Sarcasm Detection on Twitter:

bolstering lexical features with contextual clues

by

Samuel Albanie, MMath (Oxon)

Dissertation

Presented to the

University of Dublin, Trinity College

in fulfillment

of the requirements

for the Degree of

Master of Science in Computer Science

University of Dublin, Trinity College

September 2013

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Samuel Albanie

August 27, 2013

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Samuel Albanie

August 27, 2013

Acknowledgments

I would like to extend my thanks to the many people who contributed to the production and presentation of this research. To Judith Albanie, who contributed her proofreading skills at late notice, to Seàn Labastille, who offered helpful advice on all things techincal, and to Zsofia Novak who was a bastion of emotional support through the process. I would like to thank Stephen Barrett for his encouragement throughout the year and the many researchers and students who kindly contributed their time to participate in my research.

Finally, I wish to express my sincere gratitude to my supervisor, Dr Alexander O'Connor, who not only contributed his guidance and knowledge to the project, but was also prepared to offer his considerable technical skills to any problem to which I could not find the answer.

Samuel Albanie

University of Dublin, Trinity College

September 2013

Sarcasm Detection on Twitter:

bolstering lexical features with contextual clues

Samuel Albanie, M.Sc.
University of Dublin, Trinity College, 2013

Supervisor: Alexander O'Connor

One of the consequences of the unabated growth of social media has been a surge in the production of opinionated user-generated content. This data represents an immensely valuable resource if meaningful insights and trends can be accurately extracted. One of the challenges facing *sentiment analysis*, the field of research that addresses this task, is the accurate detection of sarcasm, which flips the polarity of literally-interpreted sentiments. While previous research in this area has focused on the use of *lexical feature*-based models to classify sarcasm, this research investigates the impact of using *contextual features* to improve accuracy.

A thorough review is conducted into the literature and theory relating to sarcasm detection and a corpus is generated using the *Twitter API* with a series of gathering and filtering techniques. A study is then conducted into the ability of humans to detect sarcasm and classifiers are trained to recognise sarcasm with a range of lexical and contextual feature sets. Finally, an extensive evaluation of each set of classifiers demonstrates the value of considering the contextual feature-based approach.

Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction

In this chapter, the motivation behind the research is introduced. The research objectives for the thesis are identified, together with potential research challenges that may present themselves in the undertaking of these objectives. Finally, a technical approach for the research is proposed and an outline illustrating the structure of the document is provided.

1.2 Motivation

The growth of the internet over the past decade has been characterised by the emerging dominance of social media as a platform for human interaction. As a consequence, the dynamics of communication between producers and consumers of online content have undergone a fundamental change. Now, more than ever before, anyone with basic access to the internet can publish their thoughts and spread their ideas through online communities. The result has been a surge in the creation of *user-generated content* and in particular, an abundance of opinionated material online [1].

One of the primary driving forces behind this phenomenon has been a growth in the

number of services that strive to lower the barrier to entry for ordinary users that wish to produce content. Perhaps no medium has achieved this more effectively than the global microblogging service Twitter [1]. In one hundred and forty characters or less, it offers users the chance to express their opinions to anyone who is prepared to *follow* along. Recent statistics suggest that on average, 500 million tweets are produced a day by its diverse set of users around the world [2]. As a result, Twitter data provides a vast source of publicly available human interactions, covering a broad range of topics and geographical locations.

Whether it is assessing the impact of a political speech [3] or researching potential market opportunities [4], an accurate understanding of how people feel about specific issues is an incredibly valuable asset. There has therefore been a strong demand for tools that can accurately extract useful insights and trends from this abundance of subjective data created by services such as Twitter [5]. For this reason, the field of sentiment analysis, which concerns itself with capturing the intended meaning of opinionated content, has received considerable research interest [6]. In particular, concepts and ideas from the fields of linguistics and statistics have been successfully applied to this problem, yielding significant improvements in the performance and effectiveness of automated systems for sentiment analysis.

Despite this progress, accurate detection of the literary device known as sarcasm remains a difficult problem [7]. Broadly, a sarcastic utterance is one in which the intended meaning differs from the literal meaning. For example, consider the following tweet:

I just love the new Samsung. It's so, so fast. #TheSnailPhone

A naïve sentiment analysis system that isn't capable of detecting sarcasm would inaccurately classify this tweet as a positive affirmation of the speed of the phone. Any

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

tool that strives to effectively extract the meaning of a tweet therefore must be able to accurately detect sarcasm.

Moreover, in order to operate successfully in a sentiment analysis environment, a sarcasm detection system must be capable of not only recognising sarcasm, but of the more difficult task of differentiating it from tweets that carry either a directly positive or negative sentiment. Such positive and negative tweets are known as *polar* tweets. Polar tweets form the most significant input for the tool and hold the most influence over the outcome of the analysis.

So far, the most successful work in addressing this problem has focused largely on statistical approaches in which the content of each tweet is considered as a *bag-of-words* [8]. Machine learning models are trained to associate a weighting to each word or combination of words and then use this information to determine whether or not each new tweet exposed to the model is sarcastic. This is commonly referred to as a lexical feature-based approach.

In contrast, much of the work in linguistic theory relating to sarcasm supports the perspective that an approach built on contextual features, in which the context surrounding the tweet is considered, should lead to better model performance [9]. There is therefore an opportunity to construct a model that can leverage both the successful techniques of statistical approaches together with concepts from established linguistic theory to improve performance. This opportunity motivates the research question considered in this paper.

1.3 Research Question

The aim of this thesis is to consider the problem of sarcasm detection in polar twitter data and to investigate the performance impact of bolstering existing lexical feature-based approaches with the use of contextual features.

1.4 Research Objectives

To address this research question, five specific research objectives have been defined:

1. To conduct a survey of existing approaches to sarcasm detection, together with an overview of the underlying machine learning and linguistic theory.

2. To gather a corpus of reliably labelled tweets that can be used to train and accurately evaluate the relative performance of different classifiers.

3. To provide a realistic benchmark for the difficulty of differentiating between sarcastic and polar tweets by conducting a study in which humans perform the classifying task on the dataset.

4. To design and build a range of machine learning models that utilize both purely lexical features as well as models that leverage contextual features.

5. To conduct an evaluation of each approach, assess performance implications and offer suggestions for potential future work.

1.5 Research Challenges

Alongside the objectives outlined above, there are also a number of significant research challenges that are specific to this problem domain. These must be addressed for this research to be conducted successfully:

1. Sarcasm is an ambiguous concept. We must develop a definition which can be used consistently throughout the research. Moreover, we must overcome the challenge of constructing a corpus containing tweets that accurately conform to this definition.

2. The corpus will be gathered from the *twitter API*, a service that offers public access to tweets made in the public domain. There is therefore the potential for noisy tweets generated by spammers to pollute the dataset. Such tweets must be filtered from the final corpus.

3. While the tweets are gathered from a public service, there are ethical implications to storing large quantities of user data. In particular, opinions expressed by users must be handled in a respectful manner and any personally identifying content present in the tweets should be used responsibly.

4. Contextual features are typically far more difficult to obtain than lexical features, which are readily available in the tweets themselves. An approach must be developed to extract contextual features from the corpus of tweets.

1.6 Brief Technical Approach

To address the research objectives outlined above, fundamental machine learning concepts and linguistic theory are surveyed alongside relevant research in these areas. This investigation highlights the importance of generating a high quality corpus and leads to the development of a set of filtering techniques which are then applied to the raw data gathered from the *Twitter API*. A study to investigate human performance on the classification task is then conducted on the corpus, before classifiers are trained on the dataset using a range of lexical and contextual features. Finally, a comparison of the performance of each classifier is conducted, together with an analysis of the results.

1.7 Thesis Outline

A survey and analysis of the relevant theory and related work is given in Chapter Two, together with an explanation as to how this work influences the decisions taken while performing the research. Chapter Three converts these decisions into engineering requirements and a design for the project. The key details of the implementation are discussed in Chapter Four. Chapter Five contains a discussion of the evaluation of the research together with significant datasets. Finally, in Chapter Six, conclusions are drawn about the research, a summary of the research contribution of the thesis is provided and possible extensions and future work are discussed.

Chapter 2

State of the art

In this chapter, the background for the research is investigated. To further elucidate the motivation behind the research, the potential implications of improved sarcasm detection are discussed in detail. Next we give a brief summary of the microblogging service Twitter, offer justification for its use as a source for corpus generation and give an overview of the unique challenges presented by working with the platform. The field of *machine learning* is then introduced, together with a discussion of the core concepts and principles and an evaluation of the statistical techniques that are relevant to the investigation. Following this, we consider sarcasm from a linguistic perspective, laying the foundations for the framework in which we will base our machine learning algorithms. Finally, we review the research that is research most closely related to ours from the field of computational sarcasm detection in the context of our research question.

2.1 Background

Few technology trends of the past decade have been more prominent than the rise of *Big Data*. Driven by the availability of cheap magnetic storage on a previously unprecedented scale [10], this abundance of data represents a enormously valuable resource if meaningful information can be accurately extracted. Data mining (the field of analytics that deals

with this task) has received considerable attention particularly in the field of sentiment analysis which seeks to extract and classify subjective content. The application of powerful statistical processes to traditional natural language processing techniques has yielded interesting results in many areas, ranging from modelling social power relationships [11] using a volume of data, to suicide note classification [12] on a very small dataset. However, the recognition and classification of humour, and in particular sarcasm, remains a challenging problem and indeed it is one that forms the perfect test platform for an advanced level of understanding by a computer [13].

Our research seeks to address the problem of automated sarcasm detection on the social media platform Twitter. Sarcasm has long been considered an interesting problem in linguistics, but correct identification of its presence also represents an opportunity to achieve accuracy gains in the field of sentiment analysis. Since 2001, sentiment analysis, a field of Natural Language Processing that concerns itself with determining the polarity of subjective content, has seen considerable research interest. This interest has been primarily motivated by its potentially valuable commercial applications. As demonstrated by [14], the impact of online reviews on product purchase is substantial. Targeted sentiment analysis allows a company or organization to maximize their responsiveness to such reviews by facilitating an assessment of their online reputation based on the evidence of social interactions. Moreover, accurate sentiment analysis allows a corporation to gauge the effectiveness of an advertising campaign or to test the potential of a market opportunity, both valuable strategic abilities. From a scientific perspective, it has enabled researchers to examine human responses to social phenomenon on a global scale [15]. Given these potential benefits, Twitter has naturally attracted both corporate and academic attention for conducting sentiment analysis as one of the largest and most diverse social media platforms.

2.1.1 The Twitter Microblogging Service

The concept behind Twitter is elegantly simple. A user can publish a message, known as a *tweet*, to anyone who is prepared to listen to them (a follower). The tweet can be up to 140 characters in length and there are no restrictions on the lexical ordering of the message. The user may tag their tweet with the use of a hashtag and they may target their message at a particular user with the use of an ampersat (@user) providing some meta data to their content. A full description of tweet functionality can be found at the Twitter field guide [16]. The simplicity of a tweet lowers the time cost and thought burden for the user and is one of the driving factors behind Twitter's phenomenal growth since its inception in 2006 [17].

Collectively, there are over 200 million active users of Twitter [18] and since only 11% have protected accounts [19], the vast majority of the resulting tweets are in the public domain. The release of the Twitter API has made this enormous source of user-generated content accessible (in varying degrees) to both businesses and scientific researchers.

In a commercial context, Twitter offers an opportunity for consumers to express their feelings and experiences associated with a brand and its products. Since there is a competitive advantage to be gained by an accurate understanding of the needs of clients, many companies invest in technology that allow them to extract this valuable information from the data. As the scale of Twitter and other sources of user-generated content grew, there was a need for tools that would enable corporations to rapidly analyze and make sense of subjective consumer data in vast quantities. The publication of the influential MapReduce paper [20] and the release of the open source implementation Hadoop laid the foundations for such tools, allowing businesses to make the most of this abundant

data and facilitating a better informed business strategy.

Alongside commercially significant scientific studies into sentiment analysis (such as [21] regarding politics and [22] on *socio-economics*), interest in Twitter as a research platform has also developed in many other disciplines. It has been used to investigate such areas as internet-based information credibility [23], online education [24] and the rapid spread of news in the context of natural disasters [25]. The diverse nature of these studies reflect the depth and breadth of the data generated by the service. Moreover, the content is primarily subjective and often humorous, lending itself as a natural source for corpus generation for an investigation into sarcasm detection.

There are however, significant challenges associated with working with the unique structure of Twitter data. Firstly, the brevity of tweets presents a serious challenge for accurate data mining techniques [26]. Traditional bag-of-words approaches to data mining view text as an unordered collection of words and rely on frequency analysis of individual terms to give meaning to the data. With tweets, not only is there less content to work with to extract meaning, but the character limitation often leads users to use abbreviations and create new words to save characters, presenting further difficulties for such techniques [27]. To address these challenges we draw on the well-established techniques in the research field of machine learning.

2.1.2 Machine Learning

The discipline of machine learning sets its sights on the ambitious goal of creating computer programs that are able to learn. To give an indication of the meaning of the word

learn in this context, the following definition is useful [28]:

A computer program is said to learn from experience E with respect to some class of task T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

It is a broad field that finds its origins in the work of Alan Turing when he first considered the question of whether or not a computer could think, introducing the *Turing Test* in his influential paper [29] in 1950. Perhaps the first true instance of a machine learning program was a checkers playing program published in 1959 by Arthur Samuel [30], which he demonstrated was able to improve as it encountered more opponents. From such beginnings it has now grown into a research area that blends with numerous others: *artificial intelligence, evolutionary modelling, control theory, neuroscience* and perhaps most heavily, *statistics*. We offer an overview of the basic concepts and statistical techniques used in machine learning and demonstrate how these may be applied to our system for detecting instances of sarcasm.

2.1.2.1 Core Concepts

Working from the definition outlined above, a machine learning problem may be decomposed into three parts as follows. Firstly, the task to be addressed by the system must be identified and defined. In our case, the task is to classify instances of sarcasm among a collection of polar tweets. Secondly, there must be some collected set of training examples which form the experience by which the system can learn. This will be achieved by generating a corpus via the Twitter API. Thirdly, the problem must have performance metrics with which the system may be evaluated. In this study we will measure the performance of our system against a constructed *gold standard* using a number of metrics, discussed

in detail in section 2.1.2.3.

In order to give a more formal structure to our task, we introduce some terminology to allow us to address the problem:

Suppose that we have collected a dataset, where each record has n components. We may write each record as a vector,

$$\mathbf{x} = (x_1, x_2, ..., x_n) \tag{2.1}$$

which we refer to as an *instance*. Each component x_i is known as a *feature*. There are, broadly speaking, four types of problem that may be addressed using machine learning techniques [31].

The first is referred to as *classification*. It takes a dataset of *classified instances*, that is, instances which have already been assigned to specific categories, and uses these as a training set for an algorithm that can deduce a way to classify previously unseen instances. The second is *numeric prediction*, which similarly works on training data, but which seeks a numerical value instead of a categorical class value for each instance. *Association learning* forms the third type of problem. In such cases, the goal is to discover significant associations between different features and uncover structure that may be hidden in the data. The fourth is known as *clustering*, the aim of which is to partition the instances into useful categories. *Useful* in this sense, can be difficult to define precisely and this approach often involves some subjective evaluation.

All of the approaches outlined above remain subject to the familiar limitations of statistics. Care must be taken to avoid overfitting the classifier to the training datasets and any algorithm selected for carrying out an approach should be appropriately robust

to the noise that invariably exists in the data. We provide a brief overview of some of the supervised learning algorithms that have demonstrated themselves to be effective research tools in this field.

2.1.2.2 Supervised Learning Algorithms

To illustrate in a formal sense the specific purpose of the algorithms, we suppose that there exists an unknown function *f(x)* that correctly classifies or assigns the numerical outcome to each instance. Our aim is to determine this function, to which end we define a new function *h(x)*, referred to as the hypothesis, that we aim to match to *f(x)*. For classification, we refer to *h(x)* as a classifier and for numeric prediction, as an estimator. Thus the purpose of the supervised learning algorithm is to fit *h(x)* to the values taken by *f(x)* on the training set such that it can be used as accurately as possible to estimate the values of *f(x)* for new data. There are a large number of alternatives which have been designed for this task, and each performs well under different circumstances.

2.1.2.2.1 Naïve Bayes Classifier

Based on the Bayesian theorem, this probabilistic classifier works on the (often unrealistic) assumption of the independence of features, hence the term naïve. Despite this, it often performs very well at real-world classification problems. Indeed, research has been conducted as to why it performs better than expected on complex problems given its simplistic assumptions [32]. These assumptions make it naturally resistant to overfitting and it performs particularly well as a classifier on datasets with many features. It is also computationally efficient - the training of the algorithm is linear in both the number of features and the number of instances [33]. However, despite these positives, it is often outperformed by other algorithms on data where feature dependence is prevalent. Moreover, although it works well on small datasets, performance does not scale with larger training sets as effectively as other algorithms.

