Soccer Key Event Extraction

Rahul Ashok Bhagat
Natural Language Processing
Texas A&M University
rahul.bhagat@tamu.edu

Rui Huang
Natural Language Processing
Texas A&M University
huangrh@cse.tamu.edu

Abstract

Open information extraction (open IE) has been shown to be useful in a number of NLP tasks, such as question answering, relation extraction, and information retrieval. Soccer, the most watched sport in the world, is a dynamic game where a teams success relies on both team strategy and individual player contributions. Sports events data is often compiled manually by companies who rarely make it available for free to third parties. However, social media provide us with large amounts of data that discuss these very same matches for free. We extracted and analyzed soccer commentaries of about 220 soccer matches from various leagues. From training the system with the data from these matches, we 1) classified the event as "Action" and "Not just Action", 2) classified the "Not just Action" events into further event types and 3) extracted the minutes in which those event occurs. Our results show that our approach performs well enough to fetch the key events of the match and summarize the commentaries to give a compact match summary.

1 Introduction

Soccer, the most watched sport in the world, is a dynamic game where a teams success relies on both team strategy and individual player contributions. Although soccer is by far the world's most popular sport, published work in soccer analytics has yet to achieve the same level of sophistication as analytics being performed in other professional sports. Crude summary statistics such as goals, shots, and assists are still the most common way to compare player performance analytically (Brooks et al., 2016). Many soccer fans try to keep track of their favorite teams by reading or watching game summaries. Generally, these summaries provide an overview of the minutes in which game highlights as goals, cards, and substitutions happen for both teams. This type of data is often created manually, a time-consuming and expensive process.Companies make good money selling these data to third parties (Van Oorschot et al., 2012). There is a growing trend among companies, organizations and individuals alike to gather information through web data mining to utilize that information in their best interest. Open information extraction (open IE) has been shown to be useful in a number of NLP tasks, such as question answering, relation extraction, and information retrieval. GOAL.com has 500 contributors in 50 countries producing content in more than 15 languages [1].This unparalleled global reach and in-depth coverage of the world's most popular game attracts more than 10 million passionate soccer enthusiasts from more than 220 countries. This is one of the reason for us to follow GOAL.com for a thorough understanding of soccer events. In this work, we build upon the live commentaries available for soccer matches on GOAL.com and present an approach to construct soccer match summaries from commentaries by detecting relevant events and event-minutes; and briefing the information presented in those events.

[1] http://www.goal.com/en/

2 Related Work

The majority of research in the field of soccer for event extration is based on drawing out highlights from audio and video contents. In (Rui et al., 2000), the authors explored the ability to extract highlights automatically using audio-track features alone. Audio keywords provides more intuitionistic result for event detection in sports video, specifically soccer videos, compared with the method of event detection directly based on low-level features (Xu et al., 2003). (Sacha et al., 2014) presented a system with integration of Visual Analytics techniques into the analysis process for high-frequency position-based soccer data at various levels of detail. Several work on game-related performance of the players and teams have also been presented. (Bojinov and Bornn, 2016) defines Passing as a cardinal soccer skill and utilizes this fundamental observation to define and learn a spatial map of each teams defensive weaknesses and strengths. The focus towards more sport-speciic metrics like player movement and their similarity to other players and uniqueness in terms of their in-game movements have been analyzed in (Gyarmati and Hefeeda, 2016). The research has also been done on the prediction of the outcome of soccer matches to used to bet on the winning team (van Wijk, 2012). Recent work (Sahami Shirazi et al., 2011) presented mobile application usage for real-time opinion sharing and used the collected data to exemplify the aggregated sentiments correspond to important moments, and hence can be used to generate a summary of the event. (Choudhury and Breslin, 2011) exploited various approaches to detect the named entities and significant micro-events from users tweets during a live sports event. People are already discussing the game on various social media platforms like Twitter, Facebook, Instagram etc. Realizing this, (Lanagan and Smeaton, 2011) set out to use Twitter data to mine tweets for highlight detection in soccer and rugby matches. They employed a fairly simple approach detecting 'interesting minutes' by looking at the peaks in the Twitter stream. Their results are comparable to highlight detection from audio and video signals, but still suffer from a high number of false positives.

