7-31-2019

Predicting the Emotional Intensity of Tweets

Intisar M. Alhamdan
ima7614@rit.edu

ROCHESTER INSTITUTE OF TECHNOLOGY

MASTER'S THESIS

Predicting the Emotional Intensity of Tweets

Author:
Intisar M. Alhamdan

Supervisor:
Prof. Ernest Fokoué

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Applied Statistics

in the

Schoool of Mathematical Sciences
College of Science

July 31, 2019

Declaration of Authorship

I, Intisar M. Alhamdan, declare that this thesis titled, "Predicting the Emotional Intensity of Tweets" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

_____

Date:

_____

Committee Approval Form

Prof. Ernest Fokoué **Date:**
Thesis Advisor, Full Professor, School of Mathematical Sciences

---

Prof. Robert Parody **Date:**
Committee Member, Assistant Professor, School of Mathematical Science

---

Prof. Linlin Chen **Date:**
Committee Member, Assistant Professor, School of Mathematical Science

"Life can be so difficult at times, but fighting through the pain is so worth it. It's better to feel every kind of emotion than not feel at all."

Demi Lovato

"A hidden connection is stronger than obvious one."

Heraclitus

ROCHESTER INSTITUTE OF TECHNOLOGY

Abstract

Dr. Ernest Fokoué

College of Science

Master of Science in Applied Statistics

Predicting the Emotional Intensity of Tweets

by Intisar M. Alhamdan

Automated interpretation of human emotion has become increasingly important as human computer interactions become ubiquitous. Affective computing is a field of computer science concerned with recognizing, analyzing and interpreting human emotions in a range of media, including audio, video, and text. Social media, in particular, are rich in expressions of peoples' moods, opinions, and sentiments. This thesis focuses on predicting the emotional intensity expressed on the social network Twitter. In this study, we use lexical features, sentiment and emotion lexicons to extract features from tweets, messages of 280 characters or less shared on Twitter. We also use a form of transfer learning – word and sentence embeddings extracted from neural networks trained on large corpora. The estimation of emotional intensity is a regression task and we use, linear and tree-based models for this task. We compare the results of these individual models as well as making a final ensemble model that predicts the emotional intensity of tweets by combining the output of the individual models. We also use lexical features and word embeddings to train a recently introduced model designed to handle data with sparse or rare features. This model combines LASSO regularization with grouped features. Finally, an error analysis is conducted and areas that need to be improved are emphasized.

Keywords: affective computing, emotional intensity…

Acknowledgements

The moment I commenced my Master's program, I realized that I would need all the assistance, both in my academic and personal journey, from those who have already traveled that path to become a successful graduate. My research thesis would therefore not be possible without some key people, both in my education life as well as my personal life. Starting with my instructors, I am greatly indebted to my advisor of the thesis Prof Ernest Fokoué for being my career guide, encouraging, supporting and giving me strength in conducting the project, even as I was about to give up. It is a blessing to have such an advisor and mentor who constantly supported and showed guidance. Furthermore, all of my professors in the program played a key role in my success, by being patient, kind and believing that I can accomplish my target by learning first-hand benefits of efficiency and immediate feedback. I would like to thank Prof. Robert Parody and Dr. Linlin Chen for being my committee and for their support.

I would also like to express my gratitude to my peers, who we supported each other and exchanging ideas, making the entire process to be enjoyable. Special thanks to Fadyah, who was always there for me, offering encouragements in my quest every time we talked, believing that I would manage to succeed. I value our relations a lot and I am committed to maintain them to the end of my life. Badryah was another significant individual who was more than a friend. Our fates were meant to cross so as we can be there for each other, both in our academic and personal life. I truly believe that your special friendship is a gift not everyone has deserved.

Additionally, I want to thank my family in law for their always love and support.

I am also extending my gratitude to my family, who raised me to become the person I am today. Through their upbringing, love, and care, I have managed to become a persistent person. Beginning with my father. Although he passed away, his love and support meant so much to me. You might not be physically here with me, but I believe It would make you happy and proud seeing me graduating. I would also like to thank my mother who was always there for me, being kind and helpful at all times through my study, as well as being the source of my inspiration in the pursuit of my dreams. She is and will always be the cornerstone of my life. Whenever I needed her, my sister was always there for me. I can't imagine my life without you.

Finally, I want to express my appreciation and pure love to my small family. I would like to thank and appreciate my beloved husband for being very supportive in pursuing my master's degree. Thank you for lending me a helping hand every time I needed assistance. I want to also thank my three children. Fissal, I am astonished of your optimism, confidence in my power and your belief in my success. I am motivated and oriented at achievements in consequence of your mere presence in my life. Talia, you are one of the gifts I have and I value your warm hugs that kept me going and fulfilling my dream. Mansour, you have come to my life so unexpectedly, but you managed to bring so much colour and happiness to our family. Getting to know that I am going to give life to you on the first day of the year was a sign that life is full of wonders and new discoveries, and I am grateful to you for this.

I would like to thank the 8th Annual Conference of the (UP-STAT) New York Chapters of The American Statistical Association for giving me the best student award with the honorable mention on April 27th, 2019. ...

Contents

List of Figures

List of Tables

Dedicated to my Mother.
Everything I am is because of you. You are my strength, you are my inspiration....

Chapter 1

Introduction

1.1 Introduction

Technology is often thought of as a tool or instrument. Humans use technological tools in order to achieve certain ends and the emotional state of the user is not part of this interaction. What if this changed and technology, software and hardware, were sensitive to the more human aspects of its users, their attitudes and emotions?

This study seeks to advance knowledge in one area of a larger field known as affective computing.Specifically, we design features and models that accurately gauge the emotional intensity of short texts. Social media is rich in expressions of individual moods, sentiments, and opinions. In this study we use text data, Tweets, from the social media website Twitter.

The study focuses on the detection of the emotional intensity of tweets, short texts less than 280 characters in length. In particular, we examine the performance of features used in building regression models to accurately predict emotional intensity of tweets for four different emotions: anger, fear, joy, and sadness. Three sets of features are used as the input into a suite of regression models and one

ensemble model. We analyze the performance and characteristics of the resulting models and conduct an error analysis.

The use of transfer learning in natural language processing (NLP) models is a growing area of research. Transfer learning is the use of models or features in a machine learning task different from the task they were trained on. We examine two sets of word embeddings that were trained on general language models. We compare these word embedding features to a set of baseline features consisting of sentiment and emotion lexicons along with some simple text features.

Our main conclusion is that transfer learning features can be used in detecting and estimating the emotional content of text data. Models that use word embedding features trained for a general language task are found to be competitive with the baseline features designed for gauging emotional intensity. However, not all word embeddings perform equally well. We find that the corpus on which the language model is trained is the most important factor in creating word embeddings that are useful for detecting emotions. Corpora with high emotional content and with language features similar to the features used in the prediction task appear more likely to produce word embeddings useful in emotion detection models.

Although our focus is on analyzing the performance of features used in a regression task, we also look at model performance on our set of features. In our study of feature performance we use three models per feature, a lasso model, random forests, and gradient boosted trees. For each feature and emotion, we also create an ensemble model by combining prediction results of the three base models.

We find that the random forest model performs well across all feature sets. For one feature set, the ensemble models perform uniformly better than the individual models. Uncorrelated models combine to form ensemble models better than the individual models from which they are created. We find that model correlation is dependent on the feature set used. This suggest that better ensemble performance could be gained by performing model selection per feature set.

We also analyze a recently developed model called the rare-model (Yan and Bien, 2018a). The rare model is based on the lasso model but is designed to group rare features. The model takes both a feature set, in this case, a document-term-matrix (DTM) and side information that gives the model information about how features are related. We used word embeddings as the side information that provides a metric for grouping terms in the DTM.

Our experiments with the rare-model were not conclusive. We tested the rare-model and three DTMs of varying size. Each model resulted in an intercept-only model. This model gives all tweets the same level of emotional intensity regardless of the contents of the tweet.

Our study contributes to the broader field known as affective computing. Affective computing is a field of computer science concerned with the automated interpretation of human emotion in a variety human expressions including facial expressions, gestures, and spoken and written language. The need for automated and accurate detection of human emotions that goes beyond recognition of positive or negative sentiment is becoming increasingly as people interact with systems largely governed by automated algorithms.

Human computer interaction is also becoming more widespread and personalized with the popularity of personal assistants that respond to voice commands, like those of Amazon, Apple, and Google. Many companies also offer some level of automated customer service through the use of chatbots. These tools may need to better recognize the emotional content of user interactions in order to more appropriate respond to user behavior. In other fields, such as education or marketing, the recognition of an audience's attention and emotional state can be used to improve lectures or commercial media.

The detection of emotion in text is particularly relevant to the automated monitoring of social media on websites like Facebook and Twitter. Both companies have recently come under criticism for handling of abusive speech. Accurate detection of the intensity of emotions like anger or sadness could enable better detection of abuse or bullying.

Much of the background for the theory of detecting and classifying emotional content is drawn from the field of psychology. A number of different theories of the emotions have been developed. Two of the most popular theories are the continuous and discrete theory of the emotions. In the continuous theory, the emotions have several poles and a particular emotional expression can fall anywhere within the emotional space between those poles (Mehrabian, 1996). A competing theory is that there are discrete 'basic' emotions. Other, more complex emotions are made of combinations of these discrete emotions (P. Ekman and Friesen, 1971).

Our study, and the data we use to create the models, are based

on the latter theory. Our data come from the **semEval 2018** emotional intensity regression task. The data for this task are grouped into four emotions: anger, fear, joy, and sadness. These emotions correspond to four of the 'basic' emotions as defined by Ekman (P. Ekman and Friesen, 1971). For our study, we use data that are labeled based on this discrete theory of the emotions. Whether this theory of the emotions best captures emotional nuance in texts is beyond the scope of our analysis.

Chapter 2

Related Work

Summary

In this chapter, we review the literature relevant to our study of emotion detection in texts. In order to detect emotions, we first need a well-defined idea of what it is we are detecting. We begin by looking at the psychological theory that has influenced the field of affective computing. Next, we broadly discuss affective computing, a field of computer science that attempts to detect, analyze, and respond to human emotion in a range of media. We also review some of the techniques that have been used for analyzing text data and look at specific text features and machine learning models that will be used in our study.

2.1 Theory of emotions

There is a long history of the study of the emotions. In modern times, Darwin posited that the purpose of emotions is to improve a species' reproductive fitness, allowing it to have a better chance at surviving and reproducing (Darwin, 1872). A primary example of

this theory is fear. The fight or flight response increases an organism's likelihood of surviving and reproducing if it either flees when encountering a predator, or if it fights and is able to kill the predator or adversary.

Whether the fight or flight reactions of animals are of the same nature as human emotions, is a subject of debate (P. Ekman, 1999).Whether human emotion is different from emotion-like responses in animals depends on the interpretation of what an emotional response is. Some theories posit that emotions are a physiological response to a stimulus. Cognitive theories, on the other hand, posit that thoughts which result from an experience of the stimulus leads to emotion. In the latter theory, it's the higher level cognitive processing of events which causes emotion rather than a direct external stimulus. This is supported by research that shows that emotions are the result of complex mental processes which may not exist in other animals. The substantially greater complexity associated with human brains also suggests that humans may feel and experience emotions in a qualitatively different way than other animals (Barrett, 2017).

An example of the physiological theory of the emotions is the James-Lange theory (Lange and James, 1922). In the James-Lange theory, emotions depend on an interpretation of our body's physiological reactions an external stimulus. For example, coming into contact with a wild animal would lead to a rapid heartbeat, sweating, shaking, and similar physical responses. Individuals experiencing these physiological affects would then interpret their reactions as an experience of fear. In this sense, emotion results from both the physiological response to a stimulus and the resulting interpretation of the physiological response by higher cognitive processes.

Criticisms of this theory relate to the role of higher level cognition in emotional responses. The James-Lange theory suggests that physiological indicators of an emotional response may occur because of emotion rather than prior to an emotional response. For example, a feeling of fear can come after thinking about some potential danger rather than being a reaction to a present danger. Another criticism is that physiological responses do not have a one-to-one correspondence with emotions, and that the timing physiological responses do not always match the subjective feeling of an emotion in time (D. H. Hockenbury and S. E. Hockenbury, 2010).

The Cannon-Bard theory of emotion was originally proposed in response to the James-Lange theory (Dror, 2014). This theory responded to some of the criticisms of the James-Lange theory. The Cannon-Bard theory suggested instead that physiological responses and emotions are experienced simultaneously. Continuing the example from above, an individual encountering a dangerous wild animal would experience fear while simultaneously experiencing these physiological responses of a rapid heartbeat, sweating, etc.

While solving some of the problems associated with the James-Lange theory, it was not until the 1960s when theories started to incorporate cognition as an important factor in both physiological and emotional responses. These later theories which emphasized the cognitive role of the emotions were part of the "cognitive revolution" in psychology.

