many event classification tasks in text domain has been approached as unsupervised rule based or pattern-based algorithms. These approaches require substantial time from domain experts to create a rule set and also the risk of missing some information will always be there since an event can be lexically described in more than one way and rule set may not represent all of them. Therefor in this research it was decided to approach this problem through a multi label classification algorithm.

The steps discussed in this section will be used in both model building and application building phases except for step 3.2.5.2 Train-Test split which take place only when building the model.

3.2.5.1 Pre-processing

Pre-processing of scraped had been done in two stages where in initial stage unnecessary tags such as <br>, <i> used to differentiate font were removed prior to saving commentaries in intermediate csv files. Also, since Cricbuzz had comma as the sentence separator in commentary text, it was replaced with full stop marks as otherwise it would have created an erroneous comma separated file.

In second phase of preprocessing which was done once the dataset is prepared, stop words (the default set of stop-words from NLTK library [22]) & numbers were removed and some regular expression processing was done to transform text to a standard format. As an example, substring 'll was replaced with will and 've was replaced with have etc. Next stemming is performed to derive the base form of the words which are having the same semantics. As an example, this would transform all three words 'Drive', 'Driven', 'Drove' to base form 'Drive'.

Further it could be observed that in Cricbuzz commentary start of the commentary text follows the same pattern i.e. "<bowler name> to <batsman name>, xx run(s)". e.g. "Maharaj to Kusal Mendis, 1 run,". Therefore, this doesn't seem to add any information for the classifier when vectorizing the text. Thus, this initial segment of commentary text was removed.

In addition to these steps the human names identified in actor recognition as described in Section 3.2.2 was removed from text as those cannot be considered as having an impact in describing a particular event in commentary text.

3.2.5.2 Exploratory Data Analysis & Train-Test split

For the results mentioned here a training set of 14764 records (commentary sentences) was taken and out of these 1558 sentences had at least single event tagged. Below Table 2 states the multiple tag counts per commentary sentence. From this table it can be seen that in the dataset 1558 sentences has at least one sentence tagged (i.e. 10.9% of dataset). This indicates that label matrix of the training set would be a sparse matrix.

Table 2 Multiple tags per Commentary sentence

No. of Labels Tagged	No of instances
0	13154
1	1558
2	50
3	2

Figure 9 below depicts the label distribution between tagged classes (i.e. events). It can be seen that there is a general balance between the classes even though Drive event had been more frequent than all others.

In the model building phase after pre-processing is done data set is divided in to train and test sets using 20% split ratio where training set ended up having 11811 records out of which 1313 sentences (i.e. 11.12% of train set) had at least one label tagged. Test set ended up having 2953 records out of which 297 sentences (i.e. 10.06% of test set) had at least one label tagged.

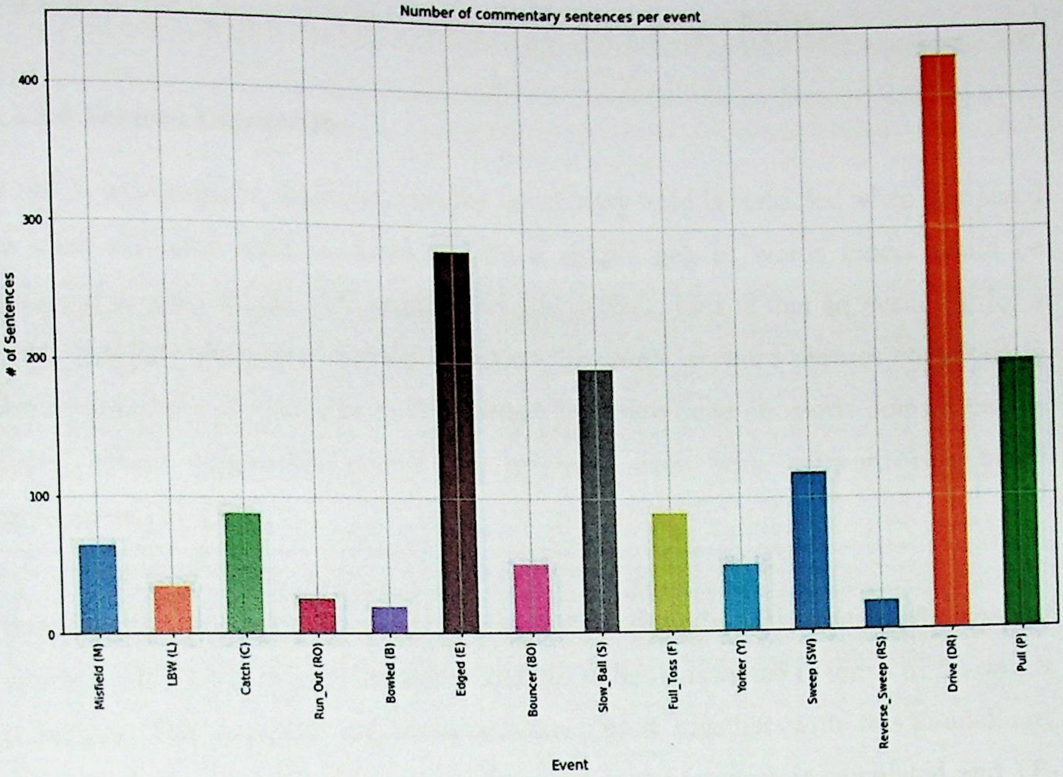


Figure 9 No of commentary sentences per event in training set

3.2.5.3 Scikit-learn Pipelines

Scikit-learn library in Python provides the functionality of Pipelines which can be used to automate the repetitive steps in a machine learning task. It streamlines the process including feature extraction and classification and allows users to use one function call instead of doing each step of the process separately. For an example a pipeline can be defined with a feature extractor (e.g. CountVectorizer) and a classifier (e.g. RandomForestClassifier). Next since pipelines are set up with the fit/transform/predict functionality, same pipeline created can be used to fit/transform training data and make predictions using test data and Vectorizer will be applied to both data sets during the process.

This functionality is extremely useful in a research like this where combinations of classifiers/ feature extractors and algorithms need to be tested, as this can be achieved by changing one or two lines of code when using a Pipeline.

3.2.5.4 Feature Extraction

Since in online sports commentaries the vocabulary used is restricted when compared to other domains such as news articles a simple bag of words model could be assumed to give results with enough accuracy. Basic idea is that an event tag for a given instance would be decided based on the words or word phrases contained in the commentary. In fact, due to this feature literature done on sports domain can be found where reasonable results are achieved even with keyword rule based approaches [9], [13] .

Based on this in feature engineering a bag-of word model was considered. A bag of words model is where each instance that need to be represented in terms of the words it carries. The common statistical measures used together with this model are, CountVectorizer in which count of each word in the instance is calculated and TF-IDF Vectorizer where the score is changed based on the rarity of the word in entire corpus/ or in all set of commentaries in this research. In our research we use both TF-IDF Vectorizer and CountVectorizer at both token level and n-gram level.

