Universiteit Leiden
Opleiding Informatica

Deep learning for Emotional Analysis

Name: Wadie Assal

Date: 30/04/2017

1st supervisor: Dr. Michael S. Lew
2nd supervisor: Dr. Erwin M. Bakker

MASTER'S THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

The detection of emotions in textual data lets us discover more about the writer. The automatic detection of emotions is useful in a large amount of applications. Furthermore, new developments in deep learning have made it effective in more domains. This research combines the two areas and explores a deep learning method for emotional analysis. The method is benchmarked against current methods using the ISEAR and a Twitter dataset. The deep learning method shows an improvement of 2% for precision and 1% for recall and F1 score compared to the current state of the art. Our research shows that the new developments in deep learning have made it viable for emotional analysis research.

Contents

CHAPTER 1

Introduction

Emotions are a crucial aspect of human life. We feel them when we communicate through different communication channels. We furthermore express emotions without realizing it. It shapes our social relationships and influences our decisions. When we write, we express our thoughts and feelings in the written text. When we read, we can often tell which feelings the writer is experiencing.

1.1 Motivation

The detection of emotion in a piece of text can tell us more about the writer. With the rise of the web there is also a rapid growth of emotion-rich textual data. Social media discussions, blog posts and forum threads generate a lot of the data. With this rich source of information about us, there is a growing need to develop techniques to evaluate it automatically. Detecting emotions in textual data is one of the tasks we can solve using this data. Automatically detecting emotions is called *sentiment analysis*, *emotional analysis* or *opinion mining*.

Detecting emotions from text has multiple applications. The following list shows some key applications that are currently being researched:

- *Health*: The ability to the detect emotions in text is useful in detecting depression. Furthermore, patterns in expressing emotion can be a key indicator of certain personality disorders like narcissism. Finally, there is also a need for robots that are aware of the patients emotions.

- *Politics*: Sentiment on blogging platforms and social networks can influence the outcome of public events. Recent studies suggest this connection. In addition, political polls can automatically be generated with the vast amounts of data available on social media.
- *Business*: In business, it is observed that the value of products, stocks or companies is highly affected by rumors and sentiment of the public on social networks and blogging platforms. Businesses naturally became interested in detecting emotions on these platforms because of these reasons. Emotional analysis gives businesses the tool to better understand their customers and react to dramatic changes in the market.
- *Education*: Emotional analysis software can be integrated with automatic tutoring systems to help detect anxiety in response to education material. Students perform better when they are in a relaxed state.
- *Social media*: We can use emotional information that has been detected, to understand the spread of emotions through a social network. This has the possible application of disaster management.
- *Literary Analysis*: There is interest in analyzing large collections of literary texts for emotional patterns. This can for example help us understand the flow of emotions in novels.
- *Human psychology*: Men and woman use other communication styles and emotions to interact socially. Emotional analysis gives us another tool to understand ourselves.

1.2 Tasks

The primary task this thesis is focused on is detecting the sentiment of the writer. We try to extract the emotion the author is expressing. Given a piece of text or sentence, we assign emotional labels such as joy or sadness in an automatic fashion. Agrawal et al. defined the task as follows [1].

Let s be a sentence and ω_s be an emotional label. Let e be a set of m possible emotion categories (excluding neutral) where $e = \{e_1, e_2, \ldots, e_m\}$. The objective is to label ω_s, where $\omega_s \in \{e_1, e_2, \ldots, e_m, \text{neutral}\}$.

However, there are many other tasks in the field of emotional analysis:

- *Detecting sentiment of other entities then the writer.* Currently, most methods focus on detecting the emotion of the writer. However, it is not always the writer, the emotions in the text is about. For example:

  Tom: *Jamie said she was angry.*

Table 1.1: Label examples

Tweet	**Emotion**
Still angry at Japan this morning for publicly attacking Obama at a joint press conference over something he has no control over.	anger
A sad day, you'll never be forgotten. Rip Muhammed Ali	sadness
Everyday all day! We won't be stopped! #GoHawks	joy

Who is angry, Tom or Jamie? Detecting sentiments towards subjects gives us the possibility to learn more about them in the text. We can also use this extra information to filter out emotions that are not about the writer.

- *Detecting sentiment towards a target.* A piece of text or a sentence can express sentiment, to multiple subjects. For example:

  Tom: *I really like the show House of Cards, but I dislike House.*

  What does Tom like? A system that could deduce what Tom likes will be useful in multiple domains. Businesses can get a more detailed impression of what customers like or dislike. For example, buyers can be very positive about the product but disliking the service.

1.3 Challenges

Most current research has been focusing on the polarity of text. Only a few efforts have been made to classify text into different emotions. This is mainly because polarity is an easier classification problem. The current methods give better results on these problems. The abundance of data sets labeled with polarity also contributes to the better performance. However, bringing emotional analysis to the same level of performance as sentiment analysis can give us much better understanding of human thought.

Emotional analysis currently faces the following main challenges:

- *Language complexity*: Language is complex. The meaning of the sentence is not just the sum of its words. Emotions are often not clearly stated. The complexities of language are a big challenge to machine learning algorithms.
- *Creative and non-standard language*: Automatic language systems have difficulty coping with creative and non-standard language. This also includes misspellings and creative abbreviations. The biggest source of data for emotional analysis is

from social media. However, social media is known for misspellings, creative use of words, hashtags and abbreviations.

- *Large datasets*: Machine learning algorithms require large amounts of training data for good performance. This data also needs to be labeled. Currently, this requires a lot of human effort. Because of this challenge, most of the work revolves around a handful of datasets.
- *Para-Linguistic Information*: When we communicate, we use subtle changes in our voice tone and pitch. For example, the same sentence is expressed differently, when we are mad then when we are happy. This information is lost in written text. The same also applies to facial expressions.
- *Cross-Cultural Differences*: How emotions are expressed can vary from culture to culture. This can be difficult even for humans. Creating systems that work for all cultures is a challenge.

1.4 Thesis structure

We start the thesis by first exploring the field of affective computing in chapter 2. In chapter 3 we will dive deeper into the field of emotional analysis. We will follow this up with an overview of current research into the field in chapter 4. Our method is presented in chapter 5 after which we will give a description of our experiments and present our results in chapter 6 and 7 respectively.

CHAPTER 2

Affective computing

The field of emotional analysis falls within the broader field of affective computing. Affect is the process of experiencing feelings and emotions. Affective computing is trying to give computers the ability to feel and recognize emotions. It represents a interdisciplinary research field on different topics, including: human computer interaction, sensing and recognizing emotions, affective user modeling and models of emotion. In this chapter we will explore this broader field of affective computing.

2.1 Overview

An influential paper by Roselin Picard [43], spurred the growth in affective computing we see today. Some consider this paper the start of affective computing in the modern era. In the paper Rosalind Picard discussed the key applications and benefits of creating an affective system. Rosalind Picard argues that if we want to make computers truly intelligent and work with us, we need to add the ability to feel and recognize emotions to computers.

2.1.1 Affective systems

If we enable computer system to recognize and express affect, what type of systems could we possibly get? In the paper, affective computing [43], Rosalind Picard present four possible systems.

- The first and most obvious system. Today most computers fall in this category. It is a system that is oblivious of the users affect an is incapable of expressing affect. These type of systems are unfriendly to the user because they lack the understanding of affect we humans have while communicating.
- The following category of systems tries to build a system that expresses affect naturally. It will be able to better express information to the user. This could be achieved by a natural voice or a realistic facial expressions.
- In this category of systems we build systems that are able to perceive affect. This would make computer systems possible that are better teachers or assistants because of the feedback they receive from the human user.
- Building a computer that can express and perceive affect maximizes the communication between human and computer. Such a system could be truly user friendly and personal. This is the last and optimal category.

2.1.2 Detecting emotions

Before a computer system can detect emotions, the computer system needs information to work with. We could possibly use the same information humans use to detect emotions in other humans. One of the most important ways of detecting emotion in humans, by humans, is sentic modulation. Sentic modulation is the physical way by which a emotion is expressed. All human emotions are also expressed with our bodies.

Because of the physical aspect of expressing emotion, we have the following sources of information we can use to detect emotions.

- *Facial affect detection*: The most widely known form of sentic modulation. Facial expression are a clear and easy way for humans to recognize emotions. Using video and image data we can build systems that recognize emotions through facial expression. Most current systems rely on machine learning to enable recognition of emotions.
- *Vocal intonation*: The tone of speech is a second example of a source of emotional information that is easily recognizable for humans. It has been shown that even young children can recognize emotions through vocal intonations [53]. While they will not understand what has been said, this is also the case for dogs. Current systems analyze digital audio with machine learning technique to recognize emotion through tone.
- *Body gestures*: Humans use body language to share all kinds of information. This also includes affect. Lifting shoulders could for example signal that the person is indifferent to the situation. Monitoring these action can provide more emotional information.

- *Physiological monitoring*: Other sources of information, while not easily recognizable for humans, are heart rate, respiration, skin conductance, and temperature. Through the use of sensors we can make this information accessible to computer systems.

Computer systems can use multiple source to help with recognizing emotions [49]. For emotional analysis we use text as an indirect source of information to detect emotions.

2.1.3 Expressing emotions

Current computer systems are completely unaware of human emotions. They can not sense it or express it. But if we wanted to build an emotion aware system, how would we let the computer express emotions? Humans should be able to clearly recognize the emotions. It has been shown that expressing emotions by computer systems can be done without giving computers human like faces [42]. A system that can move closer to a user after detecting a negative affect can give users comfort and it will seem like the system is expressing emotion. While computer systems do not need human like faces to express emotions, efforts have been made for building such a system. An example of such a system that came out of MIT's media lab is Kismet [9].

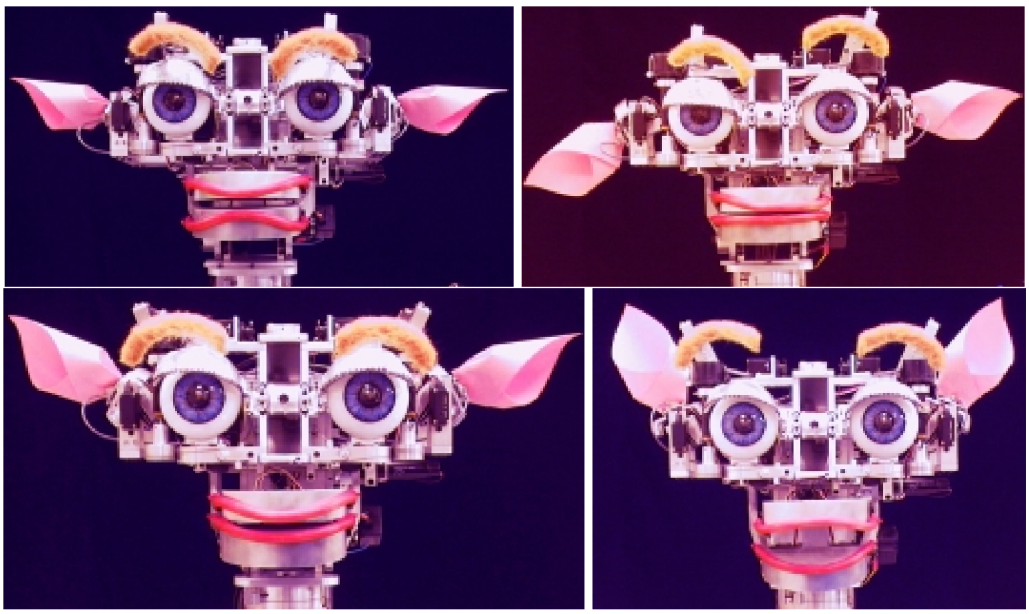

Figure 2.1: Kismet, with a number of facial expressions [10]

2.1.4 Applications

Affect plays a key role in human interaction. Understanding affect and embedding the recognition of affect in a computer system allows us to improve interactions between man-made systems and humans. The following applications are expected to be developed by solving the problems in affective computing.