An example of a cognitive theory of the emotions is Schachter-Singer's two-factor theory of emotion (Schachter and Singer, 1962).

The Schachter-Singer theory posits that physiological arousal happens first. The individual, though, experiences an emotion depending on their cognitive understanding of the source. This theory builds upon both the James-Lange and Cannon-Bard theories of the emotion but adds a cognitive element not present in the earlier theories. For example, this theory suggests different emotions can be produced on the basis of similar physiological responses. That difference depends on the cognitive processing of information related to the physiological response.

Lazarus' cognitive appraisal theory suggests that both the emotional and physiological experience of individuals will depend on how they evaluate the events they witness and experience. In this view, different emotions are a product of differing appraisals an individual makes of a situation (Lazarus and Lazarus, 1991). This theory is rooted in research conducted in 1940s which posited that different emotions result from different excitatory phenomena (Arnold, 1945). This was further developed by Lazarus who suggested there are two stages of appraisal, primary and secondary appraisal. Primary appraisal relates to the intensity of the emotions experienced by individuals and relates to how individuals assess their goals and the relevancy of the circumstances in attaining their goals (Lazarus and Lazarus, 1991). Secondary appraisal pertains to someone's evaluation of whether the resources available allow them to adequately cope with the situation. This secondary appraisal includes several facets: who should be accountable, an individual's coping potential, and what is expected of future events.

While these earlier theories related to physiological responses and cognitive processes, more recent theories inform our study. In

particular, the questions of whether emotions are universally shared, between people and cultures, and how emotions can be categorized and quantified, directly relate to if and how computational emotion detection can be performed.

Ekman and Friesen's research suggested that at least some emotions were universal (P. Ekman and Friesen, 1971). They conducted a study which found no significant difference in accuracy comparing preliterate respondents from New Guinea with those from Western cultures. In their experiment, participants had to select the correct emotional response out of three after hearing a story written to produce a specific emotion on the basis of Western cultures. On the other hand, cultural differences also exist in terms of how emotions are expressed by individuals. For example, facial expressions of an emotion may vary by individual and the appropriate expression of emotion varies by culture. One study, for example, found that the expression of emotion was encouraged in American culture, while alternatively being suppressed in Japanese culture (Miyamoto, Uchida, and Ellsworth, 2010).

The uniformity of emotional expression, at least among English speakers, is important in determining whether behavior can be seen as an expression of a particular emotion. On the other hand, how emotion can be modeled in a way that makes it suitable for quantification informs strategies for computational detection of emotions.

A basic question of psychological theory is whether emotions should be modeled as discrete or continuous. Some theorists positing there are discrete emotions also believe there is a limited set of emotions that exist in all individuals and which can be recognized across cultures. These researchers have posited that these discrete

emotions can be determined on the basis of someone's facial expression and biological processes (P. Ekman and Friesen, 1971). Ekman and Friesen's early theory posited discrete emotions such as happiness, sadness, anger, fear, disgust and interest.(P. Ekman and Friesen, 1971) Over time, variations of what the basic emotions are have been suggested.

In a survey, Ekman found that, among active researchers in the field, 88% felt that there was compelling evidence for universal features in any aspect of emotion, with 55% indicating that they felt that both discrete and dimensional models were relevant. Opinions relating to what emotions would be considered basic varied, with the most common consisting of anger (91%), fear (90%), disgust (86%), sadness (80%), and happiness (76%) (P. Ekman, 2016). Although the original theory suggests there are basic emotions, later versions allow for there to be more complex emotions which are built of mixtures of the basic emotions.

An alternative to this discrete theory models emotions as occurring in a continuous emotion space. This school of thought suggests that all emotional states come out of the same few neurophysiological signals which determine the dimensions of the emotion space (Posner, Russell, and Peterson, 2005). These models are founded on the idea that all emotions result from the same interconnected neurophysiological network. Generally, these emotion space or dimensional models use two or three dimensions. The earliest such model, proposed in the late 19th century, modeled emotions in three dimensions: pleasure, arousal, and strain (James, 1894). Scholsberg suggested replacing arousing and strain with the new dimensions of attention-rejection and level of activation (Schlosberg, 1954).

Modern models generally incorporate valence and arousal. The PAD model uses three dimensions in order to represent pleasure, arousal, and dominance, with the pleasure-displeasure dimension measuring how pleasant an emotion is, arousal-non-arousal dimension measuring intensity, and the dominance-submissiveness scale measuring how controlling an emotion is (Mehrabian, 1996).

Plutchik's multi-factor theory consists of a hybrid theory of emotions that sought to connect discrete and dimensional models, with his "wheel" of emotions incorporating eight basic emotions, acting like dimensions, which include Ekman's six emotions along with the emotions of trust and anticipation. The radius is then used to indicate intensity, drawing parallels with the dimensional approach with dyadic emotions also being present within this model, which consist of emotions that result from combinations of two or more emotions (Plutchik, 1960).

Both continuous and discrete theories allow for a quantification of the emotions. In the PAD and VAD models, and in related continuous dimensional models, any particular emotion can be represented as a point in three dimensional space, or as a three dimensional vector. Higher numbers in a particular dimension would indicate that emotional dimension places a greater role in a particular response.

The discrete emotional model, on the other hand, would label an emotional response as belonging to one (or more) emotional categories. The response could also be labeled with an intensity, as are the text in our study. If an emotional reaction is allowed to belong to more than one basic emotion, then, there is not that clear a distinction between continuous and discrete models. In both cases,

emotions can be quantified as having a continuous strength along different emotional categories, although the categories vary by theory. For example, in one theory, the dimensions are valence, arousal, and domination, and in another theory the dimensions might be fear, anger, sadness and joy.

2.2 Affective Computing

Many fields of artificial intelligence have made increased the ability of computers to perform tasks that were once thought to be uniquely human. Affective computing can be thought of as complementing these attempts to replicate human intelligence by giving computers emotional or social intelligence by attempting to imitate the ability of humans to recognize and react appropriately human emotions.

The goal of computers being sensitive to human emotions may sound far off but it has practical and current applications in a number of fields. In marketing, affective computing could be used to recognize the user's emotional response to various media. For example, affective computing techniques have been used to gauge the emotional reactions of a player to a specific game scenario based on their body movements.(Bianchi-Berthouze and Kleinsmith, 2015). Similarly, facial expressions could be used to measure engagement or frustration levels of a student to educational material, and to capture the reaction of a viewer of an advertisement(Jeffrey F. Cohn and Torre, 2015). Chatbots are widely used for customer service, and being more aware of a customers attitude could help improve

chatbot's interaction with a customer (Portela and Granell-Canut, 2017).

Below, we review work on affective computing based on the type of expression analyzed, facial expression, bodily expressions or gestures, speech, and written texts. However, there are some common issues that arise for each of these types of expression. These common issues are:

1. Choice of media used to analyze the expression;
2. Theory of emotions used to code expressions;
3. Coding procedure for creating labeled datasets;
4. Feature extraction from raw data;
5. Models used to analyze data.

We use these issues to organize our discussion of the techniques used to analyze the emotional content of various types of human expression.

2.2.1 Facial Expression

For several decades, the face has been an object of interest to computer scientists as a possible biometric, with computer vision and graphics first being used in the 1990s in order to both analyze and synthesize facial expression (Jeffrey F. Cohn and Torre, 2015). While work began with the simple recognition of the expression associated with a posed facial action, currently researchers are attempting to accurately detect expressions in more natural settings and with

naturalistic challenges such as partial occlusion, pose variation, and so forth.

The coding of facial expression is based, for the most part, on one of three models, the message-based models, sign-based models, or continuous coding models (Pantic, 2009). The message-based model is based on describing a facial expression holistically, as a single expression or emotion. The sign-based model, on the other hand, is used to code the surface expression without necessarily applying a single expression label. In general, the discrete theory of the emotions is used to label expressions in the message based model (Jeffrey F. Cohn and Torre, 2015).

The sign-based approach most commonly uses the facial action coding system (FACS) as developed by Ekman and others (R. Ekman, 1997). This method breaks up an expression into separate "signs". The way the facial expressions are categorized is largely based on facial muscle groups (R. Ekman, 1997). The sign-based coding method relies on expert coders while the message based system labelling is more easily crowd-sourced (Jeffrey F. Cohn and Torre, 2015). For either coding system, the validity of the measurement or labeling of the expressions is usually gauged by analyzing between coder agreement and consistency (Jeffrey F. Cohn and Torre, 2015).

A third method is similar to the sign-based method but uses facial landmarks to measure facial expressions (Jeffrey F Cohn et al., 2009). This method could be considered a continuous model since the movements of facial landmarks are tracked continuously and not broken into discrete positions, as in the FACS coding model.

One example of this continuous coding method is the active appearance modeling (AAM) method (Jeffrey F Cohn et al., 2009).

Most facial expression methods use video to capture expressions, either in lab-based or naturalistic settings. Analysis of facial expressions share methods in common with other computer vision applications. For example, one of the first steps in using video is to detect and then track faces (Jeffrey F. Cohn and Torre, 2015). A variety of techniques are used to track facial positions and capture the dynamics of actions in a way that accounts movement in three dimensions and individual differences of faces (Jeffrey F. Cohn and Torre, 2015).

The sign-based FACS coding system provides a relatively small and discrete set of features that can be used in machine learning applications. The AAM methods that track continuous movement produce large datasets which generally need dimension reduction methods applied in order to prepare the data for machine learning models. These features can be based on facial landmarks or by discretizing the data based on meaningful thresholds (Jeffrey F. Cohn and Torre, 2015). Other methods use typical machine learning dimensions reduction techniques, like principal component analysis of the raw data (Jeffrey F. Cohn and Torre, 2015).

The analysis of facial expression analysis has been applied to a range of problems including detecting or estimating physical pain, detecting depression, and detecting interpersonal coordination (Pantic, 2009) (Jeffrey F Cohn et al., 2009) (Jeffrey F. Cohn and Torre, 2015). The application of these methods to detecting physical pain, for example, is meant to overcome limitations of patient self-report.

There are similar benefits with respect to the diagnosis and assessment of depression and psychological pain. Other applications include being able to discriminate between subtle differences when comparing related expressions, marketing (such as responses to commercials being seen), drowsy-driver protection, and instructional technology (Jeffrey F. Cohn and Torre, 2015).

2.2.2 Body Expressions

While the analysis of facial expression has a long history, going back to Darwin and earlier, less research exists on the analysis of bodily expression. Despite evidence suggesting that some affective expressions may be more easily communicated by the body, until recently, the field has attracted fewer researchers. Recent advancements in technology used for capturing body movements has increased the interest in the automated analysis of bodily expressions (Bianchi-Berthouze and Kleinsmith, 2015).

Many applications and potential applications relevant to body expressions have been identified, including applications in the fields of security, law enforcement, health care, education, and games and entertainment (Bianchi-Berthouze and Kleinsmith, 2015). The detection of engagement as well as the emotional expressions of individuals playing games could be used for the purposes of game evaluation or the adaptation of gameplay to users emotional state (Bianchi-Berthouze and Kleinsmith, 2015). In clinical applications, detection of the emotional state through bodily expressions could be used in determining if people are suffering from depression or from physical pain (Luo et al., 2018). Doctors or nurses could use

this information about the patients emotional state could inform the treatment of the patinetn and help personalize their support. In an educational setting, systems for recognizing interest and frustration could give designers of lessons, lectures, or other educational material feedback for improving that material (Bianchi-Berthouze and Kleinsmith, 2015).

Data on body expressions has primarily been captured using vision or video-based, optical, electromechanical and electromagnetic motion capture systems. There are trade-offs between the type of technology used for gathering data and systems vary by cost, portability, and accuracy of the data captured (Bianchi-Berthouze and Kleinsmith, 2015). Video-based systems are commonly used because they are inexpensive and allow for coding of emotions based both on gesture and facial expression (Luo et al., 2018). Microsoft's Kinect technology, on the other hand, combines video with 3-D sensors and has provided researchers with an inexpensive way to capture motion data.(Zhang, 2012).

The model of the emotions used for classifying gesture and body expressions are discrete and continuous. According to Karg *et al.* the discrete method has been more popular in studies of gesture (Karg et al., 2013). The continuous models, like the pleasure-arousal-dominance (PAD) model, have also been used. Although continuous models are thought to be closer to the physiological "ground truth", the continuous models have had issues in practice. For example, manual coding of gestures using continuous models has had lower agreement than discrete models (Bianchi-Berthouze and Kleinsmith, 2015)(Karg et al., 2013).