3.2.5.5 Classification

As discussed in the beginning of Section 2.3.2 there are three types of approaches that can be used to address a multi label classification task, i.e. Problem transformation methods, algorithm adaptation methods and ensemble methods. Out of these approaches most widely used approach for text classification is OnevsRest based on Binary Relevance concept, which is a problem transformation method. This approach breaks down the multilabel classification problem in to set of binary classification problems for each class. Each binary classifier will decide whether a record belong to that class or not. Sum of output of all classifiers will be the multi label classification for that record.

The main drawback of this method is known as the ignorance of interdependencies between the labels, however in this work it was possible to good results with OnevsRestClassifier. It does not indicate that classes defined in this classification task are totally independent of each other. From cricket knowledge it can be assumed that there is a higher probability that a bouncer being hit for a pull shot than a Yorker being hit. In addition to OnevsRestClassifier, Label Powerset approach which works by creating a binary classifier for every label combination and Classifier Chains approach where labels are predicted sequentially was tried out for comparison purposes. In [24], it is stated that when best binary classifier available is used for a multi-class classification task, it will make little difference based on the approach selected to address the problem.

Therefore, algorithms to be used for classification were considered based on pass results. In Literature review few research works which contained multi-label text classification tasks were discussed and in this work few algorithms which had given good results in related work was tried out and results are all analyzed in Chapter 4 to select the best. Also, these algorithms are generally accepted to provide best results for text classification. In this paper thorough analysis in all the possible algorithms for text classification and it is stated that Boosting-based classifier committees, support vector machines and regression methods deliver top performance. Therefore, in selecting one efficiency considerations or application dependent issues should be considered [25].

SVM is generally known to work well in text classification [26], [27] and in [4], Linear SVM had given best results for multi-label classification of economic news. According to [28] SVM is more capable of solving the multi-label classification. In [5], for Cricket commentary classification three variations of Naïve Bayes classifiers were tested and Gaussian Naïve Bayes had produced poor results which they explain by the fact that Gaussian Naive Bayes is more suitable for normally distributed data set. Thus, we focused on Multinomial Naive Bayes where more emphasis is given on word-presence and word-count as it had given good results in classifying Cricket events in [14]. Ada-Boost algorithm was applied considering it had given best results for multi –label classification task done on cricket domain in this research [13]. We also tried Logistic Regression classifier and Decision trees for comparison purposes.

The selected algorithms were tried out with all three approaches OnevsRest, LP and CC. On each approach, for each classifier and feature vectorizer combination, data set was trained and tested on using Scikit-learn pipelines.

3.2.6 Application Building

Based on the results of the previous step best performing classifier was selected to be used in application building. The steps mentioned in this section are only done in the application building phase of the research. Application building phase make use of the model trained in model building phase to classify the events hidden in ball-by-ball commentaries and extract some additional information using a rule-based approach before saving all information in to the database.

3.2.6.1 Rule based Event Extraction

As mentioned in Section 3.2.5.1 commentary text in Cricbuzz follows the pattern i.e. "<bowler name> to <batsman name>, xx run(s)". e.g. "Maharaj to Kusal Mendis, 1 run", "Maharaj to Oshada Fernando, no run". This pattern was used to identify the events of 'Dot balls', '1 Run', '2 Runs' and '3 Runs' as how many dot balls faced by a batsman would give an indication of his nature of play and how many runs been run between wickets indicates their agility, these runs could give an indication of non-striker batsman too.

3.2.6.2 Attribution of Events

In Section 3.2.2 it was discussed how NER was used to identify the actors related to each sentence and in Section 3.2.5.5 events related to each sentence was identified. In this section the task of attributing these identified events to the actors identified is discussed. Unlike in soccer in game of soccer only few players out of 22 players playing in the match can be involved in a single delivery. Especially this holds true for the events we have defined i.e. type of shots, ball types etc. This makes co-reference resolution easy. Table 2 contains the players roles that can be associated with full event set identified using multi-label classification, pre-tagged in commentary and rile-based tagging.

Some ambiguity can still exist as two players can have the same name and most of the times commentary does not have the full name of the players. In this case domain knowledge is used to narrow down scope based on the possibility a player from each team can be involved in the event considered. For an example consider a scenario where there are two players having the first name 'Jason' in opponent teams and none of them are identified as batsman, non-striker or bowler. Now a sentence which contains a catch event has an actor identified with the name Jason. In this scenario based on knowledge on Cricket it can be deduced that this should be the 'Jason' in the fielding team who took the catch.

Table 3 Roles associated with all events

Event	Player role(s) involved
Misfield	Fielder
Drop	Fielder
Bouncer	bowler
Yorker	bowler
Full toss	bowler
Slow ball	bowler
Sweep	batsman
Reverse-Sweep	batsman
Drive	batsman
Pull	batsman
Edge	batsman
Century	batsman
Half Century	batsman
Six	bowler, batsman
Four	bowler, batsman
3 Runs	bowler, batsman, non-striker

2 Runs	bowler, batsman, non-striker
1 Run	bowler, batsman, non-striker
Dot Ball	bowler, batsman, non-striker
Run Out	batsman, , non-striker, fielder
LBW	bowler, batsman
Bowled	bowler, batsman
Catch	Bowler, batsman, fielder

Algorithm 1 below summarizes the approach taken to attribute events to actors. If there are two or more actors identified for a commentary sentence who are not batsman, non-striker or bowler of the delivery, the one from fielding team is sent to below algorithm as variable actor. If there are no or more than one fielding team players exists as actors nothing is passed in to below logic.

Algorithm 01 Event attribution for a commentary sentence

```
procedure eventAttribution (E, actor, batsman_id, non-striker_id, bowler_id) ▷ Event Set E
    event_actor_role ← { } ▷ data frame having fields event id, actor id and role played.
    for each event e in E do
        if e in (any ball type)
            add event_actor_role {e, bowler_id, 'bowler'}
            continue;
        else if e in (Misfield or Drop)
            add event_actor_role {e, actor, 'fielder'}
            continue;
        else
            add event_actor_role {e, batsman_id, 'batsman'}
            if e in (any shot type or 100 or 50)
                continue;
            else if e in (any run or any dismissal other than run-out )
                add event_actor_role {e, bowler_id, 'bowler'}
                if e in (SIX or FOUR or LBW or BOWLED)
                    continue;
                else if e in (any run or run-out)
                    add event_actor_role {e, non-striker_id, 'non-striker'}
                    if (any run)
                        continue;
                    else
                        add event_actor_role {e, actor, 'fielder'}
                        continue;|

            else
                add event_actor_role {e, actor, 'fielder'}
                continue;

    return event_actor_role ▷ Identified relationships will be returned
end procedure
```

Algorithm 1 Event attribution for a commentary sentence

3.2.6.3 Database mapping

After all information is extracted from scraped text as described throughout this chapter, final remaining step when building application is to save the produced data structured format. Until this stage the commentary text for each delivery was broken down to sentences in order to extract information. When saving the data these sentences and event/actor relationship would be accumulated for each delivery.

