University of Dublin

TRINITY COLLEGE

#ITK #DoneDeal #FakeNews: An Examination of “Fake News” in the context of Football-related social media content

Jibin Xavier

Master in Computer Science

Dissertation

Supervisor: Dr. Séamus Lawless

School of Computer Science and Statistics

O’Reilly Institute, Trinity College, Dublin 2, Ireland

Submitted to the University of Dublin, Trinity College, May 2018

DECLARATION

I, Jibin Xavier, declare that the following dissertation, except where otherwise stated, is entirely my own work; that it has not previously been submitted as an exercise for a degree, either in Trinity College Dublin, or in any other University; and that the library may lend or copy it or any part thereof on request

Signature: _____

Date: _____

Summary

The Summer Transfer Window is a busy time for Football clubs, supporters and the media. Clubs are continually linked with signing and releasing players; rumours circulate that clubs are interested in particular players and making moves to recruit them. A very small fraction of these rumours turns out to be accurate.

This project investigates Twitter accounts that publish rumours about player transfers in the English Premier League. The goal of the project is to use Natural Language Processing, Statistical Analysis and Machine Learning techniques on the dataset to identify patterns in the tweets and to perform an analysis of the performance of certain accounts in predicting transfers.

The proposed system first, extracts tweets from Twitter accounts that claim to be “football journalists” and "In The Know" from May 5th, 2017 to September 5th 2017. Followed by a two-stage process: first identify if a tweet a transfer rumour or not using a rule-based system that uses heuristics such as keywords to identify a transfer rumour, then verifying the veracity of the rumour. The veracity of a rumour was examined using another rule-based system which uses known facts such as completed transfers in 2017 Summer Transfer Window. The next step entails testing the accuracy of accounts and examine if there are any links such as sentiment, number of retweets and favourites to the accuracy of the accounts.

The dissertation concludes that the false rumours tend to have higher retweets and favourites than true rumours. This phenomenon has also been identified by (Vosoughi, et al., 2018) that false information propagates quicker than true information in Twitter. However, contrary to the expectation sentiment proved not influential as the tweets studied tend to have less emotive words and often are a neutral statement

SVM and Random Forest machine learning algorithms were implemented to check the veracity of a tweet, however, performed poorly mainly due to the imbalanced set and weakly informative features sets. Further feature engineering work is needed to be done to improve the performance of the algorithms.

#ITK #DoneDeal #FakeNews: An Examination of “Fake News” in the context of Football-related social media content

Jibin Xavier

Supervisor: Dr. Séamus Lawless

Abstract

Fake news in social media has been a huge problem in recent times, controversies surrounding the 2016 presidential election is just an example. This is no different in football transfers which is the core concern for this project. A transfer in football is when a player moves from one club to another, and there are constant rumours put out by Twitter accounts who claim to be close to the source

This dissertation proposes an approach to test the accuracy of Twitter accounts based on their predictions and analyse patterns related to the accuracy of an account. The proposed system uses two rule-based systems for this task, one for detecting a transfer rumour using heuristics such as keywords, and rumour veracity checking using *named entity recognition* and *entity linking*.

SVM and Random Forest algorithms were implemented to automatically capture latent features that make up a true rumour and false rumour which in turn can be used to predict veracity of new a rumour.

The dissertation concludes that false rumours tend to have higher retweets and favourites than true rumours and features such as tweet text, number of favourites and retweets, and sentiment are not informative enough for both the machine learning algorithms as they performed poorly.

Acknowledgements

Firstly, I would like to wholeheartedly thank my supervisor Dr. Séamus Lawless,

for all his help and guidance throughout the year. I would also like to thank Gary Munnelly and Yu Xu for their support throughout the course of this dissertation.

I would also like to thank my family for their continual support and friends who helped me throughout this year.

Contents

Declaration of Authorship i

Summary ii

Acknowledgements iii

Abstract iv

List of Figures ix

List of Tables x

1. Introduction 12

1.1 Motivation 12

1.2 Research Question 14

1.3 Research Aims 14

1.4 Research Challenges 16

1.5 Dissertation outline 17

2. State of the Art 18

2.1 Rumour/Fake news detection 18

2.1.1 Characterisation 19

2.1.2 Detection 21

2.1.1 Rumour classification 21

2.1.2 Football whispers 22

2.2 Data Processing Pipeline 23

2.2.1 Named Entity Recognition 24

2.2.2 Entity Linking 25

2.3 Sentiment Analysis 26

2.3.1 Comparison between Google Cloud NLP, Stanford Core NLP and VADER 27
2.4 Language Modelling 28
2.4.1 Bag of words 28
2.4.2 Word embeddings 30
2.5 Learning Algorithms 31
2.5.1 SVM 31
2.5.2 Random Forest 33
2.5.3 Unbalanced data 34
2.6 Conclusion 35
3. Design and Methodology 35
3.1 Building Corpus 36
3.1.2 Knowledge Base Building 36
3.2 System design 40
3.2.1 Data Harvesting 41
3.2.1 Rumour Labels 44
3.2.2 Transfer Rumour Detection 46
3.2.3 Rumour Veracity Checking 47
3.2.4 Sentiment Analysis 49
3.2.6 Machine Learning – pre-processing 50
3.3 Ethical Implications 51
3.4 Summary 51
4. Implementation 52
4.1 Data collection 52
4.2 Annotation Phase 53
4.2.1 Rumour Detection 53
4.2.2 Rumour Veracity Checking 55

4.2.3 Sentiment analysis 59
4.3 Modelling Phase 60
4.3.1 Sklearn 60
4.3.2 Sklearn Pipeline 62
4.4 Tools 65
4.4.1 Pip 65
4.4 Challenges encountered 65
5. Evaluation 67
5.1 Approach 67
5.1.1 General Metrics 67
5.1.2 Component Evaluation 67
5.1 General Statistics 68
5.1.1 Account level Accuracy 69
5.1.2 Follower Count 70
5.1.3 Favourites and Retweets 71
5.1.4 Influence of Sentiment 72
5.2 Performance of the components 76
5.2.1 Rumour Detection 76
5.2.2 Rumour Veracity Checking 78
5.2.3 Machine Learning component 78
5.3 Summary of results 80
5.3 Limitations and Future work 81
5.3.1 Fine Grain rumour definition 81
5.3.2 Baselines and evaluations 82
5.3.3 Improving the named entity recognition 83
5.2.4 Use of Multiple sources 83

5.3.5 Larger Knowledge Base 84
5.3.6 Scaling and Parallelising 84
5.3.7 Better feature sets for machine learning 84
5.3.8 Other Rumour Detection techniques from State of the art 86
6. Conclusion 88
7. Abbreviations 89
Appendix A 89
Appendix B 89
Appendix C 90
Bibliography 92

List of Figures

Figure 1 Characterising rumours to detection techniques (Shu, et al., 2017) 18
Figure 3 Linear separable SVM (Sun, et al., 2009) 32
Figure 4 Classification in Random Forest 33
Figure 5 Outline of overall system 36
Figure 6 Example of an entry in the knowledge base 37
Figure 7 System design 40
Figure 8 Rumour Labels decision tree 44
Figure 9 Rumour tweet which came out to be true 48
Figure 10 Pre-processing stages 50
Figure 11 Transfer labelling process 54
Figure 12 Entity Linking process 56
Figure 13 Entity linking function 57
Figure 14 Pipeline steps 61
Figure 15 Feature Union 63
Figure 16 Class weighting formula 64
Figure 17 Formulae for precision and recall 68
Figure 18 F1 score formula 68
Figure 19 Accuracy of accounts 69
Figure 20 Follower count for top 12 accounts – sorted by accuracy 70
Figure 21 Followers per account – sorted by accuracy 70
Figure 22 Number of favourites for top 5 accounts 71
Figure 23 Distribution of number retweets 72
Figure 24 Distribution of sentiment scores of true and false rumours for GCloud and VADER. The top left shows the distribution of sentiment scores for False rumours using VADER sentiment analyser and top right sentiment scores for True rumours. Similarly for GCNLP sentiment analyser false and true rumour sentiment scores distribution. 74

List of Tables

Table 1 Bag of words representation 29
Table 2 Player name variations 38
Table 3 Twitter Accounts studied in this research 42
Table 4 Non-transfer features 55
Table 5 Transfer features 55
Table 6 Previous approaches to entity linking 59
Table 7 Performance of rumour detection component 76
Table 8 Performance of Rumour Veracity Checking- rule-based system 78
Table 9 Scores for Random Forest unigram model 79
Table 10 Scores for SVM unigram model 79
Table 11: Good textual indicators for rumour veracity checking 79
Table 12 Common words in true rumours and false rumours 80
Table 13 Scores for Random Forest Bigram model 89
Table 14 Scores for Random Forest Trigram model 89
Table 15 Scores for SVM Bigram model 90
Table 16 Scores for SVM Trigram model 90

1. Introduction

1.1 Motivation

For more than a century, information was largely disseminated through independent news media which include newspapers and broadcasters, who are heavily regulated. In countries like the UK, they are bound by law to challenge views expressed in their platforms. As a result; they often acted as a gatekeeper which restricted the spread of misinformation.

However, there have been many instances where inaccurate information was reported in newspapers. From controversial reporting, such as The Sun's coverage of the Hillsborough disaster, where it included quotes from authority figures falsely suggesting Liverpool fans "picked the pockets of victims," "urinated on cops", and beat up policemen, among other inaccurate claims.[1] To relatively less controversial misquoting of company names[2] in articles. Nonetheless, typically there are correction or retraction pages published which describe the inaccuracies of the report[3].

With the advent of social media, an incredible amount of people are now consuming news online. More than two-thirds of Americans in this study[4] reported consuming news through social media. In recent times there has been an upsurge of so-called "fake news" which is essentially inaccurate reports about entities that are typically used for either political or economic gain. This misrepresentation of information is especially prevalent in social media. One such example, is a recent article by (Lion Gu, et al., 2017) from cybersecurity firm Trend Micro, where they show how easy it is to sway voter's opinion, is just an example.

---

[1] https://www.theguardian.com/media/2004/jul/07/pressandpublishing.football1 Accessed on 01/05/2018

[2] https://www.theguardian.com/uk-news/2015/jan/27/rail-passenger-satisfaction-falling-before-london-christmas-chaos Accessed on 01/05/2018

[3] https://www.thesun.co.uk/archives/news/919113/we-are-sorry-for-our-gravest-error/ Accessed on 01/05/2018

[4] http://www.journalism.org/2017/09/07/news-use-across-social-media-platforms-2017/ Accessed on 01/05/2018

Football, and specifically football transfers which are the core concern for this project, represent a microcosm of this world. A transfer in football is when a player moves from one club to another. The publication and spread of rumours related to football transfers is so common, it has been normalised. For instance, rumours about players can be used strategically by agents or clubs to unsettle the player[5][6]. Alternatively, a rumour could be used by clubs to artificially inflate a player's price or to keep their fans interested. There are so-called social media pundits that are dedicated to football transfer information.

Another motivating factor in spreading fake news is that the English Premier League is one of the most followed and richest football leagues in the world. Ths is evident in demand for broadcasting rights totalling up to $4.5 billion (for term 2017/19) alone[7]. In such a lucrative industry, offering insights which can be used as leverage, or to gain access to some of this wealth, which promises rich finacial rewards. The sheer level of interest in being one step ahead of the game allows any old social media "analyst" to charge significant amounts for their reports[8]

The Summer Transfer Window is a busy time for Football clubs, supporters and the media. This is reflected in the volume of activity on social media platforms such as Twitter. Twitter is often source of rumours; Twitter content is regularly referenced in newspaper articles, especially about football transfers. Clubs are continually linked with signing and releasing players; rumours circulate that clubs are interested in particular players and making moves to recruit them. A very small fraction of these rumours turn out to be accurate. There are lots of Twitter accounts which claim to be #ITK - "In The Know" (people who claim they are close to sources), is this actually the case?

In this project, the Twitter accounts that will be investigated are mostly sports journalists and those who claim to be "In The Know". Often, the intention is to create buzz around their accounts during the transfer window, in effect want to increase their reputation and to become a known source of information. This project explores the use of Natural

---

[5] http://www.skysports.com/football/news/11668/2272105/lampard-rumours-are-unsettling-players Accessed on 05/05/2018

[6] https://www.huffingtonpost.co.uk/jamie-spencer/football-agents_b_10336946.html Accessed on 05/05/2018

[7] https://www.forbes.com/sites/mikeozanian/2017/06/06/the-worlds-most-valuable-soccer-teams-2017/#1de30a4577ea Accessed on 05/05/2018

[8] https://crazysandra.wordpress.com/contact/ Accessed on 05/05/2018

Language Processing (NLP) and machine learning (ML) to identify patterns and analyse the performance of individual Twitter accounts in predicting the rumours linked to football player transfers. This is the core concern of this project.

1.2 Research Question

The primary research question which will be investigated as part of this dissertation is:

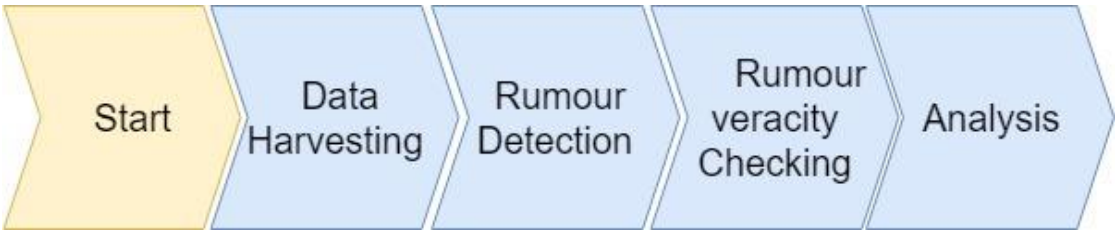
To what extent, can NLP techniques and statistical analysis be used to verify the accuracy of selected Twitter accounts in predicting football transfers?

To answer this research question, we will try to identify if there are particular patterns in tweets which can be exploited, and if NLP techniques such as Named Entity Extraction and, Sentiment Analysis can be used to support and improve the process.

A secondary research question posed by this dissertation is:

Can machine learning algorithms automatically capture latent features of tweets which can be used to predict the veracity of new tweet?

1.3 Research Aims



The primary aim of this research is to develop a method to determine how accurate certain Twitter accounts are at predicting potential football transfers. To do this, a collection of tweets has to be harvested. Following this, would involve to identify tweets that a) contain transfer information and b) contain a transfer rumour. Then using NER and entity-linking to determine the veracity of a said transfer.

Using these systems, we can further infer statistical information about the Twitter accounts in question. Then use machine learning algorithms on the annotated data from veracity checking, so that can identify and learn from latent features in the tweets and be used to predict the veracity of rumours contained in previously unseen tweets. Finally, we evaluate the performance of the rule-based systems by manually verifying a sample of the labelled dataset where the rumour detection has precision of 0.91 and rumour verification 0.98.

Scope

Rumours do not always have a clear distinction there are many granular details as outlined in the State-of-art chapter. For the purposes of this study, we focus on a deliberately narrow definition of rumour which will be defined chapter 3.2.1. This allows us to complete the necessary analysis to answer the posed research question within the time-frames allowed.

Football is a global game, played across the world. Player transfers commonly occur between professional leagues worldwide, with players moving to leagues in other countries. Furthermore, other leagues within Europe also have their transfer window at the same time as the EPL. Hence, an exhaustive study of rumours in the context of football transfers, would require these other leagues to be also considered. However, in order to make this research challenge tractable, within the given timeframe, this study focuses solely on the transfer of players to clubs in the English Premier League (EPL)

This study only works with English-speaking accounts, preforming rumour detection and verification in multilingual contexts would form a critical part of future work.

We have deliberately narrowed the definition of a rumour to that which can be verified based on the transfers listed as occurring on the official EPL transfers list. In reality, rumours are very nuanced. A story may have been posted in good faith and may in fact have been true at a particular point in time. For example, a player may have been in talks with a club during the transfer window, and close to signing a contract, without the transfer ever being completed. However, for the purposes of this research there is no way to verify this, and as such, it is deemed out of scope for this project

Important features like the age of the player, financial capital of the clubs, if the player in question will fit the squad, what sort of players are in the squad the proportion of native and non-native players that must be maintained, and many more factors cab contribute to defining the likelihood of a transfer. However, modelling this complexity is an entirely separate research challenge which would require resources far beyond the scope of this project. Hence using these features as inputs to the rumour veracity estimation process, is deemed to be out of scope.

1.4 Research Challenges

When attempting to answer the research questions, specified in section 1.2 numerous challenges were encountered.

- Issues due to limit set by Twitter on how many tweets can be extracted.
- Tweaking the rule-based system: Specifically, heuristics used in the system such as keywords used to determine the class were difficult as some can be common in both classes. Consequently, multiple iterations of the labelling process were required to fine-tune heuristics
- Twitter data is quite informal and conversational. As a result, one tweet by itself might not be useful to identify players or clubs.
- Journalists may have tweeted about their personal life or current affairs mixed with tweets about football. Identifying the distinction between football-related tweets, and non-football tweets is crucial for credible analysis
- Furthermore, the text is short and often lacks context to determine whom it is talking about without the aid of background knowledge. For example, clubs can be referred to by many aliases, including hashtags. Therefore, a knowledge base of club aliases was used to reduce the impact of this problem
- Evaluating the performance of the systems was an extremely time-consuming process. Moreover, the gold-standard dataset was labelled by me and may include elements of my own, personal biases.

1.5 Dissertation outline

Chapter 2 discusses the relevant background work for this dissertation in rumour detection, named entity recognition, sentiment analysis and machine learning techniques

Chapter 3 builds on the examination of relevant work and outlines the design of the system and the design decisions

Chapter 4 describes the implementation details of the proposed system.

Chapter 5 outlines the evaluation process and presents the results. This section also reviews the limitations and highlights the implementation issues and future work leading on from this project

Chapter 6 draws conclusions on the results in light of the research objectives and final thoughts on the results of this dissertation

2. State of the Art

This chapter describes the background to this project and introduces related research. The project has identified four essential topics to research as "state for the art", namely: Rumour detection, Named Entity Recognition, Sentiment Analysis, and Language Modelling for machine learning. These topics play a vital role in this research.

2.1 Rumour/Fake news detection

Social media is often the root of many news stories. Social media platforms provide an easy and cheap medium to disseminate information. Traditional media platforms like newspapers, radio and television, are bound by the law to ensure the content they provide must be checked and challenged[9]. However, for social media, there is no such regulatory oversight body, and it is left up to the discretion of the platform provider to remove false content. As a result, there is a significant volume of false content broadcast unchecked on these platforms.

This problem has been around for a long time, however, owing to the current political climate, and the controversy surrounding the US election in 2016, it has garnered tremendous interest in the literature, and as a result various approaches to identifying fake news have been explored. The following section outlines the different approaches found in the literature to detect false information.