2.1.2.2.2 Logistic Regression

Used to classify an instance into one of (typically) two categories, this classifier does not assume a linear relationship between the features and the target class. It operates through a combination of linear regression together with a transformation through the logistic function. It has proved to be a mainstay of statistical analysis, particularly in medical and social science research primarily for its robustness in dealing with data that does not fit with even statistical assumptions. Its primary weaknesses are its sensitivity to overfitting [34] and that the classifier requires a large sample size to stabilize.

2.1.2.2.3 Support Vector Machines

Support Vector Machines (SVM) use a linear model referred to as a maximum margin hyperplane to find the greatest separation between classes. This hyperplane is uniquely defined by minimizing the total distance (the margin) to the support vectors, the instances in the transformed space that are closest to it, illustrated in Figure 2.1 below:

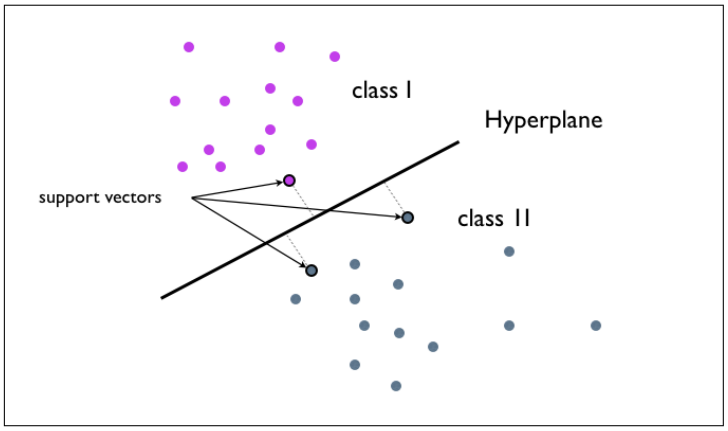

Figure 2.1: *Support Vector Machines*

Since these support vectors in some sense represent the training set, the algorithm is less vulnerable to overfitting, a problem caused by excessive flexibility in the decision plane. Support Vector Machines can resolve the problem of dealing with nonlinear

boundaries by transforming the instance space using a nonlinear transformation, a process referred to as the *kernel mapping technique*. In the transformed space, a linear model can be constructed that corresponds to a nonlinear model in the original instance space. Although this transformation would appear to introduce a great number of new dimensions to the problem, SVM functions only depend on the training and test data through the kernel mapping, which is typically far less complex. Even in the nonlinear case they are therefore able to avoid the curse of dimensionality excessive parameters which lead to overfitting and intractable complexity [35].

Although these are perhaps the three most popular choices, there are many other alternatives. A thorough review of machine learning algorithms for text classification is given in [36]. However, as suggested by the *No Free Lunch theorem* [37], there is no algorithm that exhibits superior performance across every environment. At the outset of an experiment it is difficult to ascertain which algorithm will produce the best results. It is therefore useful to perform classification tasks with multiple algorithms in order to be in a position to select the specific algorithm that maximizes performance. This is further supported by an extensive empirical study in [38] that evaluated the performance of many of the most popular supervised learning algorithms using eleven different performance metrics and found that the optimal selection varied depending on the conditions of the learning. In light of this evidence, we propose to adopt such an approach to our machine learning tasks, utilizing each of the three algorithms mentioned above.

2.1.2.3 Evaluation Metrics

A critical choice in the comparison of different machine learning techniques is the selection of metrics that will be used to evaluate each model. Different metrics lend significance to different properties of a classifier and characteristics of the dataset. It is therefore important to use metrics that are suitable for the dataset in question and that facilitate

a comparison of the classifiers from the perspective that is of most significance to the research. Each metric can be defined in terms of the *confusion matrix* for the classifier, which provides a visualization of the complete performance of a classifier, as demonstrated in Figure 2.2:

	Predicted True	Predicted False
Labelled True	*true positives (tp)*	*false positives (fp)*
Labelled False	*false negatives (fn)*	*true negatives (tn)*

Figure 2.2: *A confusion matrix for two-way classification*

2.1.2.3.1 Accuracy

Accuracy can be defined simply as the fraction of classifications that are correct. Thus in the terminology used above, we may write [39]:

$$Accuracy = \frac{tp + tn}{tp + fp + fn + tn} \tag{2.2}$$

It is a useful performance measure for providing an indication of the overall performance of the classifier, but it has the disadvantage of *skewed weighting* when used on an unbalanced dataset - the outcome is skewed in favour of the dominant class. For this reason two other metrics, *precision* and *recall*, are also used.

2.1.2.3.2 Precision

Precision is a measure of the ratio of instances that were classifed as positive that are truly positive. It indicates how effectively the classifier is performing on a specific class, irrespective of any skew in the dataset:

$$Precision = \frac{tp}{tp + fp} \tag{2.3}$$

It is therefore of greatest utility when one class holds more significance than others. However, it takes no account of the false negatives generated by the classifier. To address this shortcoming, it is often used in conjunction with *recall*.

2.1.2.3.3 Recall

Recall is a measure of the ratio of relevant instances that were retrieved for a specific class. It forms the natural complement to *precision* - together they give an overview of performance on a specific class:

$$Recall = \frac{tp}{tp + fn} \tag{2.4}$$

The classifier can be tuned to favour precision by cautiously only classifying instances as positive in cases where there is very strong evidence in favour of a positive classification. However, this will naturally lead to a very poor recall. In contrast, a classifier may classify every class as positive, obtaining a perfect recall score but very poor precision. It follows that a balance is needed between these two measures and one metric that is widely used to achieve this balance is *F-measure*.

2.1.2.3.3 F-measure

The trade-off between precision and recall can be selected by combining them with the *F-measure*, which is calculated as the weighted harmonic mean of precision and recall [39]:

$$F\text{-}measure = \frac{(\beta^2 + 1)Precision.Recall}{\beta^2.Precision + Recall} \tag{2.5}$$

where

$$\beta = \frac{1-\alpha}{\alpha}, \;\; \alpha \in (0, 1] \;\; and\, hence \;\; \beta^2 \in [0, \infty) \tag{2.6}$$

In order to create a balanced weighting, setting $\beta = 1$ yields the default *balanced F-measure*, more commonly known as F_1*-score*. Equation (2.5) may then be rewritten as:

$$F_1\text{-score} = \frac{2.Precision}{Precision + Recall} \tag{2.7}$$

Adjustment of the weighting parameters can give greater significance to either precision or recall which can be particularly useful in *Information Retrieval* tasks. However, for machine learning tasks on relatively balanced datasets, both are of equal interest and so the balanced F-measure is therefore the most appropriate.

There are also many other metrics available, each with different strengths and weaknesses (a detailed analysis is given in [40]). Among all of these, the four metrics outlined above were selected for two primary reasons. Most importantly, when used together they offer a strong overview of the performance characteristics of a classifier. Additionally, these metrics are ubiquitous amongst the literature, which allows any evaluation performed in this resarch to be compared against previous work.

After considering some of the basic concepts of machine learning, it is instructive to review the machine learning research that has been conducted on the Twitter platform.

Much of this work has been performed under the field of *sentiment analysis*, which seeks to classify the polarity (i.e. *positive, negative* or *neutral*) of user-generated content. We shall focus our attention on supervised learning algorithms which have received the most attention in both sentiment analysis and *humour recognition*. First, however, we review the literature on irony in linguistics to give us a solid basis of understanding of the target that our machine learning system will be trained for.

2.1.3 Linguistic Foundations

Automated humour detection is a field that remains very much in its infancy, particularly for the detection of sarcastic content. Sarcasm may be considered under the broader concept of irony. To build a system capable of performing this task effectively requires a solid grounding in the established theory of irony that has been developed in the field of linguistics. Before delving into this theory, it is worth reflecting on the delicate issue of definitions.

2.1.3.1 Definitions

Irony is a concept that is notoriously challenging to define. Renowned critic and author of *The Compass of Irony*, D.C Muecke, made the following humorous comment on the futility of such a task [41]:

Since Erich Heller, in his 'Ironic German', has already quite adequately not defined irony, there would be little point in not defining it all over again.

Even for the narrower concept of sarcasm there are a great number of definitions ([42] gives a detailed discussion of the linguistic concepts and challenges associated with constructing such definitions). Unlike irony, a term that can cover a range of concepts,

many people have a strong intuitive idea as to what constitutes sarcasm. However, there are regional variations that occur in the interpretation of this concept. These variations can occur even within the same country as demonstrated in [43]. We propose to address this issue in manner proposed in [44]: instead of working to a rigid definition, we will leave the concept undefined. When constructing our gold standard, the only definition of sarcasm that we consider important is that held by the author of the tweet. Moreover, when conducting our human annotator study to assess the difficulty of sarcasm detection, we will allow each participant to work according to their own understanding of sarcasm. In this way we propose to incorporate a degree of flexibility into our research that will allow us to take account of the regional variations in sarcasm that will inevitably occur in a corpus generated from a global service like Twitter. This is discussed in further detail in the design, covered in Chapter Three.

2.1.3.2 The Theory Of Irony

Much of the research on irony in linguistics has its foundations in the Theory of Conversational Implicature, developed in the subfield of pragmatics in 1975 by the philosopher of language Paul Grice. Pragmatics is the research area that seeks to understand how context contributes to meaning in language. Grice introduced the theory that participants in a conversation both contribute under a shared understanding that there is a standard manner in which to communicate, which he refers to as the *cooperative principle* [45]:

Make your conversational contribution such as is required, at the stage at which it occurs, by the accepted purpose or direction of the talk exchange in which you are engaged.

To clarify this concept, he divided this principle into four maxims, now referred to as *The Gricean Maxims*, as follows:

MAXIM OF QUALITY: *Try to make your contribution one that is true: (1) Do*

not say what you believe to be false. (2) Do not say that for which you lack adequate evidence.

MAXIM OF QUANTITY: *(1) Make your contribution as informative as is required for the current purposes of the exchange. (2) Do not make your contribution more informative than is required.*

MAXIM OF RELEVANCE: *Make your contribution relevant.*

MAXIM OF MANNER: *(1) Be perspicuous. (2) Avoid obscurity of expression. (3) Avoid ambiguity. (4) Be brief (avoid unnecessary prolixity). (5) Be orderly.*

The core idea is that if both the conversation participants are aware of these maxims then when a maxim is flouted by a speaker, the addressee is able to interpret the implied meaning. According to Brown and Levinson in their work on Politeness Theory [46], irony can be interpreted in terms of the Gricean Maxims as a violation of the maxim of quality. This view is supported by much of the subsequent literature, although [47] contends that irony may also result from violation of the other three maxims under specific circumstances.

Building on these theories, perhaps the most in-depth analysis of sarcasm (commonly referred to as *verbal irony* in the literature) is offered in [48], which proposes a new implicit display theory for distinguishing verbally ironic utterances. It identifies shortcomings in the previous approaches, such as an inability to comprehend understatement as verbal irony. The main contribution of the theory is to recognize that ironic language presupposes an ironic environment, described as a certain situational setting in the discourse context. This concept is distinct from the idea of situational irony, which refers to a different type of irony. The principle of the ironic environment motivates the techniques we employ

in our system. In particular for Twitter, we note the significance of context and shared understanding between the author of a tweet and its intended audience when sarcasm is employed.

2.2 Related Work

We propose to review the related work in the following manner. Firstly we examine the research that has been conducted into computational humour and in particular, humour recognition. Next, we consider work that has been performed using machine learning techniques on the Twitter platform and more generally on subjective content. Finally we evaluate the research that has been carried out on systems for the specific purpose of sarcasm detection.

Early work in computational humour was designed to construct a model of a particular form of linguistic humour, short pun-like sentences known as *Tom Swifties* [49]. One example of such humour is *'I like pizza', Tom said crustily.* To achieve this, the authors performed a careful linguistic analysis of all potential forms of the joke structure before designing a set of rules that were used by a computer to successfully generate new instances. Their research is grounded in linguistic theory, particularly the use of morphological *malrules* (morphological extensions that are not intended eg. *gin/gingerly*) from previous models of natural language. It therefore represents a semantic approach to humour modelling.

Following this, a formal model of a subtype of humour together with an implementation called JAPE was produced to create *punning riddles* [50], [51]. The model was built on syntactic and semantic structures, and evaluation of the performance of the implementation demonstrated the effectiveness of the model. The author notes that pun generation

requires less knowledge than pun comprehension - an indication as to why early work did not focus on humour recognition. To effectively apply such an approach to our work in the detection of sarcasm would require a very large dataset for the training of the system. We therefore do not adopt this style of matching at a structural level.

Some of the first work on humour detection was performed in [52] which proposed a system for detecting a restricted form of humour, *knock-knock* jokes. The authors followed a statistical approach based on collecting a set of N-grams from a carefully selected corpus of 195 jokes. The system is designed to validate the format of the joke, generate wordplay sequences and then test to see if a generated sequence appears in the joke being tested. They train their system on one third of their corpus and evaluate it on the remaining two thirds. While reasonable at detecting wordplay, the system ultimately performs poorly at recognizing jokes on the test set. Various possible reasons for this exist, but given that the effectiveness of the system is highly dependent on the input data, we note the significance of using a sufficiently large corpus to train humour recognition system.

More recent work [53] considered the identifying features that would facilitate the separation of humorous from non-humorous content, focusing their attention on one-liners and satirical news articles. To avoid the potential issues caused by too small a corpus, the authors collected 1,125 articles and used a web-based bootstrapping method to expand a small number of manually selected seeds into a dataset of 16,000 one-liners. They conducted their investigation using both Naïve Bayes and Support Vector Machines for text classification. They were able to detect humorous news articles with an accuracy of 96.8% with SVMs (compared to 88.0% for Naïve Bayes). As one might expect, with less content available, they were less effective with one-liners, although here Naïve Bayes narrowly outperforms SVMs (NB 79.69%, SVMs - 79.23%). This highlights the significance of using multiple algorithms for the classification task, particularly with tweets which contain very limited content. The authors determine that the most significant features

that indicate humour are human centredness (references to human-related scenarios) and negative polarity. Furthermore, they suggest that it is unwise to construct a model that searches for one defining indicator of humour, rather, following the linguistic analysis in [54], that in each instance there are a number of significant concepts involved. This research emphasizes the importance of an open approach to sarcasm detection, in which incremental experimentation with different features and algorithms is more apt to lead to a strong performance than a technique based strictly on preconceptions about what constitutes sarcasm.

The polarity of a sentiment (whether it is positive, negative or neutral) is a crucial feature for humour detection and therefore correctly ascertaining the *contextual polarity* of terms in subjective content forms a critical component of a number of pieces of research. Contextual polarity refers to the polarity of a word when considered in its context, as opposed to *prior polarity*, when it stands alone. In [55] the authors develop an approach for identification of contextual polarity at the phrase level. They first produce a gold standard based on human annotated phrases. Next, they generate a lexicon, starting with over 8,000 subjectivity clues and expanding this with a dictionary and thesaurus and then tagging the prior polarity of these terms. The *BoosTexter AdaBoost* machine learning algorithm [56] was then implemented in two experiments, each time performing 10-fold *cross-validation*. Firstly, it was used to classify instances as contextually neutral or polar, and secondly, to assert the positive or negative polarity of those instances that were not neutral. They compared the performance of the classifier when using (i) just prior polarity (ii) a small feature set of word and modification features (iii) a complex feature set that also contained sentence and structure features, for both experiments. In both tasks the larger feature set performed best managing an accuracy of 75.9% for neutral-polar classification and 65.7% for polarity classification. This research suggests that contextual polarity is a complex phenomenon that must be tackled with sophisticated feature sets. However, we note that while a feature set must have the ability to express the

complexities of the classification target, a focus on expanding the feature set brings with it a tendency towards models that do not generalize well. Therefore there is an important balance to be struck when comparing potential features for the supervised algorithms to be implemented.

Sentiment analysis on the Twitter platform has been investigated by several authors, many using feature engineering, such as in the previous paper. To achieve improved classification performance over this approach, semantic features are included into the model using interpolation in [57]. The authors introduce abstract concepts for entities and compare the use of these to *part-of-speech* tags and microblogging features as well as sentiment topic features. Three datasets are used, the largest of which contains 60,000 tweets and a baseline Naïve Bayes classifier is trained from unigrams into which they incorporate their new semantic features. The classifiers perform differently on each dataset, with semantic features proving most effective on the largest dataset and sentiment topic features offering better results on the others. This may be an indication that the semantic features are most useful in a broader topic space. The authors also make a useful observation about *stopwords* - short words that are commonly removed in the data cleaning process such as 'the' and 'and'. They find that they achieve 3.2% better accuracy when leaving the stopwords in their dataset. This again emphasizes the need for experimentation with different features when training the models. Various other machine learning approaches using novel feature sets have also been attempted for the purposes of sentiment analysis - [6] offers a thorough review of the field.