In selecting the type of database to be used after considering current trends and their advantages/disadvantages it was decided to use a Relational Database Management System (RDBMS) considering the below facts;

- Effective for analytics

The primary object of this research is to make use of online sports commentaries for analytic purposes by storing them in a structured format. Since there are strict rules defined as to what data is stored in the database and how they are related to each other, it takes less time to retrieve intended information from a RDBMS. In contrast in NoSQL databases unstructured data is stored in large volumes without any definite format or relationship among them. Thus, in order to analyze and find patterns from these data mining of such data would be required which takes more time and effort.

- Structure of data is more or less fixed for a given sport

With any given sport having a hard defined rule set the information that is derived from sports commentaries can be easily mapped to an structured format in comparison to other domains like social networks, recommendation systems and economic networks [17] which can be modeled using a database structure like a Graph database in a more effective manner. The main difference here is that relationships between entities are fixed in sports domain.

- All instances of an object would have same fields

Due to the fact that structure of data being fixed as discussed above instance of a particular entity would always have the same values e.g. a Match will always have same set of attributes describing it. In contrast in a domain like news feeds or articles, the content of an article cannot be predicted in advance, use of document databases would prove to be effective any type of data can be stored in a document.

Even though most data structures can be translated into relational schema, it might end up having many tables created due to nature of data and structure of relationships between them being not fixed as discussed above. But that can sometimes lead to the creation of many tables, across which each entity is spread. This can make accessing the data slow.

- Updates are rare

Main disadvantage of RDBMS that is being discussed is its low performance on large volume of data. The bottleneck which leads to this is the difficulty and cost of scaling write traffic as RDBMS are focused on maintaining atomic transactions which affect its ability to scale horizontally as opposed to a NOSQL database. However, in this application database is used for storage purpose of retrieved data that can be queried later and no updates are expected to be performed on them.

Table 4 list the tables and respective columns in the database created based on the ER diagram that was created to model an ODI match shown in Figure 10.

Table 4 Table Design

Table	Column set
Match	<MatchId, Date, Venue. WinningTeam, TossWinTeam>
Team	<TeamId, TeamName>
Player	<PlayerId, PlayerName, BattingStyle, BowlingStyle>
PlayingTeam	<MatchId, TeamId, PlayerId,Home/Away>
Innings	<MatchId, InningsId, BattingTeam, BowlingTeam>
Delivery	<MatchId, InningsId, DeliveryId, BallNo, Runs>
Event	<MatchId, InningsId, DeliveryId, EventId, EventType>
Actor	<MatchId, InningsId, DeliveryId, EventId, Role, PlayerId>
EventType	<EventTypeId, Description, EventCategoryId>
EventCategory	<EventCategoryId, Description>

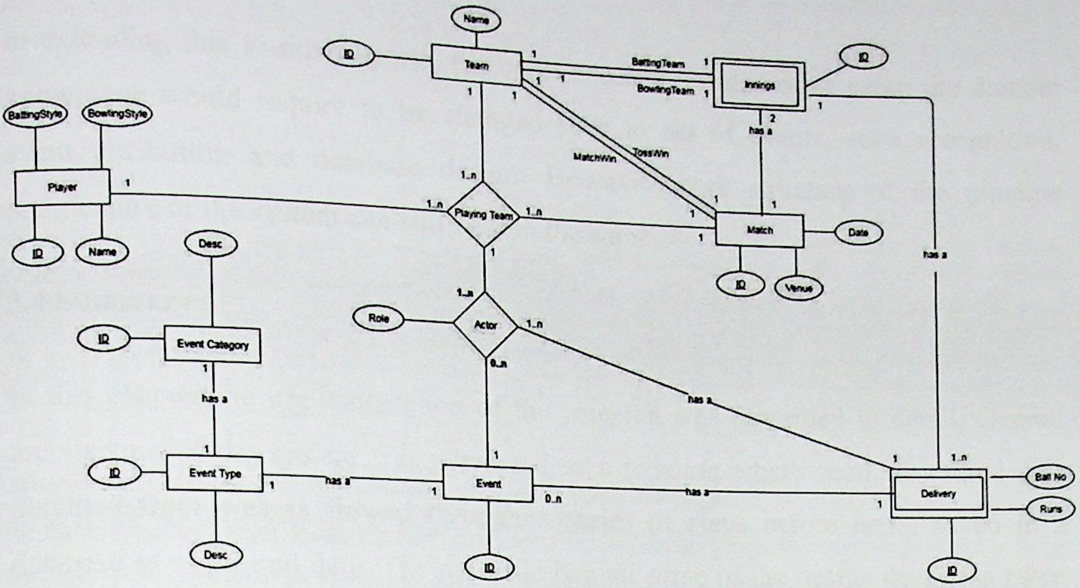


Figure 10 ER diagram

3.3 Adaptability

Since use of lexical syntactic features of Cricbuzz website was kept at a minimum level in this research it can be extended to another cricket platforms without much effort. The web scraping mechanism would need to be changed if required based on how web page is functioning. In addition, the possibility of extracting information from any pre-tagged events and by using the format of commentary text should be examined. In this research only runs scored was extracted using a rule-based approach. Further the keyword-based validation used to validate the manual tags would need to be expanded based on the vocabulary used. However, this can be built on top of current key word set used in this research.

The scope of this research was restricted to ODI Cricket matches but can be extended other formats of Cricket i.e. Test and T20 International matches with minimal changes. The event space might require being changed by adding more events e.g. innings declaration for tests. Database mapping would also need to be changed accordingly and also be expanded to hold specific structure of a Test match having four innings as opposed to limited over matches having only two.

In extending this to other sports, the sport specific designs done using the domain knowledge would require to be changed such as set of events, actor recognition, event attribution and database design. However, core structure of the pipeline architecture of the system can still remain the same.

3.4 Summary

In this chapter the implementation of the research was described in detail. Overall architecture of the system was presented as a pipeline where semi structured data obtained from web is flowed through a series of steps before being saved in a database as structured data. The rationale behind some of the design decisions taken and approaches used was also discussed. Finally, adaptability of this system to other formats and sports is discussed including components/stages which would have an impact.

CHAPTER 4: RESULTS AND ANALYSIS

This chapter analyzes the performance of the system introduced to produce structured text from online sports commentaries. A pipeline architecture is used to transform data retrieved from web scraping, until extracted information is stored in a RDBMS with defined relationships.

Event classification play a major role in the information extraction process and this was approached as a multi-label classification task. Classifiers were built with different approaches of resolving a multi-label classification task. The performance of each of classifier is evaluated by comparing the classifier labeled events with manually tagged commentaries as ground truth.