- *Entertainment*: Sensing how an audience reacts to entertainment can enable interactions between the entertainment system and the audience. A game system that is aware of the mood of the player could for example, adjust the music in the game to lift the player's mood. In a shooting game where the player should be excited and focused we could sense that the player is bored. In this case, we can add opponents to the game to add to the excitement.
- *Expression*: Affect is an important tool in human expression and communication. When we remove affect between humans, miss-communications can occur. This, for example, occurs when we try to communicate trough mail because we can not hear what the tone of the message is. If we do hear the tone but we are missing facial expressions, the same miss-communications can occur. Computers that can not show affect will have the same communication problems with the user. By giving computers the ability to feel and recognize emotions we can improve the quality of human-computer communication. This opens up a new world for applications.
- *Film/Video*: Using affective computing we could help filmmakers in retrieving and editing movie scenes. Instead of tirelessly commenting on movie clips shot by filmmakers we can let a computer annotate the interesting shots by looking at the reaction of humans to the video. While it will not give the film editors a magic retrieval tool, it will help finding the rights scene.
- *Environments*: Human's often change their environments based on their mood. For example, we light candles if we are in a romantic mood. With affective computing we could let the room react to the people inside it creating an environment that is alive. We can build buildings that learns from the affect of users that reside in them, dynamically adapting the rooms based on the user's feedback. This creates a pleasant working and living environment possibly making us happier and more productive.
- *Aesthetic pleasure*: Aesthetics are a difficult to measure characteristic we humans perceive. However, it is part of our perception of the world. Affective computing could possibly help us understand aesthetics and build a system that has an understanding of this. Imagine an image or video retrieval system. Looking for a picture with the system we find thousands of suitable images. How can we let the computer further help us with finding the right picture? Teaching the computer our personal taste or general aesthetics we can further improve the retrieval system. For example, find all the red boats that I like, can be a possible search query in the future.

- *Affective wearable computers*: Wearable computers that can measure mood will create new research opportunities and enable new health applications. Medical studies could measure the mood of patients in the real world. This would, for example, be beneficial to psychotherapist in getting a better understanding of diseases like depression. We can use the wearables to detects stress and anxiety with a range of applications. Preventing stress in the workplace or alleviating anxiety during exams in schools. Connecting the devices to medical alert services improves the care that health services can provide. The applications are however limited by how much the user wants to share about their mood. Furthermore, while there are a number of theories of emotion, our understanding is limited by the lack of real world data. Affective computing could provide us with a wealth of real word data, enabling us to understand human emotions better.

While affective computing is much broader than sentiment and emotional analysis. The technologies supports these applications because it enables the computer to recognize the emotions of human users.

2.2 Human emotions

Animals are in a constant battle with a continuous source of dangers. They need to eat enough, not become dehydrated, overheat or freeze. Furthermore, predators are always looking for there next meal. Survival is hard, but animals are well adapted to survive these dangers through their behavior. When, how en to whom, the behavior is directed is decided by motivation. Animals can have multiple motivations that's result in there observed behavior. Scientist are constantly discovering new motivations in animals that results in certain behavior. For example, to insure survival, animals must keep certain parameters within a range. Emotions are one of those motivators that ensure survival.

What defines an emotion is mostly a philosophical question. However, emotions can be described as a strong feeling that arises spontaneously in response to one's situation. Emotions drive much of our behavior. It can make humans act irresponsible and illogical. Understanding emotions is essential to understanding humans. In the book The Emotion Machine [33], Minsky argues that emotions are different modes of thinking. It enables us to increase our intelligence. Minsky challenges the difference between emotions and normal thinking. When we have a problem or get in a certain situation our brains selects the correct way of thinking (emotion) to deal with the problem. However, Minsky fails to connect his theory to physical processes in the brain.

Emotion are an important source of motivation for humans. They are central in determining reactions to situations. For example, Fridja [16] theorizes that positive emotions are triggered by events to enhance one's survival. Negative emotions are triggered by painful or threatening situations. It increases the motivation to prevent the situations from unfolding in an unpleasant way. The emotions motivate humans to fulfill certain

needs. Our mind pushes us to certain situations and pulls us away from others through the use of emotions.

Furthermore, voice, face, gestures and posture are an important way of communicating emotions to other humans [26]. By doing this, we let other people know how we feel and influence their behaviors. It is suggested by Darwin that physically signaling emotions was selected for evolutionary. Often the outcome of a act is depended on the emotions of others. Because, when we signal our emotions we influence the other person.

2.3 Models of emotion

How emotions are represented, is the first question that needs to be answered when building a system based on emotions. For this reason we need to use a model that describes human emotion. Human emotions have been researched from two viewpoints. Are emotions discrete and clearly separate states? Or can emotions be defined on a dimensional basis? We will explore the viewpoint in the following paragraphs.

2.3.1 Categorical models

The discrete or categorical model describes a finite set of discrete basic emotions. Humans seem to be capable of recognizing the different emotions easily. This suggests a discrete set of human emotions [12]. How these emotions are defined differs from each method. In the following paragraphs, we present the main dimensional models used in emotional analysis.

Ekman's six

The Ekman model [14] for human emotions is a widely used categorical model and also one of the least complex. Basic emotions described by Eckman are joy, fear, anger, surprise, and disgust. The emotions are tied to human facial expressions. Because there are no large evolutionary differences between human populations, the emotions are universal.

Paul Ekman argues that humans have a set of basic emotions, and reject that a scale from pleasant to unpleasant is enough to describe human emotions. Ekman describes basic emotions as emotions that differ from other emotions in important ways. Fear, anger, and disgust are all negative emotions. However, they differ from each other for example in their

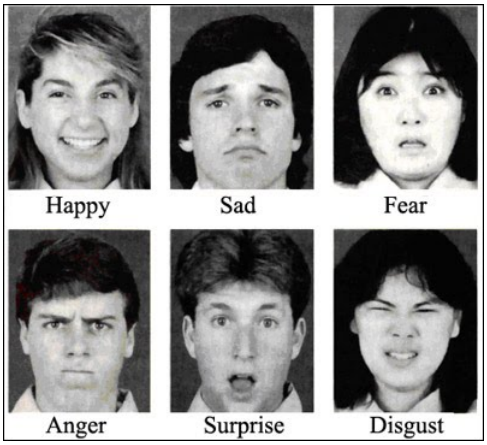

Figure 2.2: Ekman's six [15]

behavioral response and physiology. If emotions have evolved to deal with fundamental life task, Ekman argues, then it is logical to expect that in the contexts in which emotions occur some common elements are shared [13].

Tomkins model

Silvan Tomkins argues that there are nine, and only nine, discrete human affects [52]. The affects are interest-excitement, enjoyment-joy, surprise-startle, distress-anguish, anger-rage, fear-terror, shame-humiliation, dissmell-disgust. From these nine affects all other human emotions arise. Tomkins also defined the effects in pairs, where the first pair defines the milder variant of the affect [39].

2.3.2 Dimensional models

When describing emotions in the dimensional model, we create an emotional space where each human emotion can lie in. Because of this, the dimensional model also makes a distinction between basic emotions and secondary (complex) emotions. Secondary emotions arise when there is an overlap between the emotions in the emotional space.

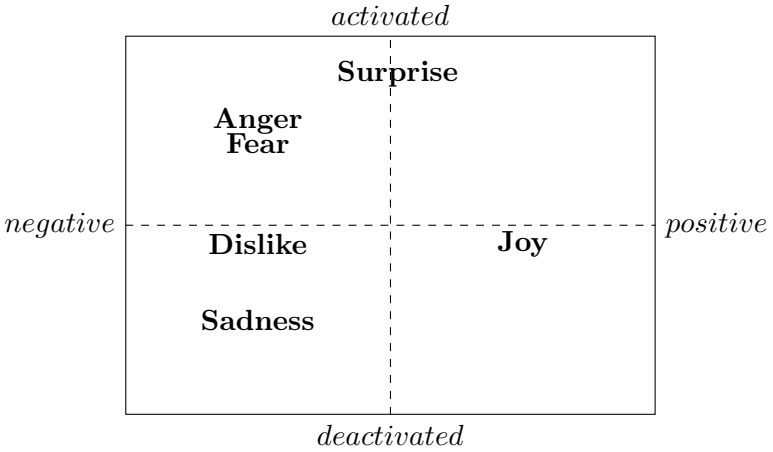

Figure 2.3: Russell's space

Russell's space

Russell's space [47] is a two-dimensional model of emotions, where emotions are described by a space of two dimensions. In this model one dimension describes polarity

and the other space describes activation. When placing an emotion in this space, we describe the emotion with positive, or negative, in the polarity dimension, and activated or deactivated in the activation dimension. For example, anger would be a negative feeling and happiness a positive feeling, while activation describes the intensity of the emotion. Furthermore, this approach creates the regions positive-activated, positive-deactivated, negative-activated and negative deactivated within the space. The following figure shows how some emotions lie within this two-dimensional space.

In this thesis, we use the term sentiment analysis or opinion mining to refer to the automatic detection of polarity as defined by Russell's space. The term, emotional analysis is utilized for the automatic detection of an affective state.