The objective of text summarization is to save a prospective reader time and effort in finding useful information in a given article or report. In (Luhn, 1958), the author has used distribution and Statistical information derived from word frequency by the machine to compute a relative measure of significance. Sentences with high significance score expresses the overall summary better. LexRank (Erkan and Radev, 2004) computes sentence importance based on the concept of eigenvector centrality in a graph representation of sentences. A connectivity matrix based on intra-sentence cosine similarity is used as the adjacency matrix of the graph representation of sentences. TextRank (Mihalcea and Tarau, 2004) utilizes Graph Based ranking algorithms like HITS and PageRank to score the sentences in graph. The sentences are built into a sparse matrix of words and then similarity matrix is constructed between sentences using tf-idf scores of words of the sentence.

We aim to work towards event detection by utilizing the live commentaries from experts available on website like GOAL, BBC sport, SportMule and many more. We have restricted our model to commentaries from GOAL.com and applied various natural language processing classification algorithms like Naive Bayes and linear model based learning method. For text summarization, we utilized LexRank and TextRank and compared the results from both these algorithms.

3 Data

In this section, we detail the data collection and preprocessing steps. Our main focus for data collection was on live commentaries. We restricted our model to GOAL.com only as it is the most vibrant soccer community in the world that provides international soccer news, commentary, and entertainment through both internet and mobile platforms. Moreover, Goal reaches over 60 million football fans around the world every month. Goals 500+ strong editorial team deliver football expertise and unique insight to thousands of pieces of content every day, in a language and style to suit fans whoever and wherever they are [2]. Goal is available in 18 languages across 38 location-based editions, mobile and interactive TV apps plus social channels with

[2] http://www.performgroup.com/brands/goal/

more than 66m fans across Facebook, Twitter, Instagram, LINE, and YouTube.

3.1 Commentary Collection

There are many methods to scrap information from the Web (Mehlführer, 2009). Since barriers to prevent machine automation are not effective against humans, the most effective method is human copy-paste. Although sometimes this is the only way to export information from a Web page, this is not feasible in practice, especially for big company projects, being too expensive. Another method is text grepping in which regular expressions are used to find information that matches some patterns. Further Web scraping techniques are HTTP programming, DOM parsing, and HTML parsers. Finally, a Web scraping method consists of making scraper sites that are automatically generated from other Web pages by scraping their content (Penman et al., 2009). For our work, we employed HTML parser to scrap commentary from GOAL.com.

We used Beautiful Soup which is a Python library designed for quick screen-scraping. The matches are tagged with Match ID which consist of league Id and other team information. For example, match between Manchester United and Chelsea on 16th April,2017 was tagged as 2242081 where 224 is the league code for Barclay's English Premier League. Since most of the high profile matches occur over the weekends, we scrapped the data of the matches over weekend. Since GOAL.com provides with the feature of presenting all the fixtures over a particular date, we utilized this to get data for a particular historical match day.

Once, we started to scrap the data, we observed that for certain low profile matches like matches in a second division league or match between low ranked teams, the commentary was very abstract. Since the information content of these matches was very low, we decided to train our model selectively for those matches which had enough data content.

We stored all the commentaries corresponding to a match in a CSV file under UTF-8 encoding named with its corresponding match id. The commentaries were ordered according to the minutes. In order to understand the commentary language better, we did not restrict ourselves with any particular league. We scrapped for the commentaries of all the matches happening across multiple leagues - Major Soccer League(USA), La Liga -Primera Divisin (Spain), Barclays premier league - (England), Serie A (Italy), BundesLiga (Germany), Ligue 1 (France), UEFA Champions League, FA Cup and International matches.