However, in evaluating the results of the overall system, there were no results published to benchmark against. Therefore, in the analysis, results are compared with the two most related works to this research. Both researches involve multi–label classification tasks in text mining where first one involves with mining soccer commentaries [5] and second one mining news feeds [4].

Section 4.1 describes the experimental set up used and Section 4.2 discusses the parameters used for evaluation. Performance of each classifier presented followed by an analysis in Section 4.3 and Section 4.4 presents a summary of the Chapter.

4.1 Experiment setup

Experiment was carried out on an Intel(R) Core(TM) i7-6820HQ CPU @2.70 GHz processor with 16GB RAM.

4.1.1 Dataset

As mentioned in Section 3.2.5.2 Exploratory Data Analysis, out of total dataset of

14764 records, training set contained 11811 records (1313 sentences having at least one label tagged) and test contained 2953 records (297 sentences having at least one label tagged). Table 5 below contains the label count for each class in both and training and test set. As it can be seen in the table few classes had very few instances in training data however those classes too were included in the classification task and results were obtained to analyze the impact of volume of training data.

Table 5 Label counts in Dataset

Class Label	No. of Labels in Training Set	No. of Labels in Test Set
Misfield	50	14
LBW	28	6
Catch	73	14
Run-out	18	7
Bowled	15	4
Edged	238	44
Bouncer	47	4
Slow-ball	159	37
Full toss	69	19
Yorker	41	8
Sweep	94	24
Reverse Sweep	19	2
Drive	343	86
Pull	166	35

4.1.2 Classifier setups

As discussed in Section 3.2.5.5 three types of problem transformation methods (i.e. OneVsRestClassifier, Label Powerset and Classifier Chains) were tried out with different algorithms identified in Section 3.2.5.5 as below.

1. OneVsRestClassifier (n_jobs[15] = 1) + Linear SVC
2. OneVsRestClassifier (n_jobs = 1) + Multinomial NB(fit_prior=True, class_prior=None)

[15] n_jobs = The number of jobs to use for the computation

3. OneVsRestClassifier (n_jobs = 1) + Logistic Regression(solver='sag', max_iter=1000)
4. OneVsRestClassifier (n_jobs = 1) + AdaBoost(n_estimators=50, learning_rate=1)
5. Label Powerset + Linear SVC
6. Classifier Chain + Decision Tree
7. Classifier Chain + Linear SVC
8. Classifier Chain + AdaBoost(n_estimators=50,learning_rate=1)

With each of the classifiers, both Count Vectorizer and TF-IDF Vectorizer were used for feature extraction. Therefor in total 8 * 2 = 16 classifiers setups were tested. During feature extraction process both unigrams and bigrams was considered because, the use of bi-grams tends to improve performance, as we provide more context to the model. Bi-gram “full toss” could be taken as an example. Since the classification was done on sentence level no higher-order n-grams were considered.

In addition to this ML-KNN was attempted as an adopted algorithm, but it did not converge and exceptions were thrown. This could be attributed to the fact that sklearn ML-kNN algorithm known to perform very slowly with high dimensional sparse matrices. In the classification task we are trying to solve, since bag of words model is used as feature vector set both feature set and label set for a given instance would be sparse which could have led to the performance constraint mentioned above.

4.2 Evaluation Matrices

As this is a multi- label classification problem where most of the instances of data set have more zero or one label, the training and predicted matrices containing vectors for each class would be sparse. Therefore, accuracy would not be a suitable measure to evaluate such classifier, as results would be dominated by the large number of true negatives present in the confusion matrix. Considering this, it was decided to measure and compare results of each classifier based on precision, recall and F1-Score values.

- Precision - This would indicate, out of the instances predicted positive by the system, how many of them are actually positive. A low precision value means instances that were not supposed to be tagged with a certain event are tagged by the classifier. When considering player analytics which is a possible use case of the proposed system, having a low precision will have an adverse effect on a player if the event being considered carry negative sentiments.

  E.g. No. of misfields by a fielder. Player will be considered to have done more than the number of misfields he actually committed.

- Recall - This measure indicates the ability of the system to capture an event once it truly present in data. This would directly impact how much of information we extract from the commentaries. Also, analogues to above, a low recall will adversely affect a player if an event with positive sentiments like a SIX is missed.

- F1 Score - This a combined measure of precision and recall calculated as:

$$\text{F1 Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \qquad (01)$$

  This measure is a good indicator to compare different models where both precision and recall is equally important, as in this system.

4.3 Results and Analysis

To start with, average performances of all classifiers are stated in the Table 6 below. When taking the averages in Table 6 the classes which had less than 50 labels in the training set as for Table 5 was ignored to mitigate the impact of having a small training set. However later in this section these classes are analyzed and it could be seen that no. of labels actually does not heavily impact the classification results.

Table 6 Average performance of all set ups across classes

Classifier setup	**Recall**	**Precision**	**F1 Score**
OnevsRest/SVC/TF_IDF	0.858	0.898	0.856
OnevsRest /SVC/Count	0.872	0.905	0.879
OnevsRest /NB/TF_IDF	0.018	0.375	**0.035**
OnevsRest /NB/Count	0.334	0.86	0.443
OnevsRest /LR/TF_IDF	0.467	0.854	0.590
OnevsRest /LR/Count	0.826	0.833	0.827
OnevsRest/ADA/TF_IDF	0.885	0.909	**0.884**
OnevsRest /ADA/Count	0.876	0.885	0.871
LP/SVC/TF-IDF	0.830	0.891	0.838
LP/SVC/Count	0.855	0.915	0.866
CC/DT/TF-IDF	0.892	0.870	0.876
CC/DT/Count	0.834	0.875	0.862
CC/SVC/TF-IDF	0.858	0.898	0.856
CC/SVC/Count	0.872	0.905	0.879
CC/ADA/TF-IDF	0.881	0.866	0.870
CC/ADA/Count	0.876	0.864	0.866

From overall results it can be seen that almost all of the classifier set ups are able to achieve more than 80% of success rate (considering all precision, recall and F1 score), except for Multinomial NB implementations. The assumption of independence between the features used in Naïve Bayes could have contributed to this, since this classification is done using a bag of words model on commentary text where generally same or similar type set of words are used to describe an event, thus features become dependent on each other. Also as for [26], it works poor, when dataset is small and in handling rare categories that contain only few training instances. This is justified by the fact that in this dataset for classes which had the lowest training instances (less than 50) Multinomial NB (with Count Vectorizer which works better with Multinomial NB, than TF-IDF as shown in Table 8 below) had obtained extremely poor results as shown in Table 7 below. Best performing classifiers had scored better results with same training set as seen on Table 9 below.

Table 7 Performance of Multinomial NB with rare categories

Class	No. of Training Instances	F1 Score with Multinomial NB
LBW	28	0
Run-Out	18	0
Bowled	15	0
Bouncer	47	0.4
Yorker	41	0.222
Reverse-Sweep	19	0

It can be seen that best F1 score is recorded for the set up OnevsRest/ ADABoost/ TF_IDF Vectorizer, with nearly the same results were obtained with setups;

- OnevsRest/Linear SVC/Count Vectorizer,
- LP/Linear SVC/Count Vectorizer,
- CC/DT/TF-IDF Vectorizer and
- CC/Linear SVC/Count Vectorizer.