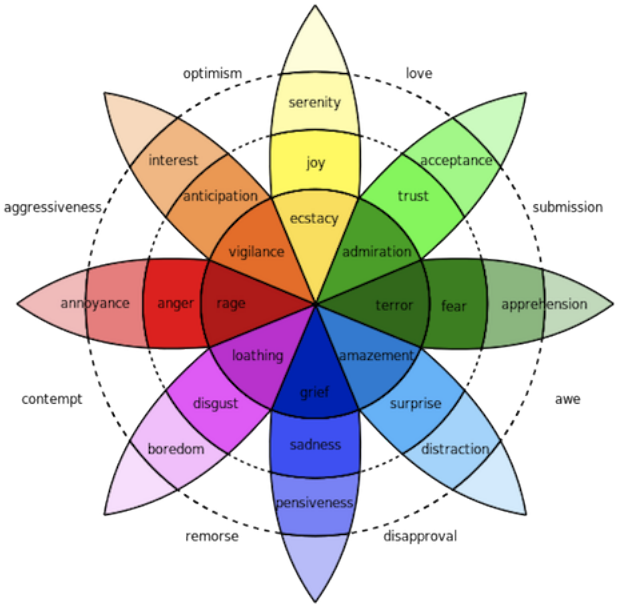

Figure 2.4: Plutchik's eight [44]

Plutchik's wheel of emotions

A popular three-dimensional model is the Plutchik's wheel of emotions [44]. Plutchik's model is often used in computer science for affective human-computer interaction or sentiment analysis. The eight basic emotions described by plutchick, are joy, fear, anger, surprise, disgust, trust, and anticipation.

The model Plutchik created is a hybrid of the dimensional model and basic-complex categorical model. The diagram in figure 2.4 shows how these emotions can be arranged on a wheel so that opposite emotions are also opposite on the wheel. Similar emotions

are placed close together on the wheel. The diagram also shows us that secondary emotions are a combination of basic emotions. We can read the model as a color wheel. All emotions are a mixture of the basic emotions just like all colors are a mixture of the primary colors.

Vector model

A two dimensional model of emotion developed in 1992 by Bradley et al. [7]. Based on the research of that time period they argue, that a two-dimensional model best describes human emotions. The model theorizes that there is an underlying arousal dimension. Valance is shown by two vectors that point in a positive and negative direction starting at zero arousal.

Lovheim cube of emotion

Based on neurotransmitters, Lovheim proposses a three-dimensional model of emotion [28]. It is argued that the levels of dopamine, noradrenaline and serotonin can model the human emotions. Dopamine, noradrenaline and serotonin levels are therefore the three dimensions in the model. The three dimensions are modeled by a cube. The eight basic emotions as defined by Tomkin, are placed in the corner of the cube. A further description of the eight basic emotions can be found in section 2.3.1.

PANA model

A model with two-dimension by Watson et al. [55]. PANA is acronym for positive activation and negative activation. The unique aspect of the model is positive activation and negative activation are separate dimensions that can be measured independently from each other. For example, an emotion can be described by a high positive activation and a high negative activation.

PAD emotional state

Another model of James Russell developed in collaboration with Albert Mehrabian models human emotion in three dimensions [31]. The dimensions are, dominance, arousal and pleasure. Intensity of the emotions is described by arousal. Pleasure describes like the name suggest, how pleasurable an emotion is. Dominance describes how dominant the emotion is experienced by others. For example, anger is a dominant emotion while fear is submissive.

CHAPTER 3

Emotional analysis

This chapter will take you through the steps needed to create an Emotional analysis system. There is currently no system that performs well on all emotional data. For this reason, we create a system for a particular kind of data. Textual data retrieved from books is different in many ways from Twitter data. Books are ordered, and the spelling is predictable. Twitter data is chaotic, and the language unpredictable. In general, when making such a system, we follow the following steps:

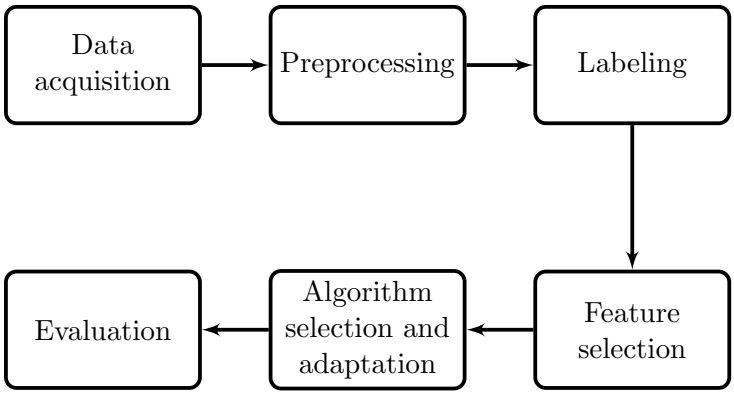

Figure 3.1: Emotional analysis workflow

3.1 Preparation

Before we can build the emotional analysis system, some preliminary steps need to be taken. Emotional analysis is a multifaceted problem that is solved in multiple steps. The following sections describe the preparation that is needed before we can start with the technical work.

Data acquisition

Data, the basis of every emotional analysis system. The data can be acquired, or existing data sets can be used. Currently, user generated data on the World Wide Web is the main source of data. When collecting data from the internet, there two approaches that can be taken:

- *API-based*: The easier approach from a programmers perspective is accessing the data through an API. Currently, a lot of social network's offer an API service from where the data can easily be accessed programmatically. An example would be the Twitter API that is offered in two flavors. The regular API for small en medium data needs and the Twitter Firehose API for applications with large data needs. The Twitter firehose API requires special permissions before access.
- *Webcrawler-based*: The second, and harder approach. Using a web crawler we collect HTML pages with the information needed. We are then able to parse the information with, for example, regular expressions, CSS selectors or XPath to extract the information. While this method works, the information obtained is not as clean as through an API. Extra steps are advised to check the information collected.

Preprocessing

Data is often too chaotic to be used by emotional systems. We can bring some order to the chaos by taking a preprocessing step. Data for example can have fields with null values or inconsistencies. Removing this invalid data gives us better performance in the final system. In the field of emotional analysis we can also choose to remove characters like punctuation marks. However, care should be taken so useful data is not removed. Remove too much, and you will have less features to work with. This results in a system that does not achieve the results that could have been achieved.

Labeling

Detecting emotion in text data is modeled as a classification problem where labels are assigned from a set of emotion labels. To evaluate prediction performance of emotional

analysis systems, we need data that is labeled with the correct categories. In systems that learn by example, this data can also be used to train the system.

Most datasets are pre-labeled and only need some adjustments. For example, we can convert the dataset from seven emotions to two emotions by removing the excessive labels. If the data is self-collected, we need to annotate the dataset ourselves.

To avoid bias, we need to annotate the dataset using multiple persons. Using the multiple annotations, we can discard weak annotators. For example, we can identify weak annotators if the agreement probability is more than two deviations from the mean. Furthermore, when setting up a questionnaire for the dataset, give the annotators a non-option. This prevents wrong data entering the dataset because annotators must provide a choice.

If not enough suitable annotators are present, crowd-sourcing can be used to supply or supplement the annotators. For example, Amazons Mechanical Turk and CrowdFlower can be used to find annotators for a small payment.

After collecting all the annotations and filtering out the weak ones, we can finalize the dataset. We label the data if we have a strong majority from the annotators. For example, data D is only labeled with label L, only if more than half of the annotates chose label L.

3.2 Feature selection

The first step in analyzing the data is feature extraction. The Emotional analysis system is going to predict the emotions based on these features. Choosing bad features will get you bad results further up the pipeline. You can see this step as laying the foundation. We, therefore, leverage features in the text, for a better performing system for to specific data we choose. For this, we first need to select the features.

- *Terms presence and frequency*: Which words are present and how often? We do a simple count of words or word n-gram.
- *Emotion words and phrases*: Words that express emotion are of course an indicator of that emotion being expressed. Examples are words like joy, happiness and furious. However, some phrases expressed emotional without using emotion words.
- *Negations*: Negation within a sentence can change the meaning. Detecting negations can improve accuracy.
- *Data specific features*: Social media has additional information like emoticons that express information. This information can also be used as a feature for detecting emotions.

3.3 Methods

Different approaches have been used to build emotional analysis systems. We will give an overview of the main methods that have been used successfully. An overview of common methods and their relation can be found in figure 3.2.

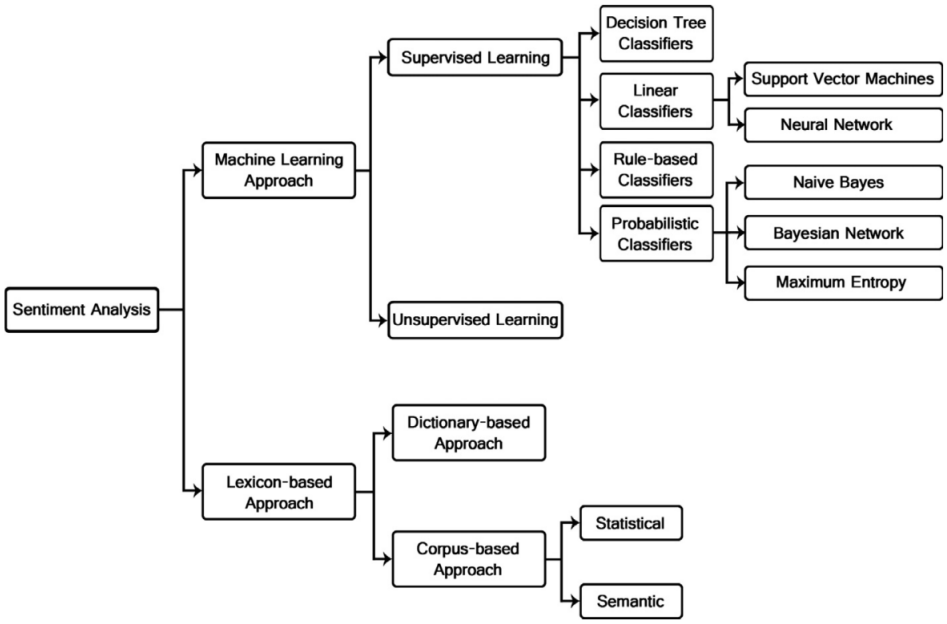

Figure 3.2: Classification techniques [30]

3.3.1 Machine learning approach

With the success of machine learning in other fields, it also got the attention in the field of emotional analysis. Currently, machine learning is the dominant method for recognizing emotions. The methods used in machine learning can be roughly divided into supervised and unsupervised methods.

We generally use supervised methods when we have a large set of labeled data. Unsupervised methods can be used when such data does not exist. This thesis will focus on the supervised methods. Some commonly used machine learning algorithms in emotional analysis will also be presented in this thesis.

Naive Bayes

Naive Bayes is a simple and fast machine learning algorithm that works well in a lot of real world applications. It is often used in sentiment and emotional analysis. With this method we calculate the probability of each word in the document or sentence of having a specific label. Too calculate the probability of the complete document or sentence we simply multiply the probabilities of all the words in the document or sentence.

The Bayesian classifier is often used because there is only a small amount of computing power needed, compared to the other method. It is a good baseline method, that is good enough for practical use.

Decision Tree

This classifier uses a tree in which the inner nodes have features as labels. It can classify data by starting at the root node and move through the inner nodes by checking which of the features are satisfied by the data. It will continue till it reaches a leave node. The data is then classified by the label of that leave node. In other words, if we have a feature F, we can make decisions to reach the right classification by comparing it to threshold T. If a comparison holds, we follow that edge to the next node. The final node holds a label that classifies the data.