The use of *pragmatics* offers another potential tool for verbal irony detection. In particular, the presence of emoticons in tweets can often be indicative of polarity. Although there is some debate as to how significant the impact of emoticons are on the meaning of the message [58], it is shown in [59] that emoticons can serve as a non-verbal surrogate for facial behavior, and thus hold real potential for sarcasm detection. In [60],

sarcasm recognition was investigated for spoken systems. Although some features (such as prosody and spectral voice analysis) are not relevant for Twitter-based analysis, they provide further research into contextual features and produce a system that can discern sarcasm as effectively as a human, focusing on the use of the phrase 'Yeah right'. They categorize the context of the use of this phrase as one of four types of speech acts and tag certain objective cues: *laughter*, *question/answer* (the phrase was used in a question or answer), *start/end* (the phrase appeared at the start or end of a statement), *pause* (the phrase appeared before or after a pause) and *gender* (whether the speaker was male or female). The contextual feature-based classifier outperformed the spectral and prosodic classifiers, supporting the view held in previous work and emphasized by the linguistic theory, namely that context is critical.

Attempts to detect verbal irony in user-generated content have taken a number of different directions. In [61], the authors note that the most useful application of a sarcasm recognition system is to improve the performance of polarity interpreters which, although often very good at identifying negative opinion [62], often perform poorly at detecting positive opinion which is more regularly affected by irony. They therefore investigate cases where a term or expression with a positive prior polarity is figuratively used to express a negative opinion (the study is carried out in their native language, Portuguese). Only eight simple linguistic patterns were used to detect sarcasm which proved to be very restrictive. As a result their coverage was extremely low resulting in a very poor recall, although some patterns achieved high precision. They found that the most effective patterns were those that involved quotation marks and indications of laughter (such as *lol* and positive emoticons). These are insights that can be incorporated into our model to improve accuracy.

The influence of *lexical features* is investigated in [44], who note that these have been largely ignored in favour of pragmatic features in the literature. The authors suggest

that lexical features will be much easier for a computer to evaluate effectively and might therefore lead to a better system. However, although they achieve some success with some specific terms such as 'gee' and 'gosh', they find via linear regression analysis that the variance due to lexical factors is just 5% which supports the perspective that pragmatics are considerably more important for such systems.

Perhaps the most extensive verbal irony detection system to date was constructed in [63]. The authors achieve good results on both a Twitter dataset of 6.9 million tweets and a set of 66000 Amazon product reviews. A semi-supervised SASI algorithm based on the k-nearest neighbour strategy is used as a classifier. They first extract features based on pattern matching and punctuation and then use a data enrichment strategy for their training sets. They compare the classifications of their algorithm to those of human workers provided by the Amazon Mechanical Turk service and obtain a remarkable accuracy of 0.947 and a recall of 0.756. However, we note that even when using only the non-enriched punctuation related features they still have an accuracy of 0.821, exceptionally high in comparison to similar previous work. This may, in part, result from the manner in which they construct their first gold standard. Firstly, as pointed out in previous studies, humans who are not the authors of the content may not perform any better than a machine learning algorithm (also found in [60] for spoken content) which would result in this gold standard inaccurately classifying significant numbers of sarcastic tweets. Moreover, to qualify as sarcastic for this standard, a tweet must achieve a certain level inter-annotator agreement . It could be that this only allows tweets that are most obvious in their sarcasm to be considered, which would explain the exceptional performance of the SASI algorithm. Following this study, they also test out the algorithm on a set of tweets that contained a sarcasm hashtag, then removed the tag. On this dataset, the algorithm does not perform nearly as well, still achieving reasonable accuracy of 0.727 but recall of only 0.436. There are several potential reasons for this. The first, as pointed out by the authors, is that by using a sarcasm hashtag, the author may feel that there is

less need to put sarcastic clues in their tweet. The other possibility is that these figures give a more realistic assessment of their approach and that the first gold standard leads to too generous a performance figure.

Finally, the work that most closely relates to our research objectives was performed in [8], in which the authors take on the more challenging task of differentiating sarcastic tweets from those that are directly positive or negative, known as *polar tweets*. They use basic pragmatic and lexical features for their approach and compare the use of Logistic Regression and SVMs on a dataset of 900 tweets. Their first approach was to experiment with differentiating between positive, negative and sarcastic tweets based on their lexical features. The best accuracy achieved by the system was poor - 0.5722, though it performed marginally better when differentiating between just two of these categories. They then undertook the same task with human judges who did not perform much better - 0.6259 average accuracy for three-way classification, indicating the difficulty of the task. Following this they performed a second experiment with a smaller dataset of 50 sarcastic and 50 non-sarcastic tweets, all of which contained emoticons. Both the algorithmic classifier (0.71 accuracy) and the human judges (on average 0.73) performed much better at this task, again demonstrating that at least under these circumstances, pragmatics are more valuable than lexical features for classification.

Given that the most valuable potential application of a system capable of detecting sarcasm would be as an accuracy enhancing accompaniment to a sentiment analysis tool, the final paper in our review offers the most interesting direction for investigation. Firstly, there is clearly much room for improvement with regards to the accuracy of a classifier. Moreover we believe that the evidence presented in the previous research indicates that this improvement may be found by more sophisticated modelling and incorporation of

context alongside lexical features.

It is therefore proposed to develop such a system for the detection of verbal irony on Twitter. Drawing together the lessons from the surveyed research, it is planned to collect a significantly larger corpus of tweets in which to train a system. The algorithms that have shown the best results in different instances in the literature are the Naïve Bayes, Logistic Regression and Support Vector Machine classifiers, so each of these will therefore be used. Given the repeated indication of the significance of context, not only individual tweets, but also the replies associated with each tweet will be collected. This will facilitate the analysis of Twitter conversations in which there is, by assumption, a degree of shared understanding and context between the author of the reply and the original tweeter, a technique that has not yet been considered in the literature. This approach will enable the exploitation of the features that take advantage of the ironic environment defined in the linguistic theory.

2.3 Summary

In this chapter, the background and state-of-the-art for the research question were discussed. We started by considering Twitter and its suitability as a source for corpus generation, contrasting the advantages of using such a popular and diverse resource for subjective content against the difficulties caused by the limited length of a tweet. This was followed by an overview of the core concepts and evaluation metrics of machine learning together with relevant algorithms for the task of text classification. It became clear that any effective approach to the learning task proposed in the research must incorporate multiple algorithms and a sufficiently expressive feature set. The foundations of linguistic theory relating to our research were then investigated, focusing on the constructs for irony

as a whole and the definition issues that surround work with sarcasm.

The previous work relating to computational humour and verbal irony recognition was then reviewed from the perspective of a sarcasm detection system. We then considered the work most closely relating to the research question outlined in Chapter One, focusing specifically on the Twitter platform and the approaches used to label the corpus and create a gold standard. Finally, conclusions were drawn from the findings of the review, laying the foundations for the design decisions covered in the next chapter.

Chapter 3

Design

The purpose of this chapter is to provide a detailed description of the design decisions for the research, building on the foundations of the related work reviewed in Chapter Two. Firstly, a set of requirements are gathered to ensure that each of the research objectives and challenges outlined in Chapter One are met. This is followed by a description of the design solution and a discussion of the implications of the decisions made in the design process.

3.1 Requirements Engineering

Due to the diverse range of research objectives and challenges, the requirements are considered in sets relating to specific aims. The first set of requirements relate to the process of corpus generation, summarized by the research objective: *Gather a corpus of reliably labelled tweets that can be used to train and accurately evaluate the relative performance of different classifier models.*

Reliable Labelling

- *The system must be capable of generating a corpus that is reliably labelled.*

It is important to carefully define what is meant by the term *reliably labelled*. This refers to the fact that the category associated with the tweet should reflect the true sentiment of the tweet as accurately as possible. Thus, every tweet that carries a positive sentiment should be labelled as positive in the corpus. Similarly, every tweet that is a true instance of sarcasm should labelled as sarcastic. As discussed in the review of related work, this is in itself a difficult task and a number of different approaches have been attempted. The two primary techniques available are:

1. The use of human judges to annotate the tweets with labels. A label is only used for a tweet when there is a sufficient degree of inter-annotator agreement. This can be done with either untrained participants or a panel of specialist linguists.

2. The use of *hashtags* within the tweets themselves to indicate the sentiment of the tweet. This relies on the assumption that the best person to label the tweet is the author of the tweet itself. Thus a tweet containing #sarcasm is labelled as sarcastic. Tweets containing #sad, #frustrated etc. are labelled as negative.

There are strengths and weaknesses to both methods. The first method is liable to miss out on difficult instances of sarcasm. As noted in the literature review, it is by no means guaranteed that even the panel of experts will detect all of the instances of sarcasm, potentially leading to a corpus containing only tweets in which the sarcasm is obvious. Moreover, in view of the dataset size requirement, it is worth noting that this manual approach is not scalable, which represents a significant issue for generating a large corpus. Conversely, the second technique may be liable to label too many instances as

sarcastic, for example, by including tweets that are about sarcasm such as:

I don't understand. Why is everyone using so much #sarcasm today?

It is therefore more likely to lead to a more conservative performance evaluation for the classifier. However, steps can be taken to remove tweets in which the tweet is about sarcasm, rather than an instance of it (for example, by only using tweets in which #sarcasm occurs at the end). Moreover, this approach is able to scale much more efficiently as the corpus grows in size. For these reasons, the #sarcasm technique will be adopted for labelling the corpus.

Corpus Size

- *The system approach to corpus generation must be scalable.*

One of the critical issues highlighted by the survey of related work was the significance of corpus size. The need for a large training set in order to build a good classifier, and a test set of sufficient size to evaluate it fairly was repeatedly noted as an important issue. This requirement does not influence any single decision, but rather is applied as a principle to all design decisions.

Corpus Quality

- *The system must be capable of generating a corpus of high quality.*

The quality of the corpus relates to a concern raised in the research challenges, namely that as a service that operates in the public domain, tweets gathered from the Twitter API are potentially 'noisy'. This refers to tweets that are generated by spam systems or fake users. They lead to the presence of duplicate, nonsensical or scam-linked tweets. There is also the issue of multilingual users tweeting in languages other than English, but

still choosing to use English hashtags, which could present issues for the corpus labelling approach.

To address this issue, the system will implement a series of filters and preprocessing methods after collecting tweets from the API. Duplicates will be removed, as well as tweets that contain multiple URLs. To address the issues relating to nonsensical and non-English tweets, an additional filter will be defined, based on an English dictionary (details of how this additional filter is implemented are provided in Chapter Four).

Corpus Composition

- *The system must be capable of generating a corpus consisting of both tweets and their replies.*

One of the research challenges highlighted in the first chapter was the task of constructing contextual features. An examination of linguistic theory suggested that this context can best be constructed by leveraging the shared understanding between the *speaker* of a sarcastic utterance and the *listener*. Consequently, by asserting that replies to tweets can indicate some level of shared understanding, gathering a corpus that contains replies will allow such features to be extracted.

The next set of requirements relate to the process of human classification, summarized by the research objective: *Provide a realistic benchmark for the difficulty of differentiating between sarcastic and polar tweets by conducting a study in which humans perform the classifying task on the dataset.*

Study Participation

- *The annotation study must be scalable.*

The purpose of the research objective is to provide a benchmark for the difficulty of the task. In order to do this as effectively as possible within the time and resource constraints of the research, it is desirable to gather a large number of participants (with English fluency) and to encourage each of these participants to annotate a large number of tweets. To maximize the human resources available, the annotation study must be scalable. It is therefore a natural choice to conduct the study online. Moreover, gamification techniques will be employed to encourage each participant to make a significant contribution.

Ethical Considerations

- *The annotation study must comply with the ethical standards for research.*

Any human study designed for the purposes of research must be approved by the Trinity College Ethical Research Committee. In order to gain approval, the study will be conducted in adherence to certain ethical guidelines. All participants in the study must participate voluntarily and must be free to leave the study at any time. Basic personal data used for identification purposes during the study will be stored securely and no personal data or recordings relating to the participants will be stored after the completion of the study. All contributions made by the participants will be made anonymous under publication.

The next set of requirements relate to the process of machine learning, summarized by the research objective: *Design and build a range of machine learning models that utilize both purely lexical features as well as models that leverage contextual features.*

Algorithm Selection

- *The system must implement three distinct algorithms.*

The importance of implementing a range of algorithms was repeatedly noted in the review of previous work in Chapter Two. Therefore, the system must implement each of the three algorithms discussed: *Naïve Bayes, Logistic Regression* and *Support Vector Machines* for each of the feature sets that are investigated.

Feature Selection

- *The system must implement both lexical and contextual features.*

Meeting this requirement is a critical component of addressing the research question set out in Chapter One, investigating the impact of bolstering lexical features with contextual features. Following the approach taken in previous work, the lexical features will be implemented with an *n-grams*-based approach. The contextual features will be constructed with the use of the tweet replies, contingent on the success of the corpus generation.

The final requirement does not apply to any specific research objective, but instead applies to all components of the system.

Data Storage

- *The system must conform to the legal and ethical requirements for data storage.*

The nature of the research undertaken leads to the collection of large amounts of user data from the Twitter API. From a legal perspective, there is therefore the need to ensure that the research complies with both the *Communications (Retention of Data) Act, 2011* [64] and the Twitter API terms and conditions - *Rules of the Road* [65]. Since the research does not constititue a commercial service, the only relevant part of the Twitter document is entitled *Respect User Privacy* which forbids the publication of confidential information, passwords or geo-location data. Compliance with both the *Retention of Data*

Act and the *Rules of the Road* can therefore be achieved by ensuring that all Twitter data collected in the research is made completely anonymous. As mentioned in the annotation requirements, similar steps will be taken with any personal data used for the purposes of conducting the human study.

3.2 Design Structure

Guided by the requirements engineering process, we are now in a position to design software that will enable the research question to be addressed. The requirements discussed above fall naturally into three categories, each representing a functionality that must be provided by the system. Consequently, in order to meet these requirements, the proposed design reflects this structure. There are three primary components to the system:

- *Gatherer* - A software component responsible for collecting the tweets from the Twitter API and implementing filters to clean the data.
- *Annotator* - An online survey tool responsible for the human classification task.
- *Learner* - A software component that implements the machine learning algorithms.

These components, together with the flow of data, are illustrated in Figure 3.1 below:

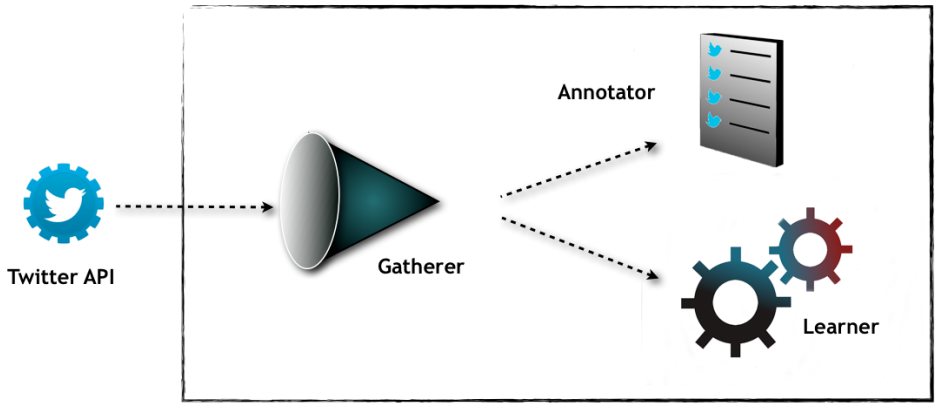

Figure 3.1: *System Components*

3.2.1 The Gatherer

In order to fulfil the requirements of tweet collection and filtering, the design of the *Gatherer* consists of a number of subcomponents, pictured in Figure 3.2:

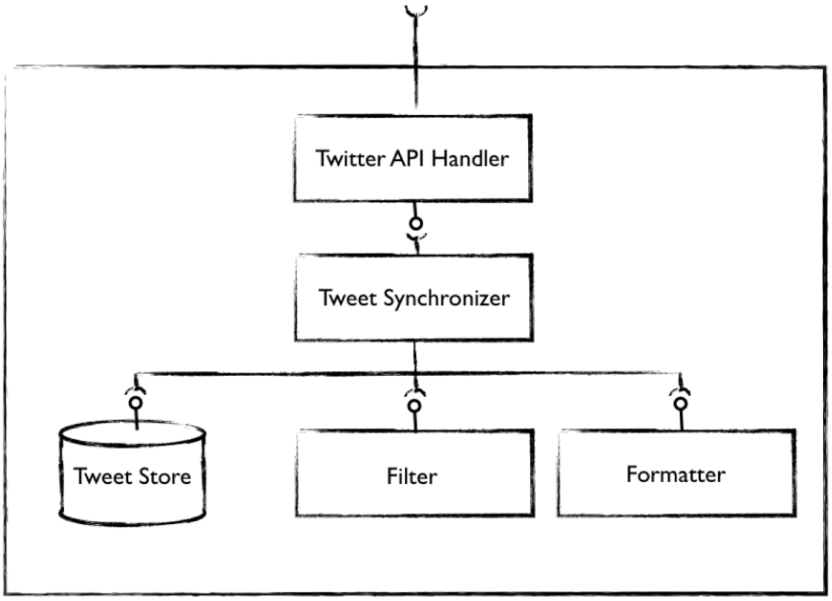

Figure 3.2: *Gatherer Components*

- *Twitter API Handler* - forms the interface between the tool and the Twitter API and is responsible for handling details of this interaction (authentication, persistent connections etc.).
- *Tweet Synchronizer* - co-ordinates the collection of tweets and their replies to ensure that the *corpus composition* requirement is met.
- *Tweet Store* - provides storage for tweets and associated metadata that have been collected directly from the API.
- *Filter* - responsible for ensuring that the *corpus quality* requirement is met through data cleaning.
- *Formatter* - outputs the results of the filtering process to the *Annotator* and *Learner* in a structured format.