There does not appear to be a single notation method that has

become accepted among body expression researchers, although a variety of methods have been proposed. Systems have been developed that are based loosely on the FACs annotation system for facial recognition. These systems, like FACS, break movements into a limited set of discrete units (Karg et al., 2013). Another method map complex motion to simpler linguistic labels, usually using some set of emotional labels based on Ekman's discrete system (Karg et al., 2013). The structural approach, on the other hand, attempts to capture the complete movements of the limbs and joints (Karg et al., 2013). Similar to continuous models of facial recognition, these models capture data on the geometric relations of landmark points over time. The structural method can be used both with motion capture data which records these landmark points or with video that uses computer vision techniques to recognize and track landmarks over time (Luo et al., 2018).

As with studies of other affective computing methods, the two most common methods of labeling data are by an expert, trained coders or through crowd-sourcing. For crowd-sourced labeling methods are used to identify quality labels, either through taking the most frequent label or by using other measures of between-coder agreement such as the Best-Worst Scaling (BWS) method (Bianchi-Berthouze and Kleinsmith, 2015).

2.2.3 Speech

Speech is another human expression that is rich in emotional content. A speakers dialects and word choice convey information about

the speakers' background, pitch and acoustic qualities convey emotional information, emphasis, and other traits of the speaker such as age, gender and health (C.-C. Lee et al., 2015). As with facial and bodily expressions, there is a great deal of variability among speakers which poses a challenge to isolating patterns that have emotional significance. The physical mechanisms that produce speech are also relatively complex and small physical differences among individuals can cause significant variations in the acoustic properties of the speech they produce (C.-C. Lee et al., 2015).

The basic theory underlying the detection of emotion using acoustic patterns of speech is that emotional processes correspond to physical changes in the muscular systems responsible for speech production (Eyben et al., 2015). Analysis of speech production data suggests that information relating to emotion is encoded acoustic properties modulated by the vocal tract and vocal source activities (C.-C. Lee et al., 2015). More detailed research has focused on the identifying mechanisms that produce acoustic traits associated with specific emotions. Analysis based on previous research in phonetics focuses on acoustic patterns associated phonetic articulation (Eyben et al., 2015).

On the other hand, machine learning researchers have often approached the acoustic signals as any other audio signal, and have used wide range of audio features to detect and analyze patterns in speech. The use of features derived from the raw audio signal can produce high-dimensional feature vectors with hundreds of features or more.

However, questions remain, such as why a large number of acoustic low-level descriptors work well, or what mechanisms are appropriately modeled using this technique. This method is also computationally expensive, and may not efficiently scale to an emotional recognizer that can be used in real-time efficiently and reliably (C.-C. Lee et al., 2015). The proliferation and variety of features used by researches limit the ability to interpret results across studies (Eyben et al., 2015). Some researchers have turned to systems that are based on more grounded understanding of human speech while not suffering from reduced accuracy of machine learning models (C.-C. Lee et al., 2015).

One example of such a limited system of parameters is the Geneva Minimalistic Acoustic Parameter Set (GeMAPS). This set of features includes parameters related to frequency, amplitude, and the spectral characteristics of the audio signal (Eyben et al., 2015). The motivation for having a limited set of features is similar to that for the FACs system for facial recognition – it provides a baseline set of features that can be compared across studies as well as allowing for potentially allowing for better inferences about the relationship between emotion, vocal mechanisms, and audio characteristics.

As with other areas of affective computing, the discrete and continuous models of the emotions are the most commonly used systems for labeling or classifying emotion in speech. There has been more success in applying continuous or dimensional models of the emotions, like PAD, to vocal expressions, where continuous properties like pitch, duration, and energy have been related to dimensions of the PAD model (Yildirim et al., 2004). However, many studies still rely on the discrete model of emotions to label categorize

specific speech productions as belonging to small set of emotions, similar to Ekman's: anger, sadness, happiness, boredom, disgust, fear, joy, and neutrality, for example (Yildirim et al., 2004)(Eyben et al., 2015).

A common technique for labelling speech production is to task actors with speaking the same set of sentences but with different emotional intonations. The Geneva Multimodal expression corpus is one such collection in which actors produced the same vowel sounds with variations on the emotional content (Bänziger, Mortillaro, and Scherer, 2012). Other speech corpora are of speech recorded in more naturalistic contexts that are then labeled annotators using either discrete emotional categories or as low-high activation and positive-negative valence. Eyeben *et al.* include a table of that lists several corpora and the labeling system used (Eyben et al., 2015).

While there is a range of techniques for collecting gesture and facial expressions, the recording of speech audio is fairly standardized. Some researchers also record video or use audio sources that include video, such as the Vera-Am-Mittag corpus that is a video recording of a daily talk show (Eyben et al., 2015).

2.3 Sentiment Analysis

The area of sentiment analysis, or opinion mining, has a long history, and a great expansion of the field has been seen alongside the growth of social media (B. Liu, 2012). Sentiment analysis covers a broad range of topics but generally refers to the automatic estimation of the positive or negative valence of text. The more general sense of sentiment analysis is the determination of one's attitude

toward some target, topic, or person, sometimes referred to as an entity (Saif M Mohammad, 2016). It has been one of the most active areas in natural language processing and has spread to related fields (B. Liu, 2012). Even though the detection of positive and negative sentiment appears simpler than the detection of the full range of human emotions, even this reduced task has proven challenging (Pang, L. Lee, et al., 2008).

The popularity of sentiment analysis over the past two decades can be explained by several trends. The spread of standardized and easy to use tools for basic natural language processing tasks has encouraged researchers to use text data. There has also been a number of practical applications, such as automated detection of consumer attitudes towards consumer products. The availability of high volume, near real-time text streams that have also been used to monitor evolving emergencies or to detect changes in investor attitudes towards stocks(B. Liu, 2012) (Saif M Mohammad, 2016) (Pang, L. Lee, et al., 2008). Sentiment analysis has also been used in a broad range of fields from public health, political science, to education, psychology, and literary analysis(Saif M Mohammad, 2016).

Sentiment analysis can be carried out at different levels of a text: at the document level, the sentence level, and the entity level (Alawami, 2016) (B. Liu, 2012). At the document level, the aim consists of determining whether a document contains a positive or negative sentiment, and assumes that each document focuses on a single entity or topic. At the sentence level, the task relates to determining whether a positive, negative, or neutral opinion is expressed in each sentence. At the entity level, the opinion itself is examined. The granularity associated with this level of examination allows for a

much greater number of qualitative and quantitative analyses to be conducted (Alawami, 2016) (B. Liu, 2012). For example, changes in sentence level sentiment can be analyzed over the extent of a text in order to see how positive and negative valence changes over the course of the text.

Classifications at entity level are most difficult, even more than those conducted at the document or sentence level (B. Liu, 2012). NLP methods are usually frequency-based and use bag-of-words methods, or use entity recognition methods to determine the entity and attitude towards that entity. Other statistical approaches include topic modeling or supervised learning with labeled data sets. (Alawami, 2016).

There are a number of challenges with correctly identifying the valence of a text. Opinions or attitudes may be either explicit or implicit. In the latter case, the valence of the text can be much harder to classify (B. Liu, 2012). Lexicon techniques, that match words to positive or negative valence, can be much more successful when there are explicit opinions expressed. For example, valence indicating words like good, amazing, bad, and terrible can make valence easier to detect (B. Liu, 2012) (B. Liu, 2007). However, sentiment lexicons rely on a bag-of-words model of language. The bag-of-words (BOW) model assumes a text can be split into words "atoms" and the meaning of the text can be determined statistically through analysis of these atoms.

For texts with implicit attitudes or emotions expressed indirectly, BOW methods and sentiment lexicons can be less effective. For example, the valence of a sarcastic phrase might be hard to detect. Consider the sentence: "We lost again, that's great!". The word "lost"

might be scored slightly negative while "great" is scored as having a strong positive valence, giving the phrase an overall additive positive valence. However, the phrase is meant negatively. Sarcasm is just one example of language subtleties that can be hard to detect with lexicon techniques (B. Liu, 2012).

Many methods exist for compiling word lists, or more generally, sentiment or opinion lexicon for use in sentiment analysis. Several examples of early lexicons were compiled by psychologists manually. Some more recent lexicons are also based on manually scoring of words for positive or negative sentiment (B. Liu, 2012) (B. Liu, 2015). The manual approach is labor intensive although crowdsourcing manual labelling has become more viable with tools like Amazon's Mechanical Turk. Lexicons that have performed well in the past, for example, in the SemEval-2013 and 2014 Sentiment Analysis in Twitter competition, have relied upon very large sentiment lexicons of 10-15,000 words that were labelled by crowdsourcing on Mechanical Turk (Nakov, Ritter, et al., 2016).

The dictionary method uses word relations to compile a larger list. The dictionary approach starts with a few sentiment words and then, using lists of synonyms and antonyms, builds an entire list. For example, a researcher can start with a negative word, like anger, and look up related synonyms. Words found as synonyms, like vexatioun, irritability, or indignation, would also be listed as words with negative valence.

More sophisticated approaches use distance measures or other bootstrapping methodologies to go from smaller seed lists to larger compilations of words. This method can also be automated. For example, a small set of negative "seed words" is chosen then a graph

of synonyms is searched and words are added to the negative list that are found on this graph. The dictionary method has the benefit of quickly and easily finding a large number of sentiment words, though can result in lists that contain many errors and require manual checking in order to correct, which is a time-consuming effort. Another limitation is that the words collected using this methodology are also independent of domain and context.

The corpus-based approach has been used to find other sentiment words from a domain corpus given a list of seed words. This method is used to create a lexicon that contains words that are specific to the domain being studied. For example, a general lexicon of sentiment words based on the Oxford English Dictionary might not be relevant for studying Twitter. In this case, using a domain dictionary based on words gathered from Twitter might be more effective. This approach tends to be useful if a very large and diverse corpus is available. (B. Liu, 2012) (B. Liu, 2015).

2.4 Language models

One of the most central statistical language models is a probabilistic or distributional model that can be used for predicting text. In general, a probabilistic language model learns a large table of probabilities of word co-occurences. This table can then be used to predict what words would occur before, after, or between other words or phrases.

The most common probability model learns the probability of a word given 1 or more preceding words. The probability of a longer

phrase or a sentence occuring can then be calculated using the conditional rule of probabilit. In it's simplest form, the probability of an event A given an event B is:

$$P(A|B) = \frac{P(A) \cap p(B)}{P(B)}.$$

The chain rule of probability gives a formula for calculating the probability of an event given more than one event:

$$P(A|B,C,D) = P(A) \cdot P(B|A) \cdot P(C|A,B) \cdot P(D|A,B,C)$$

This is applied to predicting text by considering words in a sentence as events. For a sentence with words $w_1, w_2, \cdots w_n$, the chain rule of conditional probability becomes

$$p(w_1 w_2 \cdots w_n) = \prod_t P(w_t|w_1 w_2 \cdots w_{t-1}). \tag{2.1}$$

The above model of word occurrences is called the **n-gram** model, where n refers to the number of words in a row whose probabilities are being predicted (Manning, Manning, and Schütze, 1999). The n-gram model assumes that each set of words of length n are statistically independent. For example, for a model with $n = 2$, a word would be predicted using a single preceding word; For $n = 3$, a word would be predicted using two words preceding that word. The assumption that every n words are independent is known as the **Markov** assumption.

Mathematically, the goal of the n-gram model is to estimate the probability of the occurrence of w_t, the word w at position t . This is done by calculating $P(w_t|w_{t+n-1} \cdots w_{t-1})$: this is the probability

of w_t given the $n-1$ preceding words. To accomplish this task, the probability distribution of sets of $n-grams$ are estimated based on the corpus of data D.

For a unigram model, no conditional probabilities are required and just the frequency of individual words are used. For the bi-gram model, the conditional probabilities of any two words, w_1, w_2 is estimated based on their individual and joint frequencies:

$$P(w_2|w_1) = \frac{count(w_1, w_2)}{count(w_1)}$$

For other n-gram models, additional conditional probabilities would need to be calculated.

Once each order of conditional probability has been calculated, the probability of a sentence or a given word at position t can be calculated. Using the example of the bigram model, the probability of the nth word in a sequence at position t is:

$$P(w_t|w_1 w_2 ....w_{t-1}) \approx P(w_t|w_{t-1}) \tag{2.2}$$

Like other language models we will discuss, the n-gram model is a distributional model of language. A probability distribution of word contexts, here, the $n-1$ words that precede a word, are learned from a corpus. Below we discuss similar distributional models are used to train deep neural networks.

While many existing NLP systems view words as atomic units, as in the Bag-of-Words models, this approach seems to have reached the upper limit of its effectiveness in NLP tasks (Mikolov, Chen, et al., 2013). Many texts cannot be considered or analyzed by breaking

the text down to its atomic units, words (Kong et al., 2011). This suggests that continuing to try to scale up these techniques will no longer lead to continued progress (Mikolov, Chen, et al., 2013). Vector space representations of word or embeddings learned from deep neural networks are one representation of language that attempt to better capture the meaning and contexts in which words are used beyond their occurrence in n-grams.