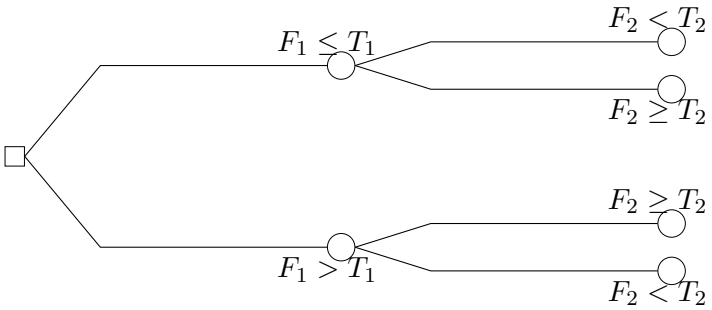

Figure 3.3: Decision Tree

Ada Boost

A classifier of the ensemble class. This classifier takes a completely different approach by leveraging weak learners to create a good performing classifier. It combines the strengths of the different type of learners. A large number of different learners can be combined with this method. The final result of the weak learners is combined with a weighted sum to get a right classification.

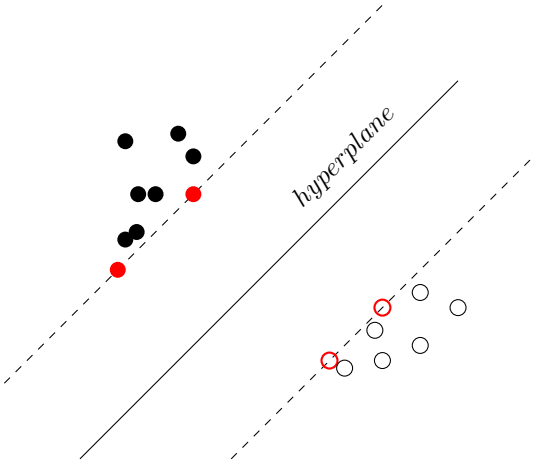

Figure 3.4: Support Vector Machine

Support vector machine

Support vector machines have been used extensively with a lot of success. It is an easy to use classifier that gives good results with a minimal amount of computational power. The classifier operates by generating a hyperplane with the maximum euclidean distance to the training data. When tasked with classifying new data, the hyperplane, dictates how the data is labeled.

In other words, training data is mapped to points in a multi-dimensional space based on its features. Based on this data a division is made so we get regions within the multi-dimensional space. Each region is labeled with the different classes. When new data is mapped to this space, the region dictates the label the data will receive.

The original support vector machine by Vapnik is a linear classifier [6], however an extension was made by Aizerman et al. [2] to enable a non-linear support vector machine by using the kernel trick. Data that is linearly non-separable in the current dimensions may be linearly separable in a higher-dimensional space. While changing the current data representation to a higher-dimensional space is possible, it will result in a higher burden for the support vector machine. However, by using a kernel function we can calculate the dot products of a higher-dimensional space, without building the higher-dimensional representation. This enables us to work computationally efficient in a higher-dimensional space and receive better prediction performance from the support vector machine.

Neural Network

While promising in different fields, neural network have not been used extensively within the domain of sentiment analysis. Neural networks are a family of different models loosely based on how animal neurons function. Because there is no formal definition of a

neural network, we will discuss the most common neural network, the single hidden-layer back-propagation network or the single-layer perception. We will further expand on this explanation with multiple hidden layers or the multi-layer perceptron (MLP).

Neural networks and especially neural networks with multiple hidden-layers have gotten a lot of attention in recent years. They have been presented as a mysterious device that can solve scientific and business problems easily. However, there is nothing magical about neural networks. Neural networks are a non-linear statistical model similar to a non-linear support vector machine.

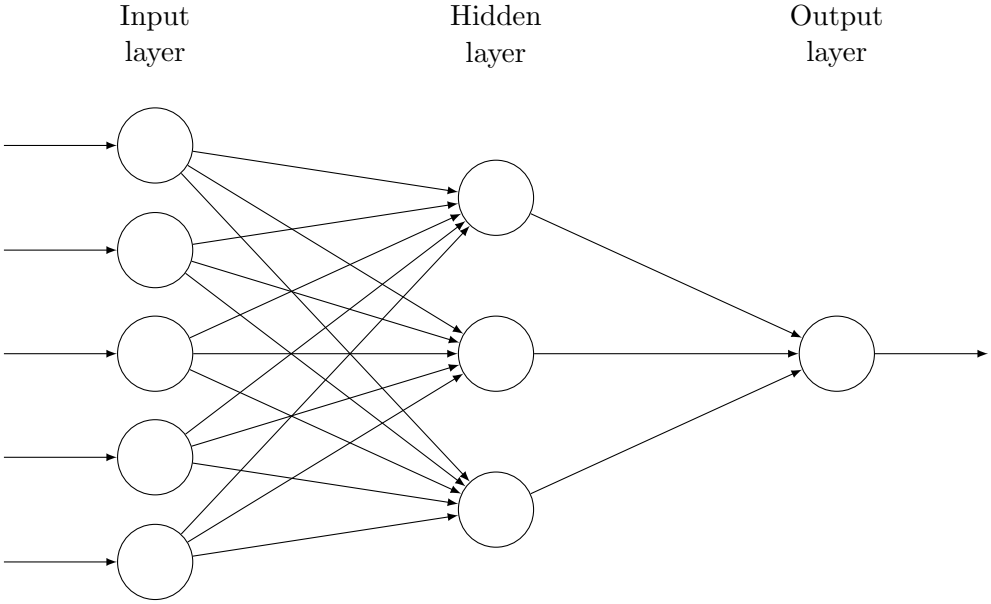

Figure 3.5: Neural network

A single hidden layer network as seen in figure 3.5, is composed of three layers. The first layer, the input layer receives the input values after which the output layer outputs the desired value. The hidden layer is placed between the input and the output layer. Each layer is connected with the next layer and only with the next layer. The connections are made by connecting the neurons in the different layers.

The key components of a neural network are the artificial neurons. Based on inputs received from the incoming connections a neuron can fire on the outgoing connections. The firing is decided on by a firing rule. For example, consider the neuron in figure 3.6 with three incoming connections. If we have a firing rule that fires if all incoming connections have input, we are responding to specific patterns of input. This makes the firing rule a powerful concept in neural networks. Working with these components, we can create a complex network that has a specific output with certain inputs. However,

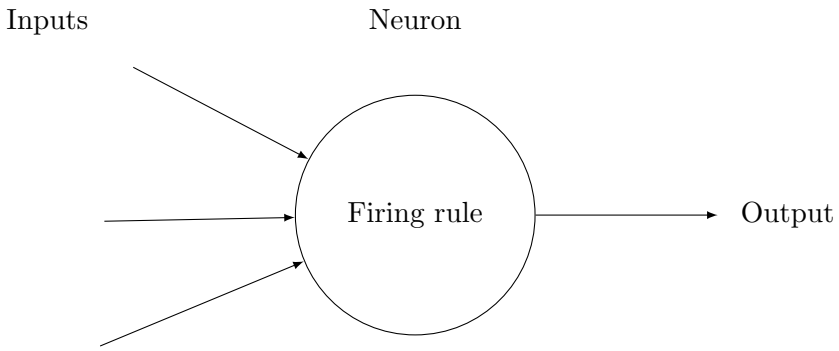

Figure 3.6: Neuron

we are not doing anything special that computer can not already do. Adding weights to the connections makes the neural network much more powerful and flexible. The neurons now only fire if the inputs adjusted by the weight exceed a certain threshold T. This makes the neural network adaptable to certain situation by adjusting the weights. For example, if we have inputs I_1, I_2 and I_3 with weights W_1, W_2 and W_3 respectively. The neuron will only fire if it satisfies the following firing rule:

$$I1 \cdot W1 + I2 \cdot W2 + I3 \cdot W3 > T \tag{3.1}$$

After building a suitable neural network architecture, we can now teach the neural network to perform some task by adjusting the weights. We must adjust the weights of the neural network in such a way that we achieve an output as close as possible to the desired output. A common method for achieving this, is the backpropagation algorithm [46]. The backpropagation learning algorithm is the key step that enables neural network to learn [11]. In simple terms, backpropogation runs the neural network in reverse to go from desired output to the current input by optimizing the weights.

For any set of input data, there is an error magnitude that can be described by an error function. By adjusting the weights and finding the global minimum in the error function we teach the network by showing it the new data. The weight are optimized using an optimization method like gradient descent. The algorithm can be run for multiple training iterations and we can combine this with weight decay. This means that after each update the weights are multiplied by a value smaller then one to prevent the weights from becoming too large. This prevents overfitting, which is further explained in section 3.3.1.

Furthermore, instead of outputting discrete values with the neurons like 0 and 1, we can also output between 0 and 1. The firing rule of a neuron can be an activation function that outputs 0 and 1 or a transfer function that will output a value within a range. An example of transfer functions is the sigmoid and softmax functions. With this adjustment, we can now also solve non-linear problems. We can further improve the non-linear neural network by adding multiple layers to the architecture. This adds more levels of abstraction that cannot be as simply contained within a single hidden layer network. It has been proven by Hornik et. al that a non-linear neural network with at least one hidden layer can approximate any function [20]. However, a neural network with an activation function is more limited [34].

Overfitting

While machine learning algorithms are extremely powerful and flexible, they all suffer from one weakness. Machine learners are pattern recognition machine's just like humans. And just like humans can see faces and animal shapes in clouds that are not there, the machine learning algorithms can see patterns in insignificant details.

This problem occurs when we train our model in such a way that it fits the training data perfectly. Insignificant details in the training data become important to the model. While it can make perfect predictions on the training data, the model does not generalize. While building a machine learning model, we should take precautions to prevent overfitting on the training data. A commonly used method is cross-validation. This method is further explained in paragraph 3.4. A second approach that helps with overfitting is regularization. Regularization discourages complex representations of the data, as it is assumed that it will not generalize well.

Vanishing and exploding gradient

The error function is essential in finding the correct weights through an optimization method like gradient descent. However, in some situations the gradient is extremely small and this makes gradient descent progress slow. This is called a vanishing gradient and as a result the network becomes much harder to train. When the opposite happens, we call this an exploding gradient. The gradient changes abruptly, resembling a cliff. We can see an example of an exploding gradient in figure 3.7. With an exploding gradient even a small gradient update step can move the parameters of the slope losing all progress.

The vanishing and exploding gradient limits the amount of hidden layers we can add to the network, because it makes the deep neural network hard to train effectively. The vanishing and exploding gradients problem is primarily caused by sigmoidal activation functions [17].