3.2 Gold Standard

GOAL.com in its live commentaries also provide detail corresponding to the event type. For example, a substitution event would also have an event type along side the normal commentary. We used to store the event type along with the commentary and minute. For simplicity's sake, we only want each minute to belong to one class of event. There were 11 different event types - action , yellow-card , substitution , assist , goal , penalty-goal , red-card , own-goal , missed-penalty , penalty-save and yellow-red. In all, we collected data corresponding to all 11384 events. Of all these events, 8626 events were of 'action' type only, constituting 76% of the events. In order to deal with this biasing, we included 2 classifiers in our model. The first classifier would distinguish between 'action' and 'Not just action' events. The second classifier would work on 'Not just Action' events and further classify them in 10 different classes.

3.3 Data Preprocessing

We collected data for 850 matches from November 2016 to April 2017. Of these 850 matches, we selected only 184 matches according to the information content in these matches. For this, we kept a threshold check to the size of each file - File size less than 10KB were filtered out. These 184 matches had 11384 events.

3.3.1 Entity Collection

In order to support the language model better to our data set, the first step was to remove the entities like Person Name, Team name and Location from our data. For this, we employed Stanford Named Entity Recognizer (NER) to get the tagged data. Before training our model, we made a list of NER tagged data corresponding to each match and stored it in a Python dictionary. We employed Stanford NER functionality through Python Natural Lan-

guage Toolkit library [3].

3.3.2 Tokenization

We first tokenized the words using space as separator. After tokenization, we all words were transformed to lowercase. We removed all the entities from these tokens using entity collection as described in the previous step. Common words were removed using the modified stopword list from the Python Natural Language Toolkit.

3.3.3 Data Set Creation

We created 2 different sets of data set for 2 different classifiers. Both the data set would consist of information content of minute, event type, commentary and tokenized words. The first data set included data set for all the events. The second data set included events other than the 'action' event corresponding to 'Not just action' class. The data set was stored in DataFrame object of Pandas Python library [4]. DataFrame is a 2-dimensional labeled data structure like a spreadsheet or SQL table, or a dict of Series objects with columns of potentially different types.

4 Game Event Classification

After filtering out the low information content data and preprocessing the data, we want to know what kind of event it is. As mentioned in section 3.2, we distinguish between 11 types of events across 2 classifiers. The first classifier segregates the 'Not just action' event from 'action' events. It is a binary classifier. We have applied various algorithms for training these classifier and evaluated them based on their results.

4.1 'Action - Not just action' Classification

We initially implemented the Naive Bayes model for this classification. The Naive Bayes classifier is a simple probabilistic classifier which is based on Bayes theorem with strong and nave independence assumptions. It is one of the most basic text classification techniques with various applications in email spam detection, personal email sorting, document categorization, sexually explicit content detection, language detection and sentiment detection. Despite the nave design and oversimplified assumptions that this technique uses, Naive Bayes performs well in many complex real-world problems.The Naive Bayes classifier is a simple probabilistic classifier which is based on Bayes theorem with strong and nave independence assumptions. It is one of the most basic text classification techniques with various applications in email spam detection, personal email sorting, document categorization, sexually explicit content detection, language detection and sentiment detection. Despite the nave design and oversimplified assumptions that this technique uses, Naive Bayes performs well in many complex real-world problems.

We tested our results on various variations of naive bayes - Gaussian Naive Bayes, Multinomial Naive Bayes and Bernoulli Naive Bayes.

Gaussian Naive Bayes is useful when continuous values associated with each class are distributed according to a Gaussian distribution (John and Langley, 1995) (Salton and McGill, 1986). Since our model has discrete term frequency associated with each word, the classifier does not perform well with an accuracy of 72% Table 1. The average precision, recall and f1-score has also been reported to be low as presented in Table 2.

