Improving Customer Satisfaction using Sentiment Analysis

Master's Thesis in Computer Science

Mohamad H. Jalloul

May 18, 2018
Halden, Norway

Østfold University College www.hiof.no

Abstract

This thesis presents a qualitative study to determine if sentiment analysis on social media can facilitate the customer and user feedback process for a software developer. We have developed a prototype for classifying tweets using an ensemble of sentiment classifiers that have been trained using three standard data sets. The developed prototype was used in the qualitative study conducted on five software developers.

The results of the study show that the prototype can help identify feedback given in social media, which may never reach the developers without a social media mining tool. The results also show that the prototype can provide an overview of user experiences of an application, a product, or other services.

The sentiment analysis prototype is general and can be used to classify tweets relating to any topic. The prototype is not only designed for product and application developers, but for anyone working with customers or towards a better customer satisfaction. The prototype can also be implemented as part of a larger system, for example a customer support system.

Keywords: Sentiment Analysis, Customer Feedback, Social Media, Customer Satisfaction, Ensemble, Support Vector Machine, Naive Bayes, Random Forest

Acknowledgements

I would like to thank my supervisor, Lars Vidar Magnusson, who has given me valuable feedback throughout this thesis. I would also like to thank Susanne Koch Stigberg, for the comments and preparation for the study, and all of the participants in the study. Finally, I wish to thank my parents for their support throughout my study.

Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

The World Wide Web is full of content such as customer reviews, social media posts and blog posts that express opinions on products, applications and other services. Customer feedback is important with respect to other users and valuable for product developers.

1.1 Motivation

It is imperative to get quick, honest and constructive feedback to act upon in order to continuously improve products. Product- and application developers have noticed that their customers and users share their feedback through social media, which could include opinions, bug reports, possible improvements and other comments. This information would allow developers to increase the satisfaction of their users, but the information might never reach them. Manually searching through social media for customer feedback would require lots of resources and would therefore be a costly endeavour for companies.

Feedback shared on social media can reach potential customers before developers can refute or comment back on them. The immediate availability of information collected from monitoring the products and applications can, if gathered, analysed and structured help shorten the feedback loop and gather information developers would normally miss out on. Their goal is after all to make the users happy and enjoy a great a product.

Customer satisfaction is one of the main goals for product and application manufacturers. Product and application owners can receive over dozens of customers reviews a day spread out on different channels such as in news, blogs, forums and social media. In this thesis, our primary objective is to determine the value, if any, of using sentiment analysis to mine customer feedback from social media so that developers and others working towards a better customer satisfaction may identify customer feedback in social media and act upon these.

1.2 Research Question & Method

RQ. What, if any, is the value of using sentiment analysis in the domain customer and user feedback in social media?

Method

We assessed the research question by reviewing related work and tools utilising sentiment analysis. We developed a sentiment analysis web-application which was used in an qualitative study. The qualitative study was performed in two parts, prototype testing and interviews, and was conducted on five software developers.

1.3 Report Outline

This thesis is based on the IMRaD (Introduction, Method, Results, and Discussion) structure. Chapter 2 Background presents some information on the topics; social media, customer feedback and sentiment analysis. We present related work in Chapter 3. Chapter 4 presents the prototype developed in this thesis. We describe how the prototype testing and interviews were prepared and planed in Chapter 5. We present the results of the study in Chapter 6. We discuss the results from the study and an improved system in Chapter 7, before we end with the conclusions in Chapter 8.

Chapter 2

Background

This chapter contains background information on the topics customer feedback, social media and sentiment analysis.

2.1 Customer feedback

Since the growth of the World Wide Web, new content such as customer feedback and blogs that express opinions on products and services, also referred to as customer reviews, has become an important source of information [34].

Barlow and Møller [6] define customer feedback as complaints and that companies must think of these as an opportunity to learn something new about their products or services. They also describe it as a statement about expectations that has not been met but is an opportunity to satisfy a customer by improving the product or service.

> *"Complaints provide a great feedback mechanism that can help organisations rapidly and inexpensively shift products, service style and/or market focus to meet the needs of the customers - who, after all, pay the bills and are the reason why we remain in the business in the first place."*
>
> \- Janelle Barlow & Claus Møller

Barlow and Møller [6] also mention that to consider these feedback as gifts, companies would have to accept that the customer always has the right to complain, and that the customer is still showing confidence and loyalty in the company by taking time to complain.

Maalej and Nabil [39] categorise customer feedback into four types:

Bug Report describes a problem with the application which should be corrected, such as a crash or a performance issue.

Feature Request is when a user ask for a missing functionality, or new ideas by adding or changing features.

User Experiences is where the user reflects on the experience with the application and its features.

Ratings are simply text reflections of the numeric star rating. Rating are less informative as they only include praise or dispraise.

Mudambi and Schuff [45] mention that customers tend to search online for product information and evaluate alternative products based on others' customer reviews. They define online customer reviews as peer-generated product evaluations posted on company or third-party websites. Websites offer the consumers the opportunity to post product reviews with content in form of numerical star ratings (1 to 5 stars) and comments about the product.

Lipsman [36] found that customers reviews have significant impact on purchase behaviour and what customers are willing to pay for a product or service.

Gallaugher and Ransbotham [22] describe the communication between customers and firms before and after social media. A company or a firm would before the emergence of social media interact with customers either *individually* or in *mass* communication. The individual communication would be as part of a purchase or at a customer service desk, which occurred through phone calls, face-to-face, email or postal mail. It was either the firm or the customer who initiated these dialogues. Mass communication included printed or broadcast advertising and were typically by the firm. The customers had limited ability to observe or influence other customers. They also explain that the changes brought by the emergence of social media created new firm-customer interactions and exposed these to others. Customers can participate in the firm-customer relationship of other customers and learn about the firm by observing others. Comments online are visible to other customers, and they can also corroborate or refute the experiences of other customers.

These studies show how important customer feedback has become for businesses and companies to consider when developing products and applications.

2.2 Social Media

Social Network sites has been around since the launch of SixDegrees in 1997 [16]. It allowed users to create profiles, list their friends and view friend lists.

Ellison et al. [16] define social network sites as web-based services which allows a user to create a public profile within the system, create a list of other users they share a connection with, and lastly view their list of connections and those made by others. They also describe how social network sites are unique by allowing users to create and show their social networks and create new connections with strangers.

Kaplan and Haenlein [30] define social network sites as applications that allow users to connect, by creating personal information profiles, inviting friends and colleagues to have access to those profiles, and sending e-mails and instant messages between each other. The profiles may include information, photos, videos, audio, and blogs. They also mention that social network sites are so popular amongst younger users, that new terms are made, such as "Facebook addict".

The user-basis and user-activity in social media has increased at an incredible speed. Figure 2.1 presents an overview of the most popular social media sites worldwide as of January 2018, ranked by number of active users a month. Facebook registered 175 million active users in 2009 [30]. That number had increased by over ten times when Facebook reported over two billion monthly active users in 2018 [18].

Facebook was founded in 2004 and state that their mission is to give people the power to build communities and bring the world closer together [17]. They also mention that people use Facebook to stay connected with friends and family, to discover what is going on in the world, and to share and express what matters to them.

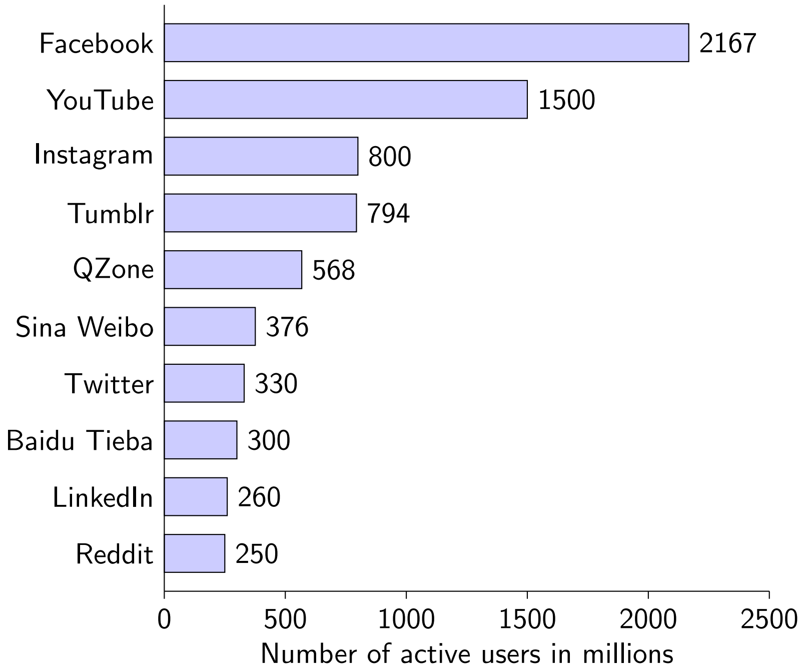


Figure 2.1: The most famous social media sites worldwide as of January 2018, ranked by active users. The numbers in the graph are retrieved from Statista [55].