With most of the times Linear SVC giving good results, justifying its' use for multi-label classification tasks [28]. As for [27] SVM works well with text classification due to high dimensional input space, few irrelevant features and sparse feature vectors. According to [29] a linear kernel is good enough since number of features is large so that data does not need to be mapped to a higher dimensional space.

In order to determine the best approach for feature engineering, average recall and precision values were calculated for each classifier set up for the two vectorization mechanisms used. As it can be seen from Table 8, best approach to be used depends on the classification algorithm. For Multinomial NB, a considerable improvement could be obtained by using Count Vectorizer when compared to TF-IDF Vectorizer even though both the approaches produce extremely low results. From other algorithms it could be noticed that Linear SVM produce better results with CountVectorizer in all three problem transformation methods. In contrast Ada Boost algorithm seems to work better with TF-IDF Vectorizer. Here also when taking the averages, the classes which had less than 50 labels in the training set as for Table 5 above was ignored.

Table 8 F1 scores based on Feature Engineering

	Recall		**Precision**	
	TF_IDF	**Count**	**TF_IDF**	**Count**
OnevsRest/SVC	0.858	0.872	0.898	0.905
OnevsRest/NB	0.019	0.334	0.375	0.86
OnevsRest /LR	0.461	0.826	0.854	0.833
OnevsRest /ADA	0.885	0.876	0.909	0.885
LP/SVC	0.830	0.855	0.891	0.915
CC/DT	0.892	0.834	0.870	0.875
CC/SVC	0.858	0.872	0.898	0.905
CC/ADA	0.881	0.876	0.866	0.864

Table 9 below provides the detailed performance of few best performing classifier set ups (as identified above), across all labels. It can be seen that despite having all labels in Table 5 included (classes having less than 50 training instances also included), all five classifier set ups carry an average F1 score above 85%.

Table 8 F1 scores per Label

	OnevsRest/SVC/ TF_IDF	**OnevsRest/ADA/ TF_IDF**	**LP/SVC/TF-IDF**	**CC/SVC/TF-IDF**	**CC/ADA/TF-IDF**	**AVERAGE**
Misfield	0.4	0.5	0.333	0.4	0.435	**0.328**
Catch	0.857	0.928	0.857	0.857	0.897	**0.845**

Run Out	0	0.308	0	0	0.286	**0.119**
Bowled	0.533	0.857	0.533	0.533	0.857	**0.857**
LBW	0.923	0.923	0.923	0.923	0.923	**0.933**
Edged	0.935	0.854	0.935	0.935	0.854	**0.900**
Sweep	0.936	0.885	0.937	0.937	0.885	**0.9110**
Reverse Sweep	0.667	0.8	0.8	0.667	0.8	**0.8**
Pull	0.943	0.958	0.928	0.943	0.943	**0.952**
Drive	1	1	0.994	1	1	**1**
Full Toss	0.974	0.974	0.974	0.974	0.974	**0.979**
Yorker	0.75	0.75	0.75	0.75	0.75	**0.794**
Slow Ball	0.987	0.974	0.973	0.988	0.974	**0.973**
Bouncer	0.857	0.857	0.857	0.857	0.857	**0.819**
Average	**0.879**	**0.884**	**0.866**	**0.879**	**0.870**	

Average F1 scores for each label depicted in right most column of Table 8. This gives an indication of how well a certain event could be identified from commentary text. Apart from the classes 'Misfield' and 'Run-out' which are having 50 and 18 labels in the training set respectively, for all the other classes average F1 score is recorded at 80% or above. This is further illustrated in Figure 11 below. Labels 'LBW', 'Bowled', 'Yorker' and 'Reverse-Sweep' having less than fifty instances in training set, had obtained high average F1 score values despite limited amount of data.

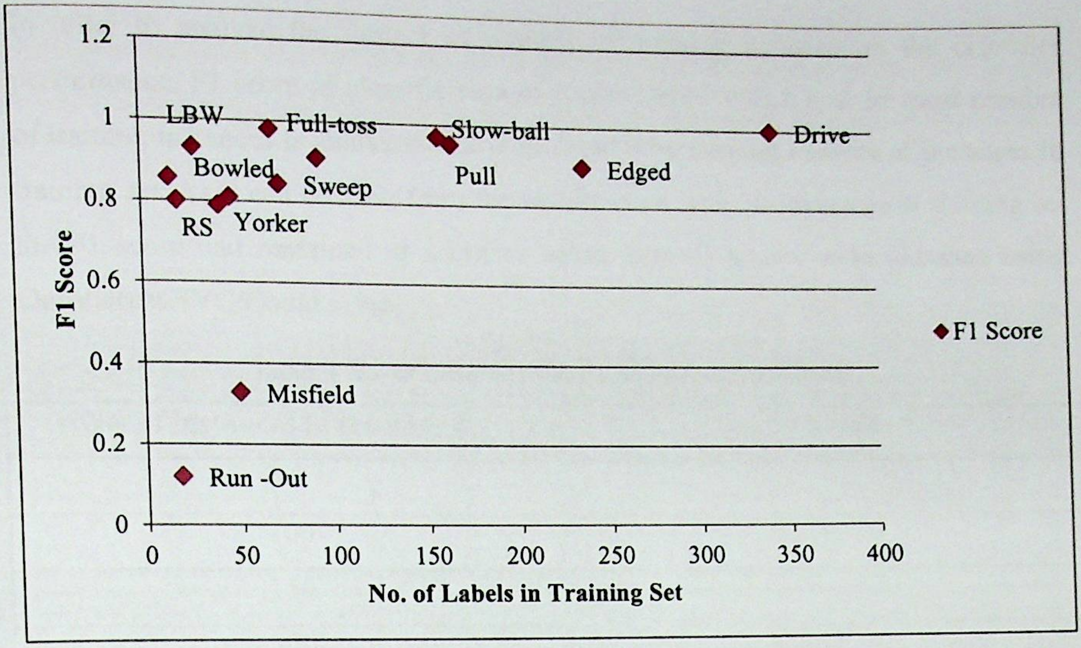


Figure 11 Comparison of F1 Score vs No. of labels in training set

Label 'Misfield' even though having slightly more number of labels than the above mentioned events, has received a low average F1 score value. This could be explained by the fact that a misfield can be described in many ways lexically. In Table 1, in key word dictionary used to validate manually tagged data it could be seen that presence of words "misfield, mess, concede, fumble, tumble, sloppy, overthrow" in commentary line could indicate a misfield event.