The Multinomial Naive Bayes variation esti-

Table 1: Evaluation Matrix for Binary Classifier using Gaussian NB in k-fold cross validation (k=10)

	Action	Not Just Action
Action	6060	2566
Not Just Action	564	2194

Table 2: Classification Report for Binary Classifier using Gaussian NB in k-fold cross validation (k=10)

	Precision	Recall	F1-Score
Action	0.46	0.80	0.58
Not Just Action	0.91	0.70	0.79
Avg/Total	0.80	0.73	0.74

[3] http://www.nltk.org/

[4] http://pandas.pydata.org/

Table 3: Evaluation Matrix for Binary Classifier using Multinomial NB in k-fold cross validation (k=10)

	Action	Not Just Action
Action	8488	138
Not Just Action	410	2348

Table 4: Classification Report for Binary Classifier using Multinomial NB in k-fold cross validation (k=10)

	Precision	Recall	F1-Score
Action	0.94	0.85	0.90
Not Just Action	0.95	0.98	0.97
Avg/Total	0.95	0.95	0.95

mates the conditional probability of a particular word/term/token given a class as the relative frequency of term t in documents belonging to class c (McCallum et al., 1998) (Salton and McGill, 1986). The probability of a document d being in class c is computed as

$$P(c|d) \propto P(c) \prod_{1 \leq k \leq n_d} P(t_k|c)$$

where $P(t_k|c)$ is the conditional probability of term t_k occurring in a document of class c.We interpret $P(t_k|c)$ as a measure of how much evidence t_k contributes that c is the correct class. $P(c)$ is the prior probability of a document occurring in class c. If a document's terms do not provide clear evidence for one class versus another, we choose the one that has a higher prior probability. $\langle t_1, t_2, \ldots, t_{n_d} \rangle$ are the tokens in d that are part of the vocabulary we use for classification and n_d is the number of such tokens in d (Salton and McGill, 1986). The results using Multinomial Naive Bayes variation are better and are expected as this is more suited for discrete word count (term frequency). The accuracy for Multinomial Naive Bayes is 95%. The confusion matrix and the classification reports are given in Table 3 and Table 4 respectively.

An alternative to the multinomial model is the multivariate Bernoulli model or Bernoulli model . It is equivalent to the binary independence model which generates an indicator for each term of the vocabulary, either 1 indicating presence of the term in the document or 0 indicating absence.The different generation models imply different estimation strategies and different classification rules. The Bernoulli model estimates $\hat{P}(t_k|c)$ as the fraction of documents of class c that contain term t. In contrast, the multinomial model estimates $\hat{P}(t_k|c)$ as the fraction of tokens or fraction of positions in documents of class c that contain term t . The accuracy of Bernoulli variation is quite close to Bernoulli Model for binary Classifier - 94.3%. The confusion matrix and the classification reports are given in Table 5 and Table 6 respectively.

Table 5: Evaluation Matrix for Binary Classifier using Bernoulli NB in k-fold cross validation (k=10)

	Action	Not Just Action
Action	8418	208
Not Just Action	435	2323

Table 6: Classification Report for Binary Classifier using Bernoulli NB in k-fold cross validation (k=10)

	Precision	Recall	F1-Score
Action	0.92	0.84	0.88
Not Just Action	0.95	0.98	0.96
Avg/Total	0.94	0.94	0.94

Later on we moved towards Multinomial logistic regression model, also known as Maximum Entropy model for our classification. Logistic regression belongs to the family of classifiers known as the exponential or log-linear classifiers. Like naive Bayes, it log-linear classifier works by extracting some set of weighted features from the input, taking logs, and combining them linearly (meaning that each feature is multiplied by a weight and then added up). Technically, logistic regression refers to a classifier that classifies an observation into one of two classes, and multinomial logistic regression is used when classifying into more than two classes (Jurafsky, 2000).The most important difference between naive Bayes and logistic regression is that logistic regression is a discriminative classifier while naive Bayes is a generative classifier. A discrimina-