Twitter has also seen an incredible increase. Weil [61] reported that Twitter users were tweeting 5000 times a day in 2007, which increased to 50 million tweets per day in 2010. Weil [62] reported a few years later that 500 million tweets were being published per day in 2014. He also describes how they were working on giving more ways to tweet beyond text, such as adding videos, photos, GIFs, and other features.

Twitter offers five main functions [58]:

Tweet A tweet is a twitter post or message that may contain photos, GIFs (Graphics Interchange Format), videos, links, and text. To post or publish a tweet on Twitter is known as tweeting.

Retweet A Tweet that you share publicly with your followers is known as a Retweet.

Follow To Follow someone on Twitter means that the user is subscribing to their Tweets as a follower. The tweets posted by the one being “Followed” will appear in the *Home Timeline*. The person being “Followed” will also be able to send the user direct messages.

Search A user can find Tweets from friends, local businesses, and everyone else. Users can search for topic keywords or hashtags, to follow ongoing conversations about breaking news or personal interests.

Hashtags A hashtag, written with a # symbol, is used to index keywords or topics on Twitter. This function was created on Twitter, and allows people to easily follow topics they are interested in.

A tweet has a few unique attributes. Go et al. [24] describe the following attributes: *Length*, the maximum length of a tweet is 140 characters. *Data availability*, the magnitude of data about a tweet. *Language*, twitter users post messages from many different media, including cell phones. The frequency of misspellings and slang in tweets is much higher than in other domains. And *Domain*, twitter users post short messages about a variety of topics.

Go et al. [24] note that tweets differ from reviews because of their purpose. Reviews represent a summary of the authors thoughts on a specific topic. While tweets are more casual and limited to 140 characters of text and are generally not as thoughtfull as reviews. But tweets still offer companies an additional method to gather feedback.

2.3 Sentiment Analysis

Wilson et al. [63] define sentiment analysis as a task of identifying positive and negative opinions and emotions. Sentiment analysis has become an active research area within natural language processing and has also been successfully applied in management science, in studies by: Archak et al. [3], Das and Chen [14], and Ghose et al. [23]. Bird et al. [7] describe that natural language processing could be as simple as counting word frequencies or as extreme as understanding and responding to humans.

> *"Sentiment analysis, also called opinion mining, is the field of study that analyses people's opinions, sentiments, evaluations, appraisals, attitudes, and emotions towards entities such as products, services, organisations, individuals, issues, events, topics, and their attributes."*
>
> - Bing Liu

Liu [38] defines sentiment analysis as a field of study that analyses people's opinion, attitude or sentiment towards entities such as products or services. He also describes three main levels in which sentiment analysis is performed, *document*, *sentence* and *aspect* level.

The classification task in document-level is defined as following [37]: In a set of documents D, it determines whether each document $d \epsilon D$ expresses a positive or negative opinion (or sentiment) on an object. In a given document d that comments on an object o, determine the orientation oo of the opinion that is expressed on o, for instance, discover the opinion orientation oo on feature f in the quintuple (o, f, so, h, t), where $f = o$ and h, t, o are assumed to be known or irrelevant.

Liu [37] mentions that other existing research on sentiment classification makes the following assumption: The opinionated document d expresses opinions on a single object o and the opinions are from a single opinion holder h. This assumption may hold for customer reviews of products and services, but he argues that it may not hold for forum and blog post because the author may express opinions on multiple products and compare them. Most existing techniques for document-level sentiment classification are based on supervised learning.

In *sentence level*, the task is to classify whether each sentence is positive or negative. Liu [37] states that there is no difference between document and sentence level classification because sentences are just short documents and may contain multiple opinions.

The classification task in sentence-level is defined as following: In a given sentence s, two subtasks are performed

1. *Subjectivity classification:* Determine whether s is a subjective sentence or an objective sentence

2. *Sentence-level sentiment classification:* If s is a subjective, determine whether it expresses a positive or negative opinion.

The two subtasks of sentence-level classification are important because they filter out those sentences that contain no opinion, and after knowing what objects and features of the objects are talked about in a sentence, the opinions on the objects and their features can be determined as positive or negative.

Aspect level looks at the opinion and is based on an opinion which consists of a sentiment and a target. Liu [37] explains that for a complete analysis of a sentence or document, one would need to discover the aspects and determine whether their sentiment is positive or negative on each aspect. He defines the aspect-level classification task as following: Identify object features that have been commented on. For instance, in the sentence, "*The picture quality of this camera is amazing*" the object feature is "picture quality". Determine whether the opinions on the features are positive, negative, or neutral.

Cambria et al. [11] describe the following two common sentiment analysis tasks, *polarity classification* and *agreement detection*. They explain that polarity classification occurs when a piece of text stating an opinion on a single issue is classified as one of two sentiments. Examples of polarity classifications are "thumbs up" versus "thumbs down" or "like" versus "dislike". They explain that agreement detection determines whether a pair of text documents should receive the same or different sentiment-related labels. After a system identifies the polarity classification, it might assign degrees of positivity to the polarity. It will help classify the sentiment when distinguishing between the subjective and the objective. A piece of text might have a polarity without necessarily containing an opinion, for example a news article could be classified into good or bad news without being subjective.

Most sentiment classification is done using supervised classification [7]. Bird et al. describe classification as a task of choosing the correct class label for a given input. In basic classification tasks, each input is isolated from other inputs, and the set of labels is defined in advance. Some supervised classification applications are:

- Deciding if an email is spam or not

- Deciding the topic of a news article

- Deciding the meaning of a word in a specific context

Bird et al. [7] describe a classifier as supervised only if its built based on training corpora containing the correct label for each input. Figure 2.2 presents an overview of how supervised classification is performed. (a) A feature extractor is used to convert each input value to a feature set during training. The feature sets capture the basic information about each input that should be used to classify it. Feature sets and labels are then fed into the machine learning algorithm to generate a model. (b) The same feature extractor is used to convert unseen inputs to feature sets during prediction. The feature sets are then fed into the model, which generates predicted labels.

Taboada et al. [56] describe two main approaches to the problem of extracting sentiment automatically, *lexicon-based* and *machine learning*. Lexicon-based approach involves calculating orientation for a document from the semantic orientation of words in the document. The machine learning approach involves building classifiers from labelled instances of text or sentences. A combination of both has also been applied in studies by: Khan et al. [31], Melville et al. [42], and Prabowo and Thelwall [49].

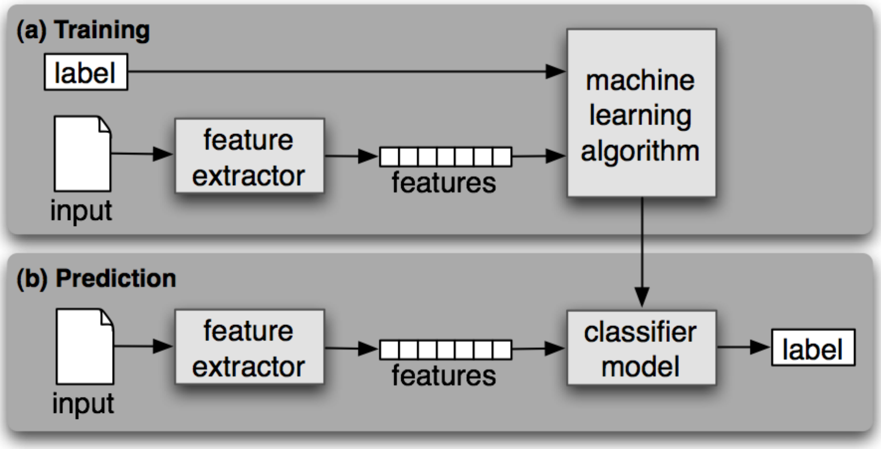


Figure 2.2: An illustration of how supervised classification is performed in training and prediction. The figure is from Bird et al. [7].

2.3.1 Lexicon-based in Sentiment Analysis

Taboada et al. [56] explain that dictionaries, also known as word list, for lexicon-based approaches can be created manually or automatically using seed words to expand the word list of words. Lexicon-based research has focused on using adjectives as indicators of the semantic orientation (SO) of text. First, a list of adjectives and corresponding SO values are compiled into a dictionary. All adjectives are then extracted from a given text, and annotated with their SO value using the dictionary scores. The scores are then aggregated into a single score for the text. The following are dictionaries used in sentiment analysis in the English language:

ANEW Affective Norms of English Words (ANEW) was developed by Bradley and Lang [9] to provide a set of normative emotional ratings for a large number of words in the English language. There are two primary scores for each word in the list for emotional assessment. *Valence*, which ranges from pleasant to unpleasant and *Arousal*, which ranges from calm to excited.

WordNet Miller [43] explains WordNet as an effective combination of lexicographic information and modern computing, and an online lexical database designed for use under program control. He also describes it as a database that links English nouns, verbs, adjectives, and adverbs to sets of synonyms that are linked through semantic relations to determine its definition. WordNet consists of more than 118,000 different word forms and more than 90,000 different word senses.