3.2.2 Gatherer Discussion

There are essentially two key design decisions that must be taken with regard to the *Gatherer*. The first relates to achieving the required functionaility of *reply collection*. This process will rely on the collection of the metadata of tweets that are gathered from the *Twitter Streaming API* and exploiting this metadata to make requests for replies to the *Twitter REST API*. The procedure motivates the use of short-term data storage (provided by the *Tweet Store*) as a temporary holding for information that is used during the collection process before it is filtered off. Since the *Streaming API* produces tweets at a varying, unpredictable rate this process must be co-ordinated, justifying the presence of the *Tweet Synchronizer*, which provides this functionality.

The second design choice relates to the choice to include the filtering process as part of the *Gatherer* tool. The filter component is responsible for the quality of the corpus, and therefore holds a significant influence over the outcome of the research. Thus it is critical to ensure that this process is conducted both effectively and consistently. It is therefore proposed to place the filtering component as part of the gathering process, rather than as two separate components in both the *Annotator* and *Learner* components. This will ensure that the corpus is generated consistently and that the human study and machine learning algorithms have access to the same quality of data. The specific details as to how this filter is implemented are given in Chapter Four.

3.2.3 The Annotator

The component structure of the *Annotator* aims to ensure that each of the requirements relating to the human study are met. An MVC approach to the design is adopted, illustrated in Figure 3.3.

- *UI* - handles all direct interactions between the system and the study participant.
- *DB* - provides storage for all participant and study-related data.

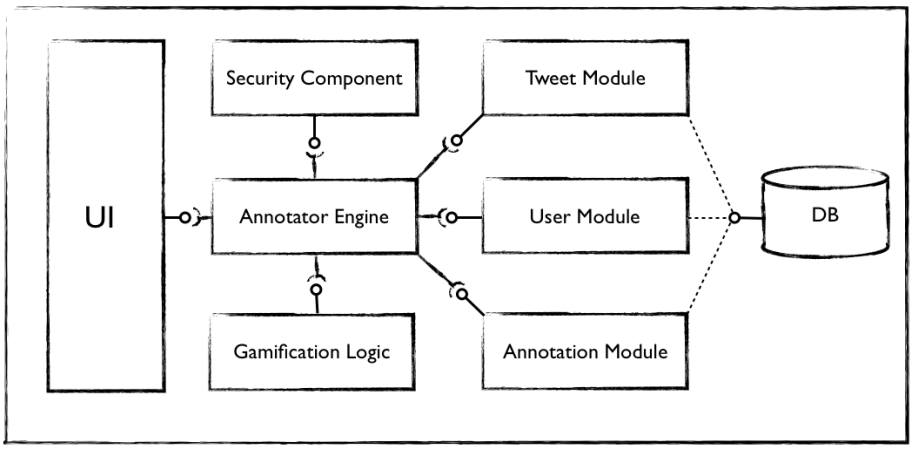

Figure 3.3: *Annotator Components*

- *User, Tweet* and *Annotation Modules* - each module handles the data and controls the functions and logic relating to the respective entities.

- *Gamification Logic* - responsible for the gamification of the study, with a view to addressing the *scalability requirement* discussed previously.

- *Security Component* - handles the security mechanisms needed to secure the application and any stored personal data to ensure that the *ethical requirement* is met.

- *Annotation Engine* - co-ordinates all other components.

3.2.4 Annotator Discussion

The purpose of the *Annotator* is to evaluate the ability of humans to perform the classification task. The tool must present tweets together with their replies and allow the participant to make a judgement. Scalability is a core requirement of the *Annotator*, since the ability of the tool to perform the evaluation in a representative manner is directly linked to the number of participant annotators and the quantity of annotations they produce. Consequently, this has been designed as an online tool, naturally suited to

the *Model-View-Controller* paradigm. Whilst creating an online tool has the benefit of increasing the scalability with regard to the number of participants, the role of boosting the number of annotations produced by each participant is addressed by the *Gamification Logic* component. The gamification of the study will be based on levels, associated with the volume of tweets annotated by the partcipant (not linked to their accuracy), to encourage productivity.

3.2.5 The Learner

Finally, in order to meet the requirements for the machine learning process, the components for the *Learner* have been designed as follows:

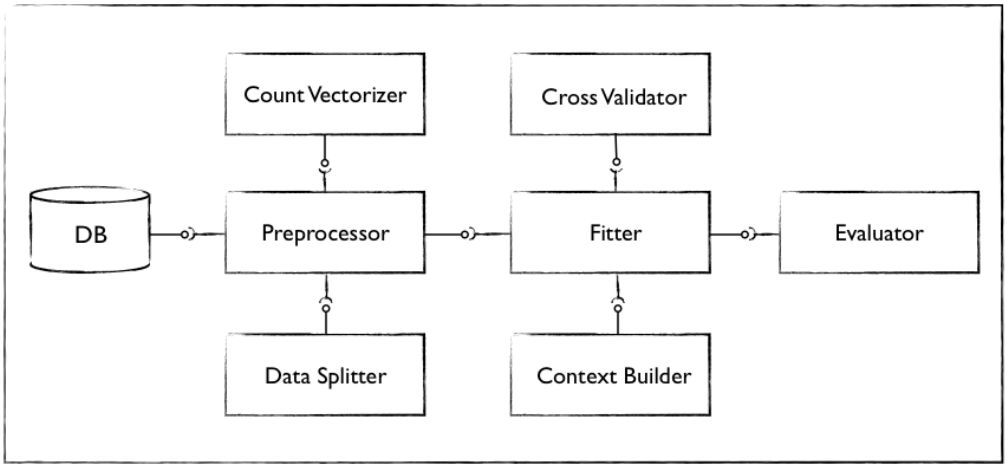

Figure 3.4: *Learner Components*

- *DB* - provides data storage functionality for the corpus and results.
- *Data Splitter* - handles the process of separating the corpus into *training* and *test* sets.
- *Count Vectorizer* - responsible for the construction of the lexical feature vector.
- *Preprocessor* - co-ordinates the data preparation procedure before algorithms are fitted.

- *Fitter* - fits each of the proposed algorithms to the constructed feature sets to ensure that the *algorithm selection* requirement is met.

- *Cross Validator* - performs 10-fold cross-validation for the classifier over the training set.

- *Context Builder* - handles the construction of contextual features, which may then be combined with the lexical feature vector to meet the *feature selection* requirement.

3.2.6 Learner Discussion

The *Learner* component is tasked with addressing the primary research question of this thesis, namely investigating the impact of bolstering lexical features with contextual clues. The design of the *Learner* represents an extension of the approaches to machine learning adopted in previous research, as discussed in Chapter Two. A range of algorithms are implemented, *10-fold cross-validation* is used in model-fitting and lexical features are constructed using a *Count Vectorizer* component. The additional element consists of the *Context Builder* component.

The *Context Builder* has the task of generating contextual features using the tweets and replies available in the corpus. Two types of contextual features will be constructed. The first will consist of the number of replies present in a conversation. The underlying assumption behind this feature is that the number of replies in some sense encodes the reaction of followers on Twitter. This reaction may contain clues as to whether or not the original tweet was sarcastic. The second type of feature is more sophisticated. It will involve splitting the *training set* into two portions and using one portion to classify the orignial tweet and the replies separately. These classifications will be encoded into a string that represents the pattern of the conversation. In this way, the feature can assert, for example, that the original tweet was positive, but that it had several negative replies, which may in turn provide a clue to the presence of sarcasm.

3.3 Summary

The objective of this chapter was to propose a design for a system that would allow the research question to be addressed. Firstly, through a process of requirements engineering, a set of requirements were gathered as a framework to achieving the research objectives. This was followed by a description of the system design at a high level, dividing the software into three components: the *Gatherer*, the *Annotator* and the *Learner* to match the functionality required. Finally, for each of these components a design was proposed and key elements of this design were discussed.

Chapter 4

Implementation

In this chapter, the implementation details of each of the three primary components of the design are discussed. The choice of technologies that are used to implement technologies used for each component are described and a justification for their inclusion is provided.

4.1 Gatherer Technologies

4.1.1 Twitter4J

The growth in popularity of the *Twitter API* has led to the development of a large number of libraries that facilitate interaction with the resource for different languages. After reviewing the available options, the *Twitter4J* Java library was selected. This decision was driven by a number of factors. Firstly, it is a mature *open source* library (it was released in 2007) that receives regular updates from a community of developers. Twitter regularly updates the API, so it is essential to use a library that interoperates with the latest changes. Secondly, unlike other alternatives, this library provides support for both the *Streaming API* and the *REST API*, with good documentation for each of the associated features. After building the *Twitter API handler* on this library, it was natural to implement the *Tweet Synchronizer* in Java, a language with excellent support for task scheduling.

4.1.2 MongoDB

Each tweet retrieved from the twitter API is associated with a large volume of metadata. This metadata must be stored to enable the *Tweet Synchronizer* to co-ordinate the reply collection process. *MongoDB* was a natural choice for this task - as a *document-oriented* database, it provides scalable storage without the need for the complex schema defintion that would be required by a SQL database to store the data in question. It therefore had the flexibility to deal smoothly with the variations that often occur in the more complex data collected from the API.

4.1.3 PyEnchant

The filtering and formatting processes rely heavily on text processing, mandating the use of *regular expressions* and dictionary manipulation. Of the number of languages suited to this task, Python was ultimately selected. Not only does it operate smoothly with regular expression construction, it enables the use of the flexible *PyEnchant* dictionary for the language filter, described below.

4.2 Filtering

The purpose of the filtering process is twofold. Firstly, it is designed to ensure the high quality of the corpus that will be made available to the machine learning algorithms. Secondly, it addresses the anonymity requirement of the data by removing content that could potentially be used to identify the author from tweets. To fulfil these roles, the filter consists of three components, illustrated in Figure 4.1. The first component addresses the issue of *duplicate tweets* (a common occurence among responses from the Twitter API). It removes any tweets that already exist in the corpus as well as removing *retweets* (a repeated post of another user's tweet), both of which could lend an unfair advantage to the machine learning process. The second filter addresses anonymity, replacing all

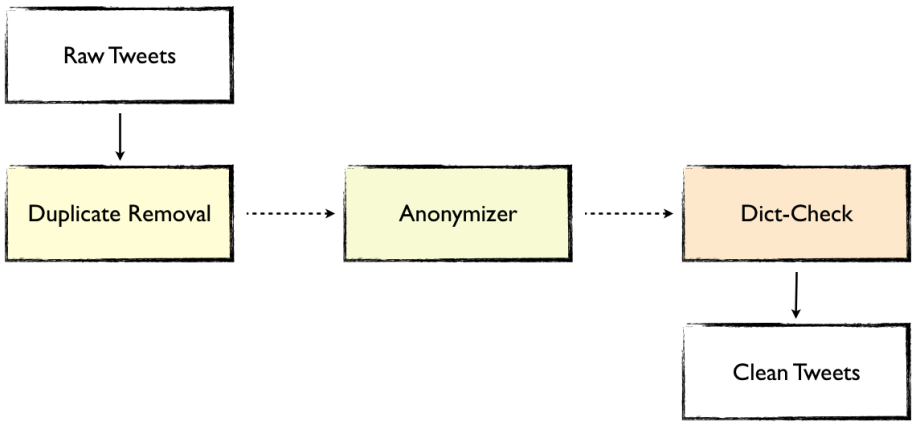

Figure 4.1: *Filtering Process*

targeted usernames in the tweets with '*@SomeUser*' and all hyperlinks with '*SomeLink*'. The final component of the filter is designed to ensure that the tweets are in English and are not simply random sequences of letters. However, the 140 character limit of tweets often encourages users to use abbreviations and words that do not occur in a dictionary. To address this issue, we define a simple metric, d: $words \rightarrow \{0, 1\}$ on each word w of a tweet as follows:

$$d(w) = \begin{cases} 1 & : w \in Dict \\ 0 & : w \notin Dict \end{cases}$$

where *Dict* is the set of words contained in the *PyEnchant* dictionary. For a tweet, t, to pass the filter, equation 4.1 must hold:

$$\sum_{w \in t} d(w) \geq \lceil (0.4).|t| \rceil \tag{4.1}$$

where $|t|$ denotes the number of words in the tweet. This mechanism, developed after a degree of experimentation, proved effective. It achieves the flexibility needed to ensure that tweets with abbreviations were still included, but tweets in other languages containing English hashtags were filtered out.

4.3 Annotator Technologies

4.3.1 Ruby on Rails

As noted in the design discussion of Chapter Three, the *Annotator* tool is based on the *MVC* paradigm. There are a number of frameworks that are designed to support development of such applications. From these, *Ruby on Rails* was selected for two reasons. Firstly, it has a vast community of active users, leading to good documentation for each aspect of the framework, easing the implementation process considerably. Secondly, it has a number of security features such as *CSRF protection* built into the framework, removing this complexity from the process of development.

4.3.2 D3.js

One of the challenges of producing a tool for conducting a human study is to give the participants a concrete understanding of what they must do for the task. Visualizations can be a powerful tool in aiding this process, clarifying concepts that are difficult to describe. The *D3.js* JavaScript library was used to produce interactive animations for this purpose. A snapshot of one such animation is provided in Figure 4.2:

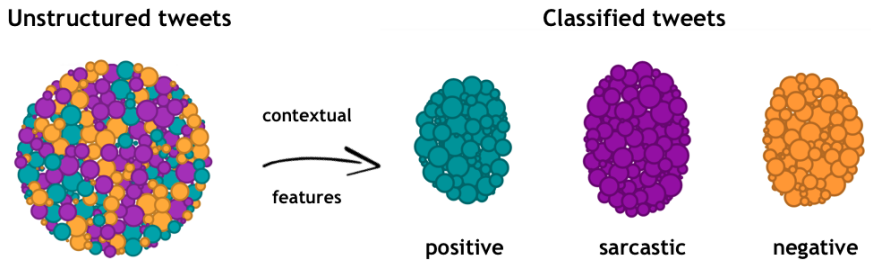

Figure 4.2: *D3 visualization*

4.4 The Classification Task

The purpose of the *Annotator* tool, is to create a benchmark for the difficulty of the classification task by evaluating the performance of humans. It is therefore critical to provide both the participants and the machine learning classifiers with identical information: the replies as well as the content of the original tweet. Figure 4.3 illustrates the user perspective for the task with a sample tweet:

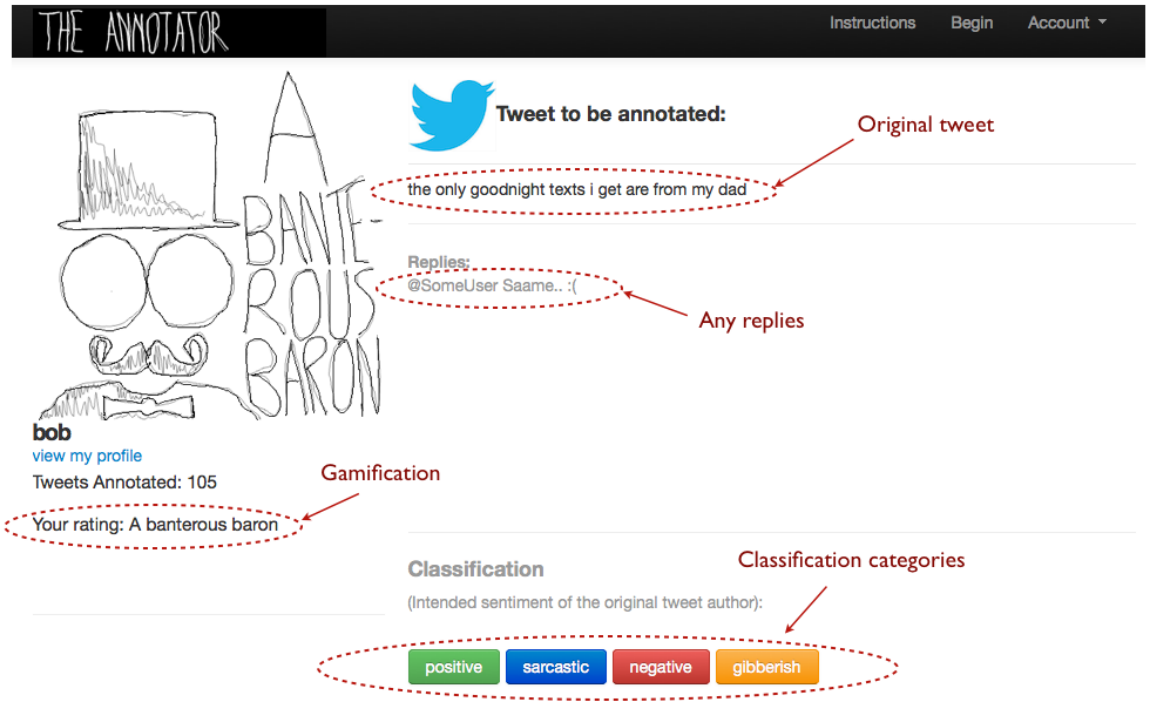

Figure 4.3: *The Annotator*

The participant is first presented with the tweet that requires classification, in this case: *the only goodnight texts I get are from my dad.* Next they are presented with any replies that were made to that tweet. For this example, only one reply was made: *@SomeUser Saame.. :(* - suggesting that the original tweet was negative. Four classification choices are available to the participant: *positive, sarcastic, negative* or *gibberish* (the last option allows them to indicate that the tweet did not make sense). Finally, note that the user is assigned a rating based on their annotations. This *gamification* technique is the subject of the next discussion.