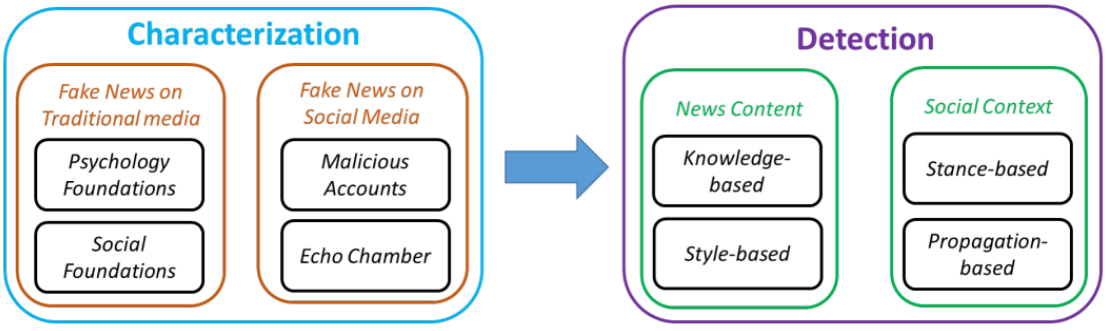


Figure 1 Characterising rumours to detection techniques (Shu, et al., 2017)

[9] https://www.theguardian.com/environment/2018/apr/09/bbc-radio-4-broke-impartiality-rules-in-nigel-lawson-climate-change-interview

Figure 1 outlines the main area of focus that has been addressed in the literature, summarised by (Shu, et al., 2017). The first task is to find the features that make a rumour, followed by a detection phase.

2.1.1 Characterisation

Studies concerned with rumours in social media, often begin by defining what constitutes as a rumour. (Zhao, et al., 2015) (Zubiaga, et al., 2018). Some are defined from social psychology literature, as in the case of (Qazvinian, et al., 2011):

"*a rumour is defined as a statement whose truth-value is unverifiable or deliberately false".*

(Zubiaga, et al., 2018) base their definition on major dictionary definitions. The Oxford dictionary describes a rumour as:

"A currently circulating story or report of uncertain or doubtful truths. " [10]

If there is a tweet about a potential player transfer to a club, there is no way to verify the veracity of this rumour before an actual signing takes place. Hence, this research is based on (Qazvinian, et al., 2011) definition and has formulated in the context of transfer rumours. A rumourous tweet about a player moving to another club which is not known at the time when it was tweeted.

2.1.1.1 Fake news on Traditional News Media

The psychology foundation tries to answer why people are strongly influenced by fakes news. The intention of this is to exploit the individuals (Shu, et al., 2017) making consumers of the content believe what they see and obstruct other rational thoughts. In addition, affirming individual's perceptions by showing content to people who have preconceived knowledge about a topic.

Another aspect of fake news on traditional media is the social foundation**,** this they describe as a system with two key players, the publisher and the consumer. They both have different goals, for the publisher the primary goal is to maximise profit, which is linked to

[10] https://en.oxforddictionaries.com/definition/rumour Accessed on 05/05/2018

number of subscribers, referred to as *short-term utility,* and their reputation, being a *long-term utility*. Whereas consumers want to acquire unbiased information referred to as *information utility* and news content that affirms preconceived knowledge - *psychology utility*. (Shu, et al., 2017) posits that fake news thrives when the:

"short-term utility dominates a publisher's overall utility [long-term utility], and psychology utility dominates the consumer's overall utility, and an equilibrium is maintained"

Interestingly, even though (Shu, et al., 2017) coined this in context of traditional media, this aspect directly applies to social media which is the focus of this research. In this context the publishers are accounts and consumers are mostly fans. Here the goal of the accounts would be to increase their follower counts, and the result of this would possibly mean more revenue and recognition. Whereas fans want to closely follow developments related to their club.

Fake news on Social Media

Fake news in social media largely falls into two categories, malicious accounts used for propaganda purposes and the "echo chamber" effect. With the first category, the primary objective of malicious accounts is to be medium a for malicious activity which includes trolls and social bots. A social bot is an account which is algorithmically controlled to publish certain types of messages. For instance, researchers from FireEye have established that thousands of Twitter accounts that campaigned against Hillary Clinton likely were controlled by automated social bots.[11]

The echo chamber effect, while not a fake-news phenomenon in and of itself, helps to exacerbate the problem of fake news. The echo chamber, or filter bubble, effect is described as a situation where people are only exposed to like-minded content or people. This can result in the dramatic polarisation of opinions. The echo chamber effect allows people to believe fake news due to the psychological factors such as confirmation bias and social credibility. People tend to be convinced that a source is credible if others, particularly

[11] https://www.nytimes.com/2017/09/07/us/politics/russia-facebook-twitter-election.html?mcubz=0&_r=0 accessed on 05/05/2018

those whom we respect, perceive it as credible. The frequency with which people encounter content also induces people into believing the fake news.

2.1.2 Detection

The first task in detection is to create a feature set that allows models to identify rumours. Survey on rumour detection on social media by (Shu, et al., 2017) have identified the following approaches for detection:

- Style-based: determine the style content in the language (i) linguistic-based which include the lexical features such word counts, the frequency of words and unique words. (ii) Syntactic features such as bag-of-words parts-of-speech tags.
- Knowledge-based where content is checked against an external knowledge base fact-check the content.

However, all the above works primarily focus on detecting rumours about topics related to current affairs (Zhao, et al., 2015). In this project, we employ a style-based approach where we use keywords to detect transfer rumour.

2.1.1 Rumour classification

In most system architectures found in the literature, a rumour classification phase follows the rumour detection phase. This task aims to predict the veracity of a given rumour. Various approaches can be seen in the literature using rule-based systems, machine learning (Kwon, et al., 2013) and probabilistic approaches (Qazvinian, et al., 2011).

The critical factor for this task is to find the right set of features to enable us to infer the veracity of a rumour. In this project, mainly due to time constraints, complex feature extraction was not possible. According to (Zubiaga, et al., 2018), (Castillo, et al., 2011)'s research has been influential on this topic. The goal of their work is to determine how accurate the authors of tweets are. In their work, they used two classifiers first to distinguish news content from conversational tweets using decision trees. Followed by another classifier to assess credibility.

They used four categories of features: message, user, topic and propagation features. Where the *message feature* includes the length of the tweet, sentiment score; *user-based* features include whether or not it is a verified account; the *topic* features include the length of tweets; and finally, *propagation* features include an indication of the initial number of tweets on a topic.

Building on from the feature set introduced by (Castillo, et al., 2011) later research (Kwon, et al., 2013) has used temporal, structural and linguistic features. The temporal features try to capture how rumours change over time. Structural features indicated network and the linguistic features. These features were shown to have performed better than (Castillo, et al., 2011) models.

For this work, the number of retweets, number of favourites, length of tweets, and tweet text were used. The machine learning algorithms performed poorly as they were not even predicting 50%, this was because the features were not informative enoug.Possible extensions inspired by state of the art and features identified given the context of football transfers to improve the performance will be discussed in section 5.

2.1.2 Football whispers

It is important to point out there are numerous tools online tools that calculate the likelihood of a player joining a club. One of them is Football Whispers (FW); which is a website that tracks rumours about possible signings, built upon tweets. This website tracks postings of rumours and ranks them. They use metrics such as “volume of whispers”, “authority of sources” and “recency of whispers”[12], and calculate a score which indicates the likelihood of completing the transfer.

Authors of the above system (Ireson, et al., 2017) have presented the approaches behind their system. This dissertation borrows some of the approaches outlined. FW is built upon extensive knowledge base made up of a combination of OptaSports, Wikidata and DBpedia to take into account misspelling and language variations. Then they use Deterministic Finite Automaton as part of entity linking process to identify the players. The

[12] http://www.skysports.com/football/news/11096/10314542/what-is-football-whispers-unique-transfer-algorithm-explained Accessed on 05/05/2018

rumour detection considers four factors: *consensus*: how often the rumour is repeated and by independent sources; *authority* reputation of sources; *time* how recent the rumours are and repeated, coherence/consistency[13].

2.2 Data Processing Pipeline

Information extraction is one of the most important parts of the project. Identifying entities and comparing them with existing knowledge bases is how this project will identify the entities and subsequently label the stated claim around a transfer to be true or false. The existing knowledge base, in this case, are transfers listed on the official English Premier League website for 2017/18 Summer Transfer Window .

(Nguyen, et al., 2014) and (Nguyen & Cao, 2015) have built systems to automatically store player transfer information by extracting information from the articles on Sky Sports and uses semantic web technologies to represent the transfer information. The overall goal was to create a data structure, where users can search for related content. Even though representing transfer rumours using the semantic web is not the concern for this project, methods described in the data pipeline have been hugely beneficial for this project.

They proposed crawling data from the Sky Sports website, then using a pre-processing step, an entity recognition step using KIM API (Popov, et al., 2003), followed by rules to detect relations related to football transfer

The use of Twitter data, rather the use of news articles, has been less researched. One such study is (Ireson, et al., 2017). Both (Nguyen, et al., 2014) and (Nguyen & Cao, 2015) have used articles written by journalists from Sky Sports. Forming a multi-stage pipeline would ensure components are modular and extra components could be added if necessary. A similar approach was followed in this project where we start by building a corpus followed by two annotation processes.

[13] http://videolectures.net/iswc2017_ciravegna_detection/ Accessed on 05/05/2018

2.2.1 Named Entity Recognition

Named entity recognition is an essential task in this project as it helps identify players in the tweet. Furthermore, it forms the basis for other components, such as the rumour veracity checking, where it labels whether a given tweet is a rumour that came out be true not.

Often NLP packages like Stanford Core NLP (SCNLP) NER are trained on news articles, which are much more formal than the Twitter content that we will be using. SCNLP's NER is based on Conditional Random Field (CRF) sequence models developed by (Finkel, et al., 2005) and they were trained on the Computational Natural Language Learning (CoNLL) collection. CoNLL is a collection of Reuters news articles annotated with four types of entities person (PER), location (LOC), organisation (ORG), and miscellaneous (MISC). (Finkel, et al., 2005). This will limit the accuracy of relation mapping when applied to Twitter data.

On the other hand, a tool called Twitter NLP[14] (Ritter, et al., 2011) was trained on Twitter data and used by (Kampaki & Adamides, 2014) for Part of Speech (POS)-tagging. However, Stanford's NER was found to perform better than Twitter NLP from a manual inspection. A blog by (COOPER, 2017) tested many entity recognisers such as Stanford Core NLP and the Twitter-specific Twitter NLP on shared task competition dataset[15]. A shared task is a competition where all the participants submit their systems to solve a particular problem on a specified dataset. According to this blog, both have very similar precision 0.5267 and 0.5240 respectively.

(Ritter, et al., 2011)'s approach is similar to SCNLP NER, however, their part-of-speech tagging, a vital component for NER, is different and is suited for Twitter data. Also, they propose an improvement to NER by creating a capitalisation classifier which identifies the difference when capital letters are used for emphasis and in acronyms, for example (NLP).

Google Cloud NLP (GCNLP) Named Entity Recogniser was also used and can be accessed through Google Cloud API. According to Google Cloud, it uses the best of Google's deep learning models. This is, however, a black-box, and no information about its workings is

[14] https://github.com/aritter/twitter_nlp
[15] https://noisy-text.github.io/2017/index.html#

available. At the initial stages, by inspection it was providing good results, however, later on, found that there were many misclassifications. The following example tweet will help motivate the reasoning behind not using the Google Cloud tool.

The entity recogniser identified the PERSON entity as "*Deal Complete Kevin Wimmer*". A sophisticated tool should be able to detect Kevin Wimmer. As a consequence, the SCNLP NER used as the NER for this research as it provided better performance compared to GCNLP.

2.2.2 Entity Linking

The pre-processing and entity recognition steps are a vital part of the data extraction aspect of this research. The proposed methods by (Nguyen, et al., 2014) and (Nguyen & Cao, 2015) have shown to be effective in their research. As a result, this project builds on top of their processes for pre-processing data. However, these steps cannot be simply ported as the text data used is structured, formal, and edited content. There needs to be further pre-processing step well as adapting to Twitter's informal text data such as identifying the club name vairations.

Twitter data has been studied widely and analysed across varies domains. (Kampaki & Adamides, 2014), investigates the use of Twitter data to predict the outcome of English Premier League games. The challenges encountered in data pre-processing step by (Kampaki & Adamides, 2014), are similar to this project.

Primarily when it comes identifying the clubs, they use predefined hashtags. Basing this approach, a list of possible club nicknames and hashtags used by fans were collected and stored in a knowledge base. Followed by a rule-based system which checks for the presence

of a team and player name. Having a list of predefined hashtags helps further improve detection of teams and hence helps improve the accuracy.

Consider a tweet *"Mohamed Salah in talks with #LFC",* where player Mohamed Salah is linked with Liverpool. In this is case Liverpool is referred to with #LFC hashtag. As a consequence, there needs to be a knowledge base of common nicknames and hashtags.

2.3 Sentiment Analysis

Sentiment Analysis is one the most challenging topics of Natural Language Processing (NLP). This aspect of the tweet will help identify whether there is any correlation between the sentiment expressed and accuracy.

Sentiment Analysis is one of the most popular fields in NLP, as it has many useful applications and data Twitter is one of most commonly explored in this field. Primarily due to the accessibility of tweets, easy to use tools and have shown to have useful applications, for example, Starbucks using sentiment expressed about their products to make informed decisions[16]

As mentioned earlier Twitter data is informal and conversational. Stanford's Core NLP are trained on movie reviews corpus introduced by (Pang & Lee, 2005) might lose out on some signals example capitalisation. However, there are tools that were specifically designed for this extracting sentiment from tweets and accommodate irregularity the language constructs in the tweet. Examples include VADER[17] which is specifically tuned to capture sentiment expressed in social media.

The most common approaches are Lexicon-based and Machine Learning based (Wei, 2012)

- Lexicon: Identifies words that best describe the sentiment
    - Advantages include: once they are built there is no need to train.
    - Disadvantages include: It is often built using WordNet corpus, which does not contain colloquial expressions. Also, it performs poorly when certain

[16] http://www.businessinsider.com/twitter-facebook-monitoring-2012-11?IR=T *accessed on 24/02/2018*
[17] https://github.com/cjhutto/vaderSentiment *accessed on 24/02/2018*

words can be either positive or negative depending on the context (A.Moreo, et al., 2012)

- Machine learning (ML): Using machine learning algorithms that learn characteristics based on data that is labelled as either positive or negative. According to (Mäntylä, et al., 2016) (Pang, et al., 2002) has been an influential study on sentiment analysis on Twitter. They used bag of words and SVM to classify the sentiment of the tweets.
    - Advantages include: Generalising better.
    - Disadvantages include: it requires considerable time and effort to label the data and train.

2.3.1 Comparison between Google Cloud NLP, Stanford Core NLP and VADER

Google Cloud NLP (GCNLP) and Stanford Core NLP (SCNLP), and VADER all offer sentiment analysis functionality. However, VADER (Hutto & Gilbert, 2014) is a rule-based system which is specifically designed to take into account social media constructs whereas SCNLP is trained on movie reviews. Meanwhile, GCNLP is a black box its workings are unknown to the public.

2.3.1.1 Google Cloud NLP (GCNLP)

This system is based on Google's complex deep learning models[18]

- For a given text it gives a score, represented by numerical score and magnitude values.
- scores are then aggregated into an overall sentiment score and magnitude for an entity
- Magnitude can be used to disambiguate

2.3.1.1 Stanford's Core NLP sentiment analyser

Stanford's Core NLP sentiment analyser is based on Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank (Socher, et al., 2013). (Socher, et al., 2013) Introduces Recursive Neural Tensor Network that is trained on Stanford Sentiment Treebank. A treebank is a hierarchical structure in which the semantic structure is encoded.

[18] https://cloud.google.com/natural-language/

The treebank was built on the corpus introduced by (Pang & Lee, 2005). The corpus consists of sentences from movie reviews, which was parsed with Stanford parser and has unique phrases from those and trees annotated by human judges.

Their methods can be seen to be performing better at capturing the sentiment and scope of negation than bag of words.

VADER[19].

VADER tool is a sentiment analyser that uses the lexical approach to map words to sentiment. It computes the scores by doing a dictionary lookup of the sentiment of phrases and sentences. Since it is adapted to social media content, it works well in detecting emoticons and internet slang. It also uses text constructs such as punctuation and capitalisation. It produces a score between -1 and 1. Where -1 is negative, 0 neutral and 1 being positive.

.

2.4 Language Modelling

Language models are a medium that can be used to represent text as numerical vectors. This particularly important for the machine learning models.

This section outlines the need for language models. Two approaches have been studied out here in this project one is count based, and the other is predictive models

The traditional methods used for sentiment analysis that used BOW methods which ignore word ordering and may use hand-designed negation feature will not capture all the details. As a result, there has been a shift in literature towards using word embeddings and neural networks, which will be discussed in the following sections

The count-based methods compute the statistics of word occurrences, whereas the predictive models give a probabilistic interpretation.

2.4.1 Bag of words

The basic model is Bag-of-words also known as unigram model. (Kampaki & Adamides, 2014) used this method to generate feature set for the machine learning models. Moreover, has

[19] https://github.com/cjhutto/vaderSentiment Accessed on 06/05/2018

been widely used in the literature while working with classification on twitter data (Culotta, 2010) (Pennacchiotti & Popescu, 2011). Also, for rumour detection tasks, ngrams have been used (Qazvinian, et al., 2011) to represent textual features.

The bags of words representation contains all unique words in the corpus. For given corpus of 2 documents

(1) I love ice-creams too
(2) Mary likes ice-creams too.

	Terms					
Doc ID	I	love	ice-creams	Mary	likes	too
1	1	1	1	0	0	1
2	0	0	1	1	1	1

Table 1 Bag of words representation

Above table shows how the corpus can be represented in a vector form. A common phrase in the transfer tweets is “Done Deal”, feature such as these might help machine learning algorithms discriminate between the classes. . A bag of words representation can be extended with an n-gram model. N-grams are sequences of n words/characters from a given text. N-gram models take into account the probabilities of seeing n-1 words.

For example, the following tweet:

"Apparently Arsenal are considering a cheeky £5M bid”.

Contains this set of bigrams:

Apparently Arsenal, Arsenal are, are considering, considering a,

a cheeky, cheeky £5M, £5M bid

TFIDF

Term-Frequency Inverse-Document Frequency (TFIDF) has two components: term frequency (TF), which reflects the importance of a word in document; and inverse document frequency (IDF) which reflects the importance of the word in the whole corpus (Manning, et al., 2008), Together it will describe a word’s importance to document in a collection.

It is calculated using this formula Tfidf = tf x idf.

The advantages of count-based such bag of words and TFIDFmethods are: [20]

- Easy to compute
- Sophisticated smoothing techniques can be used to improve the distribution

The disadvantages

- BOW can be sparse and could find it hard to capture long dependencies
- It does not take in to account the morphological aspects

Given the short text nature of the tweets, these models show to be quite useful in the modelling text.

2.4.2 Word embeddings

Another popular method in recent times in the literature is word embeddings. It is also known as context-predicting (Baroni, et al., 2014) . These techniques describes have a close connection to the Distributional Hypothesis , which states that words which occur in same contexts tend have similar meanings. This was popularised by (Firth, 1957)

Word2vec and its variations tries to capture the word similarities. It does this by predicting surrounding words of each words.

In recent times, word embeddings are popular technique when it comes to Twitter data. As noted by (Nakov, et al., 2016) for the task of Sentiment analysis on Twitter text, significant number of high performing teams have used word embeddings. However, an investigation of performance between BOW and embedding on social media rumour veracity by (Ma, et al., 2017) concluded that BOW performs better. However, this research was done on Chinese text, therefore applying to English language text could have different result.