SentiWordNet SentiWordNet is a lexical resource for sentiment analysis and opinion mining which was developed by Baccianella et al. [5]. It was generated by automatically annotating all WordNet where each synonym was assigned three sentiment scores, positivity, negativity and objectivity [53].

2.3.2 Machine Learning in Sentiment Analysis

Most of text classification research builds on classifiers trained on data set using features such as unigrams or bigrams. Taboada et al. [56] explain that classifiers built using supervised methods can reach a high accuracy in detecting the polarity of a text. However, they argue that the performance of a classifier drops when it is used in a different domain.

Classifiers

Murphy [46] defines a classifier as following: a function f that maps input feature vectors $x \epsilon X$ to output class labels $y \epsilon \{1, ..., C\}$, where X is the feature space. He assumes $X = \mathbb{R}^D$ or $X = \{0,1\}^D$, that the feature vector is a vector of D real numbers or D binary bits, and that class labels are unordered. The goal is to learn f from a labelled training set of N input-output pairs.

Naive Bayes

Murphy [46] explains Naive Bayes in document classification as following: We want to classify a document into one of C classes (e.g., positive and negative). A simple representation, called the bag or words model, is to ignore word ordering and just count the number of times each word occurs. Suppose there are D words in the language. Then a document can be represented as a p-vector of counts. Let $X = k$ mean the word occurs exactly k times, for $k = 0$: $K - 1$. For simplicity, word has count k. In this case, Murphy presents the class-conditional density as product of multinomial:

$$p(x|Y = c, \theta) = \prod_{i=1}^{D} \prod_{k=1}^{K} \theta_{ick}^{I(x_i=k)} \tag{2.1}$$

where $\theta_{ick} = p(X_i = k|Y = c)$ is the probability of observing the i-th worth having count k given that the class is c. The purpose behind this is that the number of times a word occurs in a document may provide some information about what type of document it is.

Another representation by Murphy is to just represent whether the word occurs or not, where the binary feature vector is x. In this case, he represents the class-conditional densities as a product of *Bernoulli* distributions.

$$p(x|Y = c, \theta) = \prod_{i=1}^{D} \theta_{ic}^{x_i} (1 - \theta_{ic})^{1-x_i} \tag{2.2}$$

where θ_{ic} is the probability word i occurs in class $c, x_i = 1$ means word i is present, and $x_i = 0$ otherwise.

The *Multinomial* model captures word frequencies information in documents [41]. The model will map all strings of digits to a common token if the occurrence of numbers in news articles is considered. Since every article is dated, the number token in the Bernoulli model is uninformative. However, news article about earnings have more numbers then general articles, which can help capture frequency information and the classification.

McCallum et al. [41] explain that in the multinomial model, a document is an ordered sequence of word events, drawn from the same vocabulary V. They assume that the lengths of documents are independent of the class and that the probability of each word event in a document is independent of the word's context and position in the document. Each document d_i is drawn from a multinomial distribution of words with as many independent trials as the length of d_i and N_{it} to be the count of number of times word w_t occurs in document d_i.

$$p(d_i|c_j;\theta) = P(|d_i|)|d_i|!\prod_{t=1}^{|V|}\frac{P(w_t|c_j;\theta)^{N_{it}}}{N_{it}!} \tag{2.3}$$

The parameters of the generative component for each class are the probabilities for each word, written $\theta_{w_t|c_j} = P(w_t|c_j;\theta)$, where $0 \leq \theta_{w_t|c_j} \leq 1$ and $\sum_t \theta_{w_t|c_j} = 1$.

Figure 2.3 presents an example of Naive Bayes. In Naive Bayes, each attribute node has no parent except the class node.

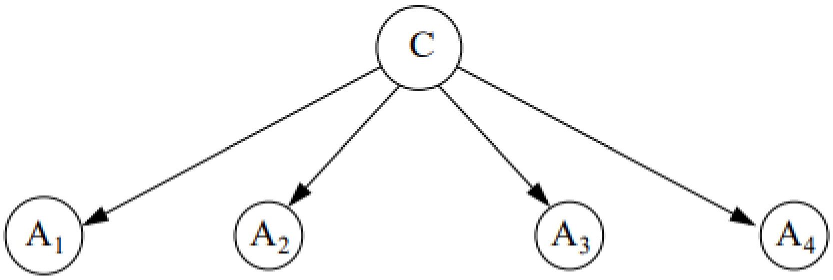


Figure 2.3: An example of Naive Bayes. The figure is from Zhang [64].

Support Vector Machine

Support Vector Machines (SVM) have been shown to be highly effective at traditional text categorisation and outperforming Naive Bayes [48]. The idea behind the training procedure in SVM is to find a hyperplane, represented by vector $\vec{w}$, which does not only separate the document vectors in one class from the others, but also for which the separation, or *margin*, is as large as possible. Pang et al. [48] explain that this search corresponds to a constrained optimisation problem; letting $c_j \epsilon \{1, -1\}$ (corresponding to positive and negative) be the correct class of document d_j, they write the solution as:

$$\vec{w} := \sum_j \alpha_j c_j \vec{d_j}, \quad \alpha_j \geq 0, \tag{2.4}$$

where the α_j's are obtained by solving a dual optimisation problem. Those $\vec{d_j}$ such that α_j is greater than zero are called support vectors, since they are only document vectors contributing to $\vec{w}$. Classification of test instances consists simply of determining which side of $\vec{w}$'s hyperplane they fall on.

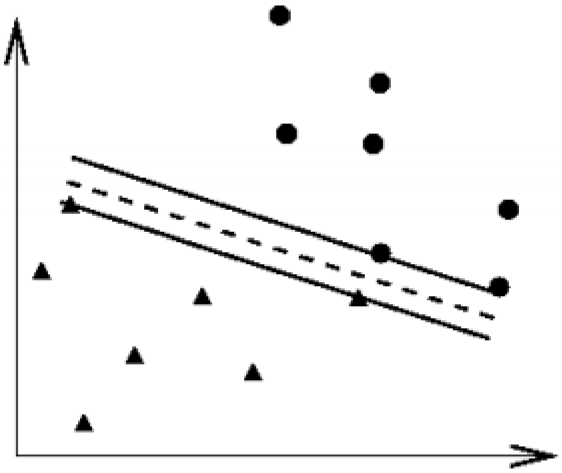


Figure 2.4: An illustration of Support Vector Machine classification trained with two classes. The figure is from Hsu et al. [27].

Figure 2.4 presents an illustration of a linear SVM trained with two classes. SVM constructs a separating hyperplane and tries to maximise the margin between the classes. SVM calculates the margins by constructing two parallel hyperplanes on each side of the initial one. These are then pushed until they reach either class.

Linear and Logistic Regression

Hosmer Jr et al. [26] explain that regression methods have become an integral component of any data analysis, describing relationship between a response variable and one more explanatory variable. Usually the outcome variable is two or more possible values. Logistic regression model can be distinguished from the linear regression model by the outcome variable in logistic regression which is *binary* or *dichotomous.* They explain that the difference between the two regressions is the choice of a parametric model and their assumptions.

In any regression problem, the key quantity is the mean value of the outcome variable, given the value of the independent variable. This quantity is called the *conditional mean* and can be expressed as $E(Y|x)$ where Y represents the outcome variable and x represents a value of the independent variable. The quantity $E(Y|x)$ can be read "the expected value of Y, given the value x". In linear regression the assumption can be expressed as an equation linear in x, such as [26]:

$$E(Y|x) = \beta_0 + \beta_1 x \tag{2.5}$$

They explain that the expression implies that it is possible for $E(Y|x)$ to take on any value as x ranges between $-\infty$ and $+\infty$

Hosmer Jr et al. [26] mention two primary reasons for choosing the logistic distribution. First, from a mathematical point of view, it is flexible and easily used function. Second, it lends itself to a clinically meaningful interpretation.

They explain that the quantity $\pi(x) = E(Y|x)$ is used to represent the conditional mean of Y given x when logistic distribution is used. They used the following logistic regression model:

$$\pi(x) = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}} \tag{2.6}$$

and a *logit transformation* can be defined, in terms of $\pi(x)$ as:

$$g(x) = \ln\left[\frac{\pi(x)}{1-\pi(x)}\right] = \beta_0 + \beta_1 x \tag{2.7}$$

The importance of this transformation is that $g(x)$ is linear in its parameters, may be continuous, and may also range from $-\infty$ and $+\infty$ depending on the range of x.

Hosmer Jr et al. [26] argue that the second important difference between linear and logistic regression models is the distribution of the outcome variable. In linear the outcome variable may be expressed as $y = E(Y|e) + \varepsilon$, where the quantity ε is called *error* and expresses an observations deviation from the conditional mean. While in dichotomous outcome variable, they express the value of the outcome variable given x as $y = \pi(x) + \varepsilon$, where ε assumes one of two possible values.

Decision Trees

A decision tree is a decision-making device [40] which assigns a probability to each of the possible choices based on the context of the decision: $P(f|h)$, where f is an element of the $future$ vocabulary (the set of choices) and h is a *history* (the context of the decision). This probability $P(f|h)$ is determined by asking questions $q_1, q_2, ..., q_n$

Magerman [40] explains that parsing a natural language sentence can be viewed as making a sequence of decisions, for example determining the part-of-speech of the words, choosing between constituent structures, and selecting labels.