Table 7: Evaluation Matrix for Binary Classifier using Linear Model - Logistic Regression in k-fold cross validation (k=6)

	Action	Not Just Action
Action	8544	842
Not Just Action	170	2588

Table 8: Classification Report for Binary Classifier using Linear Model - Logistic Regression in k-fold cross validation (k=6)

	Precision	Recall	F1-Score
Action	0.97	0.92	0.94
Not Just Action	0.98	0.99	0.98
Avg/Total	0.97	0.97	0.97

tive model discriminative model takes this direct approach, computing $P(y|x)$ by discriminating among the different possible values of the class y rather than first computing a likelihood:

$$\hat{y} = \arg\max_y P(y|x)$$

We implemented this model using scikit-learn python package [5].We used SAG (Stochastic average gradient) method as the optimization method to find the optimum of the objective function. Also, regularization strength set to 10 gave the best results. Only numerical feature were extracted using the CountVectorizer method from scikit-learn python package. Vectorization is the general process of turning a collection of text documents into numerical feature vectors. CountVectorizer implements both tokenization and occurrence counting in a single class. The number of features used were 9264. The accuracy of this model is better than all other Naive Bayes variations described above - 97.7%. The confusion matrix and the classification reports are given in Table 7 and Table 8 respectively.

4.2 Event Classification

The binary 'Action - Not just Action' classifier performed well and it was indeed needed to get a better accuracy for this classification as our main focus is towards classifying the 'Not just action' i.e, relevant events further into 10 different classes - action , yellow-card , substitution , assist , goal , penalty-goal , red-card , own-goal , missed-penalty , penalty-save and yellow-red. For this classifier too, we applied both the approaches of Naive Bayes model and MaxEnt Model. For MaxEnt model, we used the same optimization method and regularization strength as defined in 4.1. The number of features used were 4324. The comparison of evaluation metrics among different models and their variations is shown in Table 9. The evaluation metric for MaxEnt based Logistic Regression model is shown in Table 10.

[5]https://pypi.python.org/pypi/scikit-learn

Table 9: Accuracy of different classifier models for Event Classifications

	Precision	Recall	F1-Score	Accuracy
Naive Bayes				
Gaussian	0.84	0.84	0.83	0.83
Bernoulli	0.94	0.96	0.95	0.95
Multinomial	0.95	0.96	0.95	0.95
MaxEnt				
Log. Reg.	0.98	0.98	0.98	0.98

4.3 Combined Classification

Earlier, we evaluated the 'Action - Not just action' classification and Event classification independently. Now, the idea is to combine these two classifications and evaluate them. For the combined classification, we evaluated both the classifier trained with the single model as well as combination of models like Gaussian Gaussian Naive Bayes, Multinomial - Multinomial Naive Bayes, Gaussian - Multinomial Naive Bayes, Multinomial - Bernoulli Naive Bayes, Multinomial Naive Bayes - Gaussian Naive Bayes.

The MaxEnt model based logistic regression variant worked best in both the classifications and also, it has better results as compared to combination of models too. The overall accuracy for the combined classification using this model was 97.4%. The confusion matrix is shown in Table 11.

Table 10: Evaluation Matrix for Event Classifier using Linear Model - Logistic Regression in k-fold cross validation (k=10)

	Substitution	Yellow Card	Assist	Goal	Penalty-goal	Own-goal	Yellow-red	Red-card	Penalty-save	Missed-penalty
Substitution	1045	0	0	0	0	0	0	0	0	0
Yellow card	0	721	1	3	0	0	0	0	0	0
Goal	0	0	509	4	0	0	0	0	0	0
Assist	0	5	7	381	0	0	0	0	0	0
Penalty - goal	0	0	9	0	18	0	0	0	0	0
Own-goal	0	0	10	0	0	6	0	0	0	0
Yellow-red	0	6	1	0	0	0	8	0	0	0
Red-card	0	1	2	0	0	0	2	3	0	0
Penalty-save	0	1	0	0	0	0	0	0	5	0
Missed-penalty	0	0	1	0	3	0	0	0	0	6