[20] http://videolectures.net/deeplearning2017_blunsom_language_understanding/ Accessed on 05/05/2018

2.5 Learning Algorithms

In recent times Machine learning has been a popular tool to answer research questions in the literature. In machine learning there are **supervised, unsupervised** and **reinforcement learning** methods. (Bishop, 2006) Supervised machine learning algorithms a set of inputs and desired outputs also known as labels, the algorithm will try and learn by minimising the difference between the predicted and the desired output, example of algorithms include Support Vector Machines, and Random Forest. Whereas un-supervised do not require labels and it tries to find the structures itself, example of algorithms include k-means. Finally, reinforcement learning where the algorithm tries to learn by trial and error. This project focuses only on supervised learning as this project is only interested in using ML to check the veracity of rumour which has already known outcome. Furthermore, the project also has veracity checking tool which will produce the dataset.

In machine learning literature the term "features" is often used, features are characteristics of a particular observation passed into the learning algorithm. (Bishop, 2006) These features help the algorithms to learn patterns. For instance, in text classification, textual data converted into bag of words model is one of the sets of features that could be used.

2.5.1 SVM

Support Vector Machine (SVM) is a machine learning algorithm that can be used for classification and regression (Vapnik, 1995). In an SVM classifier, a separating hyperplane is drawn so that it separates the data into different classes.

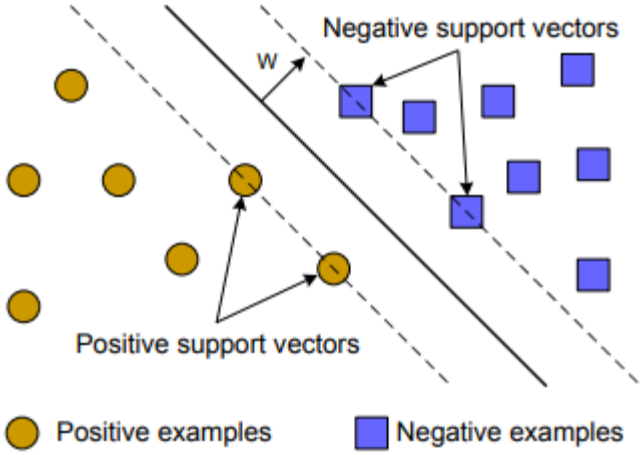


Figure 2 Linear separable SVM (Sun, et al., 2009)

The algorithm accepts a weighted feature vector xi and label yi pair. Where xi the ith example and yi is the corresponding label. Then the algorithm tries to find a linear decision surface that separates the data. A separating hyperplane is constructed which has a more maximum margin which essentially means that the hyperplane is constructed in a way that the distance from the nearest point (support vectors in Figure 2) on each side is maximised.

Two supports are created on either side of the separating hyperplane. It is better to have a larger distance between the separating hyperplane and support vectors as this will help reduce the classification error (Vapnik, 1995). Hyperplanes can be decided by minimising $\frac{1}{2}\ |\vec{w}|^2$ subject to $yi(\ \vec{w}\cdot\vec{x}-b)\geq 1\ \forall i$ (Sun, et al., 2009) where $\vec{w}$ is a vector perpendicular to the hyperplane. This decides the orientation of the hyperplane whereas b decides the position of the hyperplane.

The algorithm updates $\vec{w}$ and b, and learns the hyperplane, using the positive and negative samples. Once the learning process is complete ,a prediction can be made for previously unseen data using decision function $f(\vec{x})=\ \vec{w}\cdot\vec{x}-b$. If the decision function outputs a negative number, it is labelled negative class and vice versa. (Sun, et al., 2009)

There are several advantages and disadvantages of SVM. Advantages include, as there is a regularisation parameter for SVM, the over-fitting can be reduced. As it is a convex optimization problem, there is always a unique solution since there are no local minima.

Moreover, SVM has been shown by (Joachims, 1998) to be very useful when dealing with textual data. Using techniques like bag-of-words on textual data leads to a very large number of features. In this project textual data is a key feature as we try to see if Twitter data can be used to predict veracity as outlined in the research question. According to (Joachims, 1998) SVM is a good choice for text classification as SVMs have overfitting protection that depends on the number features and have the potential to handle a large number of features.

2.5.2 Random Forest

Random Forests is a technique of creating ensemble models of decision trees for classification and regression. Ensemble trees give more accurate results than the single decision trees. In Random Forests, a forest of trees is created by selecting the data randomly. For each tree, a random independent and identically distributed vectors are generated. (Breiman, 2001). Each generated random vectors are previously generated random vectors. Based on these vectors and dataset trees are grown using the CART methodology.

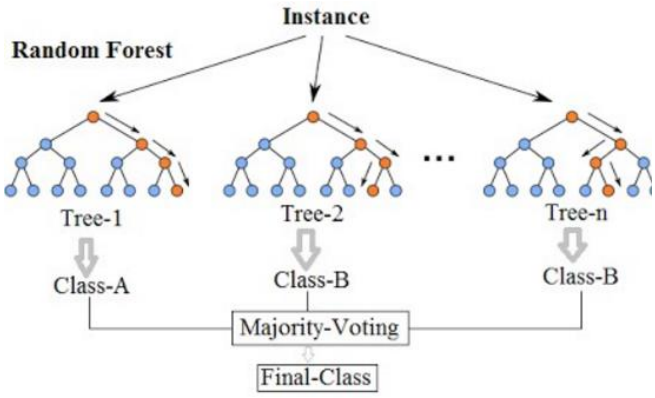


Figure 3 Classification in Random Forest[21]

For classification, the class is selected based on the poll. Each tree casts a vote for the class selection in case of Random forests classifier as shown in the above Figure 3. For regression, the result is calculated by taking the average of outputs of each of the trees. The generalisation error converges almost sure as the number of trees increases (Breiman, 2001).

[21] https://medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d Accessed on 05/05/2018

It also depends on how randomly created trees are. The error reduces when the correlation between trees decreases. There are several advantages and limitations of random forests.

Advantages are mainly the following. Even though each tree is unpruned, chances of overfitting are less as random data is selected and decision trees, are formed. It works well with on large dataset. Additionally, this algorithm can provide information on variable importance and outliers of the data. (Horning, 2010). Disadvantage include, some outputs generated are not within the range of values in case of the regression.

2.5.3 Unbalanced data

An unbalanced dataset is one where there is a disproportionate number of observations for a certain set of classes over others. This is particularly true for the dataset used in this project and is reflective of the real world, where true rumours are rare and observations of false rumours are in abundance. Unbalanced datasets are a major problem, as the algorithm will not be able to learn characteristics of the minority classes, and will predict the frequently occurring classes more regularly than the rare ones.

There are numerous methods in the literature to reduce the impact of this problem, among them is sampling and cost-sensitive learning (Weiss, et al., 2007). Sampling alters the class distribution of the data:

- **Oversampling**: Where the minority class is sampled more than the majority class. The main disadvantage according to (Weiss, et al., 2007) is that creating copies of samples makes overfitting likely, and also results in increased learning time as there are more samples.
- **Undersampling**: Where fewer majority classes are sampled. The disadvantage of thisd approach is that it could result in loss of potentially useful data.
- **SMOTE**: (Chawla, et al., 2002) suggest another technique where it creates synthetic samples based on some heuristics about how close the samples are to each other.

Another alternative to sampling is cost sensitive methods which are applied in the algorithms where it tries to penalise some misclassification errors than others. According to (Weiss, et al., 2007) this approach might not apply to all the algorithms, as result sampling is preferred.

2.6 Conclusion

This chapter defines the background for this project. The first section discusses the different types of rumour detection and classification that will help to address the research question. The next discusses different Named Entity Recognition tools that were explored, despite being trained on newspaper articles the Stanford's NER seem to perform better.

For sentiment analysis there are multiple approaches, the state-of-the-art tools were discussed and their advantages and disadvantages when it comes to Twitter data.

The choice of learning algorithms used to try and answer the secondary research question was motivated by the fact that textual data was key aspect.

3. Design and Methodology

The primary goal of the project is to provide a system using NLP techniques to test the accuracy of the accounts. In this section discusses the design of the different components, how the veracity of a rumour is checked, and the design decisions made during the duration of the project.

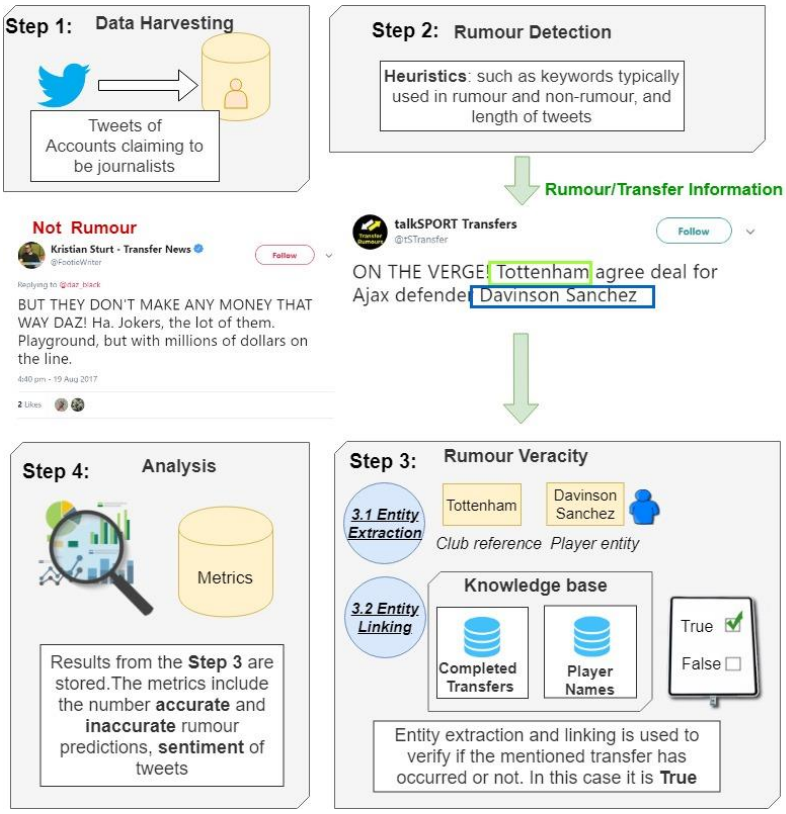


Figure 4 Outline of overall system

3.1 Building Corpus

In order to answer the research questions, additional components needed to verify the authenticity of the tweets. The following sections describe this process.

3.1.2 Knowledge Base Building

This section outlines the different knowledge bases (KB) built for purposes of this project and their use cases.

3.1.2.1 Completed Transfers

A rumour verification system requires a means to check the veracity of a rumour. In this research, we will be checking against well-known facts or ground truth.

The ground truth used in the project is the completed transfer information from the official English Premier League(EPL) website[22]. This website contains information about all

[22] https://www.premierleague.com/news/397434 Accessed on 01/05/2018

the completed transfers during the 2017 Summer Transfer Window, which includes the players, their former club and the new club. This information was extracted out and represented in a Python dictionary. Key of the dictionary is the *player name*, combined with first and second names. The value will be a tuple of *from_club*, *to_club and sign date.* All the components mentioned, are of string type and lowercased, without spaces.

The signing date was extracted from Wikipedia[23] . An essential feature of a rumour detection system, as it helps to identify whether a given tweet is an announcement after a player signed or not.

Figure 5 Example of an entry in the knowledge base

Figure 5 illustrates an example where player Jordan Pickford moved from Sunderland to Everton on 15/06/2017

To increase the chances of detecting the player when referred to by their second name, the same information with the key as the surname is added to the KB. However, there will be an issue where players have the same surname. In which case the surnames are not added to the KB. This will not have a significant impact on the performance of the system as the rumours tend to be about well-known players and these players are inserted first into the KB to ensure that their second names do not get misplaced.

The primary advantage of this approach is that it has a quick lookup of O (1) and this is crucial because components of the system often need to compare each word in the tweet with the KB. Hence, this approach substantially reduces the search times.

However, this KB is not as extensive as Football Whispers' as they consider the different variations in the player names of more than 40,000 players. The following table illustrates the problem:

[23] https://en.wikipedia.org/wiki/List_of_English_football_transfers_summer_2017 Accessed on 01/05/2018

Player Names	Discussion
Daniel Drinkwater to Danny Drinkwater	An England international player who is nearly always called Danny Drinkwater
Martina, Rhu-endly Aurelio Jean-Carlo to Cuco Martina	This is because he is often referred to as Cuco Martina hence it would increase the chance of matching the tweet.

Table 2 Player name variations

To reduce the impact of this problem, names of the players listed in this KB were manually searched and replicated the transfer information with the name variations in KB.

3.1.3.2 Player Names

Name Entity Recognition tool used as part of the entity linking, which occasionally annotates some unintelligible names. In order to help the system, find the actual players, a player's dataset was created. The following knowledge bases were represented in a Python dictionary.

Premier League Squad

A knowledge base of Premier League players squad of 2016/2017[24] was built. Here the player name is the key, and the value is the club name. This KB is used as part of the rumour veracity checking process and will be discussed in section 3.3.2. The main purpose of this KB is to make sure that the entity linking process do not make false connections, i.e. linking a player to his current club as part of a rumour. As mentioned earlier this study only focuses on player transfer to EPL, therefore, only the EPL squad was used**.**

Ideally, this system should have player names and their associated clubs, which could be found using knowledge bases such as DBpedia and Wikidata. However, we did not have the resources to extract them.

As a result, this system would incorrectly link player and clubs in rumours about other leagues. To mitigate this problem only tweets containing EPL references such as club names are considered as rumours, which will be further detailed in section 3.2.1

[24] https://www.premierleague.com/news/84136 Accessed on 01/05/2018

Tokenised Names

Data cleaning process used by the machine learning component needs to remove player references. To do this, another KB was built containing the first and second names. In order to increase the chances of finding player names, a reasonably recent EA Sports FIFA 2017[25] player names data set of 10,000 player names were used. It is a very popular game. As a result, the makers try their best to represent the player names accurately. Hence, there is a high chance of capturing names that are commonly used.

However, all these knowledge bases have their drawbacks and come with a price. Player names could be misspelt, and or use nicknames. To reduce this issue approach by (Ireson, et al., 2017) where they combined multiple knowledge bases (DBpedia, Wikidata and OptaSports) could be used to ensure that the name variations are captured.

3.1.3.2 Club names

Another issue is that Twitter text is predominately informal, hence clubs could be alluded to by their nicknames, shortened names, or hashtags. This makes it difficult to identify the teams referred. (Kampaki & Adamides, 2014) encountered a similar problem with their project. To address it, they first identified possible hashtags and nicknames of clubs in EPL and created a database. When doing their inference their algorithm checks for nicknames and hashtags to identify the club, the tweet is talking about.

This project borrows the same approach, and in addition to the EPL, clubs involved in the transfer window were also added to this knowledge base built using Python dictionary. The key was the name of the club and value is the formal name. For example, the name "toffees" is mapped to "Everton" football club. The KB was designed in this way so that it was easier to calculate the metrics for each club, as the observations related to Everton can be grouped.

[25] https://www.kaggle.com/artimous/complete-fifa-2017-player-dataset-global Accessed on 01/05/2018

3.2 System design

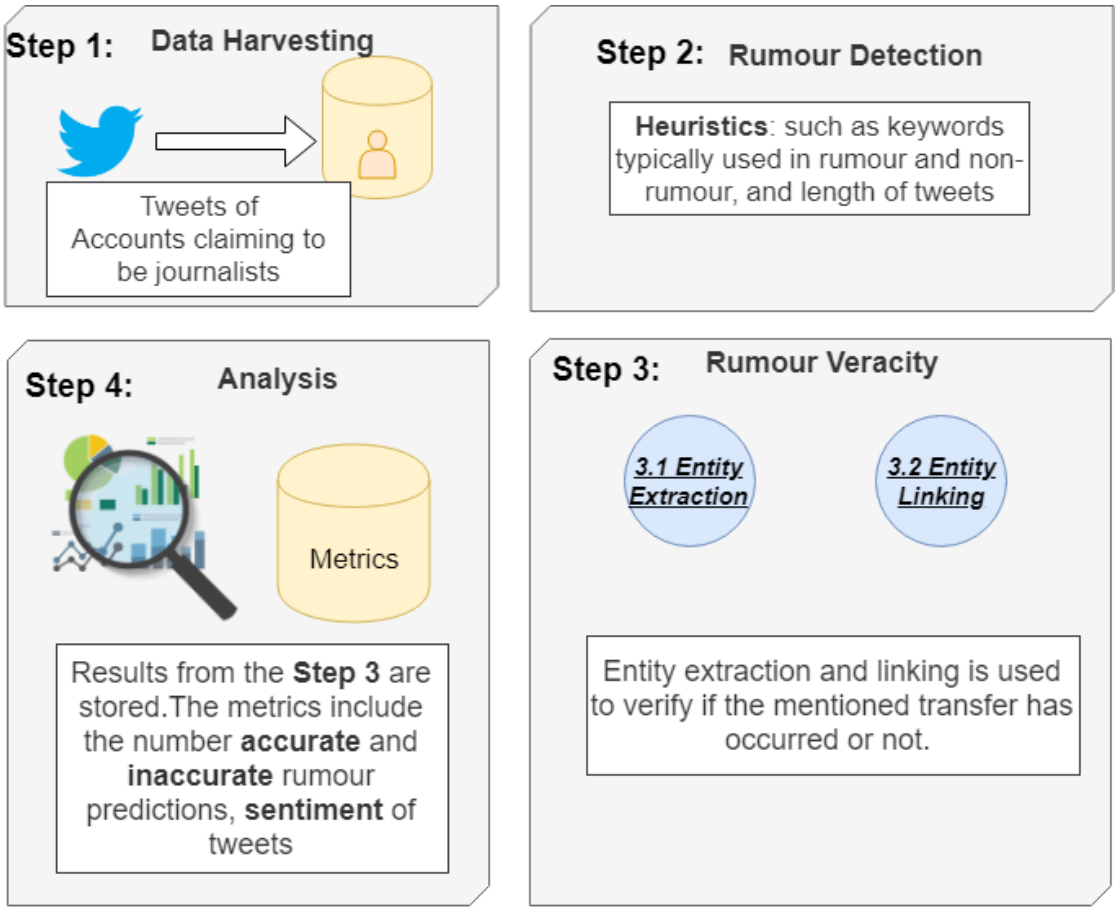


Figure 6 System design

The corpus and knowledge base building process described above play a vital role and are the building blocks of the system. The following sections outline the general overview of the components.

1. **Data harvesting**: Extracting tweets from identified accounts
2. **Rumour Detection**: Identifying if a given tweet is a rumour or not
3. **Rumour Veracity:** This leads on from the above section, checks the veracity of the rumour
4. **Analysis:** calculating the scores and metrics for each account

3.2.1 Data Harvesting

The projects main focus is only on transfers in the English Premier League. Specifically, the 2017 Summer Transfer Window, which began on June 10th and closed on September 1st [26]2017. Tweets from selected accounts were collected from 05/05/2017 to 05/09/2017. The reasoning behind the earlier timeframe is that rumours tend to start just after the final matches of the previous season and would dwindle towards the end of the transfer deadline.

The first step was collect Twitter account names that claim to be football journalists or "ITK". As often the case most of these accounts do not talk about transfer rumours. The next step was to manually identify accounts that regularly post player transfer related content. An initial set of accounts were recommended by well-known sports websites such as Sportskeeda[27], Huffington Post[28] and Bleacher Report[29] were used.