Lower F1 score obtained for Run-Out class could be attributed to the fact that class is having only few training samples, but at the same time broad definition given to that class by including event of 'stumping' to the same class could have affected the end result. Even though the two events are conceptually same i.e. batsman been out of the crease when wickets are broken, the lexical representation of the two would be different in commentary as Run-Outs are usually associated with miscommunication between two batsmen and brilliance of a fielder most of the times whereas stumping more or less based on bowler tactics or a misjudgment by batsman.

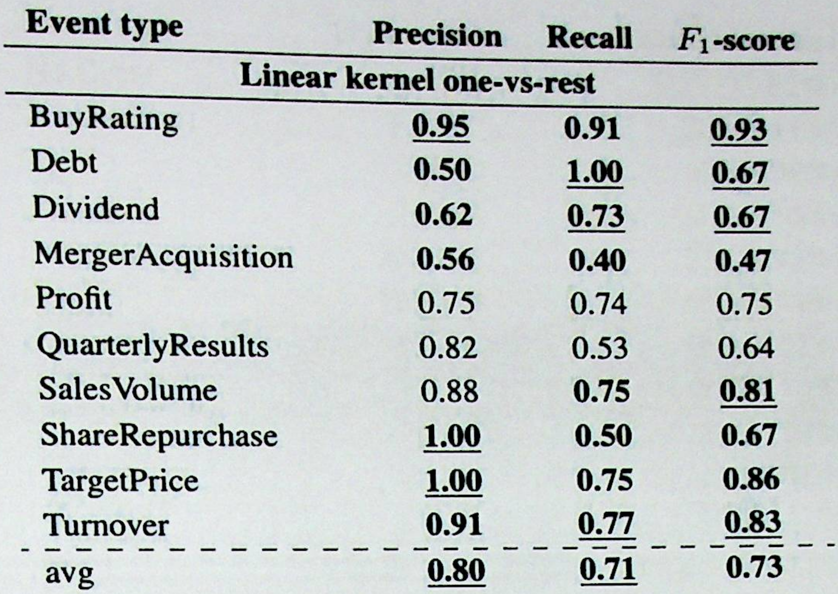
In order to analyze the impact of number of training instance on the classifier performance, F1 score of classification of Class 'Drive' which had the most number of training instances is analyzed below in Table 9 by varying number of instances in training set. As it can be seen from the results even with 50 instances in training set the F1 score had remained at a higher value. The F1 scores were obtained using OneVsRest/SVC/Count setup.

Table 9 No. of instances Vs. F1 Score for class 'Drive'

No. of instances in training set	F1 Score
50	1
100	0.974359
150	0.984127
200	0.987654
250	1
300	0.990991
350	1

Considering the analysis based on Figure 11 and Table 9 it can be observed that more than the no. of instances in the training set, other factors like wider definition of classes and multiple forms of lexical representation of a class can impact the classification results of an event.

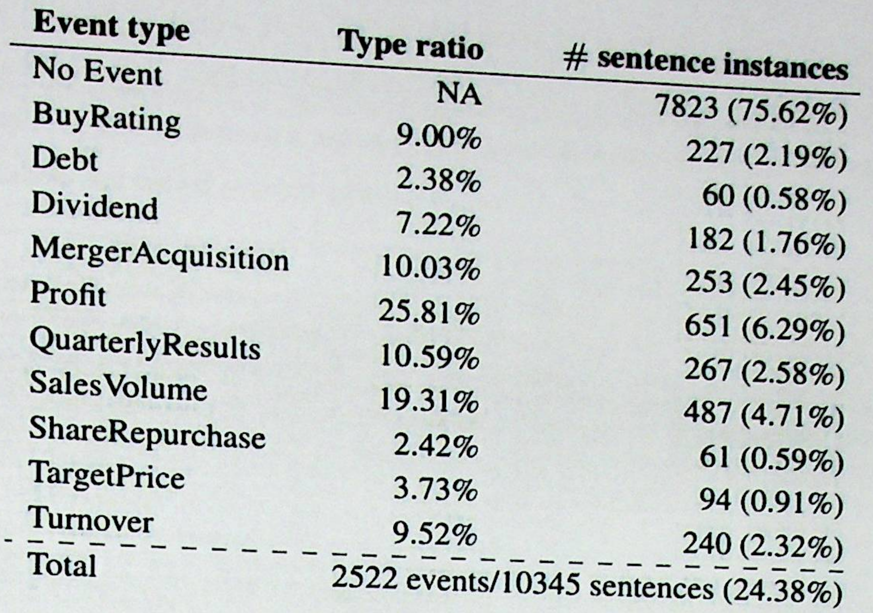
Average F1 scores of each label calculated above in right most column of Table 8, can be compared with similar values obtained in research [4] shown in Figure 12, though it is done for economic domain.



Event type	Precision	Recall	F_1-score
Linear kernel one-vs-rest			
BuyRating	**0.95**	**0.91**	**0.93**
Debt	**0.50**	**1.00**	**0.67**
Dividend	**0.62**	**0.73**	**0.67**
MergerAcquisition	**0.56**	**0.40**	**0.47**
Profit	0.75	0.74	0.75
QuarterlyResults	0.82	0.53	0.64
SalesVolume	0.88	**0.75**	**0.81**
ShareRepurchase	**1.00**	**0.50**	**0.67**
TargetPrice	**1.00**	**0.75**	**0.86**
Turnover	**0.91**	**0.77**	**0.83**
avg	**0.80**	**0.71**	**0.73**

Figure 12 Results for OnevsRest Linear SVM in [4]

It could be noted that results produced by this research are better than what is shown in Figure 12 especially considering the fact that the training data used in that research [4] had slightly higher no. of labels when compared to this research (Table 5). Figure 13 shows the label distribution of [4]. Further since they had only used Linear SVM in OnevsRest classifier set up, the corresponding results of OnevsRest/Linear SVM set of this research (first column in Table 9) can be used for the comparison. There it can be noted that in [4], Linear SVM has recorded an average F1 score of 0.73 across all labels and in comparison, this research corresponding average recorded in Table 9 is 0.8789.