2.4.1 Vector Space Representations

The n-gram model learns a probability distribution over words, or pairs or triplets of words. An alternative way to represent a language is by representing words as vectors in a vector space. Early versions of vector space models were used for solving problems related to document retrieval. In a basic vector space model, documents can be represented by a vector of terms weighted by the frequency of their occurrence, although other frequency measures are also used (Salton, Wong, and Yang, 1975).

Another method that arose in the context of document retrieval was Latent Semantic Indexing(LSI)(Deerwester et al., 1990). This method uses the co-occurrence of terms and documents as a starting point. A term-document matrix represents the frequency of term w_i in document D_j as the (i, j) entry in a matrix. The total number of terms in a set of documents can be very large. Representing a document as a vector of term weights means this can be a very high-dimensional representation of the document. In LSI the principal component decomposition (PCA) of the term-document matrix is used to reduce the dimensionality of the representation. A small

subset of vectors can be used to capture most of the variation, or information, in the term-document matrix. A different vector space representation or words, also called word embeddings, has been developed by using neural networks.

2.4.2 Neural Network Language Models

One of the innovations of the early neural network language models was to propose a method that could learn both a probabilistic representation of a word sequences, similar to the n-gram models, while also learning a low dimensional representation of the language as word vectors in the process. Bengio *et al* introduced a model that learned a word embedding in the process of being trained to learn a probability distribution over sequences of words(Bengio et al., 2003). In Bengio et al.'s model, the neural network is trained to output a probability model $f(w_t, \cdots, w_{t-n+1}) = \hat{P}(w_t|w_1, \cdots w_{t-n+1})$. This is similar to the objective of an n-gram model.

The first layer of this network is a mapping from the index of any word in the input vocabulary V to a vector in $\mathbb{R}^m$. This mapping, C, is matrix of dimension $|V| \times m$. The output probabilities are determined by functions with parameters ω, which map the word vectors output by C to a probability.

The neural network is trained to minimize:

$$L = \frac{1}{T} \sum_t \log f(w_t, w_{t-1}, \cdots, w_{t-n+1}; \theta) + R(\theta). \tag{2.3}$$

The final term, $R(\theta)$ is a regularization term applied to the weights of the neural network layers as a way to avoid overfitting (Bengio et al., 2003). In the model of Bengio *et al.*, there is a single hidden

layer which creates a non-linear map of the output of the first layer to the input of the final layer, in this case a tanh function.

2.4.3 Word Embeddings, word2Vec and GLoVE, FastText

The word embeddings learned the neural network described above come from the matrix C that makes up the first layer in the neural network. The rows of C are an m-dimensional representation of the words in the input vocabulary. In other words, the first layer of this model is the "embedding layer" since it determines the embedding of words in some m dimensional vector space. The dimension m is also a parameter of the model that can be tuned.

One of the advantages of the vector space embedding learned by neural networks is that they capture the semantic relationship of words – meaningful relationships between sets of words are represented by the geometric relationship of the vectors.(Mikolov, Chen, et al., 2013) For example, the relationship "Paris" is to "France" as "Berlin" is to "Germany" is represented by the vectors corresponding to these four words. Roughly speaking,

$$\text{Paris} - \text{France} + \text{Berlin} = \text{Germany}$$

(Mikolov, Chen, et al., 2013). The contextual information encapsulated in word embeddings appears to be one of the reasons word embeddings have performed well across a range of natural language processing tasks.

The neural network language model as described by Bengio *et*

al. was computationally complex, particularly when trained on a very large corpus. Another reason for the more recent popularity of word embeddings is that computational power has become less expensive. Research has also been done on developing models that use fewer inputs to learn word embeddings of similar quality to models trained on very large corpora.

The *word2Vec* model, for example, (Mikolov, Chen, et al., 2013) demonstrated efficient methods for learning word and phrase vector representations. The hidden layer calculations of the Bengio *et al.* model were removed, creating a "shallow" or single layer neural network. And the output layer calculations were simplified.

As well as introducing efficient computational models, (Mikolov, Chen, et al., 2013) introduced two new language models, the **Continuous Bag of Words**, or **CBoW** model and **Skip-gram** models. In the **CBow** model, a word w_t is predicted by it's context, or surrounding $2n$ words. Mathematically speaking, the model determines the probability

$$\log P(w_t|w_{t-n}, \cdots w_{t-1}, wt+1, \cdots, w_{t+n}) \tag{2.4}$$

that is, the probability of word w_t given the n words on either side of word w_t.

Similar to the **CBoW** model the Skip-gram model uses a window around a word, but instead of predicting a single word, the context is predicted. The model predicts the probability

$$\log P(w_{t-n}|w_t) + \cdots \log P(w_{t-1}|w_t) + \log P(w_{t+1}|w_t) + \cdots + \log P(w_{t+n}). \tag{2.5}$$

In other words, given word w_t, the n words preceding and the n words following the word w_t are predicted.

The success of the word2Vec model and the ability to learn word representations efficiently has lead to a proliferation of models for learning word representations. These models are trained to learn general language models based on word or phrase probabilities. Like the n-gram model, these models are designed to predict a word based the surrounding words. The word embeddings learned by these models can capture the contextual meaning of words. But the embeddings do not necessarily include information useful for predicting sentiment in texts (Maas et al., 2011).

Another approach, taken by Maas and colleagues, uses a combination of unsupervised and supervised techniques, with the goal of calculating word vectors which capture semantic information and sentiment content. Within their model, the semantic component uses an unsupervised probabilistic model which is combined with a supervised component focused on learning sentiment. In the example given by Maas *et al.*, the model learns sentiment on a labeled dataset which pushes word vectors with a more positive sentiment to one side of a hyperplane while word vectors with negative sentiment are pushed to the opposite side. The model's objective combines the two goals of learning a language model and separating words with different valence. Using these techniques, researchers were found to achieve high levels of accuracy across several sentiment tasks (Maas et al., 2011).

2.5 Sentence Embeddings

While word embeddings attempt to capture the meaning of individual words, work has also been done on creating sentence level embeddings as well. Similar to word embeddings, sentence embeddings represent the meaning of a sentence as a single vector of real values.

An approach to creating sentence embeddings that does not require building a new language model is to use word vectors trained on existing models. To embed a sentence, the average of the words in a sentence are used to create a sentence embedding. For some sentence s with words w, and v_w vector representation of word w, the sentence embedding v_s is calculated:

$$v_s = \frac{1}{|s|} \sum_{w \in s} v_w$$

The *smooth inverse frequency* (SIF) model builds on the simple averaging of word vectors(Arora, Liang, and Ma, 2016). It has been proposed as a strong baseline for sentence embeddings and is competitive with embeddings based on more complex language models. The SIF weights the averages by the probability of the words included in each sentence. It also adjusts the resulting sentence vectors by subtracting the first principal component of a matrix formed by all the sentence vectors (Arora, Liang, and Ma, 2016).

The SIF sentence vectors v_s, where words of each sentence appear in the training corpus with probability $p(w)$, are calculated:

$$v_s = \frac{1}{|s|} \sum_{w \in s} \frac{a}{a + p(w)} v_w \tag{2.6}$$

(Arora, Liang, and Ma, 2016).

The weighting term is $\frac{a}{a+p(w)}$.In (Arora, Liang, and Ma, 2016), the value a is a parameter of SIF vector and was set to 0.0001.

Since $p(w)$ is in the denominator, words that occur more frequently are given less weight, similar to TF-IDF weighting. TF-IDF or term-frequency, inverse document frequency is a widely used method for determining the importance of a word in a collection of documents. The method down-weights words that occur frequently across all documents. The idea is that uncommon words that appear in a document help identify what that document is about. Applied to sentences, this weighting method helps determine which words are most related to a sentences' meaning.

After determining v_s for all sentences in the training data, a matrix is formed with columns v_s. The first principal component, u of this matrix is calculated and each of the original v_s is updated

$$v_s = v_s - uu^T v_s$$

(Arora, Liang, and Ma, 2016). In our study, we use the SIF sentence embedding technique with Facebook's FastText word embeddings.

Other sentence embeddings are created in a way similar to word embeddings: a neural network is trained in some language task and the matrix that is the first layer of the neural network is used as the sentence embeddings. In these models, the task uses sentence input and generally predicts language output at the sentence level.

The OpenAI group and Radford *et al.* have had success creating sentence embeddings learned on a general language model that are then applied to sentiment analysis tasks (Radford, Jozefowicz,

and Sutskever, 2017). The authors observe that models trained on corpora with weak sentiment may not learn information needed to make the features useful for sentiment analysis tasks. The authors train their model on a large corpus of Amazon reviews which can be expected to express strong positive and negative valence. A linear model trained on the learned sentence embeddings were shown to perform at state-of-the-art levels on the Stanford Sentiment Treebank, a dataset based on sentences extracted from movie reviews. In particular, the OpenAI sentence embeddings were shown to have one 'neuron' or column that by itself strongly predicted text sentiment (Radford, Jozefowicz, and Sutskever, 2017). These results show that embeddings learned on a general task can be transferred to sentiment analysis tasks.

We use the OpenAI sentence embeddings as one of our features in our study. We refer to these sentence embeddings as the **sentiment-neuron** embeddings.

2.6 Machine Learning Models

A wide range of machine learning models have been applied to text data. We focus here on the models used in this thesis.

Raw text data can be considered as unstructured data. The data is not represented by a fixed set of feature columns but instead is a continuous stream of raw data: words and punctuation. The neural network models discussed above use unstructured text data to build a probabilistic language model (Bengio et al., 2003). We use the word vectors that are created in the process of training a neural network models. Since these word embeddings are structured

data, they have the same number of columns for all words, they can be used as features in models that take structured data. Structured data can also be used as input to neural networks but we did not use a neural network as one of our models in this study.

In what follows, we describe the details of models used in our prediction task. These are models that use structured data, that is, a fixed set of feature columns to predict emotional intensity of tweets.

2.6.1 Linear models

A linear model or linear regression is one of the most widely used classes of models when modeling response variables that are continuous. Simple linear regression – when there is just one predictor variable– models the relationship between a single feature or predictor x and a numerical response Y as a simple line:

$$Y = \beta_0 + \beta_1 X. \tag{2.7}$$

The coefficient β_0 of the model describes the intercept of the line and the coefficient β_1 is the slope of the line. The sign of the slope coefficient describes whether the direction in which the response variable changes is positive or negative when the predictor variable increases. The size of the β_1 coefficient describes the rate at which the response changes when the predictor increases.

For multiple predictors, the model is extended by adding a coefficient for each additional predictor. The multiple linear regression model is

$$Y = \beta_0 + \sum_{j=1}^{p} X_j \beta_j. \tag{2.8}$$

The multiple regression model assumes that the deviations of the response around the linear model are independent and normally distributed

$$Y = \beta_0 + \sum_{j=1}^{p} X_j \beta_j + \varepsilon \tag{2.9}$$

with $\varepsilon \sim N(0, \sigma^2)$ (Trevor, Robert, and JH, 2009).

2.6.2 Lasso model

Datasets with a large number of variables may contain redundant variables, variables that are highly correlated and whose inclusion biases the model results. The data may also include variables with little effect on the response. There are a number of methods for removing variables using automated selection techniques such as forward or backward variable selection. These methods generally use some metric and variable is chosen for removal or inclusion base on how this metric. These variable selection methods, though, can have high variance because the processes of adding and removing variables is discrete, rather than continuous (Trevor, Robert, and JH, 2009).

An alternative to variable selection are shrinkage methods that reduce the size of the variable coefficients in a relatively continuous way. The two most common shrinkage models for linear models are ridge regression and the lasso model. Both models add a penalty term to the basic linear regression model. However, the lasso model both shrinks and removes variables from the linear model. When using very high dimensional data, such as text data where the features are individual words, removing variables that do not affect

the response variable can be very useful. It reduces the dimensionality of the data and can make interpretation of the resulting model simpler.

The lasso model minimizes the following equation.

$$\hat{\beta}^{lasso} = \text{argmin}_{\beta} \left[\sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right] \quad (2.10)$$

(Trevor, Robert, and JH, 2009).

There are two terms that are being minimized in equation 2.10. One term is the squared difference between the observed response variable, y_i and the prediction of the linear model from equation 2.9:

$$\beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j.$$

The second term is the term that penalizes the model coefficients:

$$\lambda \sum_{j=1}^{p} |\beta_j|. \quad (2.11)$$

The model coefficients are represented by the β_j term. When the penalization parameter λ is larger, the sum of the coefficients are forced to be smaller. This results in setting small coefficients in the model to zero. This penalization term is responsible for shrinking some coefficients and removing other coefficients from the lasso model.

The value of the parameter λ determines how much shrinkage is applied to the coefficients of the model. Generally, this parameter is varied over a wide range when fitting the model and the best parameter value is chosen by cross-validation or some similar method.