However, many of these recommended accounts were either dormant or have changed their account identifiers and have to be manually checked.

Along with the above-described process, the Twitter search was used with keywords such as "Transfer news", "football transfers", "transfer rumours", "In the know", and "football journalist". These keywords were selected because focused only on English Premier League and English tweets.

---

[26] https://www.foxsports.com.au/football/premier-league/2017-summer-transfer-window-dates-when-does-it-open-close/news-story/9c30aa8283b86a1e184a0bf1a949c082?nk=22c8552ef5ad3203f903f3cf61ff33e1-1525876992 Accessed on 02/05/2018

[27] https://www.sportskeeda.com/slideshow/10-football-journalists-you-must-follow-on-twitter?imgid=31856 Accessed on 02/05/2018

[28] http://www.huffingtonpost.co.uk/2012/04/19/50-football-folk-to-follow-twitter_n_1437152.html Accessed on 02/05/2018

[29] http://bleacherreport.com/articles/2205628-50-football-writers-to-follow-on-twitter Accessed on 02/05/2018

Corpus Quality

All in all, about 120 accounts were identified. However, many did not talk about transfers or have been dormant during that period. These accounts had to be manually removed. Furthermore, the number of followers they have and how active they were influential in on picking the accounts. The following table of 31 accounts is a result of this tedious manual work.

Twitter Accounts		
Crazysandra101	Jon_LeGossip	TransferRelated
DeadlineDayLive	Jonawils	TransferSite
ed_aarons	JPercyTelegraph	TransferTrends
EPLMoves	Now__Football	TransfRumours
FB_WHISPERS	SkySportsLyall	ITTC_football
FOOTBALLITKCOM	Sport_Witness	johncrossmirror
FootieWriter	SquawkaNews	TrustyTransfers
Footy_Transferr	TEAMtalk	tsTransfer
FTransferNews	TransferBible	HITCdeadlineday
GraemeBailey	TransferMoves	
indykaila	TransferNewsCen	

Table 3 Twitter Accounts studied in this research

The language constructs of accounts are varied for instance account "*tsTransfer*" tend to have tweets like

Tweet: *"NO DEAL! Everton sources tell talkSPORT they have NOT accepted £75m offer from Manchester United for Romelu Lukaku http:// dlvr.it/PScGrn"*

The player in question in Romelu Lukaku who moved from Everton to Manchester United. This account largely speaks about transfers and the vocabulary used is similar to tabloid newspapers. On the other hand, similar comment on Lukaku's transfer from "*Crazysandra101"*

Tweet: "Oh OK so when I predicted based on Intel from my source that Lukaku was going to MUFC not Chelsea on April 13th you were asleep then"

It can be seen that tweet is much more informal, and players are often referred to by their second name.

In total there are about 58,000 tweets in the corpus. However, some of the accounts are more active than others

3.2.1.1 Get old tweets python

Twitter is data has been widely used from sentiment analysis (Hutto & Gilbert, 2014), detecting rumours (Zubiaga, et al., 2018) to predicting outcomes of matches (Kampaki & Adamides, 2014)**.**

There are numerous libraries for gathering tweets from Twitter for various languages. This discussion specifically targets Python as its the primary language used in this project. There are python-twitter –"provides a pure Python interface for the Twitter API", Tweepy – "a Python wrapper for the Twitter API", TweetPony – "A Python library aimed at simplicity and flexibility" and many more[30]. However, they are all based on Twitter API which limits the number of tweets that can be harnessed - up to 3200 for each account[31]. The project is concerned with gathering tweets from a specified period and accounts studied in this project are avid Twitter users who often tweet and exceed the limit. As a result, the above mentioned tool may not capture important tweets. GetOldTweets-python[32] was used as a workaround.

GetOldTweets-python is a Python-based scrapping tool which accepts input as a command line argument where the conditions for the extraction are specified, which include: timeframe from and to, and an account name. This tool then translates this information into a form that is consumable for Twitter Search and scrapes tweet information from the website. Which is then stored into a CSV file, one per account. The data was stored in CSV files rather than a database because it was significantly easier to inspect the data.

In this project, there were a number of accounts to collect from. Since the tool only could deal with one account at a time a Python script was written, that uses the multiprocessing library to parallelise the scraping process.

[30] https://developer.twitter.com/en/docs/developer-utilities/twitter-libraries Accessed on 01/05/2018
[31] https://developer.twitter.com/en/docs/tweets/timelines/api-reference/get-statuses-user_timeline.html Accessed on 01/05/2018
[32] https://github.com/Jefferson-Henrique/GetOldTweets-python Accessed on 01/05/2018

3.2.1 Rumour Labels

The Twitter data is noisy in nature tweets could be commenting on current affairs, sports, player transfers or just having general conversations with other users. To distinguish the football transfers rumours from other content three labels or classes was introduced:

- **Rumour True:** The tweet is *a rumour* and came out to be *True*. The project is concerned only with player transfer information. Therefore, this project will narrow the definition to "to" and "from" clubs and player associated with this transfer.
- **Rumour False:** The tweet is *a rumour* and came out to be *False*
- **Not Rumour:** This not a rumour, possibly general conversations, transfers not related to premier league or an announcement about a player joining a club.

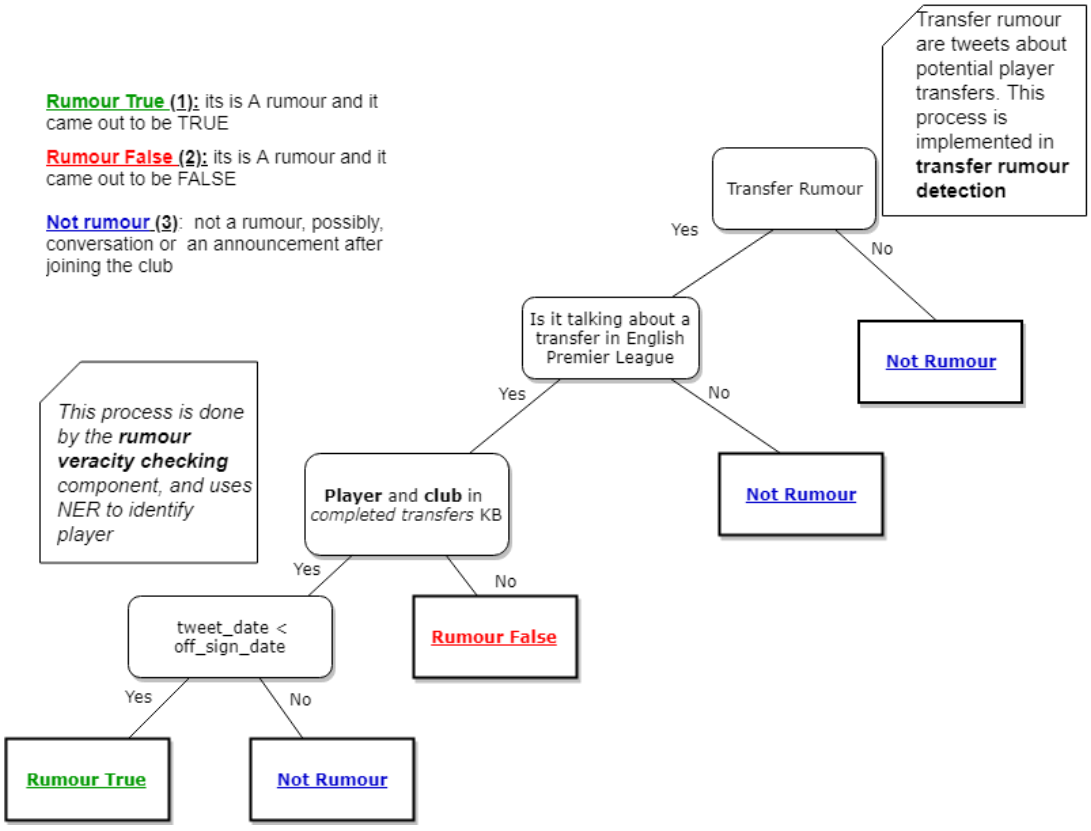


Figure 7 Rumour Labels decision tree

Figure 7 illustrates the how a particular tweet can be labelled with one of three classes. The following sections discuss each of the labels.

Rumour True:

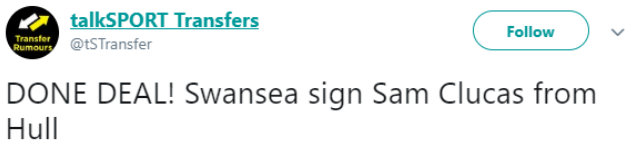
- The tweet must a player transfer rumour and related to English Premier League.
- Player *and to_club* must be in the completed transfers knowledge base
- The tweet date must be before the official sign date, retrieved from the knowledge base

Rumour False:

- Similar to the **Rumour True** it must be a transfer rumour
- The NER detected player must not be related to a club in the completed transfer's knowledge base.

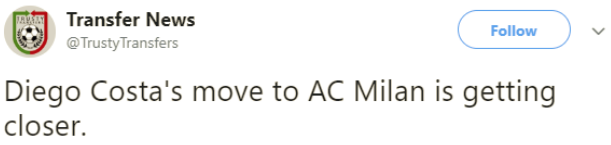
Not Rumour:

- Any tweet that is not a transfer rumour, i.e. conservations and personal opinions. Section 3.2.1 conducts a discussion of transfer rumours and how they are detected.
- Any report on transfer that dated after the official announcement, e.g..



This tweet dated 23/08/2017 which is the same day as signing date

- To cater for the conversational aspect of Twitter data, any tweet that is less than 20 characters in length is not a rumour. This is mainly because they are unlikely to carry enough informational context to infer a possible rumour.
- Consider the following tweet



This statement refers to a player named Diego Costa, who (according to reports) intended to move away from his current club – Chelsea FC. The player

subsequently moved to Atletico Madrid[33]. The statement in this tweet is, in fact, a rumour; since the move is not to a club in EPL, it is labelled as not a rumour.

- There might be cases where the tweets might be at hinting an imminent transfer, nevertheless will not mention the player in question.

There are many nuances in the definition however for this project due to the time constraints; further discussion will be conducted in the evaluation section.

3.2.2 Transfer Rumour Detection

As briefly mentioned in section 1.4, and 3.2.1, a significant amount of an account's tweets do not contain any transfer rumours. This will significantly curtail performance of the next step, the rumour veracity checking process. Therefore, there needs to be a way to differentiate between football transfer related content from other content. The first step to do this is to define a transfer rumour detection component and to build on from this a rule-based system.

A transfer rumour in the context of this project can be described as tweets reporting about a potential player transfer, or ones that claim they have inside information, or ones that quote other sources, however, in this case, we are assuming that by doing so the twitter account is supporting the claim.

The first task was to build heuristics for this component; the heuristics include keywords and regular expressions. Some of the keywords to describe a transfer include "*official*"," *loan", "medical", "contract",* and *" exclusive".* For instance, the word "medical" indicates the procedure conducted by the club who signed the player, to get as much information about the player's health condition. This occurs at the final stages; hence it is highly likely that this rumour will be true, moreover a transfer rumour.

Similarly, for non-rumours *"haha", "thank", "southgate",* and *"ladbrokes".* The name "*southgate"* refers to England national men's team manager Gareth Southgate, and

[33] https://www.independent.ie/sport/soccer/chelsea-agree-terms-with-atletico-madrid-for-transfer-of-diego-costa-36155736.html Accessed on 01/05/2018

often tweets are a criticism of the national teams. As a consequence, this became useful keyword to detect non-transfers.

Another part of the heuristics is the use of regular expressions. There are specific patterns in the tweets such as “30 million” which are highly indicative of a potential transfer rumour as they often describe a fee a club that’s interested in a player must pay to the current club to release that player.

3.2.3 Rumour Veracity Checking

This section illustrates the process in which a rumour tweet labelled by the above process is checked for its veracity. This system relies on a knowledge base of completed transfers

3.3.3.1 Named Entity Recognition

Name Entity Recognition is an essential component of the system. This provides a mechanism to identify players. The entity recognition tools try and classify named entities into a set of pre-defined categories. The categories include PERSON, ORGANISATION and LOCATION. This project only considers person entity Organisation categorisation could be used, however, a tweet containing “Chelsea” would be recognised as a person, rather than an organisation. In the context of this football transfers, organisation Chelsea is much more accurate than categorising it as Person. Therefore, the detection of the *to_club* has to be extracted out club names knowledge base. Which was done by running each word in the tweet over the KB.

The choice of NERs was influenced, firstly their ease of use and followed by the accuracy. As discussed in the State of the art the SCNLP performed better hence it was used.

SCNLP NER tokenises the sentence and then annotates each word it with entities. Tokenising is a way to chunk text into separate pieces called tokens (Manning, et al., 2008).

Example: “Danny Drinkwater undergoes medical at Stamford Bridge”. The tagger will produce: (“Danny” PERSON), (“Drinkwater” PERSON), (“undergoes” O), (“medical” O), (“at” O),…

The annotated person tags are sent to the entity linking process.

3.3.3.2 Entity-linking

This component uses the player name identified by the entity recognition system to check if this player exists in completed transfers KB. If it exists, KB returns information about the transfer. The system then scans the tweet to see if the to_club returned by KB is found in the tweet. If it does the tweet is labelled to be true.

Figure 8 Rumour tweet which came out to be true

As shown in Figure 8 Davison Sanchez is identified and sent to the knowledge base (built in section 3.1.2.1) to extract the associated information about this player. The information returned has the associated club name. Since the associated club can be found in this rumour, it is labelled to be true. More details about this process will be discussed in section 4.2.2.

This rumour definition overlooks over many of the intricate loopholes. One issue is where rumours could be labelled incorrectly if the is about transfers not in the premier league as KB only consists of EPL transfers. To reduce the impact of this problem the tweets are checked for the presence of EPL clubs by looking up knowledge base described in section 3.1.3.2. If no club is found, it is automatically labelled not a rumour. This was done prevent penalise accounts for non-EPL rumours

3.2.4 Sentiment Analysis

Sentiment analysis was employed to examine if there is any influence on the accuracy of the accounts. After concluding the above process, each tweet is then sent to sentiment analysers to calculate the scores. VADER, Stanford Core NLP, and Google Cloud NLP were utilised. Scores were normalised into three labels, (1) for negative, (2) neutral, and (3) positive. This was done to make it is easier to compare the scores from different analysers.

As noted discussed in the state of the art VADER produces scores between -1 and 1. A thresholding was employed:

- score <= -0.05 = Negative (1),
- (score > -0.05) and (score < 0.05) = Neutral (2)
- score >= 0.05 = Positive, this is in line with the documentation[34]

For Stanford Core NLP:

- score < 2 = Negative (1),
- score ==2 = Neutral (2)
- score > 2 = Positive,

For Google Cloud NLP:

- score <=- 0.25 = Negative (1),
- (score > -0.25) and (score < =0.25) = Neutral (2)
- score > 0.25 = Positive,

[34] https://github.com/cjhutto/vaderSentiment Accessed on 01/05/2018

3.2.6 Machine Learning – pre-processing

Pre-processing for machine learning

Machine learning algorithms were explored in a project to help answer the secondary research question- to investigate if ML algorithms are used to predict the veracity of a rumour. Since we wanted to examine if there are any particular textual patterns in tweets that the algorithms can learn, the player names and club names were removed from the tweets data.

This was done to avoid bias in the data, for example, there might be many references to "Liverpool" in the data. As a result, an algorithm might learn in a way that it may always favour a class when it sees a tweet about Liverpool.

Figure 9 Pre-processing stages

A multi-stage pre-processing step to clean the textual data which is common process before implementing ML algorithm was employed as shown in Figure 9. The first step is to remove tweets that are not English using tool called langid.py[35], occasionally there are non-English tweets which are due to the noisy nature of Twitter data. The text was tokenised, and each token (word) was checked against the KBs described in section 3.1.2 removed if found in player or club KB. This is followed by lemmatisation and stop-word removal.

Lemmatisation is a process in which words are conflated to remove inflectional endings and to dictionary form of a word. (Manning, et al., 2008). Stop-words are words that are quite common in the language, e.g. "for", "to", and "the", and would not give much information to the machine learning algorithms so that they can differentiate text. Lemmatisation and stop-word removal was done using a tool called Spacy.

Tweets also contain other uninformative features like hashtag and URLs these have to be removed as will increase the feature space when text is converted to vectors. Regular expressions were utilised to identify and remove them.

[35] https://github.com/saffsd/langid.py Accessed on 01/05/2018

All of the above-described processes will help reduce the feature space, and as a result, ML algorithms to learn better.

Following on from the above process the machine learning algorithms are employed which will be discussed in section 4.3.

3.3 Ethical Implications

The data examined in this research are already publicly available. Furthermore, all the accounts studied are either sports journalists or have claimed to be one (in their bio). The tweets expressed by these accounts are intended for public consumption, where they inform their followers about potential transfers. Also, their primary intention is to be influential when it comes to player transfer rumours.

As outlined in the above sections this project uses scraping like mechanism to extract the tweets from Twitter. This was done purely to for research purposes. However, for the further study, we will ensure that we fully conform to GDPR. Also, collect older tweets from paid services such as Gnip[36].

Data handling: collected data will only be stored locally, and it will be deleted after the thesis evaluation.

3.4 Summary

This chapter outlines the design of the system. It began with the knowledge bases required to infer the veracity. Following on latter sections details definition of rumour in context of this research and detailed discussion of the different components in the proposed system

[36] http://support.gnip.com/apis/ Accessed on 01/05/2018

4. Implementation

4.1 Data collection

Get old tweets- python (GOT)

Section 3.1.1 argues the need for the scrapping mechanism. However, the Python module only does one account at a time. Moreover, the scrapping is independent, as there is no interaction with other components, therefore it can be parallelised.

Firstly, GOT accepts the criterion for the tweet extraction it tries to mimic the browser and extracts the tweets. However, as mentioned earlier only one account at the time could be extracted. This problem was solved using the Multiprocessing[37] package in Python. This allows the process-based parallelism. An alternative to multiprocessing was to use threading. However, due to the Global Interpreter Lock in Python only one processor is utilised[38]. Thus, limiting parallelism. On the contrary, the multiprocessing package will utilise multiple processors on a given machine.

A simple a Python module called get_all_tweets was created. This module can be interfaced using command line arguments, which include a timeframe, i.e. from and till dates, number of processes and the file path of a CSV file containing the accounts to be extracted.

The primary job of this module is to distribute the workload across multiple processes. It first extracts all the accounts from the file provided and allocates a subset of account names to each process. It will iterate through the accounts in the subset and will

[37] https://docs.python.org/3.4/library/multiprocessing.html?highlight=process Accessed on 01/05/2018
[38] https://wiki.python.org/moin/GlobalInterpreterLock Accessed on 01/05/2018

create criterion which includes the timeframe and submit it to the GOT TweetManager. The TweetManager will then frame the URL based on the criterion and request the webpage using the URL. Then it parses each tweet from a web page and returns the content in a list to the module the extracted tweets are written into with a file with account name as its name.

The tweets collected with this process were verified against the method using Twitter API. To ensure that all the tweets were collected. This was done by iterating through the tweets and comparing the timestamps and length of the text.

4.2 Annotation Phase

4.2.1 Rumour Detection

To identify rumours a set of keywords that would be most likely to be in tweets about players transfers were gathered.

As described in section 3.2.2 there are two sets of keywords one for transfer rumours and the other for non-transfers. Some keywords such as “medical” and “deal” acted as an initial seed for creating the keyword lists.