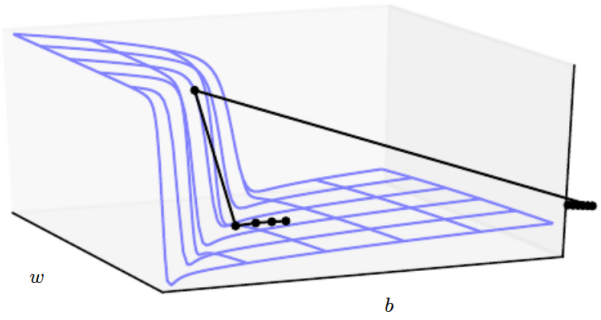

Figure 3.7: Example of an exploding gradient with a misstep by a gradient descent update [19]

3.3.2 Lexicon-based approach

Lexical analysis takes a rules-based approach to analyzing the text. It depends on finding a lexicon that contains a motion or lexicon that relates to that emotion. An example of a lexicon can be found in table 3.1.

Table 3.1: Emotions with lexicon

Emotion	**Lexicon**
anger	mad, furious, outraged
sadness	unhappy, depressed, negative
joy	happy, good, play, positive

There are three main methods for collecting the lexicon.

- *Manual approach*: Very time consuming and generally not used. If used, it is combined with some other technique. During this approach, we manually collect the lexicons or phrases that are meaningful to us.
- *Dictionary-based approach*: During this approach, a small set of lexicons is collected with known emotions. We then expand this list by looking at corpora for synonyms and antonyms. These words become part of the first list. Having a bigger list we now repeat the process multiple times until no new words are found. We now have an extensive list that can be used to analyze our text.
- *Corpus-based approach*: The corpus-based approach is based on phrases and helps to solve the problem of context. Words by themselves often do not capture the complete meaning. Using phrases we can also capture negations that completely turn around the meaning.

A commonly used lexicon for emotional analysis is the Wordnet-Affect lexicon [51]. Wordnet-Affect is an extension to Wordnet domains lexical resource with emotional information.

3.3.3 Hybrid methods

The hybrid approach takes the best of the two methods. Combining Machine learning and lexical analysis often gives the best result. This method can be implemented as a pipeline where lexical information is fed into a machine learning algorithm as features in a feature vector. Another approach is by building an ensemble of algorithms that work together in some way. We can, for example, let the different algorithms vote about an outcome. The outcome depends on various voting rules we can set. For example, we could implement a majority vote or a weighted vote where some algorithms have more voting power.

3.4 Evaluation

Before developing the system, a split is made in the datasets and divided into a training set and a test set. The test set is used for the final evaluation where as the training set is used during development of the classifier. The training set is also used for evaluating the performance during development. To prevent over-fitting on the training set, cross-validation is used.

Cross-validation is a technique to validate an approach based only on the training data [24]. It is used to predict the performance on the test set and other unseen data. We can prevent over-fitting because we do not need the test data during development. During cross-validation, we partition the data in subsets, with one subset used for training and another set used for validation. To reduce variance, multiple rounds of cross-validations are performed. An often used cross-validation strategy is k-fold cross-validation. With this approach, the data is partitioned in k sub-samples of equal size. One sub-sample is kept for validation and the rest of the k - 1 sub-samples are used for training. The validations are then performed k times, also called a fold, with every sub-sample used once as validation data.

To evaluate classification performance, precision, recall, and F1-score are used as metrics. The metrics are based on the number of true positives (T_p), false positives (F_p) and the number of false negatives (F_n). T_p are the number of items classified correctly. F_p are the number invalid items that are misclassified. F_n are the number valid items that are misclassified.

Precision (P), is the number of true positives, divided by the number of true positives plus the number of false positives. We use the metric to show the share of all the positive classification, that were made correctly.

$$P = \frac{T_p}{T_p + F_p} \tag{3.2}$$

Recall (R), is the number of true positives divided by the number of true positives plus the number of false negatives. Recall enables us to show the share of positive items in the dataset that were classified correctly.

$$R = \frac{T_p}{T_p + F_n} \tag{3.3}$$

F1-score (F_1), is defined as the harmonic mean of precision and recall. This enables a combined evaluation score of precision and recall.

$$F_1 = 2\frac{P \times R}{P + R} \tag{3.4}$$

The averages are calculated with the original scores before rounding. This prevents loss of precision in the averaging process.

CHAPTER 4

Related Work

This chapter will cover the work done on the task of detecting emotions. The chapter is organized into two sections. In the first section, we will discuss available datasets and their use. The second section will cover current methods and give some history.

Datasets

The lack of large publicly available datasets that are annotated with emotional information is holding back the field. With the lack of large datasets most work centers around ISEAR, SemEval, and self-collected Twitter datasets. The following list gives an overview of the currently used datasets in emotional analysis.

- *SemEval dataset*: SemEval dataset or also known as Text Affect dataset, is a collection of news titles acquired from new's websites like CNN, Google News and newspapers. The dataset is labeled with Anger, Disgust, Fear, Joy, Sadness, Surprise. [50]
- *Fairy tailes dataset*: The fairy tales dataset or also known as the Alm's Dataset is a dataset of sentences from fairy tails labeled with emotion. The dataset uses the Ekmans list of basic emotions and is labeled with happy, fearful, sad, surprised and angry-disgusted.
- *ISEAR databank*: A dataset collected from a big survey, by psychologist. 3000 students were asked about situations where they experienced joy, fear, anger, sadness, disgust, shame, or guilt. [32]

- *Twitter dataset*: Because of the lack of datasets, researchers often collect their own Twitter dataset. This is because of the ease of which the data can be collected through the Twitter API and the availability of relevant information. An example of this approach would be Saif et al. [36] However, because the datasets are not released, it is harder to compare the results between different studies. Furthermore, Saif [35] shows us that tweets, self-labeled with emotions through hashtags are consistent. That makes it possible to collect a twitter dataset for emotional analysis based on the self-labeled tweets.

Table 4.1: Dataset samples

Dataset	**Labelled with Sad**
SemEval	Bangladesh ferry sink, 15 dead.
ISEAR	When I left a man in whom I really believed.
Fairy tales	The flower could not, as on the pre-talesvious evening, fold up its petals and sleep; it dropped sorrowfully.

Methods

Emotional analysis started as a specialized case in sentiment analysis. However, it is now an active research area. The performance was low because of the small datasets and the larger classification problem. Multiple methods were developed, but evaluated on different datasets.

Machine learning methods like Support Vector Machines were also widely applied for emotional analysis [3, 27, 57]. This was successfully done as recently as 2015 by Saif et al. [37]. However, because of the different datasets, the approaches are not easily comparable.

A standard was set by Kim et al. in 2010 [23] when an evaluation was made of the different approaches on the ISEAR, Fairy tales, and SemEval datasets. The evaluation was performed with anger, fear, joy and sadness as emotional classification labels. The most successful methods described, use a Vector Space Model representation (VSM) in combination with the Wordnet-Affect repository. Using a dimensionality reduction method for VSM, the compute time and noise is reduced which enables better performance. To enable the final classification, cosine similarity is used to calculate the similarity to each emotional vector. Using a neutral label with a predetermined similarity threshold t, uncertain classifications are excluded for better performance. The classification result (CR) between input text I, and emotional class E_j is calculated as follows:

$$CR(I) = \begin{cases} arg\ max\Big((sim(I, E_j)\Big) & \text{if } sim(I, E_j) \geq t \\ \text{"neutral"} & \text{if } sim(I, E_j) < t \end{cases} \tag{4.1}$$

The best performing dimension reduction method is non-negative Matrix Factorization (NMF) with an average F-score of *0.468*. Dimensional estimation is the second approach evaluated by the research. It is built on emotional ratings (ANEW) where subjects reported their emotions in a three-dimensional representation [8]. Valence, arousal, and dominance are the dimensions used. This is also called the Valance-Arousal-Dominance (VAD) space. For each word w, the ANEW database provides a coordinate for the VAD affective space.

$$\bar{w} = (valance, arousal, dominance) = ANEW(w) \tag{4.2}$$

The occurrences of the words in a document of a sentence can then be used to weigh the text. This is done by averaging the values to get a single point in the VAD space.

$$\overline{emotion} = \frac{\sum_{i=1}^{k} \bar{w}}{k} \tag{4.3}$$

We are able to classify the text because the different emotions are mapped to different regions in the VAD space. The method received an average F-score of *0.392* on the ISEAR, Fairy tales, and SemEval datasets.

A rule-based approach was taken in the same year (2010) by Udochukwu et al. with great success [56]. With the added emotion classification category of disgust, an average F-score of *0.580* was achieved. A remarkable performance, because of the added category. The rules for different input variables are pre-coded after which the rules are used to evaluate the data. An example of a rule would be as follows:

If Direction = "Self" **and** Tense = "Future" **and** Overall Polarity = "Positive" **and** Event Polarity = "Positive", **then** Emotion = "Hope" (4.4)

In 2012 an unsupervised method was proposed by Agrawal et al. [1]. The method takes the same approach as Kim et al. [23], but replaces the hand-coded WordNet-Affect with a Pointwise Mutual Information (PMI) calculation. With an F-Score of *0.548* it is on par with the other methods. However, the method is more flexible and less dependent on hand-coded information of the WordNet-Affect repository.

Using several constraints Wang et al. (2015) [54] was able to further improve on the dimension reduction methods of Kim et al. [23]. Hypothesizing that documents that belong to the same topic have similar emotions and that some emotions are highly correlated the average F-score was further improved to *0.685*.

An interesting approach was performed by Perikos et al. (2016) [41] by combining multiple classifiers using a majority vote system in an ensemble classifier. A maximum entropy, naive Bayes, and a knowledge-based classifier were combined in this way. However, while trained with the ISEAR and Affective text datasets, it remains untested on

the same dataset. This makes the method incomparable to the current data without re-implementing the ensemble classifier.

Our method takes a machine learning approach that does not depend on the detection of specific features or a word repository. We take a flexible and non-domain-specific approach by leveraging a deep learner. It is assumed that deep learners are not suitable for the emotion analysis domain because of the lack of data. Our work differs from prior work by training a deep neural network on small datasets within the emotional analysis domain. We also use general features to create a flexible framework.

CHAPTER 5

Our approach

Classifiers bases on neural nets have been a bad performer on the often small emotional analysis datasets. However, they have been successful in other text classification and analysis tasks [22, 25]. We take a new approach by using a deep learning classifier for the small dataset in emotional analysis.

5.1 Network Architecture

The amount of hidden layers and neurons contained in the layers are important to the performance. Too few, and the network will not have the resources to solve the problem. Too many, and the time to train increases together with the chance of over-fitting. The network will have so much capacity it will also learn the unimportant details.