2.6.3 The rare model

The lasso model described above is commonly applied to high dimensional data in order to reduce the dimensionality of the data and to make the resulting model more interpretable. Dimensionality of the data is reduced when features are removed by the penalization term in equation 2.11.

For many types of data, however, the features can have characteristics that make it harder for the lasso to find the coefficients that truly affect the response variable. Data that is very sparse or has many rare features can cause problems for the lasso model (Yan and Bien, 2018a). For example, when building a linear regression model to analyze tweets, most tweets will have a very small subset of the words. This is the sparsity of the words or features in each tweet. In addition, there may be words that show up in very few tweets. This is rarity of any particular word. These rare features may be important in predicting the response but if they occur infrequently that information will not be incorporated into the model.

For example, if you are using a document-term matrix (DTM), each word is a feature. There may be 10,000 words in a collection of sentences or tweets from Twitter, but only only a few of these words show up in any given tweet. The feature vector that represents that sentence will will be sparse because most features or words are not included in that sentence. In the feature vector, words not in the sentence are represented as zeros.

Many words are also not used in most sentences, making those words rare. Rare words may be very important for the meaning of the text but a model will have difficulty using a rare word as a

feature if it is only used once or twice in a collection of text data.

Xiaohan Yan and Jacob Bien have recently proposed the rare model as a method for handling sparse data with rare features. Their approach is to aggregate similar rare features into a single feature. Using the text example from above, a set of words that individually appear infrequently may be grouped with a set of words with similar meaning. As an aggregate, the group of words is no longer as rare or sparse as the individual words are.

Following the example Yan and Bien's include in their paper, let's assume we have a dataset with the following words that have been determined to be similar: { 'hideous', 'ghastly', 'dreadful', 'horrendous', 'horrible'}Yan and Bien, 2018a. These words correspond to a set of features in our dataset: $X_1, X_2, X_3, X4, X_5$. In this example, the feature matrix records the counts of each word in our tweets. The column 'hideous', column X_1, has the number of times the word 'hideous' occurs in each tweet. Aggregating the features for a single tweet consists of summing up the counts for the aggregated set of words into a single new feature $\tilde{X}$,

$$\tilde{X} = X_1 + X_2 + X_3 + X_4 + X_5 \tag{2.12}$$

(Yan and Bien, 2018a).

Yan and Bien show that, at least in an ideal situation when the aggregation matches the true model, the lasso model will recover the true model coefficients. Without aggregation, the lasso will not recover the true model for any parameter λ (Yan and Bien, 2018a).

2.6.4 Feature aggregation

This leaves open the question of how to determine which words or features should be aggregated into a single feature. Yan and Bien propose using tree-based hierarchical clustering to aggregate variables. In order to create a cluster of features, information is used from outside the features used to create the linear model that is part of the rare-model. The data used to cluster the features provide some type of information about how the features are related or how "close" each feature is to the other. This "side data" is not part of the feature set but provides metadata about those features. This side data is used to build a hierarchical clustering model.

The hierarchical clustering model is built stepwise. When using an agglomerative or bottom-up hierarchical model, the model first identifies groups of individuals and then fuses those groups that are close to each other in the feature space into larger groups at the next level (Trevor, Robert, and JH, 2009). The result of this process is a tree with multiple levels where the lowest level are the original data points and at higher levels are fewer groups and larger clusters.

In the rare model, the leaves of the tree that results from clustering are the original features and nodes in this tree represent a cluster of all the child nodes of that tree (Yan and Bien, 2018a). We denote the tree formed by hierarchical clustering $\mathcal{T}$ with p leaves corresponding to each of the features of the model. In the linear model, each feature has a coefficient β_j. The rare model parameterizes the set of nodes in the tree formed by hierarchical clustering with each node associated with a parameter γ_u.

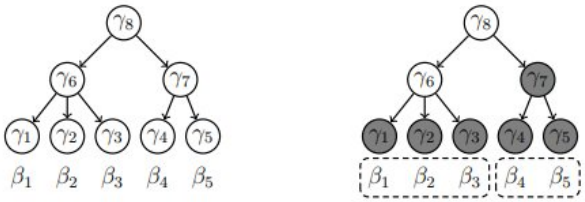

FIGURE 2.1: Representation of the tree from a hierarchical model with nodes parameterized by γ_u. From (Yan and Bien, 2018a).

Each β_j coefficient of the linear model is then expressed as a sum of the γ_u that are ancestors of that leaf:

$$\beta_j = \sum_{w \in ancestor(j) \cup \{j\}} \gamma_u.$$

The set of coefficients can then be expressed $\mathbf{fi} = \mathbf{Afl}$ where $\mathbf{A}$ is a binary matrix with:

$$A_{jk} := 1_{u_k \in ancestor(j) \cup \{j\}} = 1_{j \in descendant(u_k) \cup \{u_k\}}$$

(Yan and Bien, 2018a).

The rare model encourages sparsity using a penalty term λ then penalizes the size of the coefficients. The λ term also penalizes the number of γ_u included in the model, which encourages grouping features using higher level nodes in the hierarchical model.

The rare model introduces another term α, which determines how the penalty of the coefficient terms and the γ_u terms is balanced. The optimization problem solved by the model is:

$$\min_{\beta \in \mathbb{R}^p, \gamma \in \mathbb{R}^{|\mathcal{T}|d}} \frac{1}{2n} ||\mathbf{y} - \mathbf{X}\beta||_2^2 + \lambda \left(\alpha ||\gamma_u||_1 + (1-\alpha)||\beta||_1\right) s.t. \beta = \mathbf{A}\gamma \tag{2.13}$$

(Yan and Bien, 2018a).

The values for both α and λ are chosen by cross-validation. This method is implemented in the R language package, `rare` (Yan and Bien, 2018b).

2.7 Twitter data

Twitter has become popular among researchers in this field as it allows researchers to access very large textual datasets which include the expression of emotion. The use of data from Twitter also serves to provide a challenge, as along with the informal use of language, users also make a substantial number of grammatical errors and use emojis and hashtags in a way that is unique to short texts (Kouloumpis, Wilson, and Moore, 2011). Despite these hurdles faced in using Twitter data, the volume of texts and the diversity of users on Twitter has made it an important source of data for analyzing realtime public sentiment (Patodkar and I.R, 2016).

Twitter data has also been used in several competitions focused on sentiment analysis and emotion detection including the semEval 2014 and semEval 2018 contests (S. Mohammad et al., 2018) (Nakov, Rosenthal, et al., 2013). The data for our study comes from the semEval 2018 contest and is described in more detail in 3.

In the semEval 2018 competition, the SeerNet model had the best performance in the regression and ordinal classification tasks(S.

Mohammad et al., 2018). The SeerNet model used word and sentence embeddings as well as sentiment features from the Python package EmoInt. The EmoInt package was created by the developers of the datasets used in the semEval 2018 contest and were designed specifically to identify the discrete emotions, anger, fear, joy, and sadness, used for labeling the semEval data. The SeerNet final model consisted of an ensemble of models trained on individual feature sets (Duppada, Jain, and Hiray, 2018).

Chapter 3

Data

3.1 Data Source

The data used in this study is derived from Task 1 of the SemEval-2018 contest. This task's purpose was to estimate the emotional content of the tweets. The data we use is from the subtask focused on estimating the emotional intensity of tweets across four different emotions, anger, joy, sadness and fear. The datasets were created from tweets in three languages (English, Arabic, and Spanish). In our study we only use the English dataset, although the the procedure used for compiling the other datasets was similar to that used for the Enligsh dataset. This group of datasets was collectively named the Basic Emotion Tweets dataset.

3.1.1 Data Collection and Annotation

The Basic Emotion Tweets dataset was created by querying Twitter with a set of terms associated with each emotion. For each emotion, fifty to 100 query terms were selected. These terms represented a variety of emotion intensity levels. These query terms were selected based on selecting a few words words associated with the

basic emotions, angery, joy, sadness and fear. Several methods were then used to expand this initial set of words: synonyms of these words were selected from *Roget's Thesaurus* and words close to the initial set in word embedding space were selected (S. Mohammad et al., 2018). Once the set of query terms was set, the Twitter API was polled using these terms to create a large pool of tweets. A random selection of tweets were selected from these larger pools to form the datasets used for the emotional intensity subtask. A similar method was used for the compilation of Arabic and Spanish tweets.

For the English dataset, annotation was done by crowd-sourcing using between 118 and 220 residents of the United Sates for the datasets associated with each emotion. Scoring of emotional intensity was done using the Best-Worst Scaling (BWS) method. Best-Worst Scaling addresses issues of both rating consistency and reduces the number of rating required for reliable ratings. BWS scaling has annotators use comparative rankings between two or more items (S. Mohammad et al., 2018). For example, if among four items, A, B, C, D, where A is rated best and D rated worst, then a rating is consistent if the rating also determines A > B, A > C, D < B, D < C.

For the emotional intensity datasets, the BWS method was used with groups of 4 tweets per annotation. The BWS method gives a ranking. Rankings were transformed into a real value by taking the proportion of times the tweet was scored has having the highest intenstiy and subtracting the proportion of times the tweet had the leas intensity. The resulting set of scores was transformed to range between 0 and 1 (S. Mohammad et al., 2018).

The annotated tweets for each emotion were divided into training, development, and test sets. The training sets had betweet 1500-1700 tweets, development sets had around 300 to 400 tweets, and the test set had around 1000 tweets.

3.1.2 Bias Detection Dataset

The test set also included a large number of tweets designed for a bias detection task. The purpose of this dataset was to determine whether models systematically rated tweets associated with gender and race categories differently. The dataset was formed by creating sets of simple sentences where the only difference between examples of the sentence was the gender or name of the person in the tweet.

These "Tweets" were generated from sentence templates. These templates were the same for the datasets associated with each emotion. The templates had simple sentences, for example :"_____ feels furious." Multiple versions of each template were formed by inserting varying names and pronouns into the subject position. For example: "Heather feels furious" or "Alphonso feels furious". The words inserted into the subject position came from 6 lists. Two lists consisted of generic male and female pronouns: "Aunt", "Sister", "She", in the female gendered list and "Uncle", "Brother, "He", in the male gendered list, for example. The other lists consisted of Male and Female proper names divided into names identified as African American and more generic names. For example, the African American male name set included "Darnell", "Jamel", and

"Alonzo", while the names not identified as African American included "Roger", "Justin", and "Andrew".

Chapter 4

Methods

In order to analyze the performance of features set and model combinations, we created three sets of features:

1. Baseline lexicon and text features
2. SIF sentence embeddings
3. "Sentiment Neuron" sentence embeddings.

For each feature set, we trained 3 models and one final ensemble model for a total of 4 models per feature set. For all models, parameters were chosen by cross-validation using grid search to choose parameters. Model selection was based on the minimization of the root mean-squared error (RMSE).

4.1 Baseline features

The baseline features set included basic text and lexicon features. The set of basic text features included counts of hashtags, mentions, URLS, exclamation points and other punctuation. These features also included counts of the use of first, second or third person pronouns within the tweet. These features were created using the R

package 'textfeatures'(Kearney, 2018). Some text features were removed from the baseline set if there was near zero-variance among the feature set. Near zero-variance implies that the feature was the same or almost the same for all observations.

The AFINN and Bing sentiment lexicons were used for computing the valence of each tweet (Nielsen, 2011)(Hu and B. Liu, 2004). The AFINN lexicon scores all words on an integer scale while the Bing lexicon only labels words as positive or negative. The score of each tweet is calculated by summing the lexicon score for all words in the tweet.

A number of emotion specific features were created using lexicons from the National Research Council of Canada (NRC) collection. The most common or widely used lexicon from this collection is the Word-Emotion Association Lexicon (Saif M Mohammad and Turney, 2010). This lexicon, like the AFINN and Bing lexicons, was created by manually labelling terms. Terms in the Word-Emotion lexicon are associated with one or more of eight emotions: trust, fear, sadness, anger, surprise, disgust, and joy. In addition, words can have a positive or negative association. A tweet was given a score for each emotion by summing over the total number of words associated with each emotion or valence.

The remaining lexicons used from the NRC collection were created automatically and using the following methodolgy. Three hashtags and two emoji lexicons were created using a collection of 775,000 tweets. A set of hashtags and emojis were scored for either positive or negative valence or for association with an particular emotion. For all tweets in the dataset, the pointwise mutual information (PMI) was calculated to determine the association of words with

each labelled hashtag or emoji. Pointwise mutual information is a standard method for calculating the association of terms by measuring how often the terms occur together in a text. For postive and negative valence, the PMI socre is calculated:

$$score(w) = PMI(w, positive) - PMI(w, negative)$$

where w is a word in the lexicon (Saif M. Mohammad, Kiritchenko, and Zhu, 2013).