The probability of a complete parse tree (T) of a sentence (S) is the product of each decision (d_i) conditioned on all previous decisions [40]:

$$P(T|S) = \prod_{d_i \epsilon T} P(d_i|d_{i-1}d_{i-2}...d_1 S) \tag{2.8}$$

Figure 2.5 presents an illustration of a decision tree for part-of-speech tagging. Each question asked by the decision tree is represented by a *tree node* (oval in the figure) and the possible answers to the question are associated with branches emanating from the node. Each node defines a probability distribution on the space of possible decisions. A node where the decision tree stops asking questions is a *leaf node*. The leaf nodes represent the unique states in the decision-making problem, all contexts which lead to the same leaf node have the same probability distribution.

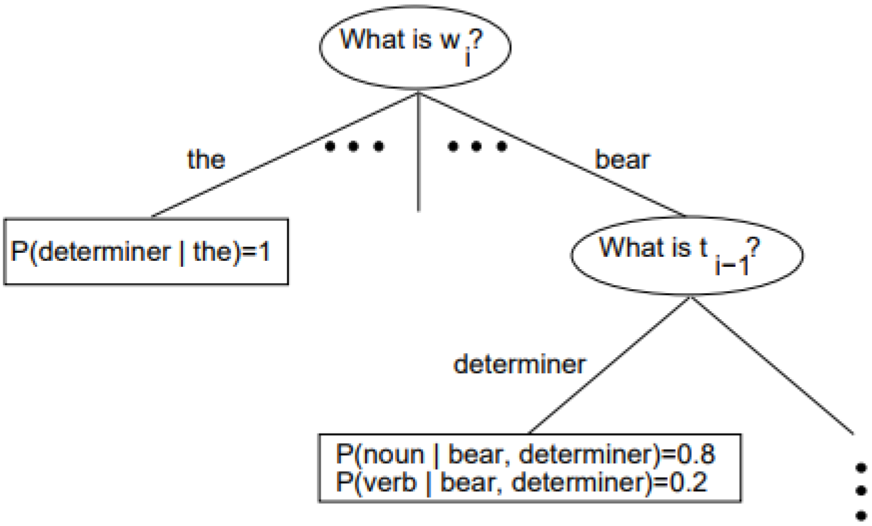


Figure 2.5: An illustration of decision tree for part-of-speech tagging. The figure is from Magerman [40].

Breiman [10] describes *Random Forest* as a combination of tree predictors, where each tree depends on the values of a random vector sampled independently and with the same distribution of all trees in the forest. He defines a Random Forest as classifiers which consists of a collection of tree-structured classifiers $\{h(\mathbf{x}, \Theta_k),\ k = 1, ...\}$ where the Θ_k are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input $\mathbf{x}$. Figure 2.6 presents an illustration of Random Forest ensemble classification.

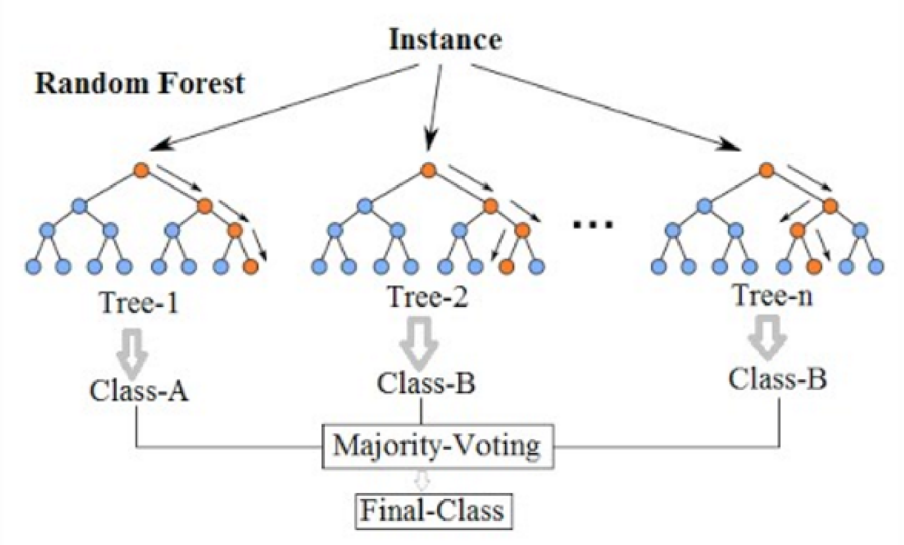


Figure 2.6: An illustration of Random Forest ensemble classification. The figure is from Koehrsen [33].

Given an ensemble of classifiers $h_1(\mathbf{x})$, $h_2(\mathbf{x})$,..., $h_k(\mathbf{x})$, and with the training set drawn at random from the distribution of the random vector **X, Y**, define the margin function as:

$$mg(\mathbf{X}, Y) = av_k I(h_k(\mathbf{X}) = Y) - \max_{j \neq Y} av_k I(h_k(\mathbf{X}) = j) \tag{2.9}$$

where $I(\cdot)$ is the indicator function. The margin measures the extent to which the average number of votes at **X**, Y for the right class exceeds the average vote for any other class. The larger the margin, the more confidence in the classification.

Neural Networks

Sebastiani [52] defines a neural network text classifier as a network of units, where the input units represent terms, the output units represent the category of interest, and the weights on the edges connecting units represent dependence relations.

He explains that for classifying a text document d_j, its term weights w_kj are loaded into the input units. The activation of these units is propagated forward through the network, and the value of the output units determine the categorisation decisions. A typical way of training neutral networks is *backpropagation*, where the term weights of a training document are loaded into the input units, and if a misclassification occurs the error is "backpropagated" to change the parameters of the network and minimise the error.

Sebastiani [52] mentions two methods for learning linear classifiers, *batch methods* and *on-line methods*. Batch methods build a classifier by analysing the training set all at once, while on-line methods built a classifier soon after examining the first training document and incrementally refine it as they examine new ones.

A simple type of neural network classifier is the *perceptron* algorithm, where the classifier for c_i is first initialised by setting all weights w_ki to the same positive value. When training example d_j is examined, the classifier built so far classifies it. If the results of the classification are correct nothing is done, but if it is wrong, the weights of the classifier are modified. So, if d_j was a positive then the weights w_ki of active terms are "promoted" by increasing them by a fixed quantity $\alpha > 0$, which is called *learning rate*, while if d_j, was a negative example, then the same weights are "demoted" by decreasing them by α.

A *multiplicative* variant differs from perceptron because of two different constants $\alpha_i > 1$ and $0 < \alpha_2 < 1$ are used for promoting and demoting weights, respectively, and because promotion and demotion are achieved by multiplying, instead of adding by α_1 and α_2.

Sebastiani [52] also mentions that other types of linear neural network classifiers with a form of logistic regression has also been proposed and tested and show good effectiveness. And non-linear neural network is instead a network with one or more additional "layers" of units, which usually represent higher-order interactions between terms that the network is able to learn but have shown very small improvements.

2.3.3 Text representation

Joachims [29] explains that the first step in text classification is to transform the text documents. These are typically strings of characters and must be transformed into a representation suitable for learning algorithms and the classification task. He also mentions that a representation scheme can lead to very high-dimensional feature spaces, and that many have noted the need for feature selection to make the use of conventional learning methods possible, to improve generalisation accuracy and to avoid "over-fitting".

The most common approach for representing text is the Bag-Of-Words (BOW) model, in which the word order does not matter. The text is broken down into words, where each word represents a feature and are thrown in a "bag", losing the sequence information in the process. Joachims [29] describes three text representation models, *Term Vector Model*, *N-Grams Model*, and *N-Grams Graphs Model*. In the next sub-sections we will look into his explanation of these.

Term Vector Model

Term vector model is employed as following in text classification: given a collection of documents D, it aggregates the set of distinct terms (words) W. Each document $d_i \epsilon D$ is then represented as a vector $V_{d_i} = (v_1, v_2, ..., v_{|W|})$ of size $|W|$ with its j-th element v_j quantifying the information the j-th term $w_j \epsilon W$ conveys for d_i.

The term information in each element can come in three forms:

- A binary value to indicate the existence or absence of a term in the corresponding document.

- A number indicating the value of occurrences of a term in a document. This is known as *Term Frequency (TF).*

- A value that takes both into account, the number of occurrences of a term in a document and its overall frequency in the entire corpus, also known as (Term Frequency-Inverse Document Frequency (TF-IDF). This is done to reduce the impact of particularly common words (stop words, such as "and", "or", "a", etc.)

N-Grams Model

The N-Grams Model comes in two forms, the *character n-grams* model, which relies on sequences of distinct letters, and the *word n-grams* model, which relies on sequences of distinct words. The set of character n-grams of a word or sentence deals with all substrings of length n of the original text. A document d_i is represented by a vector where the j-th element contain information from its n-gram for d_i.