Table 11: Evaluation Matrix for Combined Classifier using Linear Model - Logistic Regression in k-fold cross validation (k=10)

	Action	Substitution	Yellow Card	Assist	Goal	Penalty-goal	Own-goal	Yellow-red	Red-card	Penalty-save	Missed-penalty
Action	8570	0	16	35	4	0	1	0	0	0	0
Substitution	0	1045	0	0	0	0	0	0	0	0	0
Yellow card	30	0	695	0	0	0	0	0	0	0	0
Goal	103	0	0	408	2	0	0	0	0	0	0
Assist	11	0	0	0	382	0	0	0	0	0	0
Penalty - goal	4	0	0	11	0	12	0	0	0	0	0
Own-goal	13	0	0	2	0	0	1	0	0	0	0
Yellow-red	3	0	8	0	0	0	0	4	0	0	0
Red-card	4	0	0	0	0	0	0	0	4	0	0
Penalty-save	6	0	0	0	0	0	0	0	0	0	0
Missed-penalty	7	0	0	0	0	1	0	0	0	0	2

5 Commentary Summarization

For commentary summarization, we are using text summarization from Natural Language Processing. Text summarization is the process of reducing a text document in order to create a summary that retains the most important points of the original document. The research about text summarization is very active and during the last years many summarization algorithms have been proposed. A lot of online text summarizer are available nowadays like Text Summarizer [6], Text Summarization [7], t-CONSPECTUS [8] and many more.

We used the Python package Sumy [9] to perform text summarization. It has various algorithms included in the package like LexRank (Erkan and Radev, 2004), TextRank (Mihalcea and Tarau, 2004), Latent Semantic Analysis, Sum Basic and many more. We evaluated the results of TextRank and LexRank on our commentary dataset and we found that both the algorithm are unique in its way and works best on our dataset. While LexRank uses cosine similarity of TF-IDF vectors, TextRank uses a very similar measure based on the number of words two sentences have in common (normalized by the sentences' lengths).LexRank in (Erkan and Radev, 2004) has been applied to multi-document summarization. LexRank applies a heuristic post-processing step that builds up a summary by adding sentences in rank order, but discards any sentences that are too similar to ones already placed in the summary. Since, commentary at a single minute contains very similar sentences, LexRank returns the first sentence summarizing the commentary and TextRank returns the sentence ranked top. Here are a few example showing the difference in results of these two algorithms :

Commentary 1: *Goal Romelu Lukaku Menama. GOOOOOOOOOAL! LUKAKU MAKES IT 3-1! What a strike from the Belgium international, who turns smartly beyond Keane in the box and then holds his man off. He stabs the ball home with incredible force, Heaton didn't have a chance.*

[6] http://textsummarization.net/text-summarizer

[7] http://textsummarization.net/

[8] http://tconspectus.pythonanywhere.com/sumysum

[9] https://github.com/miso-belica/sumy

TextRank: *What a strike from the Belgium international who turns smartly beyond Keane in the box and then holds his man off.*
LexRank: *Goal Romelu Lukaku Menama.*
Commentary 2:*Substitution sub-out Idrissa Gana Gueye sub-in Enner Remberto Valencia Lastra . Valencia replaces Gueye, as Koeman looks to take the game to Burnley a little. Barkley moves infield as a result.*
TextRank:*Valencia replaces Gueye as Koeman looks to take the game to Burnley a little.*
LexRank:*Substitution sub-out Idrissa Gana Gueye sub-in Enner Remberto Valencia Lastra .*