4.5 Gamification

The gamification mechanism outlined in Chapter Three is based on a system of user levels. A fictional *humorous feudal hierarchy* was developed, designed as part of the process of encouraging participants to annotate greater numbers of tweets. As each participant completes certain threshold numbers of annotations, they advance through levels, unveiling new images relating to the latest levels that have been unlocked[1].

Upon starting the task, they are assigned the level of *dour serf* [66], the lowest rung of the humorous feudal society ladder. By making 15 annotations, they reach the next level in the feudal hierarchy, becoming a *humourless yeoman* [67]. At 30 annotations, they progress to a *cheerful merchant* [68]. The images associated with these levels are illustrated in Figure 4.4 below:

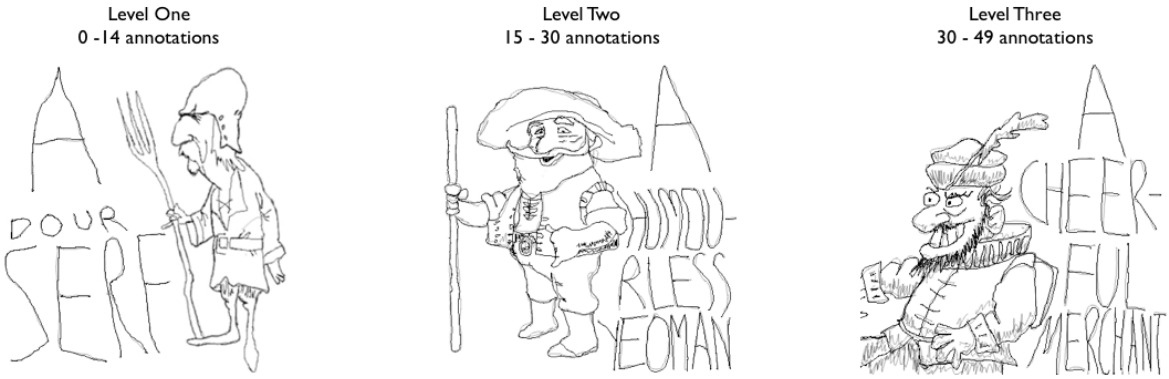

Figure 4.4: *Levels 1-3*

When 50 annotations have been made, the participant is rewarded with the level of a *mildly amusing jester* [69] (humour is considered more valuable than wealth in this fictional feudal society). The next level is that of a *witty knight* [70], awarded upon completion of 80 annotations. A pariticipant becomes a *banterous baron* [71] at 115 annotations. The relevant images are displayed in Figure 4.5.

[1]The images are stylized modifications of cartoons - a reference to each original is provided.

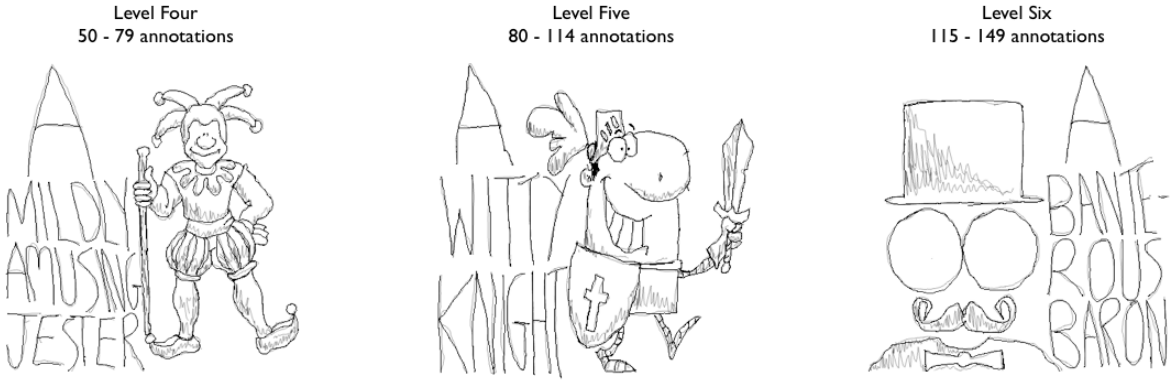

Figure 4.5: *Levels 4-6*

As the participant nears the peak of the humorous feudal society, they become royalty at 150 annotations, as a *sarcastic prince* [72]. Finally, upon completing 200 annotations, the participant reaches the highest level available, an *ironic emperor* [73], illustrated in Figure 4.6:

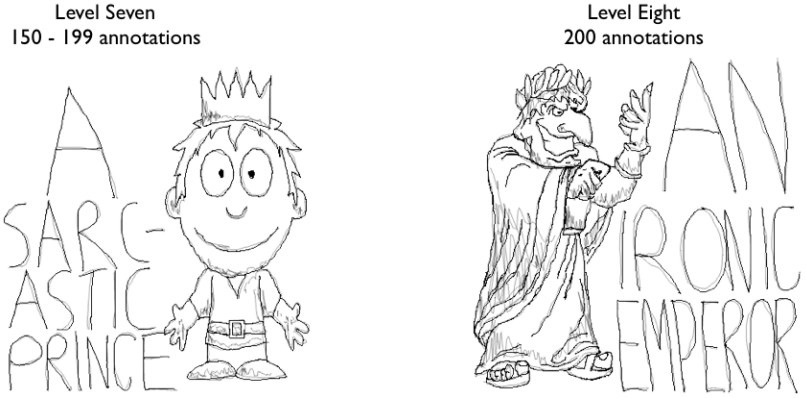

Figure 4.6: *Levels 7-8*

Ultimately, the objective of this process is to support the *benchmarking* research objective. The hierarchical implementation of the user levels reflects this goal by striving to motivate the participant to perform as many annotations as possible.

4.6 Learner Technologies

4.6.1 Scikit Learn

A number of tools have been created to facilitate the application of machine learning techniques. Popular libraries include *WEKA* [74], *NLTK* [75], *GATE* [76] and *Apache Mahout* [77], each of which have their own particular strengths and weaknesses. However, for this research it was decided that the Python *Scikit-learn* library [78] was most suited to the task. It offers support for each of the three algorithms outlined in the design, implements *cross-validation* and has excellent documentation.

4.6.2 IPython

The implementation of machine learning mechanisms often involves a great degree of tuning and frequent adjustment of subcomponents in order to achieve the greatest possible performance. For such scenarios, the *IPython* tool [79] is invaluable, allowing short sections of code to be run in isolation and facilitating incremental development. Consequently, it was a natural choice to use as the development platform for the *Learner*.

4.6.3 Classifier Training

The target of the *Leaner* is to investigate whether *contextual features* can have a positive impact on classifier performance. To achieve this, it is necessary to train classifiers on purely *lexical features* before the effect of combining these with contextual features can be evaluated.

4.6.3.1 Lexical Features

The corpus is first divided into a *training set* and a *test set* at a 60:40 ratio. The contents of the tweets are then split up into individual words, called *tokens*. The tokens are counted as word *n-grams* in various combinations - single words (*unigrams*), pairs of words (*bigrams*)

or triplets (*trigrams*). There are two popular ways of adding these counts to a feature vector. Either the counts are added directly, or the *term frequency-inverse document frequency* of each n-gram is calculated and this value is added to the vector. Other options for the feature vector include using the corpus with English *stop-words* (common English words that typically do not contribute meaning) removed from the tweet contents, and considering the tweets with all hashes removed from hashtags.

There are therefore numerous possibilities for constructing lexical feature vectors. To allow the research to be conducted within the time constraints set for the project, initial testing was performed to select the eight most promising sets of features. The combinations that were consistently yielding the best results are illustrated in Figure 4.7. Seven of these eight use *term frequency-inverse document frequency*, which initial testing showed to offer improvement over using token counts directly, except in the case when hashes were left in the corpus and the combination of unigrams and bigrams was considered.

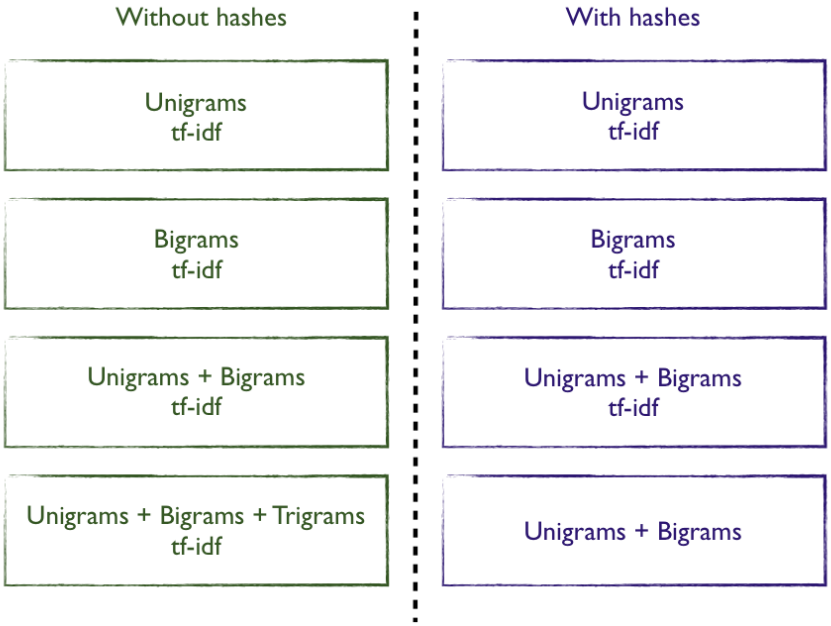

Figure 4.7: *Most promising feature combinations*

The *Naïve Bayes*, *Logistic Regression* and *SVM* classifiers were then individually trained on each of these combinations.

4.6.3.2 Contextual Features

Two types of contextual feature are investigated as mechanisms for improving classifier performance. The first, henceforth referred to as *COUNTS*, simply uses the number of replies to each tweet. Each number of replies is given a unique encoding and this is added to the feature vector.

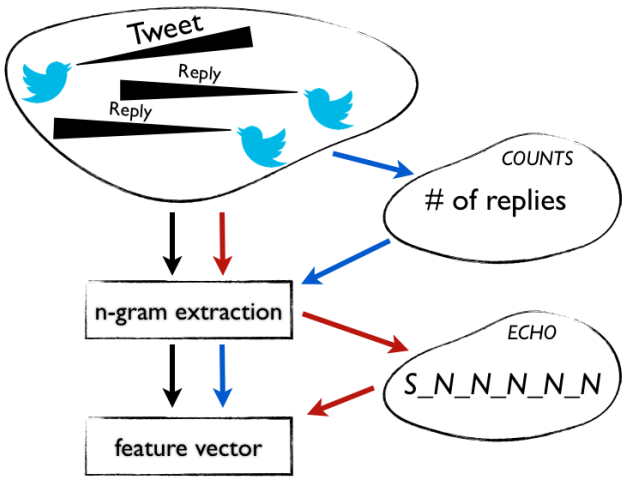

Figure 4.8: *Routes for feature construction*

The second feature type, referred to as *ECHO*, takes a different approach. Instead of dividing the corpus into a training set and test set, the corpus is divided into three, a *pre-training set*, a training set and a test set, with a 50:10:40 ratio. A classifier is trained on the pre-training and used to classify tweets and individual replies in both the training set and the test set. These classifications are used to produce *echo strings*, which represent the predicted classes of each part of a conversation. For example, if a tweet is classified as sarcastic, and each of five replies is classifed as negative, then the following output string would be produced: *S_N_N_N_N_N*. A second classifier is then trained on the training set, which now consists of these echo strings. It uses these patterns to learn whether certain types of reply can be good indicators of sarcasm, then classifies the test set with this information. The routes for feature construction for purely lexical features, *COUNTS* and *ECHO* are illustrated in Figure 4.8.

4.7 Summary

In this chapter, the technology choices for each component of the system were described and justified. The implementation details of the filtering, classification task and gamification processes were described. The chapter concluded by discussing the techniques for feature construction associated with the classifier training procedure.

Chapter 5

Evaluation

The purpose of this chapter is to evaluate the results of the research against the research objectives set out in Chapter One. To this end, a plan for the evaluation procedure is first outlined. Each component of the system is then evaluated according to this plan. Finally, the outcome of this evaluation is discussed.

5.1 Approach

The modular nature of the research objectives and consequently the design of the system built to address these naturally leads to an evaluation plan in which each objective is considered in turn. Thus, the approach is as follows:

- *To assess the corpus generation procedure and the effectiveness of the Gatherer at performing this task.*
- *To assess whether the Annotator has successfully provided a realistic benchmark for the difficulty of the task through human annotation.*
- *To assess the results of the Learner and hence address the primary research question of the thesis, investigating the impact of using contextual features together with lexical features.*

5.2 Gatherer Evaluation

In order to evalutate the *Gatherer*, it is first necessary to consider the criteria that should be used to assess *corpus generation*. Typically, each corpus is gathered with the aim of addressing a research question, and therefore different characteristics are considered important for different research questions. There is certainly a degree of subjectivity in this selection task. However, the literature review conducted in Chapter Two indicated two uniformly desirable corpus properties, namely:

- *Corpus size* - the number of tweets and replies gathered.
- *Corpus quality* - Is the dataset free of spam, duplicates and multilingual tweets?

Ultimately, the *Gatherer* must be assessed on whether it has achieved its objective of generating a corpus that enables the research question of the thesis to be addressed.

5.2.1 Gatherer Results

The raw data gathered directly from the Twitter API consisted of:

- 51,549 tweets in total
- 29,791 original tweets (8973 sarcastic, 12433 negative and 8385 positive)
- 21,758 reply tweets

It is worth noting that on average, one out of every 6.38 tweets prompted at least one reply in this raw data. The filter was then applied to remove *noisy* tweets - duplicates, spam and tweets in multiple languages with English hashtags. Tweets in which the hashtag was the subject, rather than an indication of the tone were also removed.

The application of the filter removed over half of all tweets in the raw dataset, leading to the following composition for the final corpus:

- 24,287 tweets in total
- 15,245 original tweets (5049 sarastic, 6118 negative and 4078 positive)
- 9,042 reply tweets

Figure 5.1 illustrates the transition from the original dataset to the final corpus.