This was an iterative process as illustrated in **Error! Reference source not found.** the primary keywords were inserted into the rule-based system and it labelled the tweets. Then evaluated the performance of keywords by inspecting the tweets. This process enabled to find more keywords that can also be used prune ones that gave a high number of misclassification. An example of this problem is keyword “*move*”, this is a very common word used in transfer rumours, however, and it is also widely used in other tweets.

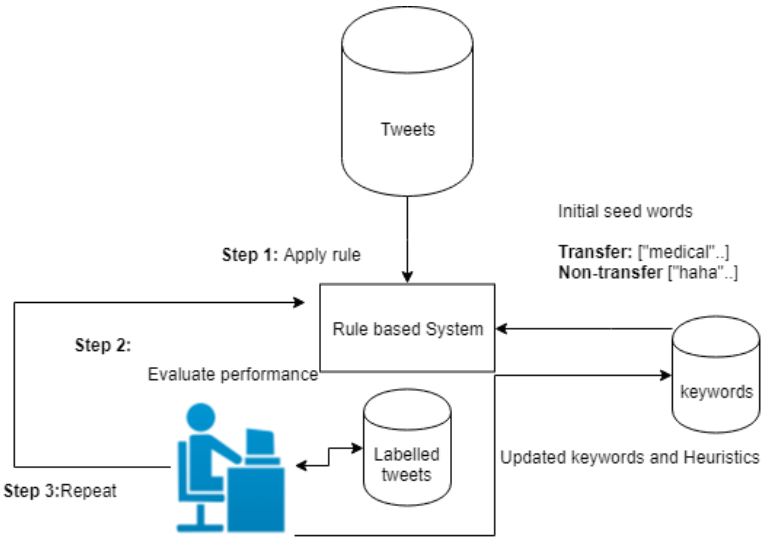


Figure 10 Transfer labelling process

The rule-based system has two indicator functions one for each class. The indicator functions count matching keywords in the tweet. This is implemented using Python string matching, where it checks if the words are a substring of the tweet. Each match has a weighting of one.

The next step for the indicator functions is to add additional feature weightings. Most of these features are represented as a regular expression. The following table describes the regular expressions and the rationale behind using them.

Additional Features – Non-Transfers			
No	**Feature**	**Explanation**	**Discussion**
1	"£\d+ "	Captures the British pound symbol followed by numerals. E.g. £100	Most accounts are British, and some tend to run the completions to engage with their followers
2	"\d+/\d+"	Numeral followed by a forward slash and a numeral, E.g. “10/1 for”	Some account offer odds for a multitude of reasons, for instance, it could be for player joining a club
3	"Amazon"	Strings that contain “amazon” substring. E.g. “amazon vouchers for”	Accounts sometimes offer Amazon vouchers to followers for various competition or talk about Amazon products
4	Length(text) < 20	Length of the tweet must be greater than 20 characters	Tweets less than 20 characters tend to be short replies or just an acknowledgement of something

Table 4 Non-transfer features

All the features have a weighting of one. However, feature four is given more weighting because for inspection of data this highly likely the case.

Like the non-transfers features, the following describes the additional transfer feature.

Additional Features -Transfers			
No	**Feature**	**Explanation**	**Discussion**
1	"([$,€,£]\d+\s?[million,m])"	Tries to capture the transfer fee of the player. E.g.	When talking about a potential transfer, accounts to mention often the cost of signing the player regarding transfer fees or wages

Table 5 Transfer features

Finally, the labelling function runs the text through the each of the indicator functions and decide to label either way depending on the indicator scores.

4.2.2 Rumour Veracity Checking

The rumour veracity checking requires two main components the entity extraction and entity linking.

4.2.2.1 Entity linking function

This takes in a tweet and returns one of the three labels ((1) True rumour (2) False Rumour and (3) Non-rumour).

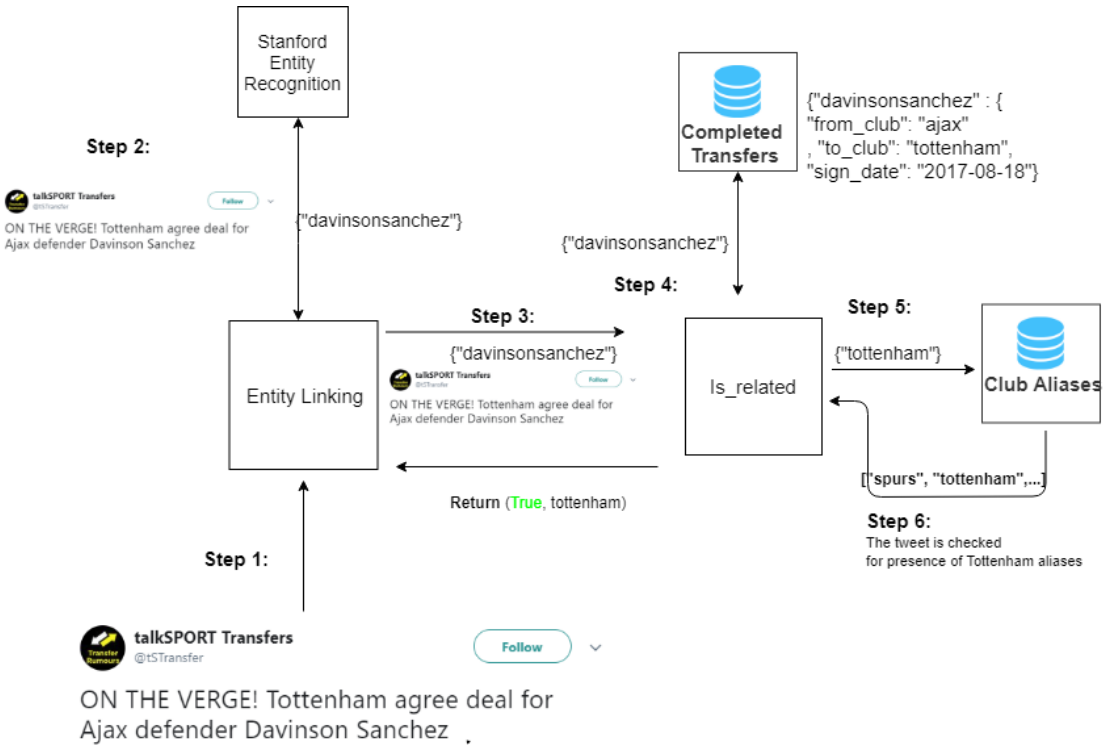


Figure 11 Entity Linking process

Figure 11 outlines the example where a rumour is true. The first step is to run the Stanford's named entity recognition on a given tweet, which it returns an output similar to the following example

Example: "ON THE VERGE! Tottenham agree deal for Ajax defender Davison Sanchez". The tagger will produce: ("ON" O), ("THE" O), ("VERGE" O), ("Tottenham" PLACE), ("agree" O), ("deal", O), ("for", O),("Ajax", "O"), ("defender", O),("Davison", PERSON), ("Sanchez", PERSON)

The entity linking function then iterates through all the PERSON tags, removes any alphanumeric character (there might be apostrophes in names for example), lowercases it and pass the player name and tweet into is_related function

Is_related takes in a person tag and the tweet and checks the knowledge base of completed transfers first, to check if this player exists, if it does the knowledge base will return the information about the "from_club", and "to_club", as illustrated in step 4 in Figure 11. The function will then use the to_club to extract club aliases and use it to check if there is

match. Finally, it returns the to_club, sign date and true if the club matched else false to the entity linking function

The entity linking function then checks the result from is_related if it is true, it then checks official sign date of the player to ensure that this not an announcement after the signing, as shown in Figure 12

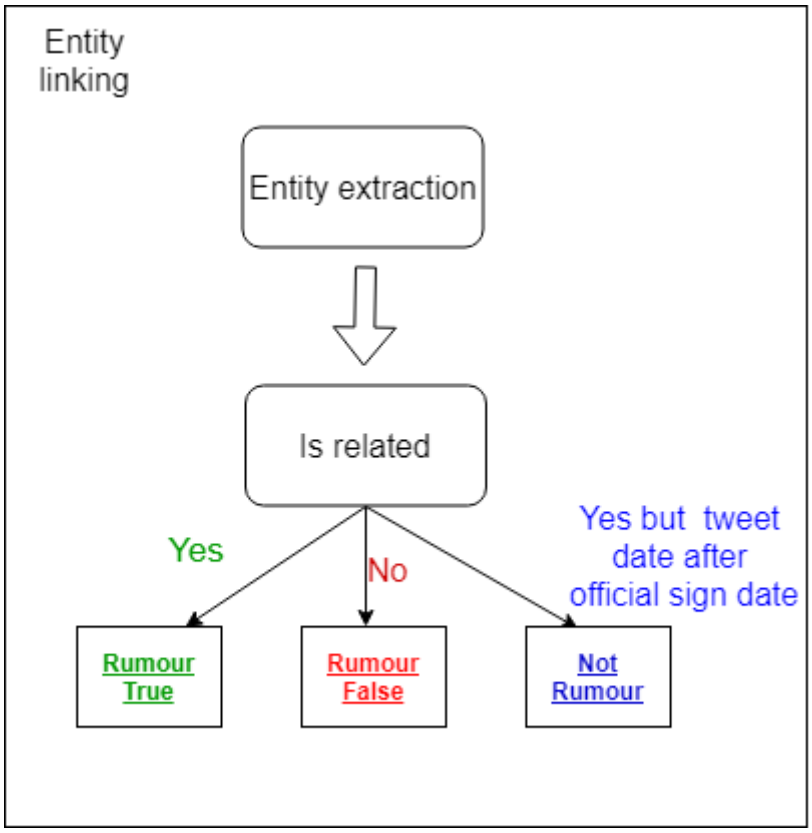


Figure 12 Entity linking function

If the player is not found in the KB, Is_related function tries its best to return the accurate to_club. This scenario describes the case where it's possibly a false rumour. Here the tweet is tokenised and each word and a bigram - current word and the following word to incorporate club names that are two words is checked to see if a premier league club is mentioned. The club and player mentioned in the tweet is checked in the EPL squad knowledge base (section 3.1.3.2) to ensure that the player does not belong this club. This is done prevent linking a player to his current club again.

Returning the to_club to the entity linking function is important because the analysis function collects metrics based on the presence of a to_club and is equally important to

make sure the correct mapping is done. The assumption here is that the tweet itself is a rumour. Hence there is less chance of incorrect mapping.

The entity linking function has two policies one is to check the second word is also a person and include that in the name. Otherwise, just use the current name. The first is to take into account if first and second names are in a tweet, second is to cater for instance - where player are referred by their second name

Failed mappings of the is_related function are stored in a dictionary with the key the name NER found and value the to_club the is_related function found. This done because the NER sometimes tag incoherent text as player names and a club may be associated with this. To avoid returning incorrect mapping, the function iterates through the dictionary and checks the name of the NER found player in the knowledge base of players (section 3.1.3.2), the first match that is found is returned. This only occurs when the rumour is false. Otherwise it would have found to_club.

The entity linking discuses different initial approaches to alleviate the NER's problems with Twitter data.

Approach	**Discussion**
Just using NER	**Advantages** • The process is much simpler: if in the KB then true else false **Disadvantages:** • Noisy NER detected may have player information that could overlook No way to find the **to_club** accurately for a **false rumour**
Iterating through EPL squad 2016 + completed transfers player names	**Notes** • Finding a player name helps reduce labelling noisy tweets. • This approach used NER and substring method to find a player **Advantages**: • Transfers rumours about for example Alexis Sanchez (rumoured to move to Manchester City) are detected as false, and to_club will be accurately found ○ Previously it would have detected it as false, but the incorrect rumours for club metric (in this instance Manchester city) could not be calculated. **Disadvantages:** • The combination of NER and the substring method for player names will find very odd matches. For example, NER

	detected “son” as a player name (not in the context of Tottenham player “Son Heung-min”) CallumWillson got matched as it has “son” in it.

Table 6 Previous approaches to entity linking

The current implementation does not use the substring approach as it results in many incorrect matches

Using a hash table like data structure substantially improves the performance as the lookup is of the order O(1). This is crucial given the informal nature of tweet; every word will have to be checked for a match.

4.2.3 Sentiment analysis

This section outlines the how sentiment was calculated for the tweets

4.2.3.1 Gcloud Sentiment analyser

For this task, a service key with permissions to access the Google Cloud Natural Language API was set up on the Google Cloud console. Then scripts were written to access the Gcloud API using the key. Finally, the sentiment of all the tweets was gathered individually using the SDK and written into a CSV file.

The scores were normalised as mentioned in section 3.2.4. The tool, however, sometimes produced exceptions this was because the text was too small for the analyser, this case -2 was the output. This will allow the metric collection tool to exclude this observation as it could not find the sentiment.

4.2.3.2 VADER

VADER provides a very simple mechanism to access the sentiment analyser. The SentimentIntensityAnalyser class has a method called polarity_scores. The text is simply passed into this function, and it returns positive “pos”, negative “neg”, neutral “neu” and compound scores. Compound represents the overall sentiment of the tweet, which is computed by “summing the valence scores of each word in the lexicon, adjusted according to the rules, and then normalised to be between -1 (most extreme negative) and +1 (most extreme positive” (Hutto & Gilbert, 2014)

Only the compound value was used as it will suffice for the scope of this project.

4.2.3.3 *Stanford Core NLP (SCNLP)*

SCNLP was setup as server. To interface it *pycorenlp*[39] library, which is a wrapper that allows easy access, was used. The *pycorenlp* class constructor accepts the URL of SCNLP server and has methods like "annotate" which accepts text and properties. The properties describe which annotator to use (here sentiment) and the output format.

The SCNLP the setup was relatively straightforward. SCNLP project was in a set of Java jars. The only other requirement is having an up to date Java version. The following command was used to start the server

java -mx5g -cp "" edu.stanford.nlp.pipeline.StanfordCoreNLPServer -timeout 10000*

4.3 Modelling Phase

This section covers the implementation of machine learning models that used the data from the Annotation phase.

4.3.1 Sklearn

Scikit Learn also known as Sklearn, is an open source Python module that has integrated state-of-the-art machine learning algorithms (Pedregosa, et al., 2011). It supports various classification and clustering algorithms which include support vector machines, random forests and, k-means. The primary reasoning behind this choice is the ease of use. It has integration of python scientific computing libraries such as Numpy and Scipy. This library

[39] https://github.com/smilli/py-corenlp

is largely written in Python and parts of it are return in Cython[40] which is designed to give performance similar to C language.

In this project, the Support Vector Classifier and Random Forest was employed. Most of the machine learning algorithms have functions called *fit* and *transform*. The fit methods learn the model parameters and transform method applies the model on unseen data.

In this project one of the most important features is text. The first step is to integrate the textual feature. As mentioned section 2.6 the textual data should be transformed into vectors for the machine learning algorithms before that, textual features first go through pre-processing step outlined in section 3.2.6. This is to clear unnecessary content.

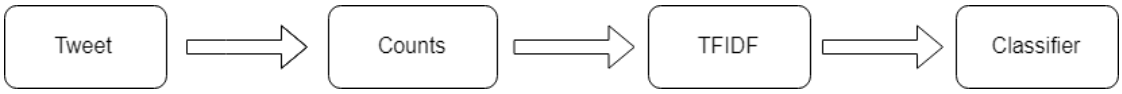


Figure 13 Pipeline steps

CountVectorizer function in Sklearn converts textual data by tokenising it and representing the information in a matrix of tokens with counts of each token/word.

This process is similar to the bag-of-words approach as illustrated in section 2.6.1. This function also accepts a parameter for the size of ngrams. For this project unigrams, bigrams and trigrams were used. The use of trigrams was inspired from Kampaki et al. suggestion that it would be interesting to try trigram gave better results, as increasing the grams gave them better results

TFIDF accepts the counts from the above CountVectorizer and converts each tweet into a vector based on TF-IDF calculation.

The goal of this task is to create a model that uses Twitter features to capture the rumour constructs that will help predict the veracity of a new tweet. Therefore, the above section only describes how to model the text; the next step is to incorporate features like retweets, favourites, and overall sentiment.

[40] http://docs.cython.org/en/latest/src/quickstart/overview.html Accessed on 01/05/2018

Table of features used

Feature name	Discussion
Text	The tweets, vectorised
Retweets	Number of retweets may indicate how popular or informative the tweet is
Favourites	The popularity of the tweets
Overall sentiment	The overall sentiment of the tweet. Intuition is that if the account is confident about a rumour, then it would be more positive

4.3.2 Sklearn Pipeline

This a Sklearn functionality makes it easier to chain multiple estimators into one. This was particularly useful when there is a fixed sequence of steps in processing data. Moreover, it has a clear declarative interface where it is easy to inspect the model. In this project, it was used for feature selection, normalisation, feature union and classification. The convenience was the most attractive point about this. After constructing this pipeline, all that needed to be done is to call function fit with the training data and predict with the test data

The above-described function was inserted into the pipeline. In order to incorporate other features, Feature Union functionality was used. Feature Union concatenates results of multiple transformer objects[41]. Hence, this mechanism provides a natural approach to append further features in the future.

[41] http://scikit-learn.org/stable/modules/generated/sklearn.pipeline.FeatureUnion.html Accessed on 01/05/2018

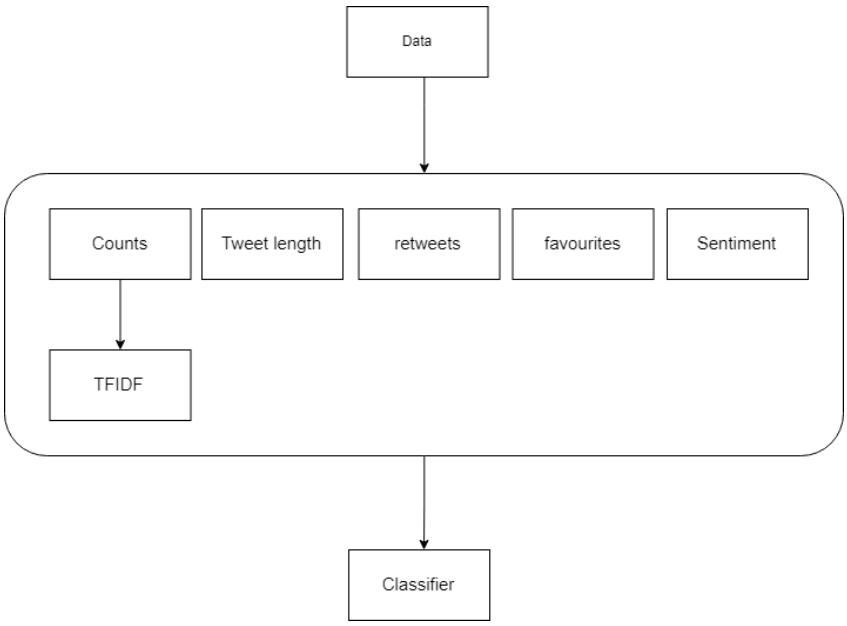


Figure 14 Feature Union[42]

However, there is a problem because, all the data is passed, i.e. text and other features, and text should be treated differently to others. Therefore, there needs to be a way to specify which part of the pipeline should be applied to what type of data.

As a result, an ItemSelector class from here[43] was used. It inherits BaseEstimator, and TransformerMixin. Only implemented the fit and transform functions. The fit function returns itself, and transform function takes in grouped data (is a dictionary), and returns feature extracted out from data using key passed in the constructor. This provides a clean and efficient way to select a subset of data at a provided key.