For this set of lexicons, associations were generated for both unigrams (single words) and bigrams (word pairs). For our feature set, we have only used the unigram lexicons. Sentiment and emotions scores for the NRC lexicons was calculated using the *syuzhet* R package (Jockers, 2015).

4.2 SIF Sentence Embeddings

FastText word embeddings were used to create sentence embeddings using the smooth inverse frequency(SIF) method (Bojanowski et al., 2017)(Arora, Liang, and Ma, 2016). The FastText word embeddings are taken from a layer in a neural network trained on a corpus of 16 billion tokens from Wikipedia, the UMBC webbase corpus, and the statmt.org dataset (*English word vectors · fastText* n.d.). These word embeddings included 1 million 300-dimensional word vectors.

SIF embeddings were created by averaging each tweets FastText word embeddings. The first principal component of the set of averaged word embeddings was then subtracted from each embedding.

The resulting SIF sentence embeddings were also 300-dimensional vectors.

4.2.1 Sentiment Neuron Embeddings

The sentiment neuron sentence embeddings come from a deep learning network trained on Amazon review data by the OpenAI group, (Radford, Jozefowicz, and Sutskever, 2017). The word embeddings come from the first layer of this network. In addition to learning the semantic relationships between words, the embeddings were found to have learned a good representation of positive and negative sentiment, (Radford, Jozefowicz, and Sutskever, 2017).

The embeddings from this model are 4096 dimensional word vectors.

4.3 Model Fitting

For each of the previous three features: the baseline sentiment and word features, the SIF sentence embeddings, and the sentiment neuron sentence embeddings, three models were fit: a random forest model, a gradient boosted trees model and a lasso model. The baseline features were pre-processed by removing near zero variance predictors. No pre-processing was applied to the sentence embedding feature sets.

Model parameters were tuned using 5-fold cross-validation over a grid of parameters. Parameter selection was based on the RMSE of each model. The `caret` package was used for parameter tuning. The parameter search was over the default grid of parameters set

by the `caret` package (Jed Wing et al., 2018). The lasso, random forest, and gradient boosted tree models were fit using the `glmnet`, `randomForest` and `xgboost` R libraries, respectively.

4.4 Ensemble

A total of nine models were created, three models for each of the three sets of features described above. We created an ensemble model for each feature set by stacking results from the three models trained on that feature set. Stacking was done by training a ridge model on the predictions of the three models for each feature set. The weights of the model were tuned by five-fold cross-validation. The ensemble model for each feature set is then based on the weighted output of the three input models.

4.5 Bias Detection

We used the bias detection dataset included with the EI-reg test dataset to test whether our models systematically scored male or female gendered sentences differently for each emotion. Male and female gendered tweets were separated based on gendered pronoun lists provided with the semEval dataset. Regular expressions were used for splitting tweets into a "male" and "female" gendered sentences where sentences varied only by the pronoun or proper name used as the subject of the sentence.

We used the three ensemble models created for each set of features to test for gender bias. Emotional intensity was predicted for

using each ensemble model on each of the four emotions and a difference in average intensity between the two emotions was tested using a randomized block ANOVA test.

4.6 Rare Model and Lasso Features

As discussed in the background, the rare model takes a document-term-matrix(DTM) as features in a penalized regression model that is related to the lasso. The DTM is a matrix whose columns are words and whose rows correspond to each tweet in the dataset. The entries in the matrix are the count of the number of occurrences of the each word in a tweet.

We created the word list used in the DTM by filtering for unique words from the training set. Numbers were removed and hashtags were stripped of the hash mark. Because the initial set of words was large, around 10,000 words in total, additional filtering was applied to the word list. To do this additional filtering, we created an index that measured the total emotional valence of each word in the dictionary. For each word, the absolute value of the sentiment and emotional valence were summed. Words with a low emotional intensity were filtered out of the word list.

This filtering process could be adjusted to create datasets of different sizes. We adjusted the filtering threshold to create three datasets of approximately 7,500, 5,000 and 1,500 words. The higher threshold resulted in a smaller set of words used in the model so, for example, the dataset with 1,500 words represented the 1,500 words with the most "emotional content", based on our filtering method.

The rare-model also uses a hierarchical clustering model which contains information about the relationship of the words in the DTM. To create the cluster model, we used a combination of word embeddings and sentiment lexicons. Facebook's FastText word embeddings, (Mikolov, Grave, et al., 2018), combined with the sentiment and lexicon data used in the baseline model. Word embeddings contain information about the semantic relationship of words, and the lexicons contain information about the emotional valence of the words. We used the combination of these two datasets for clustering in order to capture information about both the meaning of the words in the DTM and the emotional sentiment associated with the words.

The rare model has two tunable parameters: alpha and lambda, that control the penalization of the model coefficients and how that penalization is balanced between the linear model coefficient and the grouping coefficients. Ten-fold cross validation and the root mean-squared-error (RMSE) was used to choose the parameters for the rare model.

The rare model is a variation of the lasso model. To compare the performance of the rare-model to the lasso model the DTM used to fit the rare model was also fit on a lasso model. The lambda parameter of the lasso was chosen using 10 fold cross validation.

The lasso model was fit using the `glmnet` package (Simon et al., 2011). The rare model was fit using the `rare` package (Yan and Bien, 2018c).

Chapter 5

Results

5.1 Model Evaluation

The twelve models were evaluated on four test datasets of approximately 1000 examples each, one dataset for each emotion. A Pearson's coefficient measuring the correlation of the true and predicted emotional intensity was calculated for each combination of model, feature and emotion. The result is sixteen Pearson's coefficient for each feature set. The results are shown in Tables 5.1, 5.2, and 5.3.

The baseline features and the sentiment-neuron word embeddings had similar performance. The best model for these two features scored at or above 0.60 Pearson's correlation. The baseline model had the highest performance with the random forest model performing the best for the **anger** and **fear** emotions. For the sentiment-neuron embeddings, the stacked ensemble model scored best for

Emotion	Lasso	Random Forest	XG Boost	Stacked
Anger	0.51	**0.64**	0.569	0.63
Fear	0.59	0.628	0.54	**0.63**
Joy	0.57	0.608	0.55	**0.62**
Sadness	0.52	**0.65**	0.57	0.64

TABLE 5.1: Pearson's correlation for model on baseline features.

Emotion	Lasso	Random Forest	XG Boost	Stacked
Anger	0.07	0.21	0.19	**0.22**
Fear	0.13	0.159	0.18	**0.20**
Joy	0.08	**0.24**	0.179	0.3
Sadness	0.07	0.19	0.15	**0.19**

TABLE 5.2: Pearson's correlation for model on FastText SIF sentence embedding features.

Emotion	Lasso	Random Forest	XG Boost	Stacked
Anger	0.56	0.60	0.55	**0.63**
Fear	0.55	0.545	0.53	**0.60**
Joy	0.60	0.62	0.62	**0.65**
Sadness	0.55	0.57	0.54	**0.60**

TABLE 5.3: Pearson's correlation for model on sentiment-neuron sentence embedding features.

all models. The SIF features performed poorly and significantly worse than the baseline and sentiment-neuron embeddings across all models.

Leaving out the ensemble model, the random forest model tended to perform better than the lasso and gradient boosted tree models. In general, though, the three base models had relatively highly correlated predictions for the three feature sets. Tables 5.4, 5.5, and 5.6 show the Pearson's correlation of the model prediction for the emotion **joy**.

The between model correlations show generally higher correlation of all models for the baseline and SIF models. In particular, the random forest model was highly correlated with the lasso and gradient boosted trees for the SIF and baseline features. The sentiment-neuron embeddings, on the other hand, show a lower correlation of the random forest with the other two models.

For ensemble models, it is generally preferable to have uncorrelated models so that each model contributes new information to

	Lasso	Random Forest	XG Boost
Lasso	1.00	0.92	0.58
Random Forest	0.92	1.00	0.84
XG Boost	0.58	0.84	1.00

TABLE 5.4: Model correlation of baseline models on joy emotion.

	Lasso	Random Forest	XG Boost
Lasso	1.00	0.71	0.65
Random Forest	0.71	1.00	0.91
XG Boost	0.65	0.91	1.00

TABLE 5.5: Model correlation of FastText SIF sentence embeddings models on joy emotion.

the ensemble. The lower correlation of models for the sentiment-neuron embeddings is likely why the stacked ensemble models performed best across all emotions for that feature set. In particular, the random forest model, which performed better than other models, had a low correlation with other models for the sentiment neuron features.

5.2 Error Analysis

Figure 5.1 shows the distribution of the scored emotional intensity level for the test tweets for each emotion. The plot shows that the distribution is roughly symmetrical around the mean and that most tweets have an emotional intensity in the middle range and fewer

	Lasso	Random Forest	XG Boost
Lasso	1.00	0.29	0.91
Random Forest	0.29	1.00	0.44
XG Boost	0.91	0.44	1.00

TABLE 5.6: Model correlation of sentiment-neuron models on joy emotion.

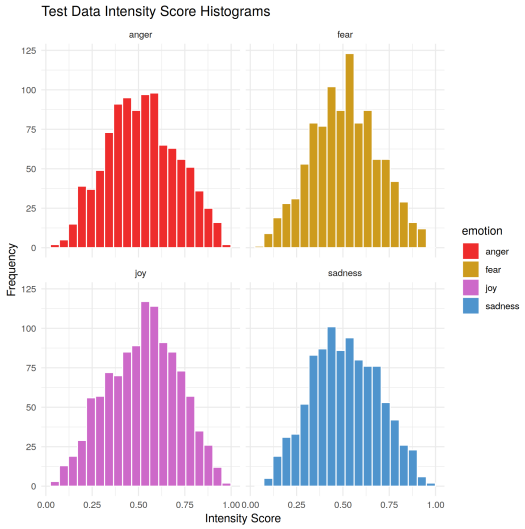

FIGURE 5.1: Tweet emotional intensity for test data.

tweets on the extremes ends of the intensity scale. The training set had an emotional intensity distribution similar to the test data.

The baseline ensemble model's predicted emotional intensity scores against the actual intensity scores are shown in Figure 5.3. As suggested by the ensemble results in Table 5.1, the pattern of predictions is similar across emotions. There do appear to be a higher number of large errors for the **joy** emotion, although the overall prediction correlation was similar for all emotions.

Figure 5.3 shows a scatterplot of the baseline ensemble model's prediction errors across against the emotional intensity of the tweet. Generally, there are larger errors in predicting tweets with more extreme emotions both those with higher emotional intensity and those with very low emotional intensity. This might be expected as there are fewer examples of emotions at the extreme. There also might be aspects of more extreme emotions that are hard to predict. We look at some of these large prediction errors in more detail below.

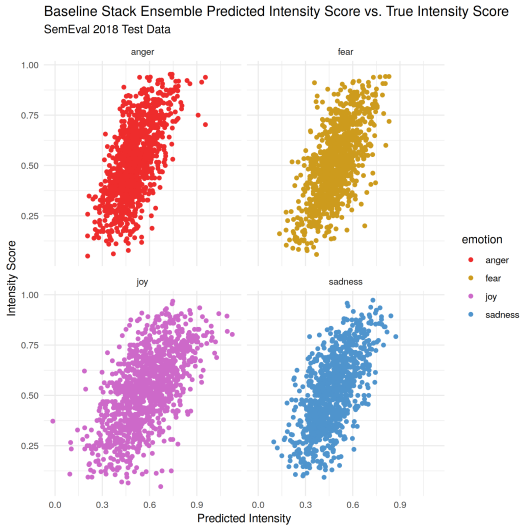

FIGURE 5.2: Sentiment-neuron predicted emotional intensity for test data vs. true emotional intensity.

The distribution of the prediction errors against the intensity level is similar for all emotions, although the model for **joy** appears to have more and larger errors on the lower end of the intensity scale. The **joy** model also has more large errors in the middle range.

The prediction errors for the sentiment-neuron embeddings are shown in Figure 5.5. The pattern is similar to that of the baseline model. Like the baseline model, there are more large errors for **joy** at the lower end of the intensity scale.

Figures 5.4 show the distribution and boxplots of just the largest 15% of the errors for the baseline ensemble. There is not a large difference in the median of the errors but the **anger** and **joy** boxplots show a slightly longer tail.

Tables 5.7 and 5.8 show several of tweets that our model's predictions had the largest absolute error from the labeled intensity. The errors for the sentiment-neuron features ensemble model show several examples where the model seems to confused by conflicting phrases.

FIGURE 5.3: Prediction error vs. intensity for baseline ensemble model.