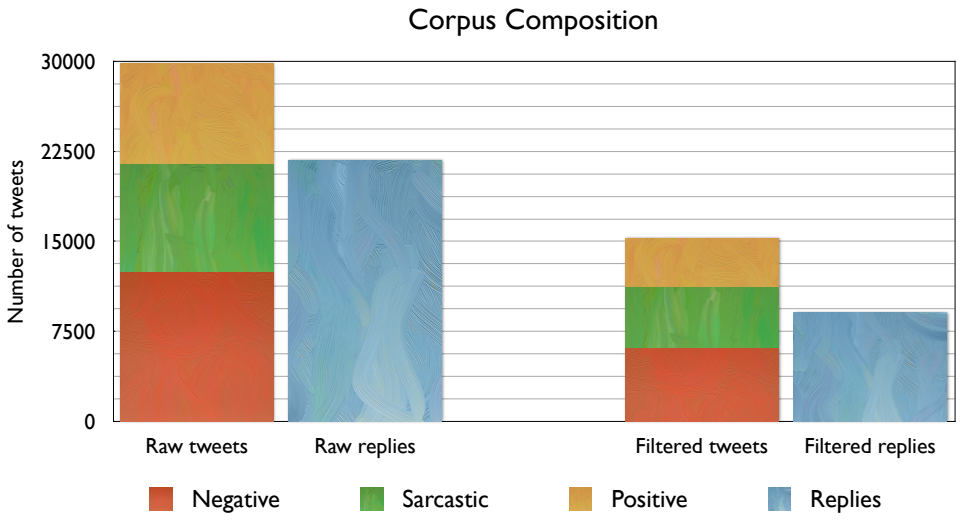

Figure 5.1: *The impact of filtering on corpus composition*

Interestingly, in the final corpus, one out of every 5.16 tweets prompted a reply, a higher ratio than on the raw dataset. This may result from the fact that Twitter users do not tend to respond to spam and nonsensical tweets. It is also worth noting that the number of the replies generated by a tweet follows an approximate power law. Figure 5.2 depicts the structure of the data in terms of the number of replies to tweets that prompt any responses. At most, five replies to a specific tweet are used in the final corpus, which, as demonstrated in Figure 5.2, equates to a cut at the 70th percentile. This decision was taken to ensure that the second of the contextual features, *ECHO*, could be applied

effectively. If tweets with up to nineteen replies were used to construct the *echo-strings*, there would be too many unique string patterns for a dataset of this size, so the classifier would inevitably *overfit* the data. This restriction enables the majority of the replies to be used in the corpus whilst at the same time avoiding this overfitting issue.

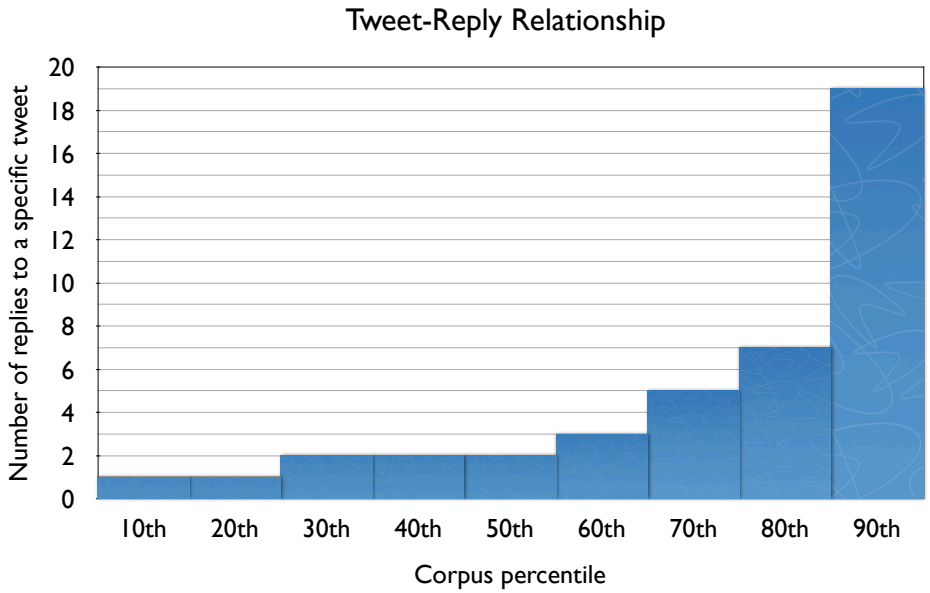

Figure 5.2: *Replies to tweets*

There were then two alternatives available for the production of the final corpus. The first was to only use tweets which have at least one reply, signficantly reducing the size of the dataset, but potentially allowing the impact of contextual features to be seen more readily. The second was to leave the dataset unedited after the filtering process, keeping those tweets that do not contain replies. The related work in Chapter Two consistently showed that working with a larger corpus not only led to stronger classifiers, but also yielded fairer evaluations. Consequently, the decision was taken to use the full set of filtered tweets as the final corpus.

5.2.2 Discussion

With the data at hand, it is now possible to apply the assessment criteria and evaluate the *corpus generation* process. A dataset of polar and sarcastic tweets has been gathered at an order of magnitude greater than the previous corpus collected for this task [8]. Although the dataset could be improved by the collection of a greater number of tweets, the *Gatherer* performed well to collect this magnitude of tweets within the tight time constraints set for the research. The *noisy* tweets have been removed from the dataset, leading to a corpus of good quality. The class weightings are reasonably (though not perfectly) balanced, given the natural fluctuations that occur in the rates of tweet collection from the *Twitter Streaming API*. Critically, tweet replies have been successfully included in the dataset. Thus the *Gatherer* has successfully achieved its objective of creating a corpus that enables the other components of the system to address the research question in more detail.

5.3 Annotator Evaluation

The purpose of the *Annotator* is to provide a benchmark for the difficulty of differentiating between sarcastic and polar tweets by evaluating the performance of humans at this task. The key assessment criteria for the effective fulfillment of this task are:

- *Sample size* - the number of participants and the quantity of annotations made.
- *Production of performance metrics* - the tool should produce representative performance metrics for the human participants that will facilitate comparison with the *Learner*.

5.3.1 Annotator Results

The *Annotator* study was conducted on 2000 tweets of the corpus (taken from the *test set* described in Chapter Four). In total, the raw data returned by the tool consisted of:

- 60 participants
- 5327 annotations in total

Thus on average, each user made a total of 89 annotations. The average accuracy of the human participants across all annoations was was 62.49%. For the class of most interest, sarcastic tweets, the participants achieved a *precision* of 78.97%, a *recall* of 49.96% and an *f-score* of 60.98%. These results are illustrated in Figure 5.3.

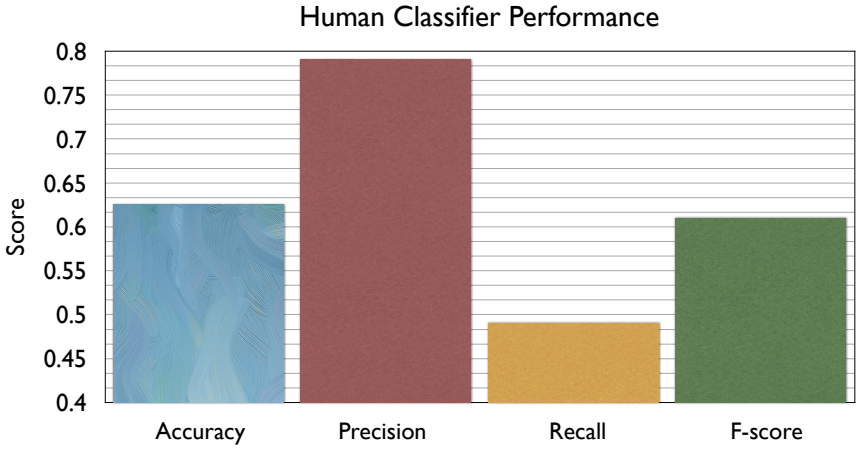

Figure 5.3: *Annotator results*

5.3.2 Discussion

The 60 participants consisted of fluent English-speaking colleagues from the Computer Science M.Sc courses at Trinity College, Dublin together with researchers from the Knowledge and Data Engineering Group (KDEG) at the same University. Each of these participants was familiar with Twitter and the associated style of communication. Although this constitutes a biased sample of the total population of fluent English speakers (by

selecting only participants from academia), the sample size was appropriate for fulfilling the objective of providing a benchmark for the difficulty of the task.

The second result to be considered is the number of annotations, 5327 in total. As noted above, this equates to 89 annotations per user. Since each participant was free to annotate as many or as few tweets as they wished, this suggests that the *gamification* process undertaken was successful. A larger number of annotations per user corresponds to a greater confidence in the representativeness of the resulting performance.

Interestingly, the mean accuracy achieved by the participants in the study (62.49%) is almost identical to the accuracy score (62.59%) of the participants in a previous study [8], who performed a similar task on a much smaller dataset. This consistency amongst results suggests that the *Annotator* has performed well in the task of providing a benchmark for the difficulty of the task. The *precision* score of the participants (78.97%) is reasonable, but the *recall* (49.96%) is poor, indicating that humans struggle to detect sarcasm in this task, but that when they see it, they are confident of asserting its presence. These results add further support to the decision to label the corpus using the *hashtag* technique rather than human judges. The low recall/high precision implies that humans tend to only recognize the more obvious instances of sarcasm, which may provide an indication as to why the performance of the *SASI* algorithm (discussed in Chapter Two) on a human labelled corpus [63] was so high (94.7%). The low *f-score* (60.98%) further reinforces the difficulty of detecting sarcasm amongst polar tweet for human judges. [1] The primary limiting factor to the study (for increasing the representativeness of the evaluation of human performance at sarcasm recognition) was the number of participants. However, given the time constraints for the research, and noting that the results are so close to the smaller previous study, it may be asserted that the *Annotator* has achieved its aim of

[1]Note that the operational environment of the *Annotator* meant that the number of participants and annotations were not known in advance, and the structure of the resulting data meant that the *Fleiss' Kappa* statistic for inter-annotator agreement was not appropriate in this case.

providing a realistic benchmark for the difficulty of the sarcasm detection task.

5.4 Learner Evaluation

The final component of the system, the *Learner*, is tasked with addressing the research question of thesis, namely to investigate the impact of using contextual features to bolster lexical feature-based models. To this end, the evaluation of the *Learner* will consist of:

- *Results analysis* - the output of the *Learner* will be analysed with a set of performance metrics.
- *Comparison* - The results will be compared against the performance of classifiers in previous research.
- *Contextual impact* - The impact of using contextual features for models will be evaluated.

5.4.1 Learner Results

To investigate the impact of contextual features, the *Learner* was used to conduct three experiments. In the first, replies to tweets were removed from the corpus and classifiers were trained with each of the eight chosen lexical feature combinations. In the second, the full corpus (including replies) was used to train classifiers with the same sets of lexical features. In the final experiment, classifiers were trained on the full corpus using the contextual features described in Chapter Four. These combinations of experiments, features and algorithms lead to numerous sets of results. Consequently, in order to portray the outcomes as clearly as possible, a protocol for displaying the results of each experiment has been developed.

Firstly, the *cross-validation* scores for every classifier used in the experiment are presented. Following this, detailed performance metrics are given for only the classifiers

which achieve the best cross-validation scores, focusing on the results that are most significant. The results for each experiment are then discussed. Finally, a comparison of the best classifiers from each type of experiment is offered.

5.4.1.1 Replies Removed

A set of classifiers were first trained on the corpus without the tweet replies, using each of the eight combinations of lexical features. The purpose of this initial experiment was to provide a baseline performance for machine learning models at this task and to facilitate direct comparison with previous research (in which replies were not used).

The *cross-validation* scores of these classifers are displayed in Figure 5.4, giving an indication of how well each model fits the training data. The best cross-validation score (69%) is achieved by both an *SVM* classifier and a *Logistic Regression* classifer. [2]

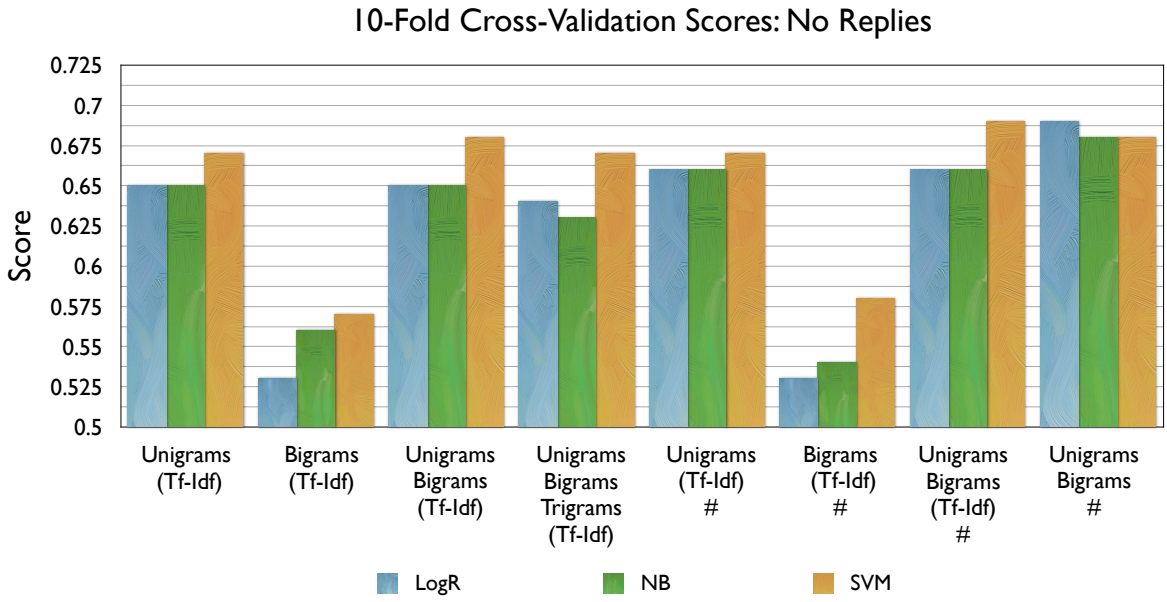

Figure 5.4: *Cross-validation scores without replies*

In-depth performance characteristics of the two classifiers achieving the highest cross-

[2]The best *SVM* classifier used *unigrams* and *bigrams* together with *tf-idf* on the dataset containing hashes, while the *Logistic Regression* classifier also used the combination of *unigrams* and *bigrams* but without *tf-idf*.

validation scores are displayed in Figure 5.5. Both classifiers obtained accuracy scores of 68%, although the *Logistic Regression* classifier outperformed the *SVM* classifier on *precision* (67% vs 66%), *recall* (66% vs 65%) and *f-score* (66% vs 67%).

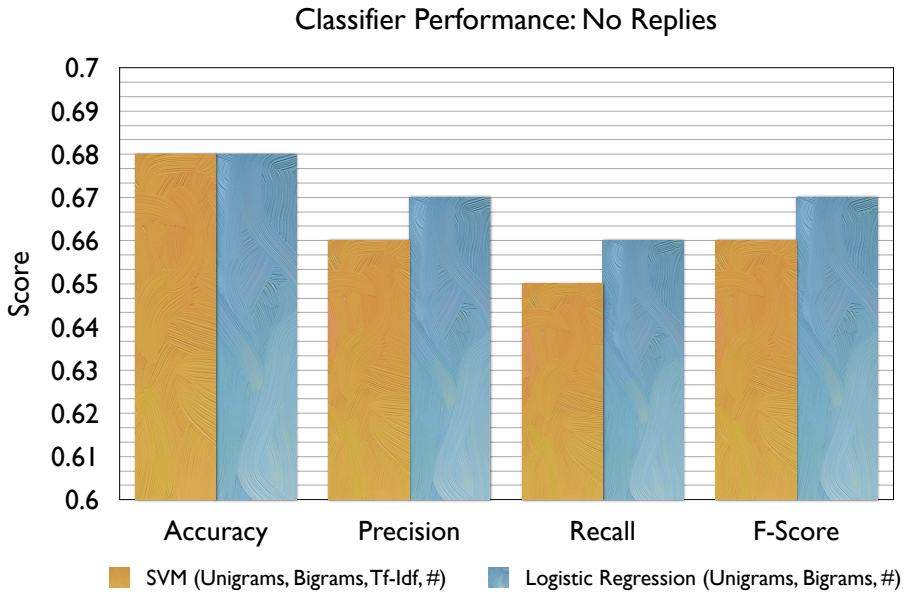

Figure 5.5: *Classifier performance without replies*

This experiment yielded a number of interesting results. Although on average the *SVM* classifiers perform best, no algorithm outperformed both of the others on every feature set. The cross-validation results in Figure 5.4 verify the importance of using a range of algorithms and features, justifying the approach taken with this research. The effectiveness of different feature sets was also shown to be largely random in nature, particularly with regards to using *tf-idf* with the feature vector, the influence of which was very unpredictable with the final combination of features. Of the two best fitting classifiers, one used *tf-idf* to load the feature vector and one did not, despite initial testing indicating that it should consistently improve results.