4.3.2.1 Classifiers

Once the features are combined the last component to be inserted into the pipeline is the classifier.

[42] zacstewart.com/2014/08/05/pipelines-of-featureunions-of-pipelines.html Accessed on 01/05/2018
[43] http://scikit-learn.org/stable/auto_examples/hetero_feature_union.html#sphx-glr-auto-examples-hetero-feature-union-py Accessed on 01/05/2018

As stated earlier multiple different classifiers were used, as a result a pipeline each was built for them. Also, within it different n-grams.

LinearSVC which is C-Support Vector Classification[44] (which uses linear kernel) was used. It was used with default parameters, which includes *class_weight="balanced"*. This is one method to deal with imbalanced classes in SVC by increasing the penalty for misclassification of minority classes. The following formula adjusts the weights:

$$w_j = \frac{n}{kn_j}$$

where w_j is the weight to class j, n is the number of observations, n_j is the number of observations in class j, and k is the total number of classes.

Figure 15 Class weighting formula[45]

Random Forest Classifier is an ensemble method which means it creates multiple models which learn and make predictions independently, in this case multiple trees trained on various sub-samples of the dataset. These predictions are combined through averaging to improve the predictive accuracy and control over-fitting.[46]

Caching

Machine learning is an iterative process, for instance, there might not be any change in the data cleansing part. Therefore it would be inefficient to repeat. Furthermore, it takes a significant amount of time as well. To address this issue, the cleaned dataset was serialised, in other words, the object containing data was converted into bytes and written to file. When needed this file was loaded back as an object.

[44] http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html Accessed on 01/05/2018
[45] https://chrisalbon.com/machine_learning/support_vector_machines/imbalanced_classes_in_svm/ Accessed on 01/05/2018
[46] http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html Accessed on 01/05/2018

4.4 Tools

4.4.1 Pip

Pip is an OS-independent package management tool used to install and manage software packages written in Python[47]. This tool was used install all the dependencies of the different components in this project.

Dependencies are all listed in a file called “requirements.txt”. All the dependencies could have been installed using following command

pip install –r requirements.txt

4.4 Challenges encountered

Numerous challenges were encountered during the implementation phase. Largely it was due to the noisy nature of Twitter data

- Disambiguation: For instance “'Transfer news: Roma want Everton target Rachid Ghezzal to replace Liverpool-bound Mohamed Salah http:// dlvr.it/PNxmBR” . The current version fails here. There is a considerable amount to disambiguate the NER detected: mohamedsalah, to_club: Liverpool. This tweet was labelled true because Mohamed Salah moved from Roma to Liverpool during 2017 transfer window, however clearly the intention was not to make claim Mohamed Salah is moving.
- Clubs are referred to by their home ground names and can be a hashtag as well, e.g. Chelsea, referred to as Stamford Bridge.
- Due to the vast number of tweets, there may be some mislabelling, and only one person labelled the data.
- Not able parallelise the sentiment analysing process because of limit on API call on GCloud.
- **Problems with the dataset**
    - The delimiter used is a semicolon, however, some tweets contained this delimiter there was a number of them and therefore decided it was best not

[47] https://en.wikipedia.org/wiki/Pip_(package_manager) Accessed on 01/05/2018

consider those tweets. These were very few tweets less than 50 tweets out more than 58,000. Moreover, these tweets were not related to the transfers

- Some of the accounts were either deleted or suspended by Twitter towards the final stages of the project.

5. Evaluation

The following sections discuss the outcomes of the project and how they help answer the research questions. This chapter begins by outlining the approaches taken to evaluate the results, then discussions on it and finally future work.

5.1 Approach

There are two parts for evaluation first is to gather general metrics followed by the evaluation of components' performance.

- **General metrics**: these include the accuracy of the accounts, the number of followers, number of retweets, number of favourites/likes and sentiment expressed in tweets
- **Component Performance:** performance of the rumour detection and rumour veracity checking components, and the machine learning approach to veracity checking

5.1.1 General Metrics

The analysis was done from the perspective of rumours about players moving to the EPL. Two types of metrics from the data was calculated, general and sentiment

The general category contains the metrics for each account and is calculated as follows: for each to_club identified, how many are labelled true rumours (1), false rumours (2) and non-rumours (3), and overall counts of each label were represented in a JSON format.

Meanwhile, sentiment category contains the average sentiment scores for each of three labels.

5.1.2 Component Evaluation

One of the primary objectives of the research question is to find out the account that is best at accurately predicting player transfers. However, the performance of the components will be a crucial indicator, on how trustworthy the results are.

The key components to be tested are rule-based systems for rumour detection, rumour veracity checking and the machine learning classifiers for rumour veracity checking. These components were evaluated using the Precision, Recall and F1 measures. For the rule-

based systems, 1000 tweets were randomly sampled, applied the systems and then manually annotated the ground truth. This was then compared against system annotated, using Sklearn's function called classification report[48]

Since the machine learning algorithm used the data annotated by the veracity checking system as the ground truth. The whole set of tweets annotated by the system was used with 70/30 which were randomly sampled with similar proportions in the test and train sets.

$$\text{Recall} = \frac{tp}{tp + fn} \quad \text{Precision} = \frac{tp}{tp + fp}$$

Figure 16 Formulae for precision and recall[49]

Precision describes how well the system found the relevant instances among the retrieved ones**.** Recall tells how many accurate results are found.

$$F_1 = \frac{2}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$

Figure 17 F1 score formula[50]

F measure is the harmonic mean of the recall and precision. This can be used a single number to describe the performance of the system.

5.1 General Statistics

This section discusses the accuracy of accounts studied in this research. TransfRumours, TransferMoves, TEAMtalk, Now__Football, and Jonawils were excluded from the analysis as they had fewer true or false observations and will skew the results.

---

[48] http://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html Accessed on 01/05/2018

[49] https://en.wikipedia.org/wiki/Precision_and_recall Accessed on 01/05/2018

[50] https://en.wikipedia.org/wiki/F1_score Accessed on 01/05/2018

TransferRelated and EPLMoves were also removed as they were suspended by Twitter towards the final stages of the project.

5.1.1 Account level Accuracy

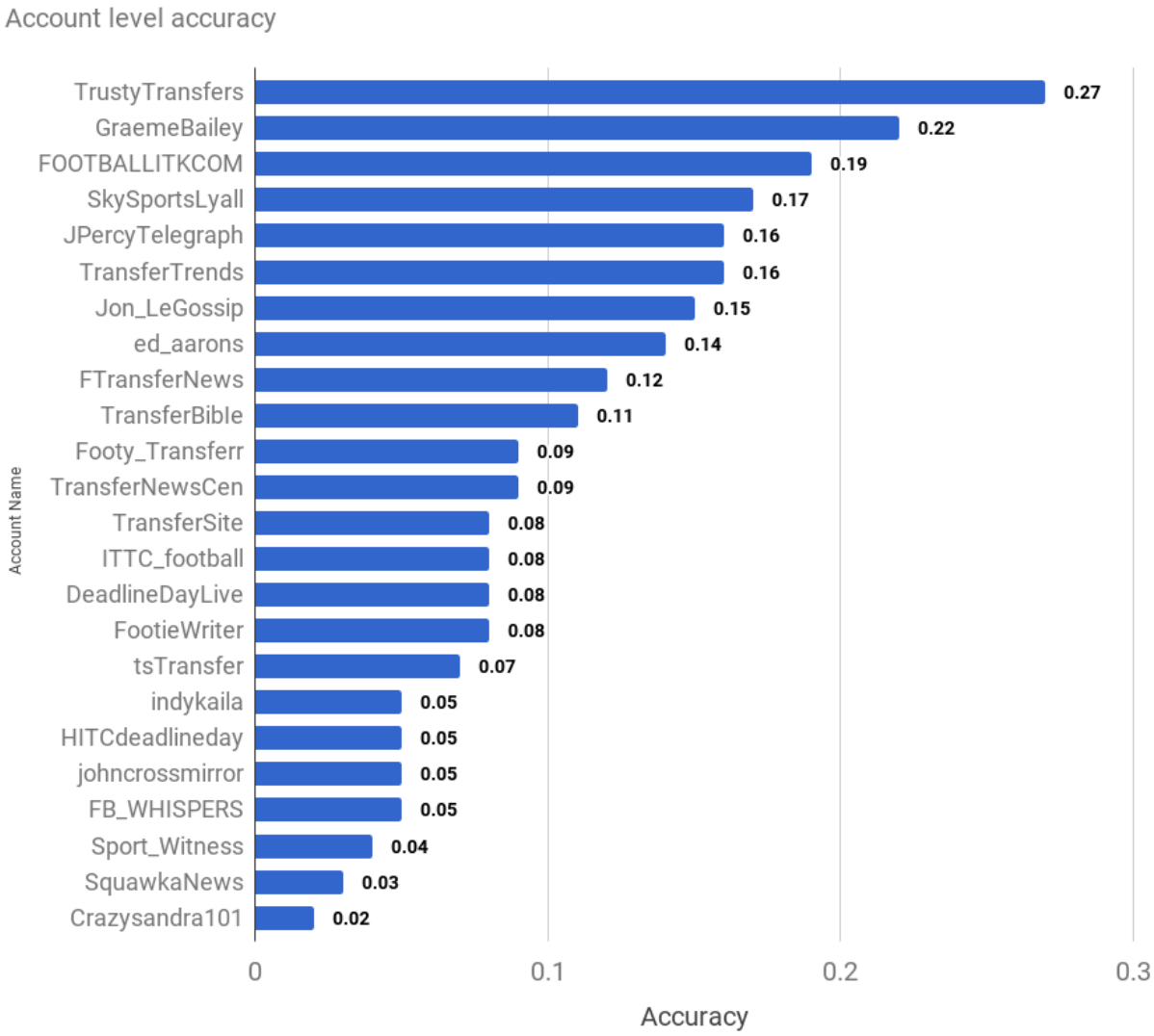


Figure 18 Accuracy of accounts

The above plot shows the account level accuracy according to the rumour veracity checking system. The x-axis shows the account names and the y-axis the accuracy in percentage. It can be seen that “FOOTBALLITCOM”, “GraemeBailey” and interestingly “TrustyTransfer” have the highest accuracy. There are some journalists that work for newspapers, and sports websites such as the Daily Telegraph and Sky Sports among the top

results. This is not surprising as they would be mindful of the fact that false content may damage their reputation.

5.1.2 Follower Count

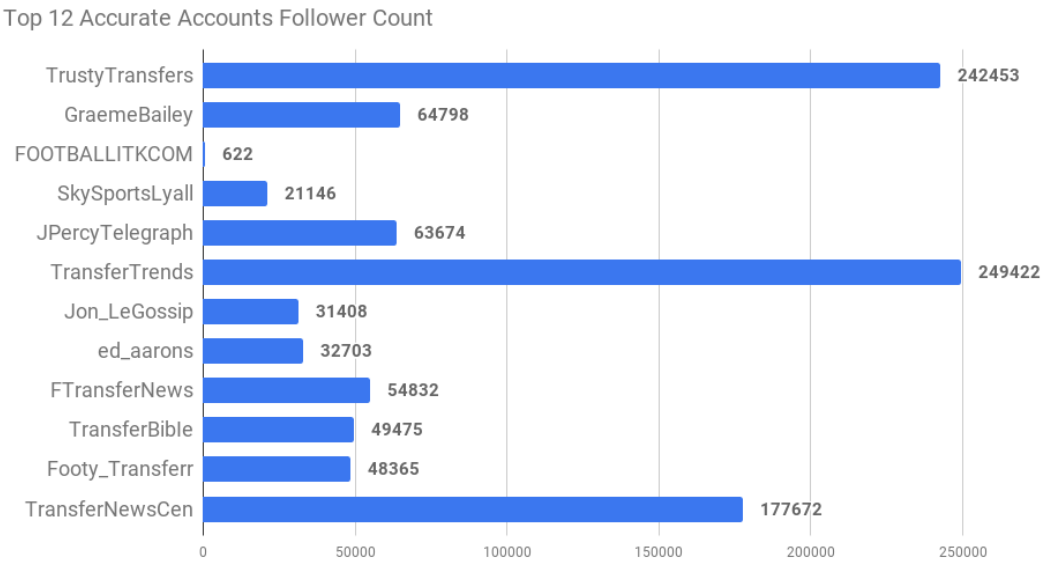


Figure 19 Follower count for top 12 accounts – sorted by accuracy

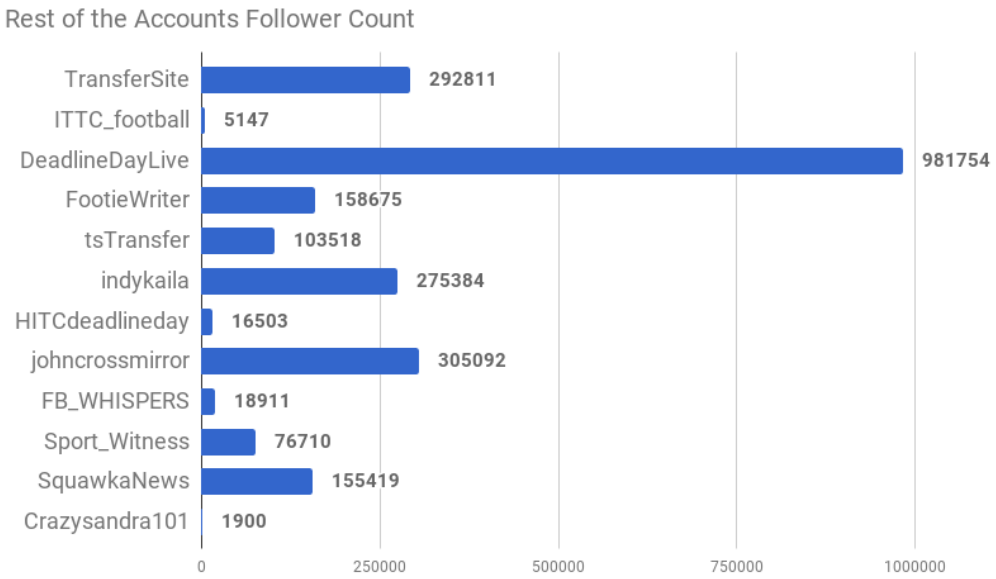


Figure 20 Followers per account – sorted by accuracy

Another possible relation could be between the follower count and the accuracy. The assumption made is that fans would follow accounts that are highly accurate. From the above two plots, there doesn't seem to be a clear-cut. For example, Deadlineday has a higher number of followers but not many accurate results. FOOTBALLITKCOM, on the other hand, has a fewer follower, however, have higher accuracy.

There can be other reasons for high follower count; the authors may frequently be appearing on sports shows

5.1.3 Favourites and Retweets

The reasoning behind exploring the relation between favourites and retweets is that false rumours tend to be more sensational, fans get excited when they see that there is a chance of an interesting player transfer. Retweets are used to repost or forward the message.Favourites/like are used to endorse a particular tweet.

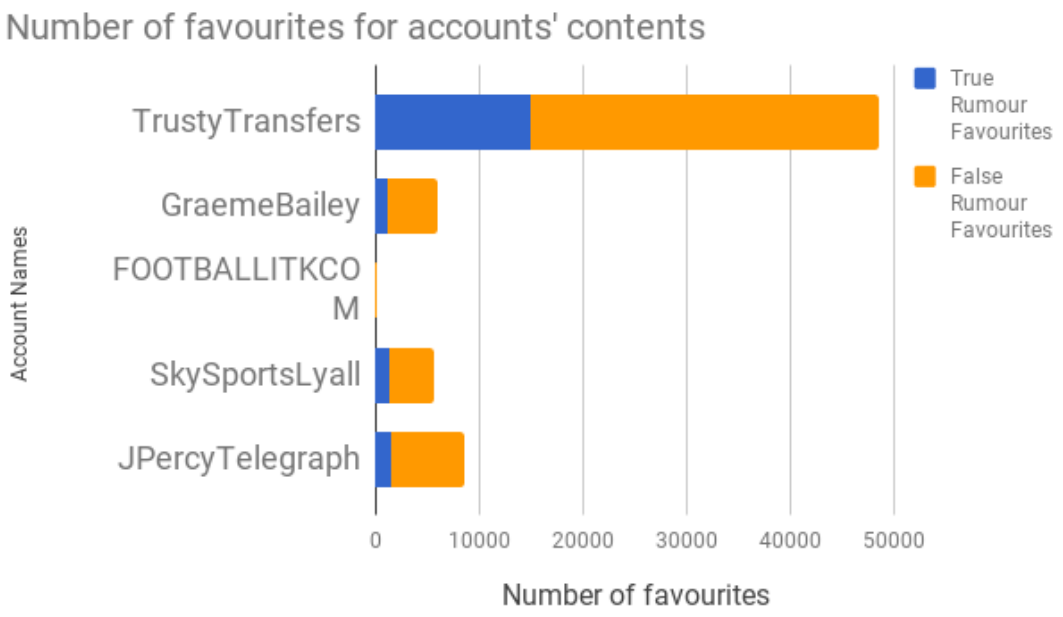


Figure 21 Number of favourites for top 5 accounts

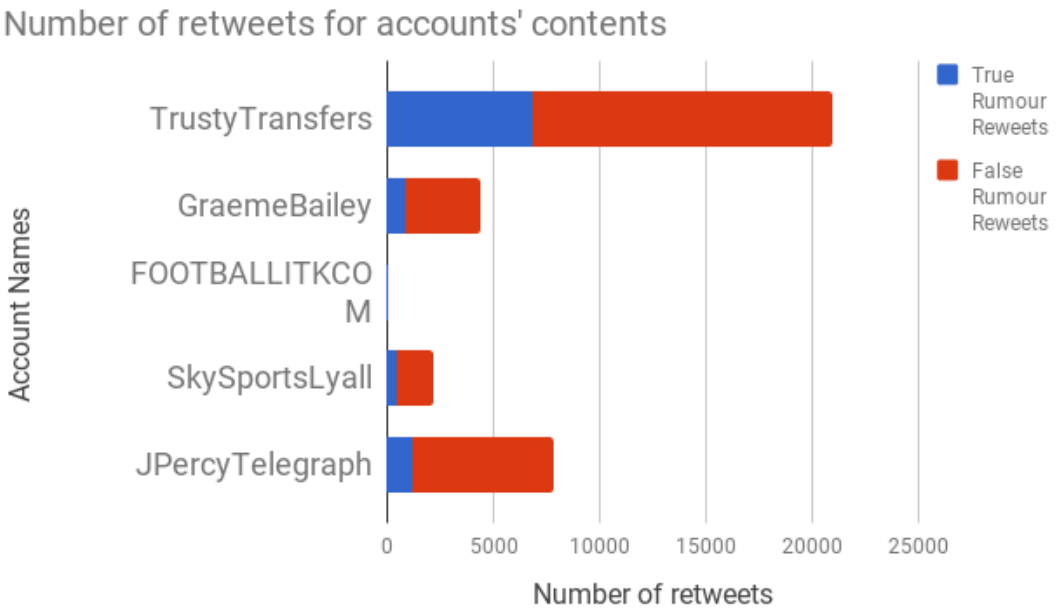


Figure 22 Distribution of number retweets

From Figure 21 there is a clear indication that there are substantially more favourites/likes for false rumours than the true rumours. Probably because they more attention-grabbing than true ones.