Unlike the term vector model, the frequency of an n-gram is commonly used to quantify this information. Typical values for n are bigrams (2), trigrams (3), and fourgrams (4). If we use "telephone" as an example, the word would consist of the following trigrams: tel, eph, one.

N-Gram Graphs Model

The idea behind N-Gram Graphs model is that the bag model of character n-grams does not consider the order of characters' appearance in the text. Resulting in words or documents with different character sequences end up having identical or similar representations. N-gram graphs model solves this problem by neighbouring pairs of n-grams with edges that represents their frequency of co-occurrence.

2.4 Summary

We present in this chapter background information on social media and its growth. We have also discussed customer feedback, how these may never reach the developers and the impact customer feedback may have on other customers. We also present the most common classifiers in text classification in sentiment analysis. Lastly, we look at the different text representation methods for the training data.

Chapter 3

Related Work

This chapter presents related work and their approach in both social media and on reviews. We also present some of the available tools that are using sentiment analysis in social media.

3.1 Sentiment Analysis in Social Media

Da Silva et al. [13] experiment on a tweet dataset with an ensemble formed by Multinomial Naive Bayes, Support Vector Machines, Random Forest and Logistic Regression. Their main contribution was to show that classifier ensembles formed by diversified components are promising for tweet sentiment analysis. Their approach was that once the classifiers had been trained, an ensemble was formed by either the average of the class probabilities obtained by each classifier or the majority voting. Figure 3.1 presents an overview of their approach. Da Silva et al. [13] concluded that ensembles formed by diversified components could provide state-of-the-art results on tweets.

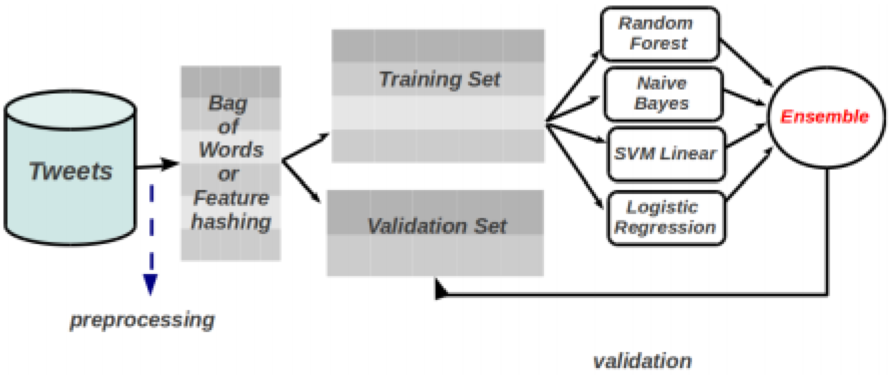


Figure 3.1: An overview of the approach by Da Silva et al. [13].

Vyrva [59] presents several machine learning techniques applied to sentiment analysis. She used three datasets containing tweets on different topics. Each of the classifiers were trained and tested separately. She used five common machine learning classification methods: Naive Bayes, Multinomial Naive Bayes, Support Vector Machine, Multilayer Perceptron Network and Random Forest classifier. She chose to compare the performance on the three twitter datasets based on the accuracy metric.

Her main goal was to compare standard machine learning methods for sentiment analysis of data collected from twitter, to find the most accurate classifiers. A summary of some of the results achieved in her study is shown in table 3.1. She found that the best performance achieved on the overall datasets were Multinomial Naive Bayes and Support Vector Machine.

Features	Count of attributes	NB	MNB	SVM	RF	MLP
unigram	2897	73.83	**80.60**	**80.60**	**80.60**	76.99
bigram	6404	74.89	76.54	76.54	76.54	**76.84**
trigram	6571	77.44	76.69	**79.69**	76.69	78.20

Table 3.1: The results achieved on one of the datasets from Vyrva's experiments. The bold-text highlights the best results.

Kiritchenko et al. [32] describe a sentiment analysis system that detects the sentiment of short informal text messages such as tweets and SMS, and the sentiment of a word or phrase within a message. Their system is based on a supervised classification approach. They obtain the sentiment features primarily from tweet-specific sentiment lexicons, which are automatically generated from tweets with sentiment-word hashtags and from tweets with emoticons. They generated a separate sentiment lexicon for negated word, to get the sentiment of words in negated contexts.

Their system ranked first in the SemEval-2013[1] shared task 'Sentiment Analysis in Twitter', obtaining an F-score of 69.02 in the message-level task and 88.93 in the term-level task. Their system also obtains state-of-the-art performance on two additional datasets: the SemEval-2013 SMS test set and a corpus of movie reviews. The F-score is a measure of a test's accuracy, where it considers both the precision and the recall of the test.

Sentiment analysis has also been applied to topics such as politics in social media. Wang et al. [60] present a system for real-time Twitter sentiment analysis of the U.S presidential election in 2012. They evaluate public tweets and news. They also mention that the system can be easily adopted and extended to other domains.

Asur and Huberman [4] demonstrate how sentiment analysis in social media can be used to predict real-world outcomes, such as box-office revenues for movies. They analyse the rate of tweets created about a topic can outperform market-based predictors and demonstrate how sentiments of the tweets can be further utilised to improve the forecasting.

3.2 Sentiment Analysis on Reviews

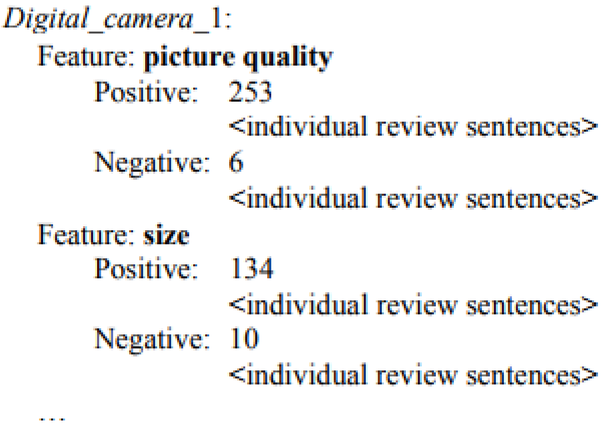
Pang et al. [48] investigate the problem of classifying documents on overall sentiment by using movie reviews as data. They examine the effectiveness of three machine learning techniques, Naive Bayes, Maximum Entropy, and Support Vector Machine classifiers in sentiment classification. They describe a challenge was that sentiment in movie reviews could be expressed in a more subtle manner. For example[2], "How could anyone sit through this movie?" which does not contain any negative words. They therefore conclude that sentiment requires more understanding.

[1] SemEval short for, Semantic Evaluation, is an ongoing series of evaluations of computational semantic analysis systems

[2] The example is from Pang et al. [48].

Hu and Liu [28] study the problem of generating feature-based summaries of customer reviews of products that are sold online. They divide the task into three steps:

1. Mining and identifying product features that customers have expressed their opinions on. Both data mining and natural language processing techniques are used to perform this task.
2. Identifying opinion sentences in each review and deciding whether the opinion is positive or negative, this process has been divided into three subtasks:
   (a) A set of adjective words (which are normally used to express opinions) is identified using a natural language processing method.
   (b) For each opinion word, they determine its semantic orientation, e.g., positive or negative. A bootstrapping technique is proposed to perform this task using WordNet.
   (c) Decide the opinion orientation of each sentence.
3. Lastly, summarising the results, in a format shown in figure 3.2



*Digital_camera_*1:
Feature: **picture quality**
Positive: 253
<individual review sentences>
Negative: 6
<individual review sentences>
Feature: **size**
Positive: 134
<individual review sentences>
Negative: 10
<individual review sentences>
…

Figure 3.2: An example of feature-based summary on a product from Hu and Liu [28]

Dave et al. [15] develop an opinion mining tool on product reviews and generate a list of product attributes and aggregating opinions about each of them. They identify unique properties in the reviews and develop a method for automatically classifying them either positive and negative.

Fang and Zhan [19] focus on the problem of polarity categorisation in data from product reviews collected from Amazon.com. They experiment on both sentence-level and document-level, with the use of Naive Bayes, Random Forest, and Support Vector Machine and achieved promising results.

Altrabsheh et al. [1] propose a system for analysing students feedback using sentiment analysis. They focus on finding the best model for automatic analysis and look at the following aspects: pre-processing, features and machine learning techniques. They collected feedback from students in lectures at their own university and from other various institutes. Students were asked to submit their feedback, opinions and feelings about a lecture.

The data was then labelled by two linguistic experts and one expert in sentiment analysis. They used Support Vector Machine, Maximum Entropy and Naive Bayes classifiers, and found that the highest results were given by SVM.

3.3 Sentiment Analysis Tools

There are various online tools available online that are using sentiment analysis in social media. We will look at some of these and what they offer in this section.