6 Discussion and Future Work

In this contribution, we presented an approach and results for detecting the most important events occurring in soccer games through mining soccer commentary. We take a three steps approach in which we first try to identify the relevant events using 'action' and 'Not just Action' binary classifier, then we further classify the data into 10 different event types according to the commentary per minute and finally we summarize the commentary of the 'Not just Action' events to produce the overall match summary. We encountered various difficulties related to encoding while developing our model. Since we included a wide range of matches in our dataset, it included players from different parts of world and it was quite challenging to deal with names of certain players like 'Zlatan Ibrahimovi', Mesut zil, Kak etc. Our approach does have its drawbacks, for example in less popular games there were too few information content available to reliably detect the interesting event minutes. Moreover, it fails to take into account the other event types like Saves, Chances and Blocks. Since Goal.com does not provide with such event types, if manually tagged data set is available for these event types, the classifier can be trained to detect these events too. Additionally, from the results of combined classifier, we find that the classification of Penalty-save, Missed Penalty and Red-card are very skewed. The reason for this inconsistency is scarcity of data for these event types.
In future work, we will focus towards mitigating these problems by integrating data from different sources as well to account for data sparsity issues for low information content matches and certain rarely occurring event types and train the classifier for more event types like 'Saves', 'Blocks' and 'Chances' by utilizing manually tagged data or commentary data from other social media source like Twitter. We can also extend our work towards evaluating the player performance by analyzing the involvement of a player in different event types.

Acknowledgments

We would like to thank Dr.Rui Huang for his suggestions and guidance in this work.

Do not number the acknowledgment section.

References

Iavor Bojinov and Luke Bornn. 2016. The pressing game: Optimal defensive disruption in soccer.

Joel Brooks, Matthew Kerr, and John Guttag. 2016. Developing a data-driven player ranking in soccer using predictive model weights. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 49–55. ACM.

Smitashree Choudhury and John G Breslin. 2011. Extracting semantic entities and events from sports tweets.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

László Gyarmati and Mohamed Hefeeda. 2016. Analyzing in-game movements of soccer players at scale. *arXiv preprint arXiv:1603.05583*.

George H John and Pat Langley. 1995. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pages 338–345. Morgan Kaufmann Publishers Inc.

Dan Jurafsky. 2000. *Speech & language processing*. Pearson Education India.

James Lanagan and Alan F Smeaton. 2011. Using twitter to detect and tag important events in live sports. *Artificial Intelligence*, 29(2):542–545.

Hans Peter Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165.

Andrew McCallum, Kamal Nigam, et al. 1998. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Citeseer.

Andreas Mehlführer. 2009. *Web scraping: A tool evaluation*. na.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into texts. Association for Computational Linguistics.

Richard Baron Penman, Timothy Baldwin, and David Martinez. 2009. Web scraping made simple with sitescraper.

Yong Rui, Anoop Gupta, and Alex Acero. 2000. Automatically extracting highlights for tv baseball programs. In *Proceedings of the eighth ACM international conference on Multimedia*, pages 105–115. ACM.

Dominik Sacha, Manuel Stein, Tobias Schreck, Daniel A Keim, Oliver Deussen, et al. 2014. Feature-driven visual analytics of soccer data. In *Visual Analytics Science and Technology (VAST), 2014 IEEE Conference on*, pages 13–22. IEEE.

Alireza Sahami Shirazi, Michael Rohs, Robert Schleicher, Sven Kratz, Alexander Müller, and Albrecht Schmidt. 2011. Real-time nonverbal opinion sharing through mobile phones during sports events. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 307–310. ACM.

Gerard Salton and Michael J McGill. 1986. Introduction to modern information retrieval.

Guido Van Oorschot, Marieke Van Erp, and Chris Dijkshoorn. 2012. Automatic extraction of soccer game events from twitter. *Proceedings of the Workhop on Detection, Representation, and Exploitation of Events in the Semantic Web (DeRiVE 2012)*, 902:21–30.

Nivard van Wijk. 2012. Soccer analytics.

Min Xu, Namunu C Maddage, Changsheng Xu, Mohan Kankanhalli, and Qi Tian. 2003. Creating audio keywords for event detection in soccer video. In *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*, volume 2, pages II–281. IEEE.