This likely shows either the fans excitement or wishful thinking. It needs to be mentioned that the number of followers also influence this count that is why the FOOTBALLITCOM is not in the frame in this plot. In light of the recent study by (Vosoughi, et al., 2018) where they found that false information propagates more quickly than true information, the results are inline with their findings.

5.1.4 Influence of Sentiment

This section discusses the impact of sentiment on the accuracy of player transfer. The sentiment of rumours that came out to be true was averaged out for each account separately, similarly for false rumours. The reasoning behind the exploration of sentiment in tweets is that accounts may use more positive words to describe a player transfer that is highly likely to occur. On the other hand, a tweet may express negative sentiment about a

potential transfer as shown in the following tweet where Romelu Lukaku joined Manchester United.

Tweet*: "NO DEAL! Everton sources tell talkSPORT they have NOT accepted £75m offer from Manchester United for Romelu Lukaku http:// dlvr.it/PScGrn"*

However, tools used to harness sentiment has returned a quite varied range of scores, there is no agreement between them. The SCNLP produced overwhelmingly negative scores, GCNLP mostly neutral, and VADER neutral and positive. However, when compared with true rumours and false rumours, there is an agreement all of the tools produce very similar as illustrated in Figure 23. Hence, nullifying the assumption that there is a clear distinction between false and true rumours. The scores for SCNLP can be found in Appendix A

The large proportion of neutral scores are probably because most of the tweets are just a statement, which is perhaps a speciality of accounts studied. Accounts such as tsTransfer ,DeadlineDay, tend not to give an opinion.

Tweet: "*Manchester City have offered £70m plus defender Jason Denayer to Arsenal for Alexis Sánchez. (Source: Daily Star) pic.twitter.com/klm85mybAH"*

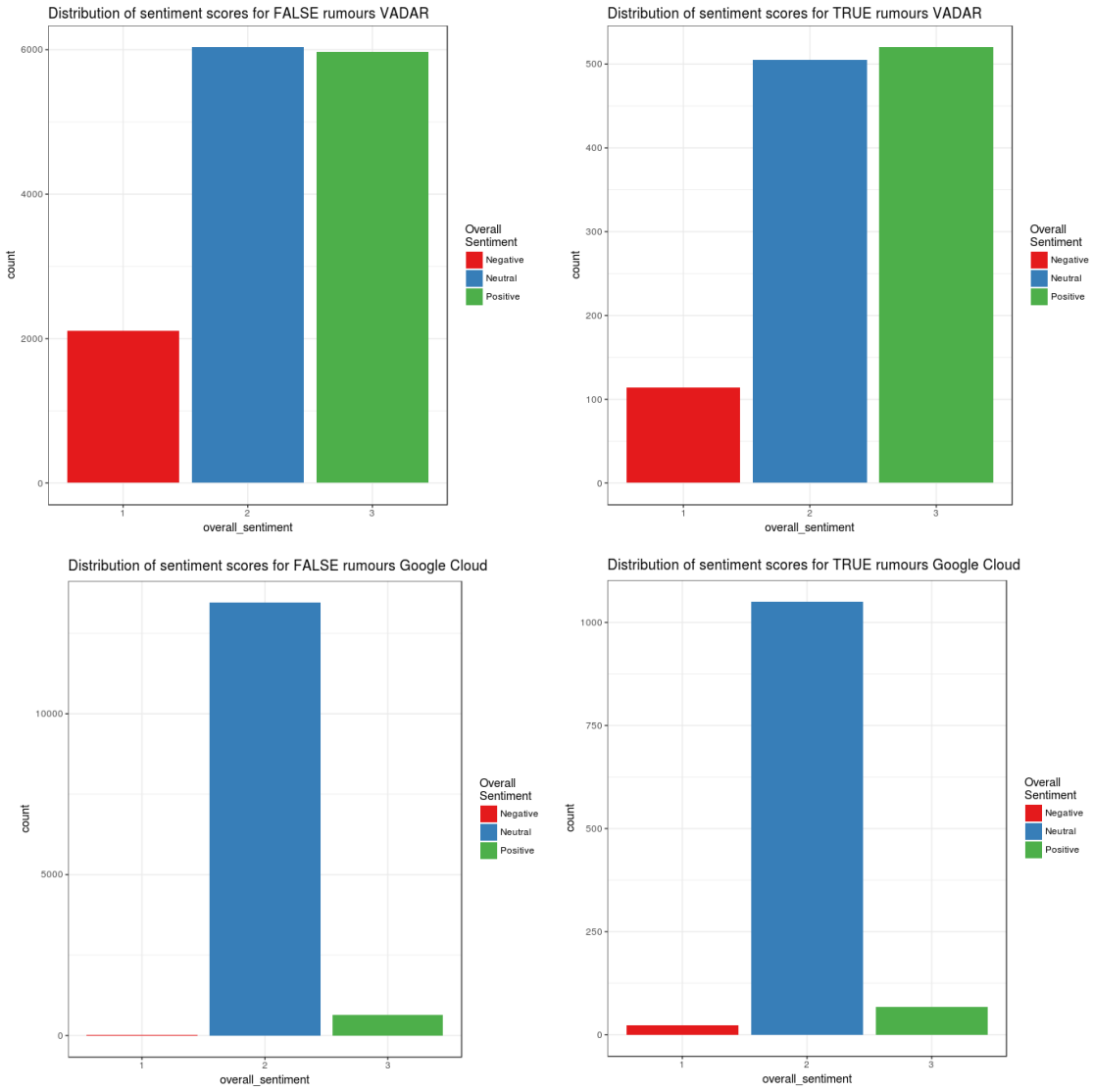


Figure 23 Distribution of sentiment scores of true and false rumours for GCloud and VADER. The top left shows the distribution of sentiment scores for False rumours using VADER sentiment analyser and top right sentiment scores for True rumours. Similarly for GCNLP sentiment analyser false and true rumour sentiment scores distribution.

From the above plots Figure 23 Distribution of sentiment scores of true and false rumours for GCloud and VADER it can be seen is not an influential feature, as there is no clear inclination towards true rumours or false rumours. Another reason is Twitter data, it is often very short and may not have the content to help the tools used in this project to gather sentiment.

5.1.4.1 Stanford Core NLP Sentiment Analysis - problems

SCNLP sentiment analysis scores were overwhelmingly negative. Further online research and examination of data reveal that some names tend to cause the overall sentiment to be negative. This is not isolated to this project, as noted by [51] and [52]. The likely explanation found on Github is that these names do not appear often enough in the training data and the decisions made by the neural network in analysers are not always explainable.

Example:

Tweet: "*Crystal Palace have had their bid for defender* ***Mamadou Sakho*** *accepted by Liverpool CPFC LFC"*

Removing ***Mamadou Sakho*** results in a neutral sentiment which is indicative of the tweet.

Another reason is that the system has particularly an issue with sentences starting with the word "Done". Consider the following tweet which upon inspection it can be said to be neutral. However, the system labelled it be negative. However, taking out the word "*Done*" results in the system outputting a neutral score

Tweet: Done Deal Marvin Zeegelaar joins WatfordFC from Sporting Lisbon on a four-year deal DeadlineDay

Phrases like "Done Deal" is quite common in the dataset and using Stanford's system will result in many being labelled negative.

---

[51] https://github.com/stanfordnlp/CoreNLP/issues/351 Accessed on 22/05/2018
[52] https://stackoverflow.com/questions/42027119/stanford-nlp-sentiment-ambiguous-result Accessed on 22/05/2018

***Tweet**: Former Coventry City goal scorer Gary McSheffrey has signed for National League side Eastleigh FC #PUSB #CCFC pic.twitter.com/pcgBncbg7*

To make the same test conditions for the three tools, no pre-processing such as removing Twitter features hashtags and also images were employed. However, this is mainly an issue for SCNLP, as they cause the sentiment to be negative. However, removing them provided more accurate sentiment. This illustrates that the SCNLP is not suitable for Twitter data without cleaning the text.

5.2 Performance of the components

This section will begin evaluating the performance of the rumour detection system and the rumour veracity checking system. Followed by a discussion on the machine learning component.

5.2.1 Rumour Detection

In this project rumour detection is treated as a binary problem, checking whether a given tweet is transfer information and a rumour or not. The following scores illustrate the performance of the system after the sampling and annotation process

Classes	Precision	Recall	F1-Score
Not a transfer rumour	0.95	0.86	0.91
A transfer rumour	0.85	0.95	0.90
Weighted Average	0.91	0.90	0.91

Table 7 Performance of rumour detection component

5.2.1.1 Discussion:

Having considered that the sample was only 1000 tweets, this system performs well in on this particular dataset. The keywords described in section 3.2.1 have been shown to be reasonably good at labelling the rumours. However, this system is built using tweets from the selected accounts and may fail to generalise when this system is applied to a new set of accounts. As it may occur that additional set of keywords would be needed to detect accurately. Tailoring towards this set of tweets explains the high precision rate.

However, there are some instances where the rumour detection fails as demonstrated in the following scenarios:

Tweet*: "Alexis Sanchez will be Manchester City player in the coming days. #AFC #MCFC".* This tweet should have been labelled as a transfer rumour, however, was not because there were no transfer rumour keywords identified.

Tweet*: "Interested in becoming a transfer news writer for http:// transfernewscentral.com ? Apply here now! https:// a.quil.la/4T64VFI pic.twitter.com/E8my24LBf2".*

This was mislabelled as transfer tweet; the word "*transfer*" is the offending word, perhaps it could be removed from the keywords but could lead to other tweets being mislabelled.

Tweet*:* "*Liverpool source: Talks are taking place for a Michael Edwards statue outside club shop. #LFC*". This tweet is not about the transfer; it is a sarcastic tweet expressing displeasure of Michael Edwards conduct as Liverpool's sporting director. Words "*Talks"* and "source" are often used by most accounts to describe a potential transfer. Therefore these keywords cannot be left out.

Some of these issues can be mitigated by having weights for features for example "*Talks",* and "source" can be given extra weighting if a player name is detected using entity recognition and entity linking to find out if this player is an active player

A machine learning approach could be investigated as the rules used in this system could become obsolete, and as a result, new rules have to be engineered. In the case of machine learning, only the data needs to be changed.

There needs to be extensive knowledge base to help differentiate the nuances in the tweets that may not be about transfers. It would be interesting to see if the complex semantic learning such as word embeddings would be able to detect that it is not about transfers

5.2.2 Rumour Veracity Checking

Results

Classes	Precision	Recall	F1-Score
True Rumour	0.94	0.73	0.82
False Rumour	0.95	0.99	0.97
Not a rumour	0.99	0.98	0.99
Weighted Average	0.98	0.98	0.98

Table 8 Performance of Rumour Veracity Checking- rule-based system

The above table represents the result of the rumour veracity checking tool. It shows high precision for all the classes.

Discussion

The high precision for True rumours is indicative of the data as most rumours tend to name the to_club along with the player name. Hence, the system can identify the association. However, this also highly dependent on how well the Named Entity Recognition finds the player.

5.2.3 Machine Learning component

Machine learning algorithms Random Forest and SVM employed in this project to predict veracity of rumours have performed poorly. This was implemented as binary classification true_rumour or false_rumour, with unigram, bigram and trigram models for text. Both algorithms produced very close results and had extremely low precision for true rumours. The results were evaluated using the 70/30 split. The n-gram models got the same results, and the results for bigram and trigram are in the Appendix B section

Results

Random forest -unigram

Classes	Precision	Recall	F1-Score	Support

True Rumour	0.12	0.06	0.09	298
False Rumour	0.93	0.96	0.94	3846
Weighted Average	0.87	0.90	0.88	4144

Table 9 Scores for Random Forest unigram model

SVM - unigram

Classes	Precision	Recall	F1-Score	Support
True Rumour	0.12	0.02	0.03	298
False Rumour	0.93	0.99	0.96	3846
Weighted Average	0.87	0.92	0.89	4144

Table 10 Scores for SVM unigram model

Discussion

12,788 tweets were labelled as false rumours and 1,025 tweets labelled true rumours. The skewness of the data plays a vital role in the poor results. Also, the features used did not provide the algorithms with the right signals that would have helped to identify intricate structures which discriminate the two classes.

The features exploited by the ML algorithms were the number of retweets and favourites/likes, text, and overall sentiment which were thought to be useful indicators. However, analysis from 5.1.4 illustrates that sentiment is not a good indicator as they do not have a clear correlation to the rumour either being true or false. The number of favourites and retweets are good indicators. However, their signals seem to be drowned by the textual features. Furthermore, the textual features may not be good indicators as previously thought.

KEYWORDS/PHRASES	DISCUSSION
ACCEPTED	Player or club accepting conditions for a transfer
COMPLETES	alluding to player completing the transfer move
DONE DEAL	Similar to *complete*
MEDICAL	Occurs at final stages of the signing. As discussed in section 3.2.2

Table 11: Good textual indicators for rumour veracity checking

The Table 11 illustrates some of the good textual indicators where found by manual inspection to be common for true rumours than false rumours

KEYWORDS/PHRASES	DISCUSSION

REPORTEDLY	More often than not used in rumours about a potential move.
AGREED	Indicating both player and clubs have agreed to a deal.
INTERESTED	Often indicate the initial stages of a possible transfer
CONFIDENT	Similar to *interested*
CONFIRMED	Similar to *agreed*

Table 12 Common words in true rumours and false rumours

This table illustrates the problem where good textual indicators about potential transfers tend to occur in both false rumours and true rumours tweets. Moreover, as mentioned earlier the noisy nature of the Twitter data further contributes to the problem by making the feature set sparse and hence drowning signals from phrases shown in Table 11**.** The sparsity is arisen from conversion of text to bag-of-words to vectors. This also helps explain why there was no significant difference in performance of the n-gram models.

However, the algorithms have shown in state of the art to perform better with more complex features (Castillo, et al., 2011) and also can be improved by doing hyper parameter tuning on the algorithms. Future research should consider employing more complex feature sets and algorithms. A further discussion on features sets is conducted in the section 5.3.7

5.3 Summary of results

In summary, the features like follower count, and sentiment contrary to expectations did not provide meaningful insight into their impact on the veracity of rumours. However, there seems to be a pattern that false rumours tend to have a higher number of retweets and favourites, which is in line with the study by (Vosoughi, et al., 2018) where they found that false information travels faster than true information on Twitter.

It has been shown that sentiment is not a particularly useful indicator of how accurate the accounts are. Alternatively, a better feature to explore is the stance towards a particular rumour, i.e. are they against or for.

Since VADER is specifically tuned for Twitter data, the result seems to pick this up as they have more positive and neutral. Whereas GCNLP scores are mostly neutral and not that informative. Finally, this is a small study therefore only limited inferences can be made.

The components were evaluated using precision and recall scores. The rule-based systems were tested by randomly sampling 1000 tweets and manually annotating them. Followed by calculating the scores. The scores for each of the rule-based components have shown to have high precision and recall.

Machine learning component performed poorly mainly because the features were not informative, and the dataset was significantly imbalanced.

The results of accounts used in this analysis is available in Appendix C

5.3 Limitations and Future work

The following sections identify limitations of the project and propose solutions to them as part of future work.

5.3.1 Fine Grain rumour definition

There are numerous issues with current rumour definition:

- The current definition does not consider the stance in the tweets such as are they supporting or not? The current definition is not capturing the accurate semantic meaning of the tweet. Just by looking at player name and club is just a start. A more fine-grained approach should be considered for future work. Consider this following tweet: *"Swansea reject £27m bid for Gylfi Sigurdsson from #Everton (ESPN) http://www. espn.co.uk/football/socce r-transfers/story/3148236/swansea-reject-27m-pound-bid-for-gylfi-sigurdsson-from-everton-sources …"*
    - Gylfi Sigurdsson, a Swansea player, was reportedly linked to Everton, where he eventually joined. However, in this work, we are only validating the linking part, not understanding the semantic meaning. Here, the tweet talks about Everton's bid being rejected, this information is as it could be a possible indicator that the player might not move.

- As mentioned in chapter one a player might be linked to the club and may have been in talks. As a result, the rumour is true at that time but have joined another club. However, we do not know if it was accurate.
- There are transfers like Naby Keita move to Liverpool for example where the clubs have agreed on the player transfer for the following year 2018/19 window[53]. However, the deal was completed during the 2017/18 window. As a result this was considered a false rumour as it did not appear in official transfer window list.
- Often it can be seen that twitter accounts reference other accounts about transfers. In the context of this project, it is considered as referee making the statement. However, a future system should take this difference into consideration.
- Rumours about a player can be double counted. An account may talk about player and club multiple times. In this project, it is counted as different rumours. New research should consider, incorporating the stance expressed in the tweet. For example, an account makes a strong claim but may reduce the severity of the claim at a later stage, which is crucial information that could lead to better veracity prediction.
- There are potentially more nuances that have to be investigated in the future work, these include, detecting sarcasm: transfer rumour may be expressed in a sarcastic tone and wrongly attribute that to the author endorsing it.
- Often due to the conversational nature of the tweets, one tweet sometimes cannot be looked at on its own. This is particularly an issue where the player name is mentioned in a tweet and following tweet talk about a possible transfer to a club, while referring to the player as "him". The current system does not consider this. A better solution should be able to manage that to a certain degree.
- There are many instances where the URLs in the tweets give much more information than tweets. For example, "this could be interesting https://x.com/sanchez-might-join-mancity", it can be seen that the URL is much more informative. This is mainly because accounts reference other sources and it may have information about potential rumours. This information can be parsed out using regular expressions or specific rules.

[53] https://www.bbc.com/sport/football/41075272 Accessed on 01/05/2018

Finally, as outlined in the scope expand the project include other leagues and fine tune tools that considers the multilingual aspects.

5.3.2 Improving the named entity recognition

Due to the informal nature of Twitter, players are often referred by hashtags, nicknames, and second names. As a result, the entity linking process will fail.

Often, as stated in state of the art these systems are built by using annotated newspaper articles. Which has significantly different language constructs compared to social media content. This is an issue also applies to club names as well for example club Chelsea is recognised as a person rather than a club by the SCNLP NER. In this case, a special rule was set to avoid this problem.

Also, the character limit on Twitter substantially restricts the ability of the entity recognisers as there is often lack context to determine an entity's type without the aid of background knowledge. The system can be improved by first collecting text from social media, then manually annotating them, and finally training it on systems like SCNLP. By doing so, the algorithms can understand the patterns in a social media text, hence provide better accuracy. However, it would take a substantial amount of time and resources.

Moreover, often tweets contain many distinct named entities. Which are often relatively infrequent, hence even a large sample of manually annotated tweets will contain few training examples.

By improving the named entity recognition, the rumour labelling mechanism implemented here would have substantial improvement in the performance, and one can be more confident about the precision values.

5.2.3 Use of Multiple sources

Auxiliary sources can be used along with tweets to infer the veracity of a rumour for example:

- Impact of this particular tweet among the well-known newspapers or online publications, will they publish this particular rumour
- Considering an account's stance overtime towards the player

5.3.4 Larger Knowledge Base

The entity linking process is highly favoured towards detecting rumours of players within EPL as discussed in section 4.2.2.1. This is a result of the amalgamation of multiple problems in the NER, and not be able to amass all the player names.

One way to reduce this bias is by building a database of player names and their current club across top European leagues as the first step. This will improve the detection of the players, hence better labelling.