By considering the shape of the performance metrics graph in Figure 5.5, the character of the best classifier (produced by this experiment) can be established. The higher accuracy score suggests that it performed slightly better at classifying positive and nega-

tive tweets than sarcastic tweets, but with only a 3% spread over *accuracy*, *precision* and *recall*, the classifier can be seen to be well-balanced.

With the replies from the final corpus removed for this experiment, the results can be compared to the performance of classifiers in previous research on the same task. The best classifier achieved an accuracy of 68%, which may be contrasted with a best score of 57% in previous work [8], a considerable improvement. This may be primarily attributed to the difference in the size of the corpus used to train the classifiers (15,245 tweets vs 900 tweets). The literature review repeatedly underlined the significance of using a large dataset for precisely this reason. The extensive filtering process performed by the *Gatherer* tool may also have contributed to the improvement in performance.

5.4.1.2 Purely Lexcial Feautres

The second experiment conducted with the *Learner* tool trained classifiers with purely lexical features on the full corpus (including replies). As with the previous experiment, every algorithm is applied to each of the eight feature-combinations. The cross-validation scores for classifiers on the dataset with replies are illustrated in Figure 5.6.

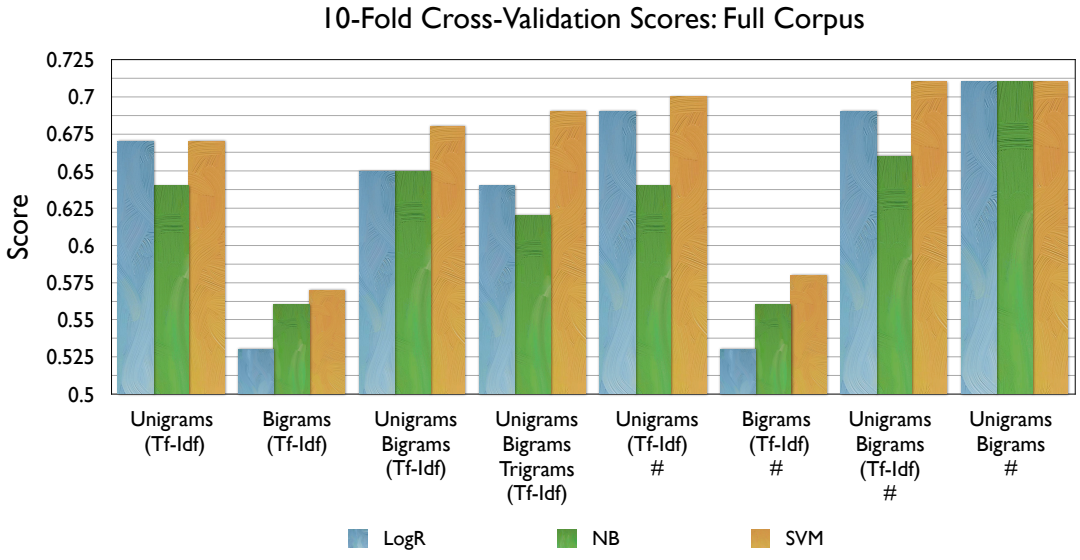

Figure 5.6: *Cross-validation scores with replies*

Four classifiers achieve a cross-validation score of 71%, with each of the three algorithms attaining the same score on the final lexical feature set. The performance characteristics for the four best-fitting classifiers are displayed in Figure 5.7.

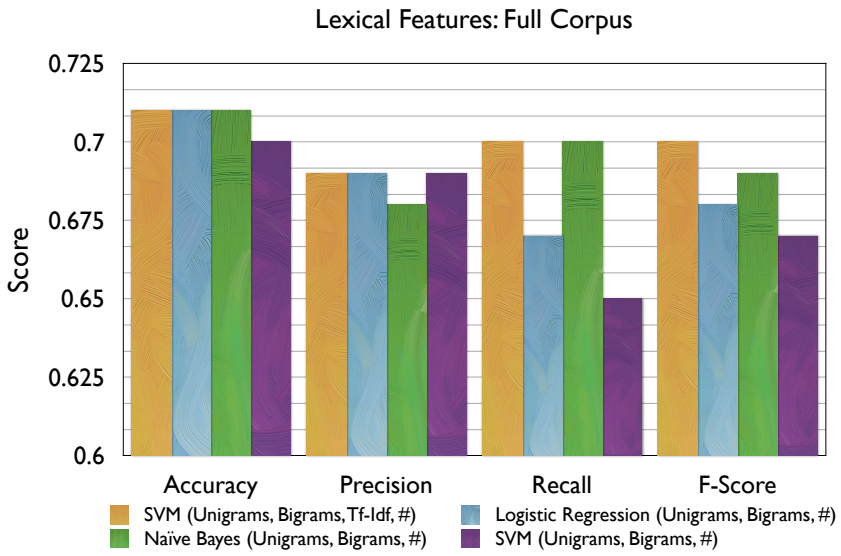

Figure 5.7: *Classifier performance with replies*

The best performance on the full corpus with purely lexical features was achieved by the *SVM* classifier [3] with a *mean accuracy* of 71%, a *precision* of 69% and both a *recall* and *f-score* of 70%.

A comparison of the results of the first and second experiments indicates that simply giving the classifier more data to work with for each classification (now it can use the tweet and its replies) results in a marked improvement in performance. Interestingly, for the best classifier in both experiments, the precision score for the sarcasm class is 2% lower than the accuracy. This suggests that although the overall performance has improved, the replies have not made improved the ability of the classifier to detect sarcasm relative to the other two classes when only lexical features are considered, highlighting one of the concerns about the limitations of this approach.

[3]This classifier used *unigrams, bigrams, tf-idf* and *hashes*.

5.4.1.2 Contextual Feautres

The final experiment used the *COUNTS* and *ECHO* contextual feature constructions to train classifiers on the the full corpus. As described in Chapter Four, both of these approaches work in combination with a lexical feature set. Out of the eight groupings of lexical features examined, the three most promising combinations were used with each approach, once with each algorithm.

The best cross-validation score (72%) was achieved by the *COUNTS* approach with both the *SVM* and *Logistic Regression* classifiers [4]. The cross-validation scores for all of the six classifiers are depicted in Figure 5.8

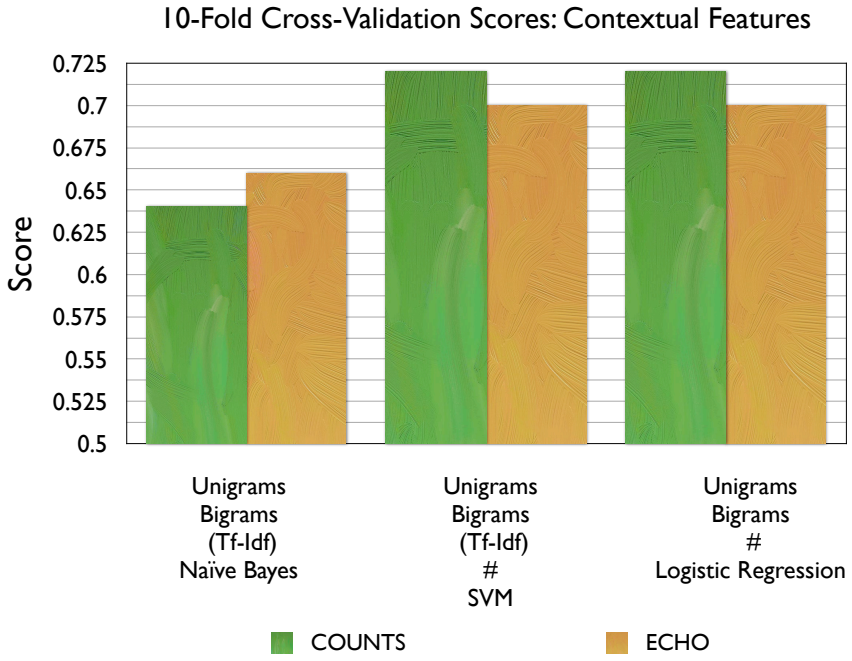

Figure 5.8: *Cross-validation scores for contextual features*

The strongest COUNTS-based classifier significantly outperformed the best ECHO-based classifier on *mean accuracy* (73% vs 69%), *precision* (73% vs 68%) and *f-score* (72% vs 70%) performance characteristics of the best classifier for each type of feature is

[4]This *SVM* classifier used *unigrams, bigrams* and *tf-idf* on the corpus with hashes, while the *Logistic Regression* classifier used *unigrams* and *bigrams* but not *tf-idf*.

displayed in Figure 5.9.

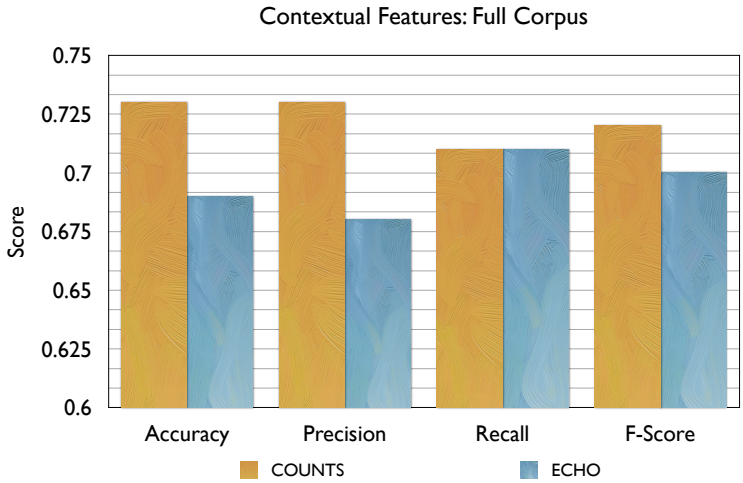

Figure 5.9: *Performance scores for contextual features*

The results of the third experiment show improvement in performance when using contextual features with the *COUNTS* approach. It outperforms the best lexical feature-based classifier on every score. Perhaps more importantly, the *precision* score for this classifier matches the *accuracy* score of 73%, implying that unlike the previous classifiers, this classifier is able to detect sarcasm as well as positive and negative tweets. Despite the simplicity of the features, the ability to leverage the content of replies in context has had an impact on the performance of the classifier.

The *COUNTS* classifier did not perform so well, proving to be less effective than the best of the purely lexical feature-based models. There are two possible reasons for this. Firstly, it relies heavily on the quality of information it can glean from the classification of the replies. However, as noted in the evaluation of the *Gatherer*, only one out of every 5.16 tweets received a reply, limiting the effectiveness of the approach on this corpus. Secondly, the implementation of this technique the splitting the data into three, a *pre-training set*, a *training set* and a *test set*, rather than the two sets used by the other approaches. This meant that in effect, rather than being used to increase the size of the training set of the classifier, 10% of the dataset was instead used to construct reply patterns. Trading this

quantity of training data in order to leverage context did not prove to be worthwhile on a corpus of this size.

5.4.1.3 Discussion

The three experiments conducted with the *Learner* allow the research question posed in this thesis to be addressed. The first experiment provided a baseline performance for the task and demonstrated the quality of the corpus. The second showed that lexical feature-based models improve when provided with extra data in the form of replies, although the relative performance of detecting sarcasm does not improve as might be expected. Finally, the third experiment illustrated that contextual features can indeed have a positive impact on classifier performance, not only in overall accuracy but also in specifically detecting instances of sarcasm. The relative performances of different classifiers considered in this research are illustrated in Figure 5.10 which shows the performance of humans, the best lexcial feature-based classifier, the best *COUNTS* classifier and the best *ECHO* classifier.

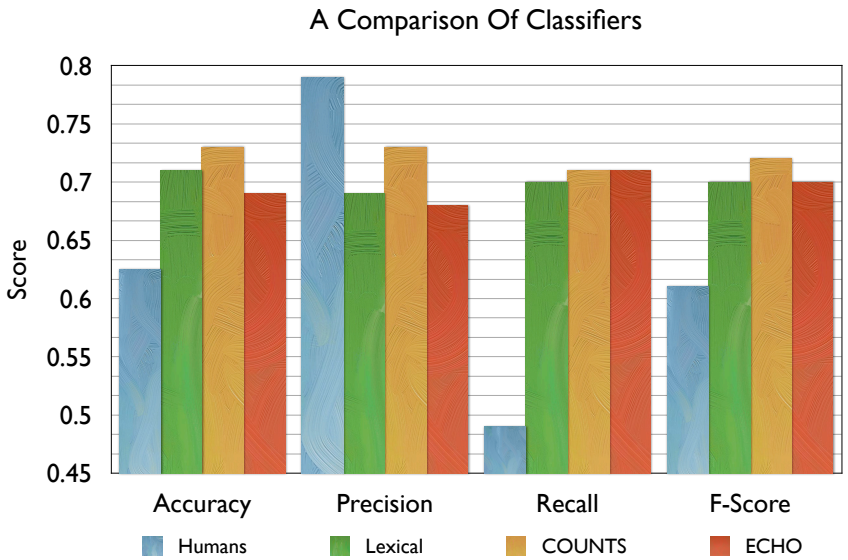

Figure 5.10: *A comparison of classifiers*

The performance gains are small, but it is worth noting that while the lexical approach has received a great deal of research interest and optimization, contextual features have remained largely ignored in this task. The fact that of just two contextual features

that were considered, one offered a performance improvement, suggests that there is the potential for a great deal of further progress to be made in this area, particularly with the application of simple features such as the *COUNTS* mechanism.

As the datasets used for sentiment analysis continue to grow and the lexical feature-based models reach a performance plateau, there is also the possibility that the more sophisticated *ECHO* approach may prove to be effective. Whilst uneconomical with data (rendering it ineffective on a corpus of this magnitude), it is capable of extracting more complicated interactions from the data which would be a powerful asset on much larger datasets. The evaluation has demonstrated that contextual features can be used to improve classifier performance and illustrated the potential that exists in this direction, but there is a great deal of further work to do be done before contextual features are exploited to their full potential.

5.5 Summary

In this chapter, an evaluation plan was developed to assess each component of the system from the perspective of the research question of the thesis. The first component of the system to be evaluated was the *Gatherer*, which was demonstrated to have satisfied the research objective of effective corpus generation. The contribution of the *Annotator* was then evaluated, specifically with regard to the quality of the benchmark it provided for the difficulty of the task. The final part of the evaluation considered the contribution of the *Learner* tool and its role in answering the research question posed by the thesis. The results demonstrated that the impact of contextual features and indicated the potential for further improvements in the task of sarcasm detection.

Chapter 6

Conclusions

The purpose of this chapter is to demonstrate that the research objectives and challenges described in Chapter One have been met, to summarise the research contribution of this thesis and to discuss future work. To this end, the extent to which each research objective has been addressed is discussed as part of a broader consideration of the research as a whole. This is followed by a summary of the most significant contributions of the thesis and a discussion of the potential for future work.

6.1 Objective Assessment

6.1.1 Sarcasm Detection, A Review

Research objective: *To conduct a survey of existing approaches to sarcasm detection, together with an overview of the underlying machine learning and linguistic theory.*

Targeting this objective, Chapter Two began by reviewing Twitter, identifying both the benefits and the unique research challenges associated with working with Twitter content, particularly those due to the brevity of tweets. It was noted that as a consequence of this brevity, Twitter users are inclined to make heavy use of abbreviations, an observation that impacted later design decisions in the filtering process. To provide a framework for

the machine learning task, the goals of machine learning and the concepts of *classification*, *supervised learning* and *evaluation metrics* were considered in detail. Two observations were noted with regard to this field: the importance of avoiding *overfitting* and the critical requirement that a range of algorithms and feature sets should be implementated in the machine learning task, if good performance is to be achieved. Next, the linguistic theory relating to sarcasm and the broader field of *irony* was investigated, examining the main challenges of working with the concept of sarcasm and the fundamental importance of context in sarcasm detection. A review of the related work facilitated an examination of the strengths and weaknesses of techniques that have been applied to this problem. The two key conclusions drawn from this process were that contextual features could offer the potential to achieve greater ressults than the lexical features that had been studied previously, and that the size of the dataset used to train the classifiers used for detection has a critical impact on performance. Consequently, it was decided to gather as large a corpus as possible within the time constraints of the research, and morever, that the corpus should include the replies to tweets as a mechanism for extracting context.

6.1.2 Corpus Generation

Research objective: *To gather a corpus of reliably labeled tweets that can be used to train and accurately evaluate the relative performance of different classifiers.*

The most significant influence on the performance of a classifier is the quality of the corpus used to train it. Multiple labelling techniques were considered for the dataset, but after review, the *#sarcasm* labelling method was used. Although this leads to pessimistic performance scores for the classifiers, it reflects the sentiment of the original tweet author and represents the most reliable approach available. To allow contextual features to be constructed, replies to tweets were gathered as part of the corpus. A scalable tool, the *Gatherer* was built to conduct each of these steps in a scalable manner, a task in which

it performed well. The result was a clean, balanced corpus of appropriate size with the properties required to allow the research question to be addressed.