Event type	Type ratio	# sentence instances
No Event	NA	7823 (75.62%)
BuyRating	9.00%	227 (2.19%)
Debt	2.38%	60 (0.58%)
Dividend	7.22%	182 (1.76%)
MergerAcquisition	10.03%	253 (2.45%)
Profit	25.81%	651 (6.29%)
QuarterlyResults	10.59%	267 (2.58%)
SalesVolume	19.31%	487 (4.71%)
ShareRepurchase	2.42%	61 (0.59%)
TargetPrice	3.73%	94 (0.91%)
Turnover	9.52%	240 (2.32%)
Total	2522 events/10345 sentences (24.38%)	

Figure 13 Label distribution of [4]

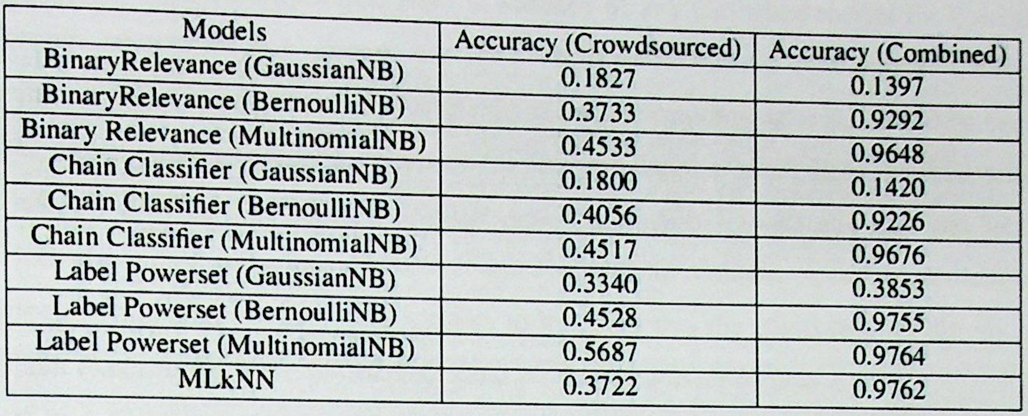
Analyzing results further, in the Table 10 below, results of best performing algorithms in each of the problem transformation methods i.e. OnevsRest, Label Powerset and Classifier Chains are shown. Even though the general norm is that OnevsRest approach is less accurate as it ignores the dependencies between the labels in a multi label classification task, in this research it was possible to obtain results which are as good as other approaches or even better for some set ups as shown in Table 10 below. These results justify the statements made by [30] and [24] where it is stated that good results could be obtained using Binary Relevance once the used with the correct algorithm.

Table 10 Performance of algorithms in different problem transformation methods

	OnevsRest	LP	CC
SVC – TF_IDF	0.858	0.830	0.858
SVC – Count	0.872	0.855	0.872
ADA – TF_IDF	0.885	-	0.881
ADA –Count	0.876	-	0.876

As next step of the analysis, the results obtained in this research are compared with the results available in similar research work discussed in Chapter 2 Literature

Review. Figure 14 below state the results obtained in [5], in which a multi-label classification task is used to classify events in soccer commentaries. They have used two data sets, one with crowd sourced labels only and other being a combination of expert labels and crowd sourced labels.



Models	Accuracy (Crowdsourced)	Accuracy (Combined)
BinaryRelevance (GaussianNB)	0.1827	0.1397
BinaryRelevance (BernoulliNB)	0.3733	0.9292
Binary Relevance (MultinomialNB)	0.4533	0.9648
Chain Classifier (GaussianNB)	0.1800	0.1420
Chain Classifier (BernoulliNB)	0.4056	0.9226
Chain Classifier (MultinomialNB)	0.4517	0.9676
Label Powerset (GaussianNB)	0.3340	0.3853
Label Powerset (BernoulliNB)	0.4528	0.9755
Label Powerset (MultinomialNB)	0.5687	0.9764
MLkNN	0.3722	0.9762

Figure 14 Accuracy of the classifiers tested in [5]

Even though, in this research, 'Accuracy' was not considered as a suitable parameter for evaluation of classification task as discussed in Section 4.2, the accuracy values of best performing classifier set ups were calculated as shown in table 11 below, for comparison purposes.

Table 11 Accuracy scores of best performing classifier set ups

Classifier set up	Accuracy
OneVsRest/SVC/Count	0.9987
OneVsRest/ADA/TF_IDF	0.9986
LP/SVC/Count	0.9814
CC/SVC/Count	0.9821
CC/ADA/TF-IDF	0.9807

When accuracy figures are compared with the accuracy of combined data set in Figure 10, it can be seen that the classifier setups used in this research had been able to obtain higher accuracies (almost touching 100%) when compared to [5]. However as discussed earlier this measure could not be taken as a fair attribute to represent the system as it is dominated by large no. of true positive instances.

4.4 Summary

In this chapter the experiment set up used for the multi label classification task is introduced and results are presented from different aspects covering different approaches that were tried out. Also in absence of any published results for Cricket domain, an attempt was taken to compare results with research done outside Cricket using the same event extraction technique as ours. From the analysis it could be seen that better results could be achieved for Cricket domain when benchmarked with what is published for soccer commentaries and economic news domain even with relatively small data set. However there could be domain specific challenges affecting this comparison. Also it needs to be noted that the controlled conditions in which experiments were performed could be different to each other.

No dominant performances could be noted with few classifier set ups i.e. OnevsRest/ ADABoost/ TF_IDF Vectorizer, OnevsRest /Linear SVC/Count Vectorizer, LP/Linear SVC/Count Vectorizer, CC/DT/TF-IDF Vectorizer and CC/Linear SVC/Count Vectorizer performing almost at the same level in achieving high recall and precision values.

CHAPTER 5: CONCLUSION AND FUTURE WORK

5.1 Summary

With increasing competition in professional sports and commercial values associated with league tournaments around the world sports team have no other option but to turn in to sports analytics to keep up with the pace, make more informative decisions and then make correct moves at right time so that maximum benefit is obtained. This is applicable for player bidding at auctions, opposition player analysis, team selection and many more depending on the sport.

Much work had been done in terms of research in sports analytics however; work focusing on online available live commentaries is very thin especially for Cricket domain. For Cricket, like baseball, being a game which is dealt with numbers by nature with team scores, batsman averages, strike rates, economy rates of bowlers etc. analysis has always focused on number crunching. However there are many events that do not go in to the scoreboard of a match such as dropping a catch, which describes characteristics of players.

Out of few researches being done on sports commentary, there are research work focusing on producing structured data [11], [12], but they only use the events that come pre-tagged from website and do not focus on extracting additional information from commentary text. There are also researches which process commentary text however they do not focus on storing the information extracted in structured format [5], [13].

In this research an end to end system was proposed to produce structured data from online commentaries. This is implemented by scraping ball-by-ball commentaries from web, identifying the events and actors mentioned in a commentary text for each delivery and saving the commentary and derived information in a database delivery wise for each match. This data then could be queried for analytical purposes.

5.2 Future Work

This research was carried out with a relatively small set of data due to time and effort required in manually tagging sports commentaries by humans who are familiar with the game. With a larger set of tagged data the accuracy of classification could be further improved and also possibility of splitting already identified events in to sub classes can be considered for a more fine-grained event set. For an example an 'attempted yorker' and 'successful yorker' can be considered as two separate events.

The scope of this implementation was limited to Cricket domain and Cricbuzz website. With ground architecture now in place, this now have the possibility of extending to other websites and sports with appropriate changes in respective stages.

The algorithm used for event attribution to actors have limitations especially when a commentary sentence having multiple labels or multiple actors attached. The algorithm could fail by attributing an event to a non-relevant player. An approach using PoS tags could be considered as done in EEQuest [17] in which actors are identified first by using NER and then nearest consequent verb to that actor is identified as an event.