3.3.1 Tweet Sentiment Visualisation

Healey and Ramaswamy [25] had a specific goal to visualise and present basic emotional properties in text and measure the confidence in the estimates. They developed a web-application that visualise the sentiment of tweets posted on Twitter. Figure 3.3 presents an example of the web-application showing sentiment visualisation of the keyword “iPhone”. Each circle’s colour, brightness, size and transparency visualise different details about the sentiment of its tweet. The colour represents the overall pleasure of the tweet. Green are pleasant, and blue are unpleasant. The brightness represents the overall arousal of the tweet. Active tweets are brighter, and subdued tweets are darker. The size is a measure of how confident the tool is on the tweet’s sentiment is. Larger tweets represent more confident estimates. Transparency is another measure of how confident the tool is. Less transparent tweets represent more confident estimates.

Each of the tabs at the top of figure 3.3 visualise the tweets in different ways [25]:

Sentiment Each tweet is shown as a circle positioned by sentiment, an estimate of the emotion contained in the tweet’s text. Unpleasant tweets are drawn as blue circles on the left and pleasant tweets are green circles on the right. Active tweets are drawn as brighter circles on the top and sedate tweets are drawn as darker circles on the bottom.

Topics Tweet about a common topic are grouped into topic clusters. Keywords above a cluster indicate its topic. Tweets that do not belong to a topic are visualised as singletons on the right.

Heatmap Pleasure and arousal are used to divide sentiment into a grid. The number of tweets that lie within each grid cell are counted and used to colour the cell, red for more tweets than average, and blue for fewer tweets than average.

Tag Cloud Common words from the emotional regions. Upset, happy, relaxed and unhappy are shown. Words that are more frequent are larger.

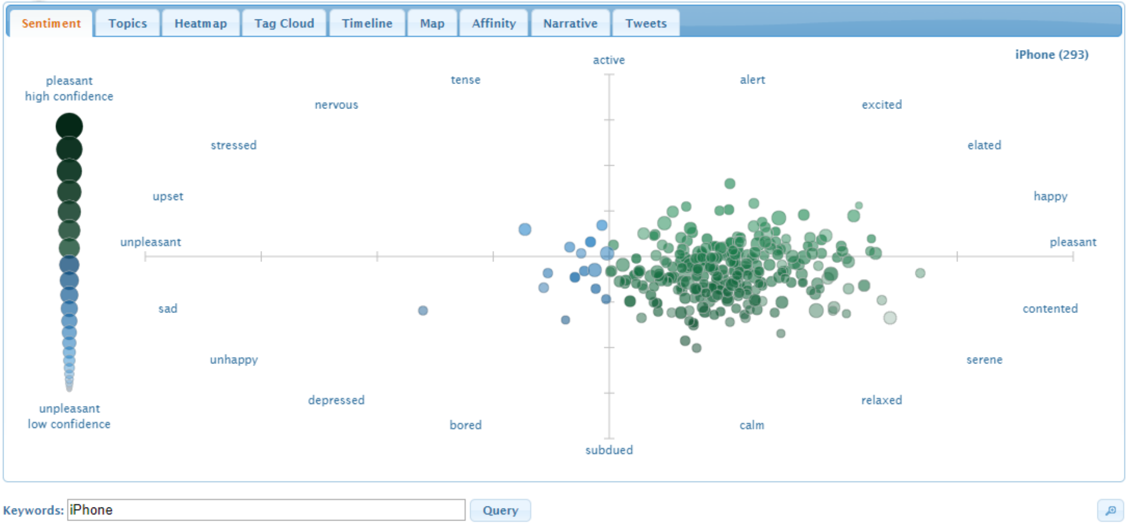
Timeline Tweets are drawn in a bar chart to show the number of tweets posted at different times. Pleasant tweets are shown in green on the top of the chart, and unpleasant tweets are shown in blue on the bottom.

Map Tweets are drawn on a map of the world at the location where they were posted.

Affinity Frequent tweets, people, hashtags, and URLs are drawn in a graph to show important actors in the tweets and any relationship or affinity they have to one another.

Narrative Displays a time-ordered sequence of tweets that form conversations or narrative threads passing through a selected tweet.

Tweets Tweets are listed to show their date, author, pleasure, arousal, and text.



Keywords: iPhone

Figure 3.3: An example of the web-application, Tweet sentiment visualisation, showing results on the keyword “iPhone”.

To estimate the sentiment, they use a dictionary that report the sentiment of a set of words along one or more emotional dimension. Their sentiment dictionary provides measures of valence and arousal for over 10 000 words, where each word is rated on scale ranging from 1 to 9. Ratings for a word are combined into a mean rating and a standard deviation of the ratings for each dimension. For example, given the word **house**:

$$\textbf{house}, v = (\mu : 7.26, \sigma : 1.72), a = (\mu : 4.56, \sigma : 2.41), fq = 591$$

This shows that **house** has a mean valence v of 7.26 and a standard deviation of 1.72, a mean arousal a of 4.56 and a standard deviation of 2.41, and a frequency fq of 591 ratings.

Given their dictionary, they use the following steps to estimate an overall valence and arousal for each tweet:

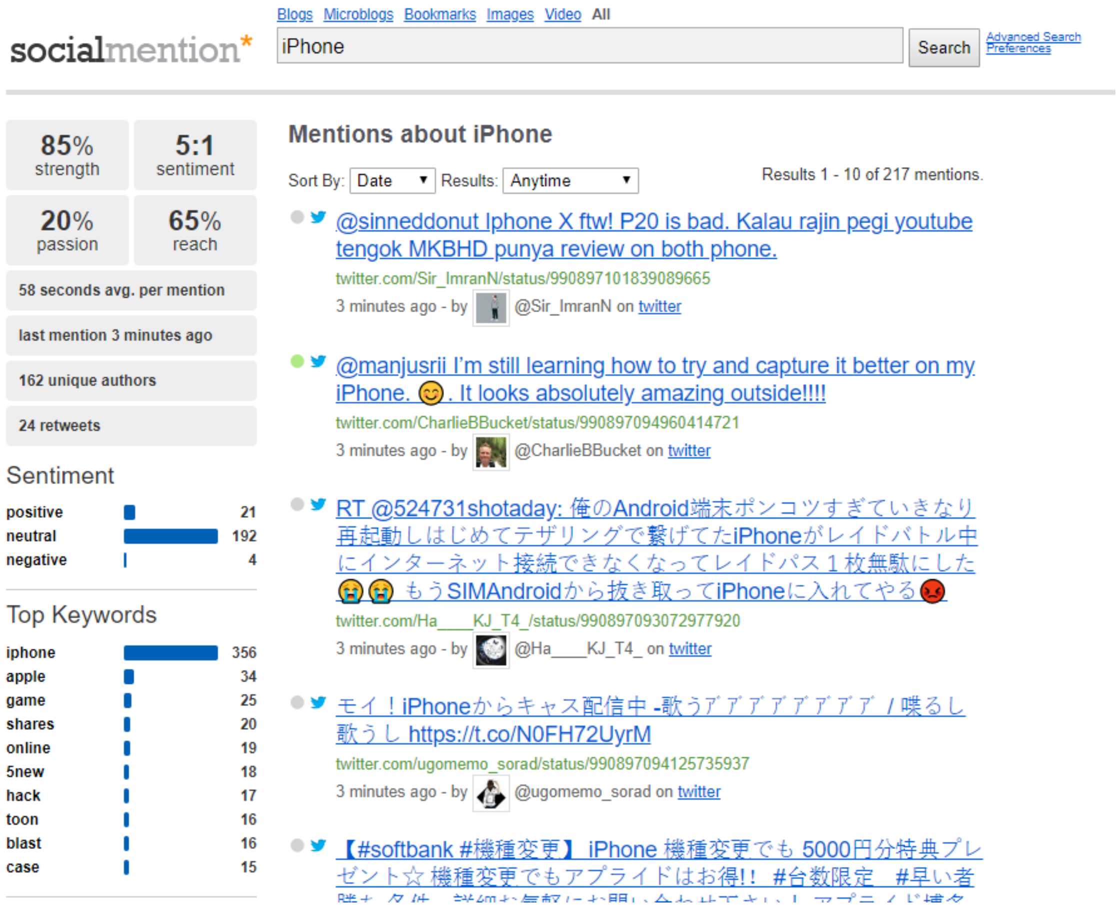
1. For each word w_i in the tweet that exists in the sentiment dictionary, save the word’s mean valence and arousal $\mu_{v,i}$ and $\mu_{a,i}$ and standard deviation of valence and arousal $\sigma_{v,i}$ and σ_{a_i}.

2. If a tweet contains less than $n = 2$ sentiment words, they ignore the tweet for having an insufficient number of ratings to estimate its sentiment.

3. Statistically average the n means and standard deviations to compute the tweet’s overall mean valence and arousal M_v and M_a.

3.3.2 Social Mention

Social Mention is a social media search engine for user-generated content across multiple platforms such as blogs, social networks and forums. It allows users to track a keyword in social media and displays sentiment, strength, passion and reach. Figure 3.4 presents an example of the tool showing results of a search using the keyword "iPhone". *Social Mention* explain the measurements as following:



iPhone

85% strength
5:1 sentiment
20% passion
65% reach
58 seconds avg. per mention
last mention 3 minutes ago
162 unique authors
24 retweets

Sentiment

positive	21
neutral	192
negative	4

Top Keywords

iphone	356
apple	34
game	25
shares	20
online	19
5new	18
hack	17
toon	16
blast	16
case	15

Results 1 - 10 of 217 mentions.