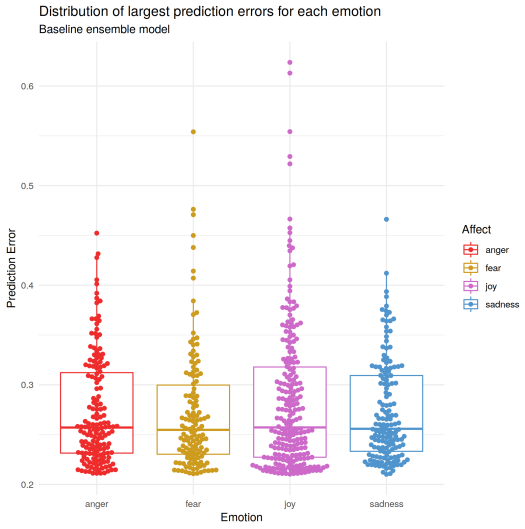

FIGURE 5.4: Distribution of largest 15% of baseline ensemble model errors.

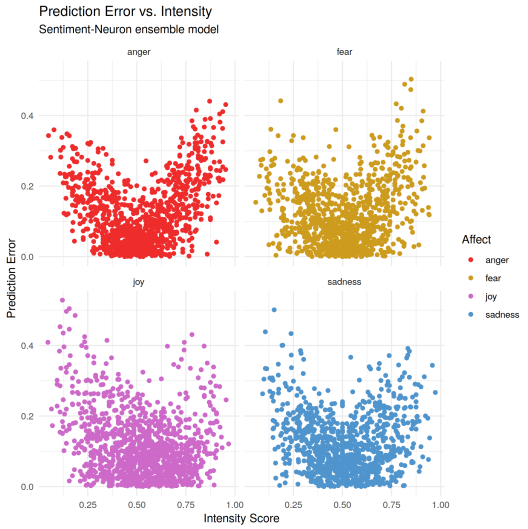

FIGURE 5.5: Prediction error vs. intensity for sentiment-neuron word embedding ensemble model.

In the first Tweet in 5.7, the religious message is hash-tagged with a happy sentiment, but the tweet includes the phrases "joyless" and "died". The hashtags show that the message is supposed to be one of faith and is intended to be joyful. We treated hashtags as regular words, stripping the "#" mark in order to be able to match the words to word-vectors. This means the hashtags would not be interpreted by the model any differently than a regular word.

A similar misinterpretation of terms may have confused the baseline model in the first example in Figure 5.7. The tweet includes the phrase "I am fine with" but also "blood tests terrify me". This is a statement where the first phrase is contrasted with the second phrase, but the emphasis is on the second phrase. The "AAAAAH-HHH" which was an emotional intensifier would also not have been included in the baseline model because the word was not captured as existing in the lexicons used.

The second tweet in Table 5.7 shows a possibly similar misunderstanding of hashtags. Here, the hashtags "good" and "great" appear to be sarcastic, while the tweet itself expresses "anxiety". In a similar way, the 4th tweet in 5.7 is about being out of toilet paper, but includes the term "Wonderful" but meant sarcastically. Some work has been done on sarcasm detection and sarcasm detection in tweets, in particular (Rajadesingan, Zafarani, and H. Liu, 2015). Incorporation of techniques detecting sarcasm would like improve emotion detection for tweets where the apparent emotional content is not really intended.

Recognition of named entities also appears to be a problem for our model. The second tweet in the baseline model table, Table 5.8 says: "4 years rest in piece #coreymontheith #glee". The name "Corey Monteith" contained in the hashtag was not included in our lexicons so would have been omitted. The name is that of a now deceased actor in the television show "Glee". The model would have understood "glee" as a simple positive word, but in this context the emotion of the tweet depends on understanding the "glee" and "#coreymontheith" as named entities. Relating proper names or named entities to the emotional content of the tweet would be a challenging problem.

5.3 Bias Detection

Previous results have shown that word embedding features can contain systematic gender bias (Bolukbasi et al., 2016). The semEval2018 emotional intensity regression datasets contained around 16,000 "Tweets" or sentences designed to test for model bias.

Tweet	Affect	Intensity	Prediction
A joyless faith is not one for which Jesus died. #thegospel #joy #Jesus #happiness	sadness	0.151	0.652
my blood sugar is 399 and my anxiety is wrecking me and i have court like right now and i made us late bc im fighting w my mom #great #good	joy	0.156	0.661
@COFFEECOWal Really Sad News, it's been a pleasure over the years, all the best for the future.	joy	0.186	0.671
Down to one roll... Wonderful I need more toilet paper. #personalassistant	joy	0.141	0.638
things that terrify me: remembering my bf follows me on twitter	fear	0.817	0.328
@DannyMcguire6 sad day Danny you have been and still are a true Leeds rhino you have been brilliant at the club I will miss you	joy	0.156	0.602

TABLE 5.7: Large prediction errors from the baseline sentiment-neuron word vectors ensemble model.

Tweet	Affect	Intensity	Prediction
Considering I am 101% fine with getting tattoos, blood tests terrify me and I AM HAVING TO GET ONE AAAAAHHHH	fear	0.845	0.374
4 years rest in piece #coreymontheith #glee	joy	0.047	0.671
@Afilicious I have at 7am so I have to get up around 5am to get ready! N also 7am classes are never pleasant	joy	0.125	0.738
@idktaehyng the possibilities terrify me	fear	0.911	0.357
@hamilou23 oswade I won't with you a happy birthday till you pay my 12000 shillings you borrowed on 12th January 2017.	joy	0.113	0.642
I just wanna be okay!! Like I know I can be all smile and giggles, but hot diggity dang, if that wasn't a mask, life would be swell!	joy	0.258	0.78
Racing all around the seven seas Chasing all the girls and making robberies n'Causing panic everywhere they go Party-hardy on Titanic...	fear	0.200	0.676

TABLE 5.8: Large prediction errors from the baseline ensemble model.

	Gender	Emotion
Baseline	$p = 0.294$	$p \ll 0.001$
SIF	$p = 0.407$	$p \ll 0.001$
Sentiment-Neuron	$p = 0.852$	$p \ll 0.001$

TABLE 5.9: ANOVA p-values for each ensemble model.

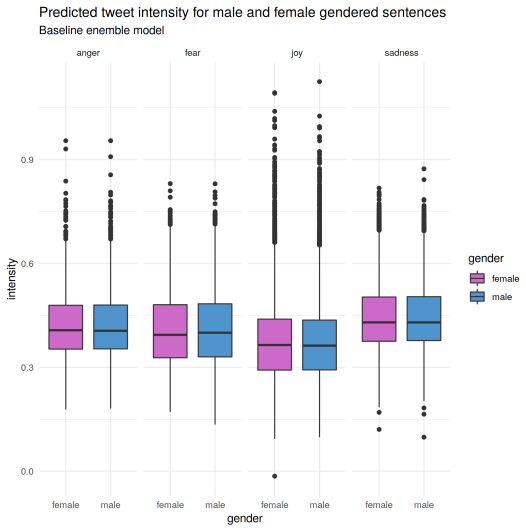

FIGURE 5.6: Box plots of emotion intensity distribution for the baseline features ensemble model for each emotion.

Box plots of the distribution of emotional intensity for male and female gendered sentence templates are shown in Figures 5.6, 5.7, and 5.8. The box plots show that the median emotional intensity for all emotions is very similar for both male and female gendered sentences.

A randomized block ANOVA test was run to test for an average difference between genders among any of the emotions. The p-values for the ANOVA test for emotion and gender are shown in Table 5.9. There was no significant difference found between the emotional intensity predicted for male and female gendered sentences. The difference in intensity between emotions was significant for each ensemble model,

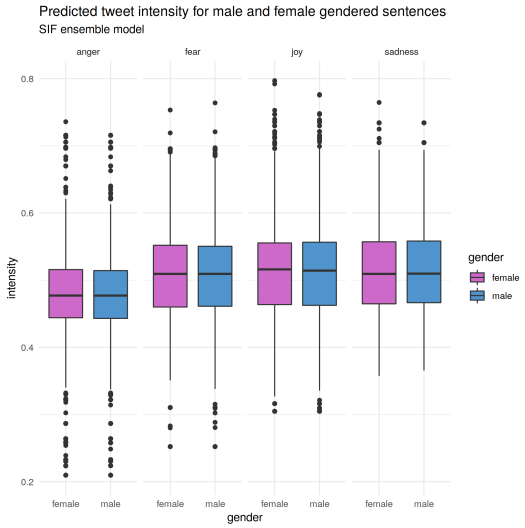

FIGURE 5.7: Box plots of emotion intensity distribution for the SIF feature ensemble model for each emotion.

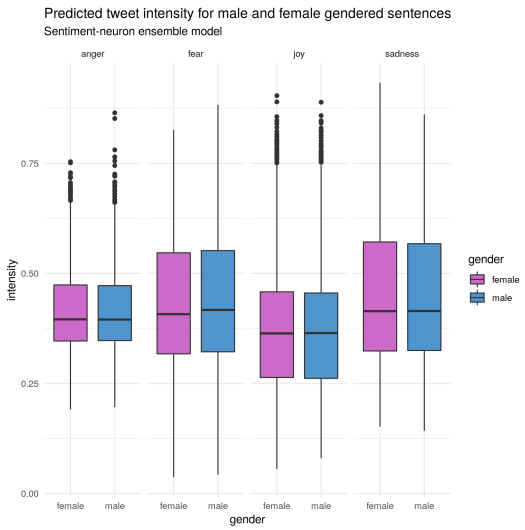

FIGURE 5.8: Box plots of emotion intensity distribution for the sentiment-neuron feature ensemble model for each emotion.

5.4 Rare Model

The rare model was trained on three sets of features, document-term-matrices with around 7,500, 5,000 and 1,500 words. The training of these three models had similar results. As a reminder, the rare model optimizes by grouping words in a document term matrix while also regularizing the model by shrinking coefficients to zero using a procedure similar to the lasso model.

The rare model optimization procedure resulted in models with all terms grouped into a single parameter. We obtained this result for each of the three models.

We also trained a lasso model on the three datasets and the result was similar: for each dataset the model with the lowest mean-squared error was an intercept only model. Although the rare model had a single coefficient in addition to the intercept term, the lasso and rare models essentially predicted a single value for all tweets in the test set.

Chapter 6

Discussion

The comparison of the model results for word embedding features and lexicon features designed specifically for detecting emotion in text reveals several things. Word embeddings, like the sentiment neuron embeddings which were trained on a general language task, can learn language features that reveal the emotional nuance of texts as well as specially designed emotion lexicons. Word and sentence embedding features, though, do not perform equally well. In particular, the performance difference between the **FastText** word embeddings and the OpenAI **sentiment-neuron** sentence embeddings show that embeddings are not equally successful at learning the emotional content of words.

The sentiment-neuron sentence embeddings performed similarly to the baseline lexicon features which were designed specifically for detecting emotions. The sentence embeddings also performed at a similar level across emotions. The Pearson's correlations for the sentence embedding ensemble model was at 0.60 or above for each emotion.

The OpenAI model which generated these sentence embeddings was trained to predict language at the byte level, or character by

character (Radford, Jozefowicz, and Sutskever, 2017). Although the OpenAI model was trained as a general language model, the authors trained the model on Amazon reviews and showed that the resulting language representation – the sentence embeddings – were also be useful for sentiment analysis (Radford, Jozefowicz, and Sutskever, 2017). Our results suggest that the language representation learned by their model is also useful for detecting emotions beyond positive and negative sentiment.

The language representation or embeddings learned by a model may not always be useful for sentiment analysis or emotion detection. For example, the FastText word embedding features did not perform well on our task. The FastText embeddings were also learned by a model trained on a general language task to predict sub-word length strings (Bojanowski et al., 2017). Although these FastText word embeddings were shown to perform well on a general text classification tests, they did not perform well in the emotion detection task (Joulin et al., 2017).

The FastText model was trained on a large corpus of news and Wikipedia articles. These texts were not likely to be rich in emotion, since news and Wikipedia articles are often intended to be neutral in tone. These results suggest that general embeddings, either word or sentence embeddings, will perform better if trained on corpora with richer emotional content. The OpenAI sentiment neuron model was trained on reviews that likely leaned toward positive and negative emotions, but it still appeared to learn a richer set of emotional content. Using other corpora that are created to represent a wider range of emotions may be useful for learning word embeddings that can be used to detect and quantify emotions in text.

The language features of the texts used to train deep learning models may also play an important role. We did not use any word embeddings that were trained on Twitter data. Other results suggest that embeddings trained on Twitter data would improve the performance of embeddings analyzing Twitter data. In particular, the SeerNet model, the top performer on several 2018 semEval tasks, used the DeepEmoji embeddings that were trained on Twitter Data (Duppada, Jain, and Hiray, 2018) (Felbo et al., 2017). Of the feature sets used by SeerNet, the models trained on DeepEmoji features performed best (Duppada, Jain, and Hiray, 2018). In particular, the DeepEmoji word embeddings appear to capture certain types of emotional nuance, including sarcasm(Felbo et al., 2017). Our error analysis suggested that our models struggled detecting sarcastic language, so DeepEmoji features may have helped with these errors. Beyond that, having features trained on Twitter data would likely help capture aspects of language as used on Twitter that is not present in corpora with longer texts, like hashtags, user handles, and use of emoji.