5.3 Conclusion

In this research an end to end system was proposed to extract structured data from online sports commentaries. While transforming data from semi structured format as obtained from web in to structured data, information is extracted from commentary sentences using a multi label classification approach. As it is shown in Chapter 4 Results and analysis, we were able to obtained good results for classification task when compared with other published results of multi label classification tasks done for event extraction from text. These results were obtained with relatively small data sets and it was observed that the classifier performed well even for classes with very few training data.

Both SVM and Ada Boost algorithms gave comparatively higher results when compared to Multinomial Naïve Bayes algorithm. It could be seen that no. of training instances had very light impact on the performances for a given class, rather when the class definition was broad (e.g. class Run-Out with stumpings included in the same class) or when the class event can be described in many ways lexically (e.g. Class misfield) the results had been low. This work can now be extended to other formats of Cricket, other commentary platforms as well as to other sports with required domain specific adjustments being made.

References

[1] S. P. C. V. J. K. Pramod Sankar, "Text Driven Temporal Segmentation of Cricket Videos," in *Computer Vision, Graphics and Image Processing*, S. P. Prem Kalra, Ed., Springer-Verlag Berlin Heidelberg, 2006, pp. 433-444.

[2] C.-W. L. T.-H. C. W. H. Liang-Chi Hsieh, "Live Semantic Sport Highlight Detection Based on Analyzing Tweets of Twitter," in *2012 IEEE International Conference on Multimedia and Expo*, Melbourne, VIC, Australia, 9-13 July 2012.

[3] J. G. B. Smitashree Choudhury, "Extracting Semantic Entities and Events from Sports Tweets," in *8th Extended Semantic Web Conference, ESWC2011*, Heraklion, Crete, 30 May 2011.

[4] E. L. a. V. H. Gilles Jacobs, "Economic event detection in company-specific news text," in *The 56th Annual Meeting of the Association for Computational Linguistics*, Melbourne, Australia, 2018.

[5] R. A. BHAGAT, "Towards Commentary-Driven Soccer Player Analytics," 2018.

[6] F. C. D. J. Lewis, "A fair method for resetting the target in interrupted one-day cricket matches," *Journal of the Operational Research Society,* vol. 49, no. 3, pp. 220-227, March 1998.

[7] G. H. W. William Elderton, "Cricket Scores and Some Skew Correlation Distributions: (An Arithmetical Study)," *Journal of the Royal Statistical Society,* vol. 108, pp. 1-11, 1945.

[8] T. B. Swartz, "Research Directions in Cricket," in *Handbook of Statistical Methods and Analyses in Sports*, 2016.

[9] A. &. A. H. M. H. &. M. R. Abdul Halin, "Event Detection in Soccer Videos through Text-based Localization and Audiovisual Analysis," *International Journal of Digital Content Technology and its Applications,* vol. 6, no. 15, pp. 164-170, August 2012.

[10] H. C. R. P. S. Osama K. Solieman, "Web Sports Data Extraction and Visualization," in *SPORTS DATA MINING*, Springer, 2012.

[11] D. M. J. S. M. Patil, "SEMANTIC INFORMATION RETRIEVAL USING ONTOLOGY AND SPARQL FOR CRICKET," *International Journal of*

Advances in Engineering & Technology, vol. 4, no. 2, pp. 354-363, September 2012.

[12] D. M. J. S. M. Patil, "Semantic Search using Ontology and RDBMS for Cricket," *International Journal of Computer Applications,* vol. 46, no. 14, pp. 26-31, May 2012.

[13] M. Gupta, "CricketLinking: Linking Event Mentions from Cricket Match Reports to Ball Entities in Commentaries," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SANTIAGO- CHILE, 2015.

[14] D. Ahn, "The stages of event extraction," in *ARTE '06 Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, 2006.

[15] M. R.-S. Chinatsu Aone, "REES: A Large-Scale Relation and Event Extraction System," in *ANLC '00 Proceedings of the sixth conference on Applied natural language processing*, 2000.

[16] F. F. F. K. U. a. D. J. F. Hogenboom, "An overview of event extraction from text," *Workshop on Detection, Representation, and Exploitation of Events in the Semantic Web (DeRiVE 2011) at Tenth International Semantic Web Conference (ISWC 2011),* vol. 779, pp. 48-57, October 2011.

[17] H. B. S. R. Prerit Jain, "EEQuest: An Event Extraction and Query System," in *COMPUTE '16 Proceedings of the 9th Annual ACM India Conference*, Gandhinagar, India, October 21 - 23, 2016.

[18] D. K. D. G. S. D. Gjorgji Madjarov, "An extensive experimental comparison of methods for multi-label learning," *Pattern Recognition,* vol. 45, no. 9, pp. 3084-3104, September 2012.

[19] V. H. Els Lefever, "A Classification-based Approach to Economic Event Detection in Dutch news text," in *Tenth International Conference on Language Resources and Evaluation (LREC '16)*, 2016.

[20] "@Alexa - An amazon.com company," [Online]. Available: https://www.alexa.com/siteinfo/. [Accessed 23 February 2019].

[21] S. P. T. R. M. Pratiksha Ashiwal, "Web Information Retrieval Using Python and BeautifulSoup," *International Journal for Research in Applied Science & Engineering,* vol. 4, no. 6, 2016.

[22] S. Bird, "NLTK: The natural language toolkit," in *21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, Sydney, Australia, 2006.

[23] International Cricket Council, "ICC Rules and Regulations," [Online]. Available: https://www.icc-cricket.com/about/cricket/rules-and-regulations/playing-conditions. [Accessed 24 02 2019].

[24] A. K. Ryan Rifkin, "In Defense of One-Vs-All Classification," *The Journal of Machine Learning Research archive,* vol. 5, pp. 101-141, 12/1/2004.

[25] F. Sebastiani, "Machine Learning in Automated Text Categorization," *ACM Computing Surveys (CSUR),* vol. 34, no. 1, pp. 1-47, March 2002.

[26] C. N. M. Vandana Korde, "Text classification and classifiers: A survey," *International Journal of Artificial Intelligence & Applications,* vol. 3, no. 2, p. 85, 2012.

[27] T. Joachims, "Text categorization with Support Vector Machines: Learning with many relevant features," in *European Conference on Machine Learning*, Chemnitz, Germany, 1998.

[28] X.-k. W. Yu-ping Qin, "Study on Multi-label Text Classification Based on SVM," in *FSKD 2009, Sixth International Conference on Fuzzy Systems and Knowledge Discovery*, Tianjin, China, 2009.

[29] C.-C. C. C.-J. L. C.-W. Hsu, "A Practical Guide to Support Vector Classification, Department of Computer Science, National Taiwan University," 2003. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/papers.html. [Accessed 25 February 2019].

[30] J. D. B. J. d. C. B. Oscar Luaces, "Binary relevance efficacy for multilabel classification," *Progress in Artificial Intelligence,* vol. 1, no. 4, p. 303–313, December 2012.