@sinneddonut Iphone X ftw! P20 is bad. Kalau rajin pegi youtube tengok MKBHD punya review on both phone.
twitter.com/Sir_ImranN/status/990897101839089665
3 minutes ago - by @Sir_ImranN on twitter

@manjusrii I'm still learning how to try and capture it better on my iPhone. 😊. It looks absolutely amazing outside!!!!
twitter.com/CharlieBBucket/status/990897094960414721
3 minutes ago - by @CharlieBBucket on twitter

RT @524731shotaday: 俺のAndroid端末ポンコツすぎていきなり再起動しはじめてテザリングで繋げてたiPhoneがレイドバトル中にインターネット接続できなくなってレイドパス１枚無駄にした😭😭 もうSIMAndroidから抜き取ってiPhoneに入れてやる😡
twitter.com/Ha____KJ_T4_/status/990897093072977920
3 minutes ago - by @Ha____KJ_T4_ on twitter

モイ！iPhoneからキャス配信中 -歌うア゛ア゛ア゛ア゛ア゛ア゛ア゛ア゛ / 喋るし歌うし https://t.co/N0FH72UyrM
twitter.com/ugomemo_sorad/status/990897094125735937
3 minutes ago - by @ugomemo_sorad on twitter

【#softbank #機種変更】 iPhone 機種変更でも 5000円分特典プレゼント☆ 機種変更でもアプライドはお得!! #台数限定 #早い者

Figure 3.4: An example of the web-application, Social Mention, showing results on the keyword "iPhone".

Sentiment: is the ratio of mentions that are generally positive to those that are generally negative.

Strength: is the likelihood that the keyword is being discussed in social media. They calculate this on phrase mentions within the last 24 hours divided by total possible mentions.

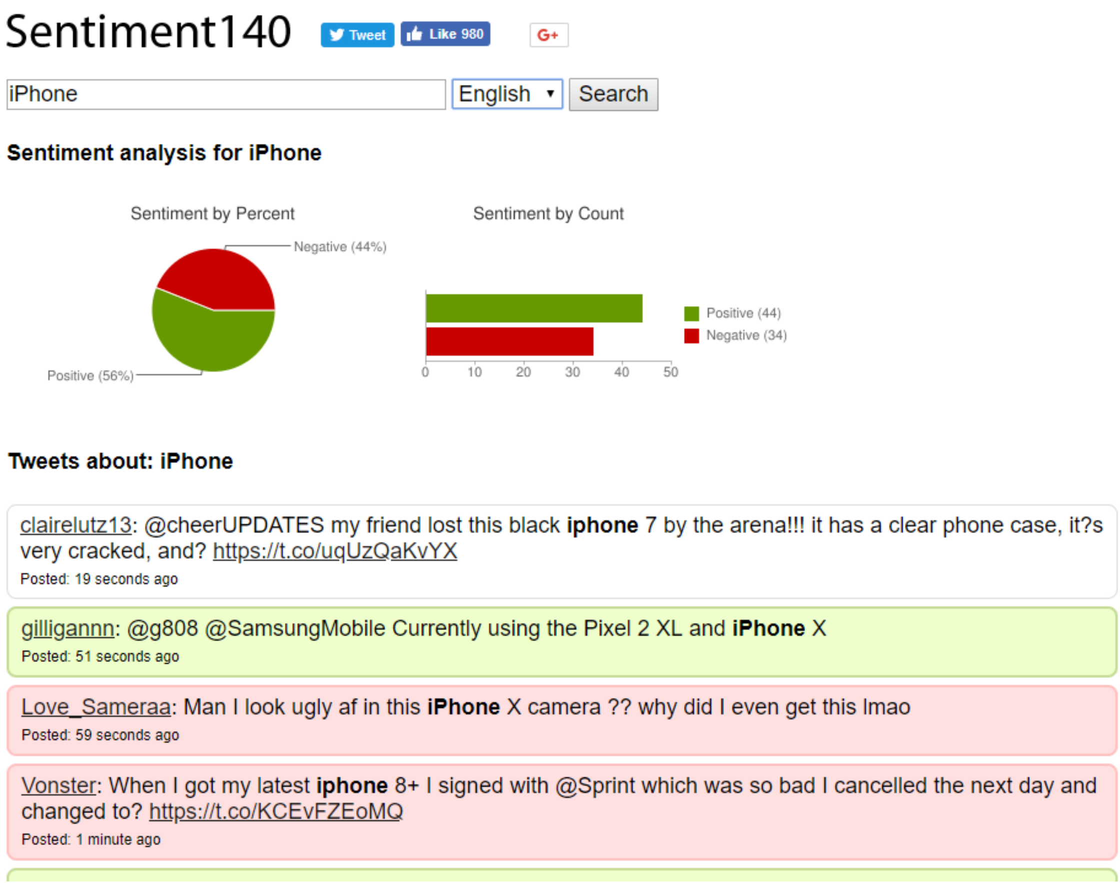
Passion: is the measure of likelihood that individuals talking about the keyword will do so repeatedly. For example, if there is a small group who talk about the specific keyword all the time, the passion score will be higher. And conversely, if every mention is written by a different author, the passion score will be lower.

Reach: is the measure of the range of influence. It is the number of unique authors referencing the keyword divided by the total number of mentions.

This tool includes several measurements. It searches forums, blogs, and social media sites. It includes several statistics on the results, such as top keywords, sentiment count and more. The results are shown in a list sorted on date. The sentiment of the text is displayed next to it in a small circle coloured grey for neutral, green for positive, or red for negative.

3.3.3 Sentiment140

Go et al. [24] developed an application using sentiment analysis on tweets in Twitter. The application support English and Spanish language. Figure 3.5 presents an example of the web-application *Sentiment140* showing results based on the keyword "iPhone". They use Maximum Entropy classifier for classifying the tweets. Their training data was automatically created and was not annotated by humans. Their approach was to assume that any tweet with emoticons, such as ":)", were positive tweets, and tweets with negative emoticons, such as ":(", were negative. They collected these using the Twitter Search API.



Sentiment analysis for iPhone

Tweets about: iPhone

clairelutz13: @cheerUPDATES my friend lost this black **iphone** 7 by the arena!!! it has a clear phone case, it?s very cracked, and? https://t.co/uqUzQaKvYX
Posted: 19 seconds ago

gilligannn: @g808 @SamsungMobile Currently using the Pixel 2 XL and **iPhone** X
Posted: 51 seconds ago

Love_Sameraa: Man I look ugly af in this **iPhone** X camera ?? why did I even get this lmao
Posted: 59 seconds ago

Vonster: When I got my latest **iphone** 8+ I signed with @Sprint which was so bad I cancelled the next day and changed to? https://t.co/KCEvFZEoMQ
Posted: 1 minute ago

Figure 3.5: An example of the web-application, Sentiment140, showing results on keyword "iPhone".

In the application, they visualise the results by percent and by count. The red colour for negative tweets and green colour for positive tweets. We can also see that the first result in the figure, has a white background, which represent neutral class. They have decided to only include 10% of the neutral results in the application.

This application visualises the results in a simple design. All tweets are listed and sorted on the date it was published. Each tweet displays the author of the tweet, the date it was published, the tweet itself and a background colour to represent if the tweet is positive or negative. They also include a pie chart and a bar graph which gives the user an overview of the results.

3.4 Summary

We have in this chapter presented some related work on sentiment analysis in social media and on customer reviews and feedback. Most of the related work focus mainly on finding the best classifiers or achieving higher accuracies. We also present three online tools that are using sentiment analysis in social media and discuss how they visualise the results.

Chapter 4

Prototype

This chapter presents the prototype developed for the study in this thesis.

4.1 Requirements

We decided to set some requirements for our prototype. We have set the following requirements:

- The user will need the ability to perform a search in social media with a specific keyword.
- The user can view the results and statistics of the search with the sentiment.
- The prototype has to implement an ensemble of the classifiers, Support Vector Machine, Multinomial Naive Bayes and Random Forest.
- The classification will be performed on document-level to classify the overall sentiment.

We developed our prototype in Python. Python is an interpreted, interactive, object-oriented programming language [50], and offers a wide variety of third-party extensions. We have used the following available extensions: *Flask*, *Tweepy*, *Scikit-learn* and *Pandas*.

Flask is a microframework for Python and includes a built-in server and debugger with integrated unit testing support [20]. This extension allowed us to easily set up and create a web-interface. The web-interface allowed the user to search for a specific keyword and view the results.

Tweepy is an easy to use Python extension for accessing the Twitter API [57]. This extension was used in our prototype when performing a search with the keyword specified by the user. The extension simplified the process of authentication for accessing the Twitter API and also allowed us to send parameters, such as query, date and language, when accessing the Twitter API.

Scikit-learn is a machine learning extension for Python and offers efficient tools for data mining and data analysis [51], such as classifications, regressions, clustering and more. The extension includes all the classifiers that were needed to build the ensemble. It also allowed us to easily train the classifiers with the datasets.