To get the right size, a practical guideline for building neural nets was laid down by Masters [29]. The geometric pyramid rule described by Masters, can be used for calculating a number of hidden neurons. The neurons in the different Layers will follow a geometric progression. If we create a three-layer network with n input neurons and m output neurons, we can calculate the amount of neurons in the hidden layer by multiplying the number of input neurons with the number of output neurons and squaring the result. Calculations for more than one hidden layer are a little more involved. We take a four-layer network with n input neurons and m output neurons. The amount of neurons in hidden layer $NHID_1$ and $NHID_2$ can be calculated as follows.

$$r = \sqrt[3]{\frac{n}{m}} \tag{5.1}$$

$$NHID_1 = m * r^2 \tag{5.2}$$

$$NHID_2 = m * r \tag{5.3}$$

Calculating the number of the neurons in each hidden layer, we get a neural network shaped as a pyramid.

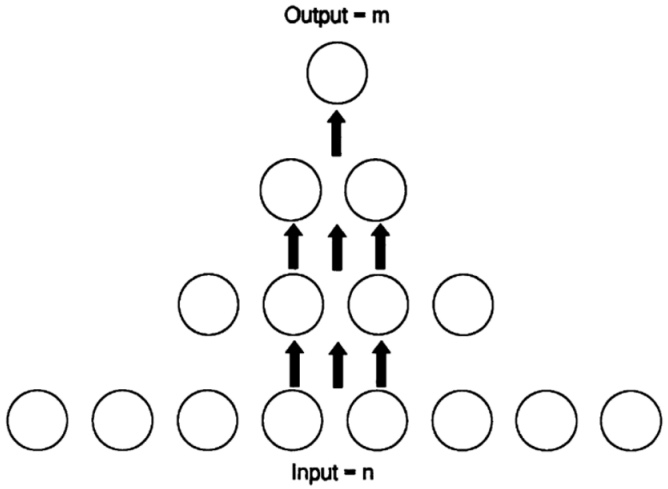

Figure 5.1: Geometric neural network [29]

Because we have a variable input layer, this practical guideline is not suitable for our solution. We, therefore, use our own empirically derived guideline for building the neural network architecture. We calculate the number of hidden neurons N in each hidden layer L starting from the output layer. Adjustments in the top hidden layer may be needed to optimize the results. The output layer has the same amount of nodes as the amount of classification categories.

$$\mathrm{N}(L) = \begin{cases} 10 & \text{if } L = 1 \\ 2^{L-1} + 10 & \text{if } L \geq 2 \end{cases} \tag{5.4}$$

Furthermore, we build our deep neural net based on a multi-layer perceptron. We use this type of neural network because it has been shown that it performs at least as well as other common methods used in emotional analysis [4]. For the hidden layers, we use the rectifier function for better training performance compared to the sigmoid function [18]. For the output later we use the softmax function to get a value between 0 and 1 for

each class. We classify the data by labeling it with the class that output's the highest value.

During the initial phase of our research, small exploratory experiments were done with alternative neural network geometries and activation functions. The tahn and linear activation function had the same bad performance as the sigmoid activation function. Neural networks not following the geometric pyramid rule performed worse then a support vector machine.

5.2 Vectorizer and Scaler

When building a machine learning classifier most methods use a specific set of features to help the machine learning algorithm to find the right solution. These features are often language and domain dependent. It is often the only clear way to build a creative solution because configuring the machine learning algorithm is rather limited.

In our approach, we will only use methods that are domain and language independent. This enables the method to be extended to other domains and languages in the future.

To extract language and domain independent features we combine multiple methods based on word count. These methods are generally called vectorizers because they extract a feature vector from data. The first and most basic method is the count vectorizer. We convert the text to a matrix of token counts.

The second method is based on the term frequencyinverse document frequency or tf-id. We use the method to transform a text to a matrix of tf-id features. The goal is to lessen the impact of common terms like "I" or "and" that are not informative. The tf-id feature matrix can be calculated based on the term frequency and the inverse document frequency. Term frequency or tf is the number of times a term t occurs in document d and is calculated as follows:

$$\mathrm{tf}(t, d) = 0.5 + 0.5 \cdot \frac{t, d}{max} \tag{5.5}$$

Inverse document frequency or idf diminished the weight of terms that occur too frequently like "the" or "and". Idf is calculated based on term t on and all documents D.

$$\mathrm{idf}(t, D) = \log \frac{t, d}{d} \tag{5.6}$$

Combining the two metrics we can find terms that have a high term frequency but a low document frequency retrieving only terms that are important to the document. This is called the term frequencyinverse document frequency or tf-id and is calculated based on the results of tf and idf.

$$\text{tfidf}(t, d, D) = \text{tf}(t, d) \cdot \text{idf}(t, D) \tag{5.7}$$

Furthermore, a term in the tf-id extraction process can be a character as well as a word. A term can also be created with multiple words or characters called n-grams. In our approach, we use tf-id to extract features on the character level and word level. We also use uni-grams and bi-grams on the character and word level. We concatenate the results of the different feature extraction approaches to get one big feature for our machine learning system.

The final step in the feature extraction process is scaling the feature values between the values -1 and 1 without shifting or centering the data. Scaling the features tend to improve training time in multi-layer perceptrons. Following the steps will result in an algorithm as described by algorithm 1.

Algorithm 1: Training the neural network

```
   Data: Documents
 1 initialization;
 2 foreach document in Documents do
       /* tfidf of words (uni-grams and bi-grams)                    */
 3     W = foreach wordTerm do
 4     |  tfidf(wordTerm, document, Documents)
 5     end
       /* tfidf of characters (uni-grams and bi-grams)               */
 6     C = foreach characterTerm do
 7     |  tfidf(characterTerm, document, Documents)
 8     end
       /* Term frequencies of words (uni-grams and bi-grams)         */
 9     A = foreach wordTerm do
10     |  tf(wordTerm, document)
11     end
12     D = (W, C, A);
13     V = Scale D between -1 and 1;
14     Feed V to input layer and train;
15 end
```

CHAPTER 6

Experimental setup

This chapter deals with the experimental setup and implementation of our method. We follow the workflow described by figure 3.1. The results are presented in chapter 7.

6.1 Datasets and labeling

To make our results comparable to the other methods we use the popular ISEAR Databank [32] to run our experiments. We furthermore, collect our own Twitter dataset to show that our method is generally applicable. We clean the twitter dataset by removing retweets and tweets with only hyperlinks. Only English language tweets are kept for the final dataset. The ISEAR Dataset is pre-labeled. However, our collected data is labeled by using hashtags as emotional labels as described by Saif [35]. The hashtags are removed from the resulting dataset. This labeling method allows us to collect a large labeled dataset in a short amount of time and is shown to give good results.

6.2 Machine learning pipeline

Because of the sequential steps that needs to be taken to process the data, we have implemented our system as a pipeline. The neural network is implemented in Theano [5]. For ease of experimentation, we use Scikit-learn [40] as an experimentation framework. Scikit-learn enables us to easily train our model and evaluate it. Scikit-learn also provides

features for reading the data sets, and extract features. We use the native facilities of Scikit-learn to extend the framework with our own implementation.

6.2.1 Dataset conversion

The datasets are converted to a folder structure on the filesystem so they can be read by the framework. Conversion is done by python scripts written for the dataset. The folder names get the names of the different label categories which contain text files with the actual text data. The filenames of the text-files are unique id's without an extra meaning.

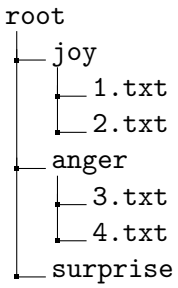

Figure 6.1: Dataset structure

6.2.2 Feature implementation

We implement the features as described by paragraph 5.2. Because we are using Scikit-learn as our experimentation framework, this is easy to do. We use the available vectorizers to implement our combined feature vector. We extract the features separately and concatenate them afterward to one big feature. We concatenate the different features starting with tf-id features on the word level. We then add the tf-id features on the character level and finish with the feature vector of the count vectorizer. For the tf-id features, we use uni-grams and bi-grams. We set the maximum document frequency to 0.95. This prevents commonly used words like "I" becoming a feature and influencing the results. We finally scale the feature values between -1 and 1.

6.2.3 Learning algorithm implementation

We implement the multilayer perceptron with the architecture described in paragraph 5.1. The neural network is implemented using Theano [5]. Furthermore, an integration with scikit-learn [40] is achieved by interfacing with the scikit-learn classification interface in the Python programming language. This enables us to train and predict with our neural network from within scikit-learn. An overview of the system can be found in figure 6.2. The implementation of the multi-layer perceptron was inspired by an implementation example provided by Theano.

6.2.4 Evaluation

With the classification interface implemented we can also use scikit-learn's evaluation abilities. We split the datasets into a training and a testing set. The training set contains

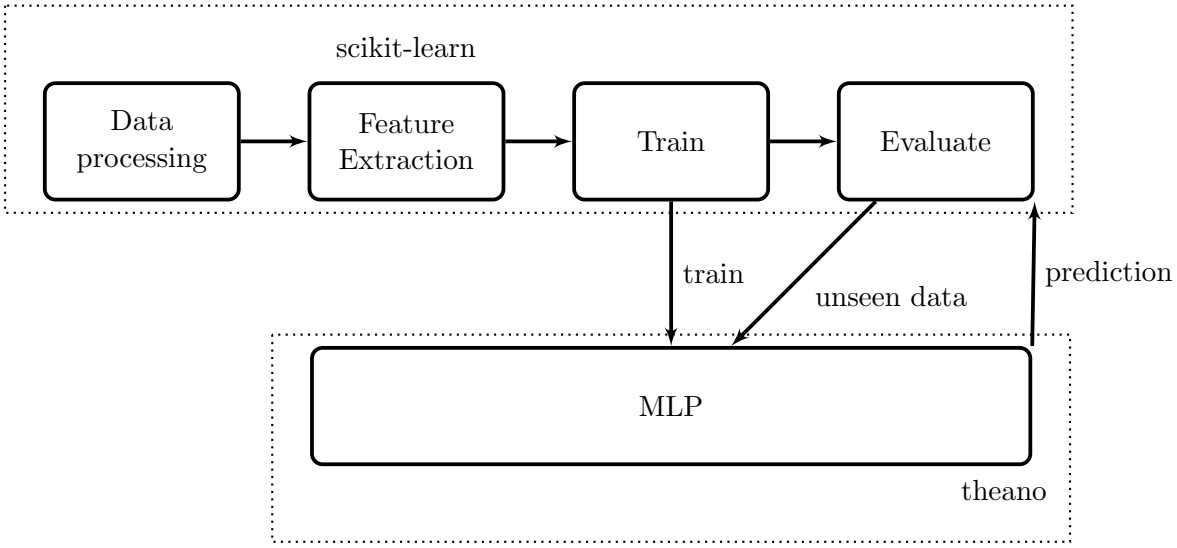

Figure 6.2: Machine learning pipeline

80% of the data, and 20% is left for the testing set. The testing set is set aside for the final evaluation. During development, we use 10-fold cross-validation to get preliminary results while preventing over-fitting on the testing set.