Another approach to improve the performance of entity linking other databases such as the DBpedia used by (Popov, et al., 2003) and (Ireson, et al., 2017) . DBpedia stores Wikipedia content in a structured form. An API call can be made to check for example the aliases a player might have.

5.3.5 Scaling and Parallelising

Currently, the system takes a substantial amount of time to compute the results. There are many components that can be parallelised. The use of StanfordNERTagger is highly resource consuming. Stanford's entity tagger is built in Java, and however, in this project, it is interfaced through NLTK a Python library, which requires a lot serialisation to communicate with Java component. A better, more scalable approach will entail the use of Stanford's Core NLP, that can be set up as a server and interfaced using an API call.

Furthermore, threads could be used. Multiprocessing library in Python creates a separate process which uses significantly more resources than threads which is comparatively more lightweight. Threads were not used previously because the resource intensive Java component would have a considerable effect on the performance.

5.3.6 Better feature sets for machine learning

Kampaki et al has demonstrated in their research that incorporating non-twitter data into the analysis improves the prediction remarkably. Inspired by this fact, adding in some

metric such as a player's market value, could significantly improve the inference. There might be a relationship between the release clauses and the spending power of the clubs. For example, Neymar jr's transfer from Barcelona to Paris Saint-Germain, costing more than €222 million[54] would have been only possible if the club had remarkable financial backing.

Previous spending record is also a useful indicator for the model. Additionally, club's spending capacity at a particular period during transfer window can also be used.

Other potential features could include:

- Player contract information: how long are they signing for
- Average number of goals per season: For example for a striker, average number of goals scored per season would be a good indicator as high performing players are always in demand.
- Is this an international player? – It is more likely that clubs would be interested in them. If they are capped international players, often it means that they exhibited good performance in the leagues they are currently in.
- Age of the player could be an influential predictor. As clubs will be more willing to pursue players that are early starlets or players who reached their peak than players close retirement. An example of this Anthony Martial at the age of 19 was bought by Manchester United and cost them £57.6 million[55]
- Experience in premier league: playing styles are different across leagues. Premier league clubs would want players that have experienced this style, which ensures that they can adapt quickly
- Type of players already in the squad: clubs sign players based on their skill need
- Restrictions regarding how many native and non-native players can be in the squad
- Is it financially viable for the club as explored by (Ireson, et al., 2017)

[54] https://www.aljazeera.com/news/2017/08/neymar-signs-psg-deal-complete-world-record-transfer-170803202033228.html

[55] http://www.espn.co.uk/football/soccer-transfers/story/2594161/anthony-martial-could-cost-man-united-57-6-million-monaco Accessed on 01/05/2018

5.3.6.1 Use of complex machine learning models

- Use of word embeddings: collect a large corpus of Twitter data train them into word embeddings and use it for machine learning.
- This will help for predicting future transfers
    - Provide much better generalisation for prediction
    - This is much better than the TF IDF. As it would learn the context much better
- Along with word embeddings deep neural networks could be used as they will be able to learn more complex aspects of the text, not just the count information. There is GLOVE embedding trained on Twitter data; it would be interesting to see how well it performs[56]
- Recurrent neural networks with long short-term memory LSTMS could consider the temporal features

5.3.6.2 *Use of sampling techniques*

Explore the sampling techniques described in state of the art. These techniques have often been used in cases where the dataset is imbalanced, which is true in this case here. This was not investigated in this research due to the time constraints.

5.3.8 Other Rumour Detection techniques from State of the art

Techniques such as the propagation factor, the stance of accounts could further enhance the overall system. As discussed in section 3.2.1 tweets contain more fine-grained information. Sometimes tweets could mention and might not be assertive. An interesting inference will be to consider the level of assertiveness, similar to the approach followed by (Vosoughi, 2015)**.**

Another interesting, approach would be to track the propagation of rumours. If one account mentions a possible transfer, how quickly others take it up. Some accounts maybe reluctant to put it out as it might not have enough credibility and might think this would damage their reputation

[56] https://nlp.stanford.edu/projects/glove/ Accessed on 01/05/2018

This research could be further improved and put together to form a system that will be able to say if a rumour could be true or false in a probabilistic form.

Finally, an entirely end to end system that can be built with these components, where the system's predictions can be evaluated at the end of each transfer window, this error information can then be fed into the learning algorithm. Which in turn will improve the performance overtime.

6. Conclusion

This section concludes the dissertation by assessing the results and research objectives.

6.1 Objective Assessment

Research Objective: To find accurate twitter accounts for transfer rumour predictions

The primary goal of this project was to develop a system to check the accuracy of selected accounts. The proposed system provides a method to identify the accounts that are most accurate when it comes transfer prediction

The drawback of this is that the definition of a rumour in the project is quite narrow, and as mentioned earlier rumours can be true at initial stages but turn out to be false.

Research Objective: To evaluate components of the system

The rule-based systems have high precision, this largely due to the narrow definition of the rumour set out in this project.

Research Objective: Find patterns within Twitter features

Another goal was to find patterns in the results and was able to identify that false rumours tend to have retweets and favourites, which is similar to (Vosoughi, et al., 2018)'s findings. It was shown that sentiment did not have any influence on the accuracy of the accounts. However, this was a small study. Hence very few inferences can be made.

Research Objective: examine machine learning approaches

Finally, machine learning algorithms performed poorly and can conclude that the features sets use in this study alone is not useful approach for veracity prediction. However, as shown in the literature (Castillo, et al., 2011) more complex feature can result in better performance.

Final remarks

This study contributes a method to test the accuracy of Twitter accounts in predicting football transfer rumours and proposes various approaches to improve the current system.

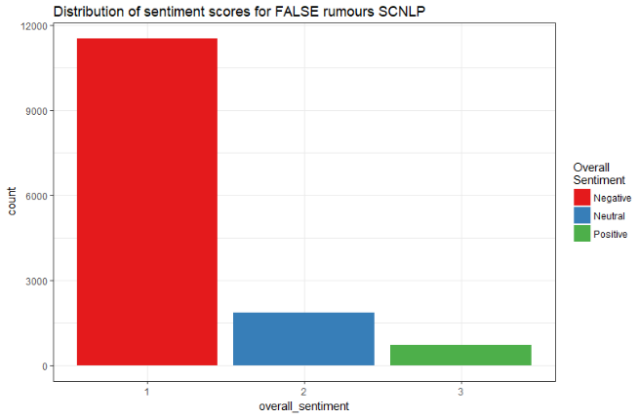
7. Abbreviations

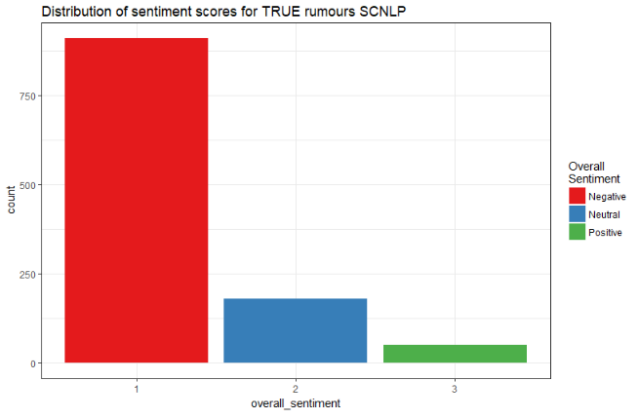
KIM – Knowledge and Information Management

EPL – English Premier League

Appendix A

Sentiment Scores distribution using SCNLP





Appendix B

Results – Machine Learning

Random forest bigram

Classes	Precision	Recall	F1-Score	Support
True Rumour	0.11	0.05	0.07	298
False Rumour	0.93	0.97	0.95	3846
Weighted Average	0.87	0.90	0.88	4144

Table 13 Scores for Random Forest Bigram model

Random forest trigram

Classes	Precision	Recall	F1-Score	Support
True Rumour	0.09	0.05	0.07	298
False Rumour	0.93	0.96	0.95	3846
Weighted Average	0.87	0.90	0.88	4144

Table 14 Scores for Random Forest Trigram model

SVM - bigram

Classes	Precision	Recall	F1-Score	Support
True Rumour	0.13	0.02	0.03	298
False Rumour	0.93	0.99	0.96	3846
Weighted Average	0.87	0.92	0.89	4144

Table 15 Scores for SVM Bigram model

SVM - trigram

Classes	Precision	Recall	F1-Score	Support
True Rumour	0.15	0.04	0.07	298
False Rumour	0.93	0.98	0.95	3846
Weighted Average	0.87	0.91	0.89	4144

Table 16 Scores for SVM Trigram model

Appendix C

Name	Accuracy	True rumours	False rumours	Non rumours	True Rumour Favourites	False Rumour Favourites	True Rumour Tweets	False Rumour Tweets	Followers count
TrustyTransfers	0.27	30	80	431	14840	33552	6807	14158	242453
GraemeBailey	0.22	82	289	995	1071	4917	837	3551	64798
FOOTAccount BALLITKCOM	0.19	7	29	136	0	2	1	0	622
SkySportsLyall	0.17	18	88	1070	1242	4420	472	1720	21146
JPercyTelegraph	0.16	7	36	74	1433	7159	1200	6610	63674
TransferTrends	0.16	30	157	297	4322	19317	1452	6316	249422
Jon_LeGossip	0.15	6	35	2830	189	683	166	649	31408
ed_aarons	0.14	14	85	650	486	7499	423	6851	32703
FTransferNews	0.12	82	582	1171	5016	33693	2660	18037	54832
TransferBible	0.11	53	422	562	3919	25073	1490	9917	49475
Footy_Transferr	0.09	9	86	185	409	3210	320	2395	48365

TransferNewsCen	0.09	100	959	2805	3663	23632	2154	13846	177672
TransferSite	0.08	46	502	510	750	7332	324	3586	292811
ITTC_football	0.08	11	121	1475	12	106	6	41	5147
DeadlineDayLive	0.08	183	2102	1345	151641	1554010	86794	853548	981754
FootieWriter	0.08	47	571	2130	1700	11651	862	5866	158675
tsTransfer	0.07	30	372	194	118	817	41	286	103518
indykaila	0.05	28	527	1145	6125	104256	3261	51951	275384
HITCdeadlineday	0.05	252	4745	3249	271	4714	573	10192	16503
johncrossmirror	0.05	8	151	1772	2044	24061	1157	17884	305092
FB_WHISPERS	0.05	7	133	660	31	638	15	217	18911
Sport_Witness	0.04	9	244	887	71	3041	61	2559	76710
SquawkaNews	0.03	17	608	9547	4748	79644	3953	54790	155419
Crazysandra101	0.02	1	45	5758	4	109	4	56	1900

Bibliography

A.Moreo, M.Romero, J.L.Castro & J.M.Zurita, 2012. Lexicon-based Comments-oriented News Sentiment Analyzer system. *Expert Systems with Applications,* 39(10), pp. 9166-9180.

Allport, G. W. & Postman, L., 1946. AN ANALYSIS OF RUMOR. *Public Opinion Quarterly,* 10(4), p. 501–517.

Auria, L. & Moro, R. A., 2009. Support Vector Machines (SVM) as a Technique for Solvency Analysis. *DIW Berlin Discussion Paper No. 811.*

Baroni, M., Dinu, G. & Kruszewski, G., 2014. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics,* Volume (Volume 1: Long Papers) , p. 238–247 .

Baroni, M., Dinu, G. & Kruszewski, G., 2014. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. *Annual Meeting of the Association of Computational Linguistics.*

Bishop, C. M., 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics.* s.l.:Springer-Verlag.

Blei, D. M., 2011. *Introduction to Probabilistic Topic Models.* [Online]
Available at: http://menome.com/wp/wp-content/uploads/2014/12/Blei2011.pdf
[Accessed 05 May 2018].

Blei, D. M., Ng, A. Y. & Jordan, M. I., 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research,* Volume 3, pp. 993-1022.

Breiman, L., 2001. *RANDOM FORESTS.* [Online]
Available at: https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf
[Accessed 17 May 2018].

Castillo, C., Mendoza, M. & Poblete, B., 2011 . Information credibility on twitter. *WWW '11 Proceedings of the 20th international conference on World wide web,* pp. 675-684 .

Cawley, G. C. & Talbot, N. L., 2010. On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation. *The Journal of Machine Learning Research,* Volume 11, pp. 2079-2107.

Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P., 2002. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research 16 (2002),* pp. 321-357.

COOPER, G., 2017. *Named Entity Recognition for Twitter.* [Online]
Available at: http://blog.thehumangeo.com/twitter-ner.html
[Accessed 16 May 2018].

Cortes, C. & Cortes, C., 1995. Support-Vector Networks. *Machine Learning ,* Volume 20, pp. 273-297.

Culotta, A., 2010. Messages, Towards Detecting Influenza Epidemics by Analyzing Twitter. *Proceedings of the First Workshop on Social Media Analytics,* pp. 115-122.

Finkel, J. R., Grenager, T. & Manning., C., 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics ,* pp. 363-370.

Firth, J., 1957. A synopsis of linguistic theory 1930-1955. *Studies in Linguistic Analysis (special volume of the Philological Society),* Volume 1952-59, pp. 1-32.

Horning, N., 2010. *Random Forests : An algorithm for image classification and generation of continuous fields data sets.* [Online]
Available at: http://gisws.media.osaka-cu.ac.jp/gisideas10/viewabstract.php?id=342
[Accessed 17 May 2018].

Hutto, C. & Gilbert, E., 2014. VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. *Eighth International Conference on Weblogs and Social Media (ICWSM-14).*

Ireson, N., Ciravegna, F., Cox, V. & Beltrame, K., 2017. FootballWhispers: Transfer Rumour Detection. *CEUR Workshop Proceedings. 16th International Semantic Web Conference.*

Joachims, T., 1998. Text categorization with Support Vector Machines: learning with many relevant features. *ECML'98 Proceedings of the 10th European Conference on Machine Learning,* pp. 137-142 .

Kampaki, S. & Adamides, A., 2014. Using Twitter to predict football outcomes. *ArXiv e-prints.*

Kwon, S. et al., 2013. Prominent Features of Rumor Propagation in Online Social Media. *IEEE 13th International Conference on Data Mining, Dallas, TX, 2013,* pp. 1103-1108.

Lion Gu, V. K., Yarochkin, F., Leopando, J. & Estialbo, J., 2017. *Fake News and Cyber Propaganda: The Use and Abuse of Social Media.* [Online]
Available at: https://www.trendmicro.com/vinfo/us/security/news/cybercrime-and-digital-threats/fake-news-cyber-propaganda-the-abuse-of-social-media
[Accessed 15 May 2018].

Ma, B., Lin, D. & Cao, D., 2017. Content representation for microblog rumor detection.. *Advances in Computational Intelligence Systems. Advances in Intelligent Systems and Computing,* Volume 513, p. 245–251.

Manning, C. D., Raghavan, P. & Hinrich Schütze, 2008. *Introduction to Information Retrieval.* s.l.:Cambridge University Press.

Manning, C. D., Raghavan, P. & Hinrich Schütze, 2008. *Introduction to Information Retrieval.* s.l.:Cambridge University Press..

Mäntylä, M. V., Graziotin, D. & M. K., 2016. *The Evolution of Sentiment Analysis - A Review of Research Topics, Venues,.* [Online]
Available at: https://arxiv.org/ftp/arxiv/papers/1612/1612.01556.pdf

Mark A. Hall & Smith, L., 1991. Feature Selection for Machine Learning: Comparing a Correlation-based Filter Approach to the Wrapper.

Nair, G., 2016. *Text Mining 101: Topic Modeling.* [Online]
Available at: https://www.kdnuggets.com/2016/07/text-mining-101-topic-modeling.html
[Accessed 03 May 2018].

Nakov, P. et al., 2016. SemEval-2016 Task 4: Sentiment Analysis in Twitter. *Proceedings of the 10th International Workshop on Semantic Evaluation ,* pp. 1-18.

Nguyen, Q.-M. & Cao, T.-D., 2015. A novel approach for automatic extraction of semantic data about football transfer in sport news. *International Journal of Pervasive Computing and Communications,* Volume 11, pp. 233-252.

Nguyen, Q.-M., Cao, T.-D. & Nguyen, T.-T., 2014. Automatic creation of semantic data about football transfer in sport news. *Proceedings of the 16th International Conference on Information Integration and Web-based Applications & Services,* pp. 356-364.

Pang, B. & Lee, L., 2005. Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales. *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics,* pp. 115-124.

Pang, B., Lee, L. & Vaithyanathan, S., 2002. Thumbs up? Sentiment Classification using Machine Learning. *EMNLP '02 Proceedings of the ACL-02 conference on Empirical methods in natural language processing ,* Volume 10, pp. 79-86.

Pedregosa, F. et al., 2011. Scikit-learn: Machine Learning in Python. *The Journal of Machine Learning Research,* Volume 12, pp. 2825-2830 .

Pennacchiotti, M. & Popescu, A.-M., 2011. Democrats, Republicans and Starbucks Afficionados: User Classification in Twitter. *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* pp. 430-438.

Pennington, J., Socher, R. & Manning, C. D., 2014. GloVe: Global Vectors for Word Representation. *EMNLP,* Volume 14, pp. 1532-1543.

Popov, B. et al., 2003. KIM -- Semantic Annotation Platform. *The Semantic Web - ISWC 2003,* pp. 834-849.

Qazvinian, V., Rosengren, E., Radev, D. R. & Mei, Q., 2011. Rumor has it: identifying misinformation in microblogs. *Proceedings of the Conference on Empirical Methods in Natural Language Processing,* pp. 1589-1599.

Ritter, A., Clark, S., Mausam & Etzioni, O., 2011. Named Entity Recognition in Tweets: An Experimental Study}. *EMNLP.*

Shu, K. et al., 2017. Fake News Detection on Social Media: A Data Mining Perspective. *IGKDD Explor. Newsl.,* Volume 19, pp. 22-36.

Socher, R. et al., 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing,* pp. 1631-1642.

Sun, A., Lim, E.-P. & Liu, Y., 2009. On strategies for imbalanced text classification using SVM: A comparative study. *Decision Support Systems,* 48(1), pp. 191-201.

Vapnik, V., 1995. *The Nature of Statistical Learning Theory.* 1 ed. s.l.:Springer-Verlag New York.

Vosoughi, S., 2015. *Automatic Detection and Verification of Rumors onTwitter.* [Online] Available at: https://www.media.mit.edu/cogmac/publications/Soroush_Vosoughi_PHD_thesis.pdf [Accessed 23 May 2018].

Vosoughi, S., Roy, D. & Aral, S., 2018. The spread of true and false news online. *Science ,* 359(6380), pp. 1146-1151.

Wei, F., 2012. *Sentiment Analysis and Opinion Mining.* [Online] Available at: http://web.nchu.edu.tw/~jodytsao/MarkegingG/IIR10-Sentiment%20Analysis.pdf [Accessed 14 January 2018].

Weiss, G. M., McCarthy, K. L. & Zabar, B., 2007. Cost-Sensitive Learning vs. Sampling: Which is Best for Handling Unbalanced Classes with Unequal Error Costs?. *DMIN.*

Zhao, Z., Resnick, P. & Mei, Q., 2015. Enquiring Minds: Early Detection of Rumors in Social Media from Enquiry Posts. *Proceedings of the 24th International Conference on World Wide Web,* pp. 1395-1405.

Zubiaga, A. et al., 2018. Detection and Resolution of Rumours in Social Media: A Survey. *ACM Computing Surveys (CSUR) ,* 51(2).