6.1.3 Human Study

Research objective: *To provide a realistic benchmark for the difficulty of differentiating between sarcastic and polar tweets by conducting a study in which humans perform the classifying task on the dataset.*

The study conducted using the *Annotator* tool supported the assertions of previous work, namely that humans perform poorly at this task. The average accuracy of 62.5% underlined how challenging the process of separating sarcasm polar tweets was for the 60 participants. The integrity of this accuracy score was supported by a small scale study in earlier research [8], in which 3 judges scored an average accuracy of 62.6%. Perhaps the most significant consequence of this result is that it illustrates the severe limitations of a human annotated dataset, leading to a corpus consisting of only the most obvious types of sarcasm.

6.1.4 Machine Learning

Research objective: *To design and build a range of machine learning models that utilize both purely lexical features as well as models that leverage contextual features.*

Both lexical and contextual features were investigated with three algorithms for the classification task. The range of models created by the *Learner* tool produced promising results, with the *COUNTS* approach outperforming all of the lexical feature combinations with an accuracy of 73%. The more sophisticated *ECHO* approach did not fare so well. It was uneconomical in its use of the training data, which proved too much of a penalty over any performance gains that might have been achieved by a more accurate modelling

of context.

6.1.5 Final Remarks

Research objective: *To conduct an evaluation of each approach, assess performance implications and offer suggestions for potential future work.*

The meaningful output of this research has been a demonstration of the considerable potential of contextual features as part of a mechanism for sarcasm detection. As noted in the evaluation, the performance gains over the state of the art lexical feature-based approaches are small. However, while lexical techniques have received a great deal of research attention, this thesis suggests that there is more to be gained through (largely neglected) contextual approaches, which have by no means been optimized to their full potential. The growing importance of sentiment analysis as a tool to extract useful insights from this data means that the problem of sarcasm detection is likely to receieve a great deal more attention. Recent news reports indicate that governments and major corporations are now investing significant sums of money in services that provide sarcasm detection for social media [80]. As the aggregation of opinionated content continues on a global scale, the data becomes rich enough to justify the use of techniques that can leverage linguistic theory in a more meaningful way, suggesting that contextual approaches represent the future for sarcasm detection systems.

6.2 Summary Of Contribution

The contributions of this research are as follows. A detailed literature review has been provided into the field of sarcasm detection, together with an analysis of the theory from the associated fields of linguistics and machine learning. A filtering mechanism for cleaning Twitter data based on a dictionary metric, which may be easily generalized to other types of user-generated content, has been proposed. A filtered corpus of polar and

sarcastic tweets together with their replies has been constructed. By conducting a human study, a benchmark for the difficulty of the classification task has been provided. An approach has been described for combining contextual features with lexical features and in particular, the *COUNTS* and *ECHO* techniques have been proposed as a mechanism for feature construction. Finally, the research contributed results demonstrating that contextual features can outperform lexical features when applied to the task of sarcasm detection.

6.3 Future Work

There are a number of potential extensions to the research undertaken in this thesis.

6.3.1 Large Corpus Generation

The time constraints of this research placed a restriction on the size of the corpus that could be generated. While the more sophisticated *ECHO* technique for feature construction did not perform well on a small dataset, its true value comes from its ability to extract more complicated contextual clues than the shallow feature-based appraoch, suggesting that to unlock its potential, much larger datasets are needed. It would therefore be an interesting research project to apply *ECHO* and other such approaches to a corpus that is several orders of magnitude greater in size than the one constructed in this research. An exciting complementary study would be to investigate the point at which the corpus is big enough that simply adding more tweets to the training corpus no longer yields significant improvements to lexical feature-based approaches.

6.3.2 Generalising To Other Literary Devices

The core principles underpinning this research apply not only to sarcasm, but also to many other literary devices, most notably irony. Given the significance of context in

every form of humour, further research could investigate the possibility of generalising the approaches developed with the *Learner* tool to other literary devices.

6.3.3 Geographical Variations In Sarcasm

Since completing the implementation of this research, work has begun on the construction of another corpus to investigate a slightly different phenomenom, namely, the regional variations that occur in the usage of sarcasm, discussed in [43]. As personalization becomes an increasingly significant component of web interaction, the ability to model and detect sarcasm specific to the region in which it occurs would be a powerful tool in improving the performance of sarcasm detection systems. Consequently, this idea offers a particularly intriguing avenue for future research.

Appendix A

Abbreviations

Short Term	Expanded Term
API	Application Programming Interface
CSRF	Cross-Site Request Forgery
D3	Data-Driven Documents
JS	JavaScript
KDEG	Knowledge and Data Engineering Group
MVC	Model-View-Controller
NB	Naïve Bayes
SVM	Support Vector Machines
Tf-Idf	Term frequency-Inverse Document Frequency
UI	User Interface

Bibliography

[1] Y.Liu-Thompkins and M.Rogerson, *Rising to Stardom: An Empirical Investigation of the Diffusion of User-generated Content*, Journal of Interactive Marketing 26.2, [2012]

[2] D. Terdimna, *Report: Twitter hits half a billion tweets a day*, http://news.cnet.com/8301-1023_3-57541566-93/report-twitter-hits-half-a-billion-tweets-a-day, [2012]

[3] A. Balahur, Z. Kozareva and A. Montoyo, *Determining the polarity and source of opinions expressed in political debates*, Lecture Notes in Computer Science, 5449, 468-480, [2009]

[4] O. Netzer, R. Feldman, J. Goldenberg and M. Fresko, *Mine your own business: Market-structure surveillance through text mining*, Marketing Science, 31(3), 521-543, [2012]

[5] R. Feldman, *Techniques and applications for sentiment analysis*, Communications of the ACM 56.4, [2013]

[6] B. Pang and L. Lee, *Opinion mining and sentiment analysis*, Foundations and trends in information retrieval, 2.1-2, pp1-135, [2008]

[7] A.Reyes and P. Rosso, *On the difficulty of automatically detecting irony: beyond a simple case of negation*, Knowledge and Information Systems, [2013]

[8] R. Gonzlez-Ibez, S. Muresan and N. Wacholder, *Identifying sarcasm in Twitter: a closer look*, Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers. Vol. 2. [2011]

[9] B.C. Wallace, *Computational irony: A survey and new perspectives*, Artificial Intelligence Review, 1- 17, [2013]

[10] D. Bollier, C. M. Firestone. *The promise and peril of big data*, Aspen Institute, Communications and Society Program, [2010]

[11] P. Bramsen, A. Patel, M. Escobar-Molano and R. Alonso *Extracting Social Power Relationships from Natural Language*, ACL. [2011]

[12] J. Pestian, H. Nasrallah, P. Matykiewicz, A. Bennett and A. Leenaars *Suicide note classification using natural language processing: A content analysis*, Biomedical informatics insights, [2010]

[13] M. Taylor, *Ontology-based view of natural language meaning: the case of humor detection*, Journal of Ambient Intelligence and Humanized Computing 1.3, pp221-234, [2010]

[14] N. Hu, L. Liu, and J.J. Zhang, *Do online reviews affect product sales? The role of reviewer characteristics and temporal effects* Information Technology and Management 9.3 (2008): 201-214.

[15] N. Godbole, M. Srinivasaiah and S. Skiena, *Large-Scale Sentiment Analysis for News and Blogs*, ICWSM 7, [2007]

[16] *The Tweet Field Guide*, Twitter API, https://dev.twitter.com/docs/platform-objects/tweets

[17] A. Java, T.Finn, B. Tseng, X.Song, *Why we twitter: understanding microblogging usage and communities*, Proc. of the 9th WebKDD, ACM, [2007]

[18] *Celebrating #Twitter7*, Official Twitter Blog, March 21, https://blog.twitter.com/2013/celebrating-twitter7 [2013]

[19] *An Exhaustive Study of Twitter Users Across the World*, Bevolve, 10 Oct, http://www.beevolve.com/twitter-statistics/, [2012]

[20] J.Dean, S. Ghemawat. MapReduce: simplified data processing on large clusters, OSDI04, San Francisco, CA, [2004]

[21] A. Tumasjan, T.O. Sprenger, P.G Sandner, and I.M Welpe, *Predicting elections with twitter: What 140 characters reveal about political sentiment*, In Proc. of the fourth international AAAI conference on weblogs and social media, pp.178-185, [2010]

[22] J. Bollen, A. Pepe, and H. Mao, *Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena*, In Proc. of the Fifth International AAAI Conference on Weblogs and Social Media, pp. 450-453, [2011]

[23] C. Castillo, M. Mendoza, and B. Poblete, *Information credibility on twitter*, Proceedings of the 20th international conference on www, ACM, [2011]

[24] G. Grosseck and C. Holotescu, *Can we use Twitter for educational activities?*, 4th international scientific conference, eLearning and software for education, Bucharest, [2008]

[25] J. Li, and H. R. Rao, *Twitter as a rapid response news service: An exploration in the context of the 2008 China Earthquake*, The Electronic Journal of Information Systems in Developing Countries 42, [2010]

[26] L. Barbosa, J. Feng, *Robust sentiment detection on twitter from biased and noisy data*, Proceedings of the 23rd International Conference on Computational Linguistics: Posters, ACL, [2010]

[27] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, M. Demirbas, *Short text classification in twitter to improve information filtering*, Proc. of the 33rd international ACM SIGIR conference, ACM, [2010]

[28] T.M. Mitchell, *Machine learning*, WCB, [1997]

[29] A.M Turing, *Computing machinery and intelligence*, Mind, 59(236), 433-460, [1950]

[30] Arthur L. Samuel, *Some studies in machine learning using the game of checkers*, IBM. J. Res. Development 3, [1959]

[31] I.H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*, Morgan Kaufmann, [2005]

[32] H. Zhang, *The optimality of naive Bayes*, A1. no. 2, [2004]

[33] E. Frank, M. Hall and B. Pfahringer, *Locally weighted naive bayes*, Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence, MKP Inc., [2002]

[34] A. Genkin, D. D. Lewis and D. Madigan, *Large-scale Bayesian logistic regression for text categorization*, Technometrics 49.3, pp291-304, [2007]

[35] C.J.C Burges, *A tutorial on support vector machines for pattern recognition*, Data mining and knowledge discovery, 2.2, pp121-167, [1998]

[36] F. Sebastiani, *Machine learning in automated text categorization*, ACM computing surveys (CSUR), 34.1, pp1-47 [2002]

[37] D.H Wolpert, *The lack of a priori distinctions between learning algorithms*, Neural computation 8.7, pp1341-1390, [1996]

[38] R. Caruana and A. Niculescu-Mizil, *An empirical comparison of supervised learning algorithms*, Proceedings of the 23rd international conference on Machine learning, ACM, [2006]

[39] Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schtze. Introduction to information retrieval. Vol. 1. Cambridge: Cambridge University Press, 2008.

[40] G. Forman, *An extensive empirical study of feature selection metrics for text classification*, The Journal of machine learning research 3: 1289-1305, [2003]

[41] D.C. Muecke, *The compass of irony*, London: Methuen, [1969]

[42] E. Camp, *Sarcasm, Pretense, and The Semantics/Pragmatics Distinction*, Noûs 46.4: 587-634, [2012]

[43] M.L. Dress, R.J Kreuz, K.E. Link and G.M. Caucci, *Regional variation in the use of sarcasm*, Journal of Language and Social Psychology 27.1: 71-85, [2008]

[44] R.J. Kreuz and G.M. Caucci, Lexical influences on the perception of sarcasm, Proceedings of the Workshop on computational approaches to Figurative Language, ACL, [2007]

[45] H.P Grice, *Logic and conversation*, pp41-58, [1975]

[46] P. Brown and S.C. Levinson, *Politeness: some universals in language usage*, [1987]

[47] L.A. Juez, *Verbal irony and the Maxims of Grice's cooperative principle*, Revista alicantina de estudios ingleses 8, pp25-30, [1995]

[48] A. Utsumi, *Verbal irony as implicit display of ironic environment: Distinguishing ironic utterances from nonirony*, Journal of Pragmatics 32.12. pp1777-1806, [2000]

[49] G. Lessard and M. Levison, *Computational modeling of linguistic humour: Tom Swifties*, In ACH Joint Annual Conference, Oxford, pp. 175178, [1992]

[50] K. Binsted, *Machine humour: An implemented model of puns*, University of Edinburgh, [1996]

[51] K. Binsted and G. Ritchie, *Computational rules for generating punning riddles*, Humor 10.1: 25-76, [1997]

[52] J. Taylor and L. Mazlack, *Computationally recognizing wordplay in jokes*, Proceedings of CogSci, [2004]

[53] R. Mihalcea, and S. Pulman, *Characterizing humour: An exploration of features in humorous texts*, Computational Linguistics and Intelligent Text Processing. Springer Berlin Heidelberg, pp337-347, [2007]

[54] G. Ritchie, *The linguistic analysis of jokes*, Vol. 2. Routledge, [2002]

[55] T. Wilson, J. Wiebe and P. Hoffmann, *Recognizing contextual polarity in phrase-level sentiment analysis*, Proc. of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, ACL, [2005]

[56] R.E. Schapire and Y. Singer, *BoosTexter: A boosting-based system for text categorization*, Machine learning 39.2-3, pp135-168, [2000]

[57] H. Saif, H. Yulan and H. Alani, *Semantic sentiment analysis of twitter*, The Semantic WebISWC, Springer Berlin Heidelberg, pp508-524, [2012]

[58] J.B. Walther and K.P. DAddario, *The impacts of emoticons on message interpretation in computer-mediated communication*, Social Science Computer Review 19.3, pp324-347, [2001]

[59] D. Derks, A.E.R Bos and J. Von Grumbkow, *Emoticons and online message interpretation*, Social Science Computer Review, 26.3, pp379-388, [2008]

[60] J. Tepperman, D. Traum and S. Narayanan, *'Yeah right': Sarcasm recognition for spoken dialogue systems*, InterSpeech ICSLP, [2006]

[61] P. Carvalho, L. Sarmento, M.J. Silva and E. de Oliveira, *Clues for detecting irony in user-generated contents: oh...!! it's so easy;-)*, In Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion, pp. 53-56, ACM, [2009]

[62] P. Carvalho, L. Sarmento, M.J. Silva and E. de Oliveira, *Automatic creation of a reference corpus for political opinion mining in user-generated content*, Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion, ACM, [2009]

[63] D. Davidov, O. Tsur and A. Rappoport, *Semi-supervised recognition of sarcastic sentences in twitter and amazon*, Proceedings of the Fourteenth Conference on Computational Natural Language Learning, ACL, [2010]

[64] *Communications (Retention of Data Act)*, Office of the Attorney General, http://www.irishstatutebook.ie/2011/en/act/pub/0003/print.html [2011]

[65] *Twitter Developer Rules of the Road*, https://dev.twitter.com/terms/api-terms

[66] *A Serf*, Calbuzz Cartoons, http://www.calbuzz.com/2011/?w=3

[67] *Yeoman Cartoon*, Writers' Village, http://www.writers-village.org/john-yeoman.php

[68] *The E-merchant Of Venice*, Goddard, http://www.cartoonstock.com/directory/m/merchants

[69] *Jester*, Phillip Martin, http://worldhistory.phillipmartin.info/worldhistory_jester.htm

[70] *Proud Knight*, Stock Toons, http://stocktoons.com/design/stock-cartoon-of-a-proud-knight-walking-by-hit-toon-81

[71] *Man With Top Hat*, ShutterStock, http://www.shutterstock.com/pic-139707175/stock-vector-man-with-top-hat-and-large-sunglasses.html?src=p-113824567-8

[72] *A Prince*, http://kootation.com/how-to-draw-prince-frnovember.html

[73] *Emperor*, http://web.viu.ca/webquests3/2001/emperors/Emperorvirtues.htm

[74] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I.H. Witten *The WEKA Data Mining Software: An Update*; SIGKDD Explorations, Volume 11, Issue 1, [2009]

[75] S. Bird, E. Loper and E. Klein, *Natural Language Processing with Python*, OReilly Media Inc, [2009]

[76] H. Cunningham et al, *Text Processing with GATE (Version 6)*, Univ. of Sheffield Dept. of Computer Science, [2011]

[77] *Apache Mahout*, http://mahout.apache.org

[78] Pedregosa et al, *Scikit-learn: Machine Learning in Python*, JMLR 12, pp. 2825-2830, [2011]

[79] F. Prez and B.E. Granger, *IPython: A System for Interactive Scientific Computing*, Computing in Science and Engineering, vol. 9, no. 3, pp. 21-29, [2007]

[80] *Authorities use analytics tool that recognises sarcasm*, http://www.bbc.co.uk/news/technology-23160583