6.1 Future Work

In this work we focused on comparing models trained on small set of features. In particular, we compared the performance of models trained using word and sentence embeddings as features. The results show that some embeddings perform better than others and suggest that embeddings trained on emotion rich data are more likely to learn a representation that captures emotional content of language.

These results could be explored further by comparing word or sentence embeddings learned on either additional tasks or trained on different corpora. As mentioned above, the Deep Emoji embeddings are one example of embeddings that would likely be suited for models trained on Twitter data, since the embeddings were learned on a Twitter Corpus.

In addition, we compared the performance of these features on only one of the semEval 2018 tasks. Comparing a wider range of embeddings on additional tasks would allow us to compare feature performance on a broader range of tasks.

We also did not explore the affect of using averaged word embeddings compared to sentence embeddings. The FastText data was trained on word embeddings that were transformed to sentence embeddings using the smooth inverse-frequency (SIF) method. The poorer performance of the SIF embeddings may also be partly due to using transformed word embeddings rather than full sentence embeddings. Additional research would be needed to show whether the word to sentence embedding process had a significant effect on the performance of embeddings transformed in this way.

Embeddings are a vector representation of words or sentences. Past research shows that, for words, arithmetic with these vector representations carries contextual information about the words. For example, the result of the vector arithmetic: vec("Madrid") - vec("Spain") + vec("France") is close to vec("Paris"). It may be interesting to compare a similar arithmetic at the sentence embedding level. Our tests for gender bias in our features did not appear to show bias between male and female gendered sentences. This might be able to be explored in more detail by looking at the difference in vector space of

identical sentences or tweets where gender was changed between sentences. A similar test could be performed by varying the emotion word in the bias detection corpus of sentences that were part of the semEval test data.

Finally, the lasso and rare models trained on document term matrices resulted in intercept or single coefficient models. Additional work would be needed to test if using a different subset of the words in the Twitter training data would improve these models. For example, in the paper introducing the rare model, only adjectives were used in the document-term matrix (Yan and Bien, 2018a). It may be that short length and the sparsity of features in each tweet were responsible for the poor performance of these models. More work would need to be done to test these models suitability for working with short texts like Twitter.

Bibliography

Alawami, Alawya (2016). "Aspect Terms Extraction of Arabic Dialects for Opinion Mining Using Conditional Random Fields". In: *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, pp. 211–220.

Arnold, Magda B (1945). "Physiological differentiation of emotional states." In: *Psychological Review* 52.1, p. 35.

Arora, Sanjeev, Yingyu Liang, and Tengyu Ma (2016). "A simple but tough-to-beat baseline for sentence embeddings". In:

Bänziger, Tanja, Marcello Mortillaro, and Klaus R Scherer (2012). "Introducing the Geneva Multimodal expression corpus for experimental research on emotion perception." In: *Emotion* 12.5, p. 1161.

Bengio, Yoshua et al. (2003). "A neural probabilistic language model". In: *Journal of machine learning research* 3.Feb, pp. 1137–1155.

Bianchi-Berthouze, Nadia and Andrea Kleinsmith (2015). "Automatic recognition of Affective Body Expressions". In: pp. 151–169.

Bojanowski, Piotr et al. (2017). "Enriching Word Vectors with Subword Information". In: *Transactions of the Association for Computational Linguistics* 5, pp. 135–146. ISSN: 2307-387X.

Bolukbasi, Tolga et al. (2016). "Man is to computer programmer as woman is to homemaker? debiasing word embeddings". In: *Advances in neural information processing systems*, pp. 4349–4357.

Cohn, Jeffrey F. and Fernando De la Torre (2015). "Automated Face Analysis for Affective Computing". In: *The Oxford Handbook of Affective Computing*, pp. 131–150.

Cohn, Jeffrey F et al. (2009). "Detecting depression from facial actions and vocal prosody". In: *2009 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops*. IEEE, pp. 1–7.

Darwin, Charles (1872). *The Origin of Species: By Means of Natural Selection Or the Preservation of Favored Races in the Struggle for Life*. Vol. 1. Modern library.

Deerwester, Scott C. et al. (1990). "Indexing by Latent Semantic Analysis". In: *JASIS* 41, pp. 391–407.

Dror, Otniel E (2014). "The Cannon–Bard thalamic theory of emotions: A brief genealogy and reappraisal". In: *Emotion Review* 6.1, pp. 13–20.

Duppada, Venkatesh, Royal Jain, and Sushant Hiray (2018). "SeerNet at SemEval-2018 Task 1: Domain Adaptation for Affect in Tweets". In: *arXiv preprint arXiv:1804.06137*.

Ekman, Paul (1999). "Basic emotions". In: *Handbook of cognition and emotion*, pp. 45–60.

— (2016). "What scientists who study emotion agree about". In: *Perspectives on Psychological Science* 11.1, pp. 31–34.

Ekman, Paul and Wallace V Friesen (1971). "Constants across cultures in the face and emotion." In: *Journal of personality and social psychology* 17.2, p. 124.

Ekman, Rosenberg (1997). *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*. Oxford University Press, USA.

English word vectors · fastText. URL: https://fasttext.cc/docs/en/english-vectors.html.

Eyben, Florian et al. (2015). "The Geneva minimalistic acoustic parameter set (GeMAPS) for voice research and affective computing". In: *IEEE Transactions on Affective Computing* 7.2, pp. 190–202.

Felbo, Bjarke et al. (2017). "Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm". In: *arXiv preprint arXiv:1708.00524*.

Hockenbury, Don H and Sandra E Hockenbury (2010). *Discovering psychology*. Macmillan.

Hu, Minqing and Bing Liu (2004). "Mining and summarizing customer reviews". In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 168–177.

James, William (1894). "Discussion: The physical basis of emotion." In: *Psychological review* 1.5, p. 516.

Jed Wing, Max Kuhn. Contributions from et al. (2018). *caret: Classification and Regression Training*. R package version 6.0-81. URL: https://CRAN.R-project.org/package=caret.

Jockers, Matthew L. (2015). *Syuzhet: Extract Sentiment and Plot Arcs from Text*. URL: https://github.com/mjockers/syuzhet.

Joulin, Armand et al. (2017). "Bag of Tricks for Efficient Text Classification". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Association for Computational Linguistics, pp. 427–431.

Karg, Michelle et al. (2013). "Body movements for affective expression: A survey of automatic recognition and generation". In: *IEEE Transactions on Affective Computing* 4.4, pp. 341–359.

Kearney, Michael Wayne (2018). *textfeatures: Extracts Features from Text*. R package version 0.3.0. URL: https://CRAN.R-project.org/package=textfeatures.

Kong, Liang et al. (2011). "Mining event temporal boundaries from news corpora through evolution phase discovery". In: *International Conference on Web-Age Information Management*. Springer, pp. 554–565.

Kouloumpis, Efthymios, Theresa Wilson, and Johanna D. Moore (2011). "Twitter Sentiment Analysis: The Good the Bad and the OMG!" In: *ICWSM*.

Lange, Carl Georg and William James (1922). *The emotions*. Vol. 1. Williams & Wilkins.

Lazarus, Richard S and Richard S Lazarus (1991). *Emotion and adaptation*. Oxford University Press on Demand.

Lee, Chi-Chun et al. (2015). "Speech in Affective Computing". In: pp. 170–183.

Liu, Bing (2007). *Web data mining: exploring hyperlinks, contents, and usage data*. Springer Science & Business Media.

— (2012). "Sentiment analysis and opinion mining". In: *Synthesis lectures on human language technologies* 5.1, pp. 1–167.

— (2015). *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge University Press.

Luo, Yu et al. (2018). "ARBEE: Towards Automated Recognition of Bodily Expression of Emotion In the Wild". In: *CoRR* abs/1808.09568.

Maas, Andrew L et al. (2011). "Learning word vectors for sentiment analysis". In: *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*. Association for Computational Linguistics, pp. 142–150.

Manning, Christopher D, Christopher D Manning, and Hinrich Schütze (1999). *Foundations of statistical natural language processing*. MIT press.

Mehrabian, Albert (1996). "Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament". In: *Current Psychology* 14.4, pp. 261–292.

Mikolov, Tomas, Kai Chen, et al. (2013). "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781*.

Mikolov, Tomas, Edouard Grave, et al. (2018). "Advances in Pre-Training Distributed Word Representations". In: *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Miyamoto, Yuri, Yukiko Uchida, and Phoebe C Ellsworth (2010). "Culture and mixed emotions: co-occurrence of positive and negative emotions in Japan and the United States." In: *Emotion* 10.3, p. 404.

Mohammad, Saif M (2016). "Sentiment analysis: Detecting valence, emotions, and other affectual states from text". In: *Emotion measurement*. Elsevier, pp. 201–237.

Mohammad, Saif M., Svetlana Kiritchenko, and Xiaodan Zhu (2013). "NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets". In: *Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013)*. Atlanta, Georgia, USA.

Mohammad, Saif M and Peter D Turney (2010). "Emotions evoked by common words and phrases: Using Mechanical Turk to create an emotion lexicon". In: *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*. Association for Computational Linguistics, pp. 26–34.

Mohammad, Saif et al. (2018). "Semeval-2018 task 1: Affect in tweets". In: *Proceedings of The 12th International Workshop on Semantic Evaluation*, pp. 1–17.

Nakov, Preslav, Alan Ritter, et al. (2016). "SemEval-2016 task 4: Sentiment analysis in Twitter". In: *Proceedings of the 10th international workshop on semantic evaluation (semeval-2016)*, pp. 1–18.

Nakov, Preslav, Sara Rosenthal, et al. (2013). "SemEval-2014 Task 9: Sentiment Analysis in Twitter". In: *SemEval@COLING*.

Nielsen, Finn Årup (2011). "A new ANEW: Evaluation of a word list for sentiment analysis in microblogs". In: *arXiv preprint arXiv:1103.2903*.

Pang, Bo, Lillian Lee, et al. (2008). "Opinion mining and sentiment analysis". In: *Foundations and Trends® in Information Retrieval* 2.1–2, pp. 1–135.

Pantic, Maja (2009). "Machine analysis of facial behaviour: Naturalistic and dynamic behaviour". In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 364.1535, pp. 3505–3513.

Patodkar, Vaibhavi N and Sheikh I.R (2016). "Twitter as a Corpus for Sentiment Analysis and Opinion Mining". In:

Plutchik, Robert (1960). "The multifactor-analytic theory of emotion". In: *the Journal of Psychology* 50.1, pp. 153–171.

Portela, Manuel and Carlos Granell-Canut (2017). "A new friend in our smartphone?: observing interactions with chatbots in the search of emotional engagement". In: *Proceedings of the XVIII International Conference on Human Computer Interaction*. ACM, p. 48.

Posner, Jonathan, James A Russell, and Bradley S Peterson (2005). "The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology". In: *Development and psychopathology* 17.3, pp. 715–734.

Radford, Alec, Rafal Jozefowicz, and Ilya Sutskever (2017). "Learning to generate reviews and discovering sentiment". In: *arXiv preprint arXiv:1704.01444*.

Rajadesingan, Ashwin, Reza Zafarani, and Huan Liu (2015). "Sarcasm detection on twitter: A behavioral modeling approach". In: *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*. ACM, pp. 97–106.

Salton, Gerard, Anita Wong, and Chung-Shu Yang (1975). "A vector space model for automatic indexing". In: *Communications of the ACM* 18.11, pp. 613–620.

Schachter, Stanley and Jerome Singer (1962). "Cognitive, social, and physiological determinants of emotional state." In: *Psychological review* 69.5, p. 379.

Schlosberg, Harold (1954). "Three dimensions of emotion." In: *Psychological review* 61.2, p. 81.

Simon, Noah et al. (2011). "Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent". In: *Journal of Statistical Software* 39.5, pp. 1–13. URL: http://www.jstatsoft.org/v39/i05/.

Trevor, Hastie, Tibshirani Robert, and Friedman JH (2009). *The elements of statistical learning: data mining, inference, and prediction*.

Yan, Xiaohan and Jacob Bien (2018a). "Rare Feature Selection in High Dimensions". In: *arXiv preprint arXiv:1803.06675*.

— (2018b). *rare: Linear Model with Tree-Based Lasso Regularization for Rare Features*. R package version 0.1.0. URL: https://github.com/yanxht/rare.

— (2018c). *rare: Linear Model with Tree-Based Lasso Regularization for Rare Features*. R package version 0.1.0. URL: https://github.com/yanxht/rare.

Yildirim, Serdar et al. (2004). "An acoustic study of emotions expressed in speech". In: *Eighth International Conference on Spoken Language Processing*.

Zhang, Zhengyou (2012). "Microsoft kinect sensor and its effect". In: *IEEE multimedia* 19.2, pp. 4–10.