6.2.5 Learning algorithm configuration

With the testing capabilities of scikit-learn, we can now use 10 fold cross-validation to optimize our neural network. The optimal configuration for training the neural network has been acquired experimentally.

Table 6.1: Configuration overview

Configuration	
Iterations	5
Update rule	AdaGrad
Regularization	L2
Weight Decay	0.0001

For the backpropagation algorithm, we use the AdaGrad update rule. Because of the adaptive nature of AdaGrad, we are able to train our neural network much faster. We run the learning process for five iterations with weight decay. To prevent overfitting we use L2 regularization. Finally, we calculate the optimal architecture of our network with our method proposed in section 5.1. Starting from the output layer we have the following amount of neurons in the five hidden layers. The amount of neurons in the last hidden layer has been optimized experimentally.

$$(10, 12, 14, 18, 31) \tag{6.1}$$

CHAPTER 7

Results

In this chapter we will present the results of each experiment individually, after which the strengths and weaknesses of the approaches are evaluated. We will follow this up with a high level analysis of the data.

Table 7.1: Evaluated approaches

Abbr.	**Approach**
DNN-our	Our deep neural network with five hidden layers
SVM-non-L-our	Our non-linear support vector machine
NN	Plain neural network with one hidden layer
COA	Constrained optimization [54]
SVM-non-L	Non-linear support vector machine [54]
CB	Context-based [1]
Lexicon	Lexicon based model [54]
NMF	NMF based model [23]

7.1 Experiments

While setting up the experiments, we made sure the results were comparable to previous studies. Running the experiments has given us positive results. In this chapter we present the results by each category. We compare our results to the following algorithms, and if possible, also on different categories. As a control for our method, a one hidden layer

neural network is added to the experiments. To complement on missing data, our own non linear support vector machine was added where experimental data was missing.

7.1.1 ISEAR Four Emotions

Most research within the field of emotion analysis has been based on the four emotions from the ISEAR databank. This makes it the most relevant category to compare our approach to. We present the results in table 7.5. While small, the results show a one percent improvement over the previous state-of-the-art. While COA outperforms our method in the sad category, DNN-our has a better precision in the remaining categories. However, COA is much more stable across the different emotional categories compared to DNN-our. A reason for this might be the small dataset where a deep neural network is at a disadvantage.

Table 7.2: Results summary

Model		P	R	F
ISEAR four emotions				
Avg.	DNN-our	**0.76**	**0.74**	**0.75**
	NN	0.68	0.53	0.51
	COA	0.74	0.73	0.74
	SVM-non-L	0.60	0.57	0.58
	CB Wikipedia	—	—	0.54
	Lexicon	0.20	0.21	0.20
	NMF	0.69	0.64	0.66
ISEAR seven emotions				
Avg.	DNN-our	**0.52**	0.42	**0.44**
	NN	**0.52**	**0.44**	0.43
	SVM-non-L	0.43	0.41	0.42
	CB Wikipedia	—	—	0.43
	CB Gutenberg	—	—	0.40
	CB Wiki-Guten	—	—	0.41
Twitter six emotions				
Avg.	DNN-our	**0.58**	0.53	**0.54**
	NN	0.44	0.46	0.38
	COA	0.43	**0.67**	—
	SVM-non-L-our	0.16	0.14	—
	Lexicon	0.33	0.27	—
	NMF	0.40	0.47	—

7.1.2 ISEAR Seven Emotions

The second experiment is on the ISEAR databank with seven emotions. The experiment is not popular, but still, has a few studies using it. The result of the experiment can be found in table 7.4. Again, we see a small improvement of one percent over the previous state-of-the-art. However, in this case our deep neural network is only slightly better then the network with one hidden layer. This could be the result of the larger amount of classification categories. We hypothesize, that there is simply not enough training data, to give the deep neural network an edge over the one hidden layer neural network on this larger set of categories. Looking at the results of the twitter six emotions experiments with more training data we can see that DNN-our again outperforms the single hidden layer NN. The results supports our lack of data hypothesis.

7.1.3 Twitter Six Emotions

Because there is no standard twitter dataset, all studies collect their own. This makes the results in this category difficult to compare. Because the different studies use the same set of emotions, we can still make a comparison to get a indication of the performance. The emotions used are anger, disgust, fear, joy, sadness and surprise.

DNN-our, NN-our and SVM-Non-L-our is tested on our self collected dataset. The results from COA, SVM-non-L, and Lexicon approaches were obtained by Wang et al. on a self collected dataset [54]. NMF was obtained by Kim et al. also on a self collected dataset [23]. The results suggest that a deep neural network can make a significant improvement compared to current methods. Because deep learning methods tend to work better on larger datasets, the expectation is they will perform better with more data.

7.2 Neural Network size

Only after training, can we find the final size of our neural network (DNN-our). The size of the input layer depends on the dataset that is used for training. The size of the output layer depends on the amount of classification categories. For each dataset the final size of neural network is listed in table 7.3.

Table 7.3: DNN-our network size

Dataset	**Input layer**	$\mathbf{L_1}$	$\mathbf{L_2}$	$\mathbf{L_3}$	$\mathbf{L_4}$	$\mathbf{L_5}$	**Output layer**
ISEAR four emotions	53811	31	18	14	12	10	4
ISEAR seven emotions	67210	31	18	14	12	10	7
Twitter six emotions	223527	31	18	14	12	10	6

7.3 Analysis

The difficulty in training a deep neural network has always been the limiting factor for neural networks with state-of-art performance [17]. Part of the problems were caused by the vanishing and exploding gradients problem explain in section 3.3.1. While overfitting can easily be solved by correctly using cross validation and carefully train the model, the vanishing and exploding gradients problem where persistent. Methods like gradient clipping were developed to counter the problems with the exploding gradients [19].

However, breakthroughs in deep learning research have made it possible to train deep neural networks and avoid the vanishing and exploding gradients problem. In 2011 it was shown by Glorot et al. [18] that the rectifier activation function enables the training

of deep neural networks and also solves the vanishing and exploding gradient problem. The rectifier activation function allows for faster and effective training.

Furthermore, the training of neural networks has always been slow, forcing researchers to use smaller datasets and models. By taking advantage of the parallel power of modern graphical processing units (GPU), we can use much bigger datasets and larger models [45]. GPU computing has made the technology available to more researchers.

By leveraging these new technologies, we are able to train a deep neural network within the domain of emotional analysis. Our approach outperforms traditional methods within the field but also provides greater flexibility. The method is language independent and doesn't need hand picked features. We expect the deep neural network to perform better with more data.

Table 7.4: ISEAR: Seven Emotions

		ISEAR		
	Model	P	R	F
Anger	DNN-our	0.31	0.42	0.35
	NN	**0.33**	0.34	0.33
	SVM-non-L	0.22	**0.71**	0.33
	CB Wikipedia	—	—	0.41
	CB Gutenberg	—	—	**0.42**
	CB Wiki-Guten	—	—	0.41
Fear	DNN-our	0.41	0.44	0.43
	NN	0.46	**0.76**	**0.58**
	SVM-non-L	**0.63**	0.42	0.50
	CB Wikipedia	—	—	0.52
	CB Gutenberg	—	—	0.44
	CB Wiki-Guten	—	—	0.48
Joy	DNN-our	0.75	0.50	0.60
	NN	**0.78**	**0.64**	**0.70**
	SVM-non-L	0.75	0.39	0.51
	CB Wikipedia	—	—	0.51
	CB Gutenberg	—	—	0.50
	CB Wiki-Guten	—	—	0.50
Sad	DNN-our	0.66	**0.54**	**0.60**
	NN	**0.89**	0.16	0.27
	SVM-non-L	0.75	0.31	0.43
	CB Wikipedia	—	—	0.40
	CB Gutenberg	—	—	0.25
	CB Wiki-Guten	—	—	0.29
Disgust	DNN-our	**0.68**	0.37	**0.48**
	NN	0.50	**0.39**	0.44
	SVM-non-L	0.49	0.37	0.42
	CB Wikipedia	—	—	0.43
	CB Gutenberg	—	—	0.43
	CB Wiki-Guten	—	—	0.47
Shame	DNN-our	**0.62**	0.19	0.29
	NN	0.26	**0.46**	0.33
	SVM-non-L	0.53	0.29	**0.38**
	CB Wikipedia	—	—	0.40
	CB Gutenberg	—	—	0.40
	CB Wiki-Guten	—	—	0.40
Guilt	DNN-our	0.22	**0.51**	0.31
	NN	0.39	0.33	0.35
	SVM-non-L	**0.41**	0.38	**0.40**
	CB Wikipedia	—	—	0.39
	CB Gutenberg	—	—	0.37
	CB Wiki-Guten	—	—	0.33
Avg.	DNN-our	**0.52**	0.42	**0.44**
	NN	**0.52**	**0.44**	0.43
	SVM-non-L	0.43	0.41	0.42
	CB Wikipedia	—	—	0.43
	CB Gutenberg	—	—	0.40
	CB Wiki-Guten	—	—	0.41

Table 7.5: ISEAR: Four Emotions

		ISEAR		
	Model	P	R	F
Anger	DNN-our	**0.75**	0.82	**0.78**
	NN	0.59	**0.91**	0.71
	COA	0.71	0.68	0.69
	SVM-non-L	0.46	0.58	0.51
	CB Wikipedia	—	—	0.63
	Lexicon	0.19	0.59	0.29
	NMF	0.58	0.64	0.61
Fear	DNN-our	0.76	0.72	0.74
	NN	0.57	0.70	0.63
	COA	**0.78**	**0.78**	**0.78**
	SVM-non-L	0.68	0.58	0.51
	CB Wikipedia	—	—	0.59
	Lexicon	0.22	0.09	0.12
	NMF	0.69	0.69	0.69
Joy	DNN-our	0.80	0.80	**0.80**
	NN	**0.91**	0.13	0.23
	COA	0.71	**0.81**	0.76
	SVM-non-L	0.61	0.63	0.62
	CB Wikipedia	—	—	0.56
	Lexicon	0.20	0.21	0.12
	NMF	0.66	0.70	0.68
Sad	DNN-our	0.71	0.63	0.67
	NN	0.65	0.38	0.48
	COA	**0.77**	**0.67**	**0.72**
	SVM-non-L	0.64	0.51	0.57
	CB Wikipedia	—	—	0.41
	Lexicon	0.20	0.06	0.09
	NMF	0.71	0.54	0.65
Avg.	DNN-our	**0.76**	**0.74**	**0.75**
	NN	0.68	0.53	0.51
	COA	0.74	0.73	0.74
	SVM-non-L	0.60	0.57	0.58
	CB Wikipedia	—	—	0.54
	Lexicon	0.20	0.21	0.20
	NMF	0.69	0.64	0.66

CHAPTER 8

Conclusions

Advances in deep neural networks training have made it possible the apply the method in emotional analysis. The rectifier activation function solves the vanishing and exploding gradient problem that made deep layer networks hard to train. Through GPU computing, the method has become accessible to more researchers [45]. Our research suggest that deep learning could significantly improve on current method's within the field of emotional analysis. However the lack of a large dataset in this field is holding further development back. For example, a large twitter dataset annotated with emotions, will provide more opportunity for developing new methods and comparing them to other research.

We have shown that deep learning can successfully be applied to the field of emotional analysis with state-of-the-art results. At the time of writing, this is the first and only deep neural network developed and tested on the topic of emotional analysis that we are aware of. Deep learning provides signification advantages over current methods. Instead of handpicking the features we need for each domain, we let the neural network discover the relevant features. This makes the approach suitable for different datasets and languages with only changes needed in the neural networks architecture. We furthermore present a guideline for building deep learning architectures for the purpose of emotional analysis classification tasks.

Small exploratory experiments during the initial phase of the research were done using different activation functions and geometries. Sigmoid, tahn and the linear activation function gave similar results with the rectifier activation function being superior. Neural network geometries based on the geometric pyramid rule also gave superior results. While the changes gave better performance on their own, combining the rectifier activation

function with the geometric pyramid rule raised the performance dramatically. It seems that this change made it possible to raise the performance to rival the other machine learning algorithms.

Futher work

Our research focused on evaluating deep learning for emotional analysis classification tasks. However, based on our results and experiences the follow subjects would be interesting for further research:

- *Unsupervised learning*: Current methods rely on big datasets with labeled items to build classifiers for emotional analysis. Building the dataset relies on a large amount of human labor. A method that can be trained with minimal amount of labeled data would decrease the need for large amounts of labeled data.
- *Ensemble classifier*: The results show that our neural network is performing not as well on certain categories as the other more tradional methods. Combining different methods in an ensemble classifier could solve these deficiencies. If enough computational power is available an ensemble of neural networks with different architectures and parameters would be interesting to explore.
- *Bigger and standardized datasets*: Comparing results is an important way of verifying the performance of one's method. To be able to do this, publicly available datasets for emotional analysis are crucial. To get traction the dataset could be presented as a SemEval task in the future [38].
- *Relevance feedback*: In modern content-based image retrieval systems, relevance feedback is a technique used to augment the size of small datasets to improve performance. Users of a system give feedback on the relevance of the results [21, 48]. This information is then used to improve the following query made by the user. Using this technique could improve the performance of emotional analysis systems.
- *Other neural networks*: Deep learning has become a broad field with different type of neural networks. It would be interesting to see how other neural networks perform within the field of emotional analysis and why. An example would be different types of restricted bolz machines.

CHAPTER 9

Bibliography

[1] Ameeta Agrawal and Aijun An. Unsupervised Emotion Detection from Text Using Semantic and Syntactic Relations. *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, pages 346–353, 2012.

[2] A Aizerman, Emmanuel M Braverman, and LI Rozoner. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and remote control*, 25:821–837, 1964.

[3] Saima Aman and Stan Szpakowicz. Using roget's thesaurus for fine-grained emotion recognition. In *IJCNLP*, pages 312–318. Citeseer, 2008.

[4] Les Atlas, Ronald Cole, Yeshwant Muthusamy, Alan Lippman, Jerome Connor, Dong Park, M El-Sharkawai, and RJ Marks. A performance comparison of trained multilayer perceptrons and trained classification trees. *Proceedings of the IEEE*, 78(10):1614–1619, 1990.

[5] James Bergstra, Olivier Breuleux, Frederic Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU and GPU math compiler in Python. *Proceedings of the Python for Scientific Computing Conference (SciPy)*, (Scipy):1–7, 2010.

[6] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

[7] Margaret M Bradley, Mark K Greenwald, Margaret C Petry, and Peter J Lang. Remembering pictures: pleasure and arousal in memory. *Journal of experimental psychology: Learning, Memory, and Cognition*, 18(2):379, 1992.

[8] Margaret M Bradley and Peter J Lang. Affective norms for english words (anew): Instruction manual and affective ratings. Technical report, Technical report C-1, the center for research in psychophysiology, University of Florida, 1999.

[9] Cynthia Breazeal and Brian Scassellati. How to build robots that make friends and influence people. In *Intelligent Robots and Systems, 1999. IROS'99. Proceedings. 1999 IEEE/RSJ International Conference on*, volume 2, pages 858–863. IEEE, 1999.

[10] Cynthia L Breazeal. *Sociable machines: Expressive social exchange between humans and robots*. PhD thesis, Massachusetts Institute of Technology, 2000.

[11] Yves Chauvin and David E Rumelhart. *Backpropagation: theory, architectures, and applications*. Psychology Press, 1995.

[12] Giovanna Colombetti. From affect programs to dynamical discrete emotions. *Philosophical Psychology*, 22(4):407–425, 2009.

[13] Tim Dalgleish and Michael J Power. *Handbook of cognition and emotion*. Wiley Online Library, 1999.

[14] Paul Ekman. An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200, 1992.

[15] Paul Ekman and Wallace V Friesen. Constants across cultures in the face and emotion. *Journal of personality and social psychology*, 17(2):124, 1971.

[16] Nico H Frijda et al. *Emotions are functional, most of the time*. Oxford University Press., 1994.

[17] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Aistats*, volume 9, pages 249–256, 2010.

[18] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Aistats*, volume 15, page 275, 2011.

[19] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[20] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

[21] Mark J Huiskes and Michael S Lew. Performance evaluation of relevance feedback methods. In *Proceedings of the 2008 international conference on Content-based*

image and video retrieval, pages 239–248. ACM, 2008.

[22] Ozan Irsoy and Claire Cardie. Opinion Mining with Deep Recurrent Neural Networks. pages 720–728, 2014.

[23] Sunghwan Mac Kim, Alessandro Valitutti, and Rafael a Calvo. Evaluation of Unsupervised Emotion Models to Textual Affect Recognition. *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, (June):62–70, 2010.

[24] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145, 1995.

[25] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. Recurrent Convolutional Neural Networks for Text Classification. *Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2267–2273, 2015.

[26] Robert W Levenson. Human emotion: A functional view. *The nature of emotion: Fundamental questions*, 1:123–126, 1994.

[27] Kevin Hsin-Yih Lin, Changhua Yang, and Hsin-Hsi Chen. What emotions do news articles trigger in their readers? In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 733–734. ACM, 2007.

[28] Hugo Lövheim. A new three-dimensional model for emotions and monoamine neurotransmitters. *Medical hypotheses*, 78(2):341–348, 2012.

[29] Timothy Masters. *Practical neural network recipes in C++*. Morgan Kaufmann, 1993.

[30] Walaa Medhat, Ahmed Hassan, and Hoda Korashy. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4):1093–1113, 2014.

[31] A. Mehrabian. *Basic dimensions for a general psychological theory: implications for personality, social, environmental, and developmental studies*. Social Environmental and Developmental Studies. Oelgeschlager, Gunn & Hain, 1980.

[32] Gerold Mikula, Klaus R Scherer, and Ursula Athenstaedt. The role of injustice in the elicitation of differential emotional reactions. *Personality and social psychology bulletin*, 24(7):769–783, 1998.

[33] Marvin Minsky. The emotion machine. *New York: Pantheon*, 2006.

[34] Marvin Minsky and Seymour Papert. Neurocomputing: Foundations of research. chapter Perceptrons, pages 157–169. MIT Press, Cambridge, MA, USA, 1988.

[35] Saif Mohammad. #Emotional Tweets. *Proceedings of the First Joint Conference on Lexical ...*, pages 246–255, 2012.

[36] Saif M. Mohammad, Xiaodan Zhu, Svetlana Kiritchenko, and Joel Martin. Sentiment, emotion, purpose, and style in electoral tweets. *Information Processing & Management*, 51(4):480–499, 2015.

[37] Saif M. Mohammad, Xiaodan Zhu, Svetlana Kiritchenko, and Joel Martin. Sentiment, emotion, purpose, and style in electoral tweets. *Information Processing & Management*, 51(4):480–499, 2015.

[38] Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov. Semeval-2016 task 4: Sentiment analysis in twitter. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval 2016), San Diego, US (forthcoming)*, 2016.

[39] John C Nemiah. Shame and pride: Affect, sex, and the birth of the self. *American Journal of Psychiatry*, 151(5):776–777, 1994.

[40] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[41] Isidoros Perikos and Ioannis Hatzilygeroudis. Recognizing emotions in text using ensemble of classifiers. *Engineering Applications of Artificial Intelligence*, pages 1–11, 2016.

[42] Rosalind W Picard. Toward machines with emotional intelligence. In *ICINCO (Invited Speakers)*, pages 29–30. Citeseer, 2004.

[43] R.W. Picard. *Affective Computing*. MIT Press, 2000.

[44] Robert Plutchik. The nature of emotions human emotions have deep evolutionary roots, a fact that may explain their complexity and provide tools for clinical practice. *American Scientist*, 89(4):344–350, 2001.

[45] Rajat Raina, Anand Madhavan, and Andrew Y Ng. Large-scale deep unsupervised learning using graphics processors. In *Proceedings of the 26th annual international conference on machine learning*, pages 873–880. ACM, 2009.

[46] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, DTIC Document, 1985.

[47] James A. Russell. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39(6):1161–1178, 1980.

[48] Gerard Salton and Chris Buckley. Improving retrieval performance by relevance feedback. *Readings in information retrieval*, 24(5):355–363, 1997.

[49] Mohammad Soleymani, Maja Pantic, and Thierry Pun. Multimodal emotion recognition in response to videos. *IEEE transactions on affective computing*, 3(2):211–223, 2012.

[50] C. Strapparava, C. Strapparava, R. Mihalcea, and R. Mihalcea. Semeval-2007 task 14: Affective text. *Proc. of SemEval-2007*, (June):70–74, 2007.

[51] Carlo Strapparava, Alessandro Valitutti, et al. Wordnet affect: an affective extension of wordnet. In *LREC*, volume 4, pages 1083–1086, 2004.

[52] Silvan S Tomkins. *Affect imagery consciousness: Volume I: The positive affects*, volume 1. Springer publishing company, 1962.

[53] Renee Van Bezooijen. *Characteristics and recognizability of vocal expressions of emotion*, volume 5. Walter de Gruyter, 1984.

[54] Yichen Wang and San Jose. Detecting Emotions in Social Media : A Constrained Optimization Approach. (Ijcai):996–1002, 2015.

[55] David Watson and Auke Tellegen. Toward a consensual structure of mood. *Psychological bulletin*, 98(2):219, 1985.

[56] René Witte and Ralf Krestel. Natural Language Processing and Information Systems. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6177:36–47, 2010.

[57] Changhua Yang, Kevin Hsin-Yih Lin, and Hsin-Hsi Chen. Emotion classification using web blog corpora. In *Web Intelligence, IEEE/WIC/ACM International Conference on*, pages 275–278. IEEE, 2007.