Pandas is an open source Python extension providing high-performance, easy to use data structures and data analysis tools [47]. This extension allowed us to load and prepare the datasets for training the classifiers.

4.2 Datasets

Three datasets were used in the prototype. We decided to use the following datasets based on their topics and what they contain:

Dataset I - Tweets emojis This dataset contains 10000 tweets labelled positive and negative and was retrieved from the NLTK corpus "twitter_samples"[1]. The tweets were collected in July 2015 by searching against a list of emoticons (such as: :-), <3, :D for positive and :-(, ;(, >.< for negative). We refer to this dataset as *Dataset I* in this thesis.

Dataset II - Sanders tweets This dataset contains 5386 hand-classified tweets from Sanders Analytics[2] labelled positive, negative, neutral and irrelevant. But only the positive and negative were used, resulting in a total of 1091 tweets. The dataset contains tweets on topics from Apple, Google, Microsoft, and Twitter. We refer to this dataset as *Dataset II* in this thesis.

Dataset III - Android application reviews This dataset contains 19655 android application reviews. The dataset was uploaded to GitHub by Amit Tripathi[3]. The dataset was divided in two, where all the positive was in one file and all negative in another. The files were merged and randomised with their label intact. We refer to this dataset as *Dataset III* in this thesis.

4.2.1 Preprocessing

Some datasets included unnecessary additional data and some datasets were separated in two files. Dataset I was separated in two files, one file with all the positive labelled data and another with all the negative labelled data. It also contained all meta-data the Twitter API provides for a tweet. We extracted only the tweet text from both files and labelled it according to the file it was in. We then shuffled the dataset.

Dataset II included some tweets labelled *neutral* and *irrelevant*, which in our case was not needed. The tweets with these classes were not used and were filtered out. The dataset also contained some unnecessary data, such as tweet id, date, and topic, which had no use for us and were ignored.

Dataset III was also separated in two files, where all positive was in one file and all negative in another. We merged the two files and then shuffled the dataset.

Table 4.1 presents an overview of the datasets and what they contain.

[1] http://www.nltk.org/howto/twitter.html#Using-a-Tweet-Corpus

[2] http://www.sananalytics.com/lab/twitter-sentiment/

[3] https://github.com/amitt001/Android-App-Reviews-Dataset

Dataset	Total	Positive	Negative	Information
Dataset I	10000	5000	5000	Tweets containing emoticons.
Dataset II	1091	519	572	Tweets containing Apple, Google, Microsoft, or Twitter.
Dataset III	19655	9935	9720	Application reviews from the android app store.

Table 4.1: An overview of the three datasets and what they contain.

4.3 Social Media Integration

Several social media platforms allows integration using an application programming interface (API) to fetch data. An API is a set of subroutine definitions, protocols, and tools for building application software [2]. There are several types of APIs, but we have used a web-based system in our prototype. An API is typically defined by a set of specifications, such as a Hypertext Transfer Protocol (HTTP) request message, along with a response message, which is usually in an Extensible Markup Language (XML) or JavaScript Object Notation (JSON) format. The APIs are usually documented to facilitate usage and implementation. Examples of APIs are: News API[4], Twitter API[5] and Facebook API[6].

We only implement the *Search API* provided by Twitter in this prototype. It allowed us to specify a few parameters, such as query, date(s) and language in the request message to get better search results in the response message. The Search API returns the response in a JSON format with meta-data on each tweet. The Search API offers three tiers[54]: *Standard*, *Premium* and *Enterprise*.

Standard This search API searches against a sampling of recent tweets published in the past 7 days. Part of the 'public' set of APIs.

Premium Paid access to either the last 30 days of tweets or access to tweets from as early as 2006. Built on the reliability and full-fidelity of our enterprise data APIs.

Enterprise Paid access to either the last 30 days of tweets or access to tweets from as early as 2006. Provides full-fidelity data, direct account management support, and dedicated technical support to help with integration strategy.

The *Standard* version was more than good enough for our prototype. It returned the tweet, author of the tweet and the date it was posted, which is what we wanted. It also returned user information, such as profile picture, tweet URL, and much more meta-data, which was irrelevant for this study.

4.4 Classification Process

We have created an ensemble of classifiers with scikit-learn. The ensemble consists of Support Vector Machine, Multinomial Naive Bayes and Random Forest classifiers. Each

[4] `https://newsapi.org/`

[5] `https://developer.twitter.com/`

[6] `https://developers.facebook.com/docs/graph-api`

of the algorithms were trained on the three datasets using the bag-of-words model. This gave us an ensemble with nine unique classifiers as shown in table 4.2.

The classification of a tweet received from the Twitter API was done as following:

1. The tweet is represented using the bag-of-words model.

2. Each of the classifiers in the ensemble will then predict the sentiment of the tweet.

3. All of the predictions are then stored in an array.

4. The array will be looped through and a simple majority vote will be performed to decide whether to label the tweet positive or negative.

Figure 4.1 presents an illustration of how a tweet is classified.

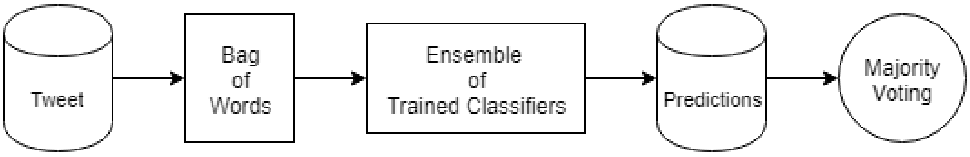


Figure 4.1: The classification process of a tweet.

For example, if we have the following text: “I just bought a new phone and it’s amazing!”. All nine classifiers will give their predictions. If five of the predictions are positive and four predictions are negative, then positive will be returned.

Classifier	**Train set**
SVM1	Dataset I
SVM2	Dataset II
SVM3	Dataset III
MNB1	Dataset I
MNB2	Dataset II
MNB3	Dataset III
RF1	Dataset I
RF2	Dataset II
RF3	Dataset III

Table 4.2: An overview of all of the solvers and their train set.

We decided to use an ensemble of nine classifiers to achieve the best classification results. To confirm that our ensemble gave better results, we compared the accuracy on the datasets. Each classifier was tested on each dataset, as well as the ensemble.

Table 4.3 presents a comparison of the results on the different datasets from each classifier and the ensemble. The ensemble achieved almost 4% better accuracy then RF on Dataset I and over 5% better accuracy on Dataset III. Table 4.4 presents the average accuracy from the ensemble and the classifier on all of datasets.

	Dataset I	Dataset II	Dataset III
SVM:	0.7552	0.8242	0.8708
MNB:	0.7652	0.8388	0.8889
RF:	0.7364	0.8315	0.8413
Ensemble:	**0.7756**	**0.8498**	**0.8915**

Table 4.3: The accuracy from each of the classifiers and the ensemble on each of the datasets. Bold text highlights the best achieved results.

	Average accuracy
SVM:	0.8167
MNB:	0.831
RF:	0.8031
Ensemble:	**0.839**

Table 4.4: The average accuracy from each of the classifiers and the ensemble on all of the datasets. Bold text highlights the best achieved results.

4.5 Prototype Processes

Figure 4.2 presents the processes in the prototype. The "Pre-processing" will activate once the prototype launches. This includes loading and preparing the datasets and training our ensemble of classifiers.

The prototype will then wait for the user to input a search query before connecting to the Twitter API and fetching tweets based on the query. The received tweets will be classified as mentioned in previous section, by our ensemble before returning the tweets and their sentiment to the user interface. The user may then perform a new search.

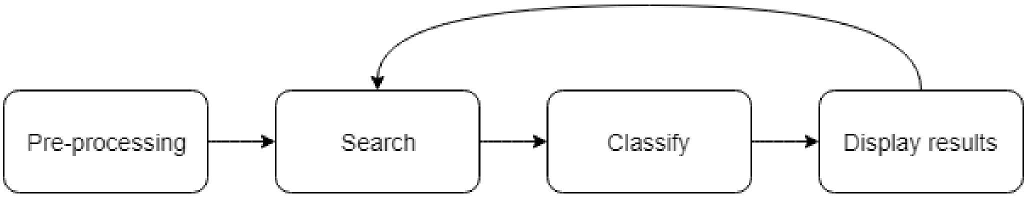


Figure 4.2: An overview of the processes in the prototype.

4.6 User Web-Interface

The user web-interface was developed in HTML, CSS and JavaScript. We describe the interface and the tools in this section.

4.6.1 Interface

The user web-interface was accessible from any web-browser. When first visiting the interface at launch it will look as shown in figure 4.3. At the top we have a navigation bar with the name of the prototype, "*Tweet Sentiment Analyser*". Just below it, we have the input text field for the user to type in a specific query to search for. On the right side of the input field there is a blue button, "*Search Twitter and analyse*", which activates our function for connecting to the Twitter API and passes the query as a parameter in the request message.

The interface also includes some examples to help the user in constructing a more advanced query for filtering the results. The examples are shown in figure 4.4.

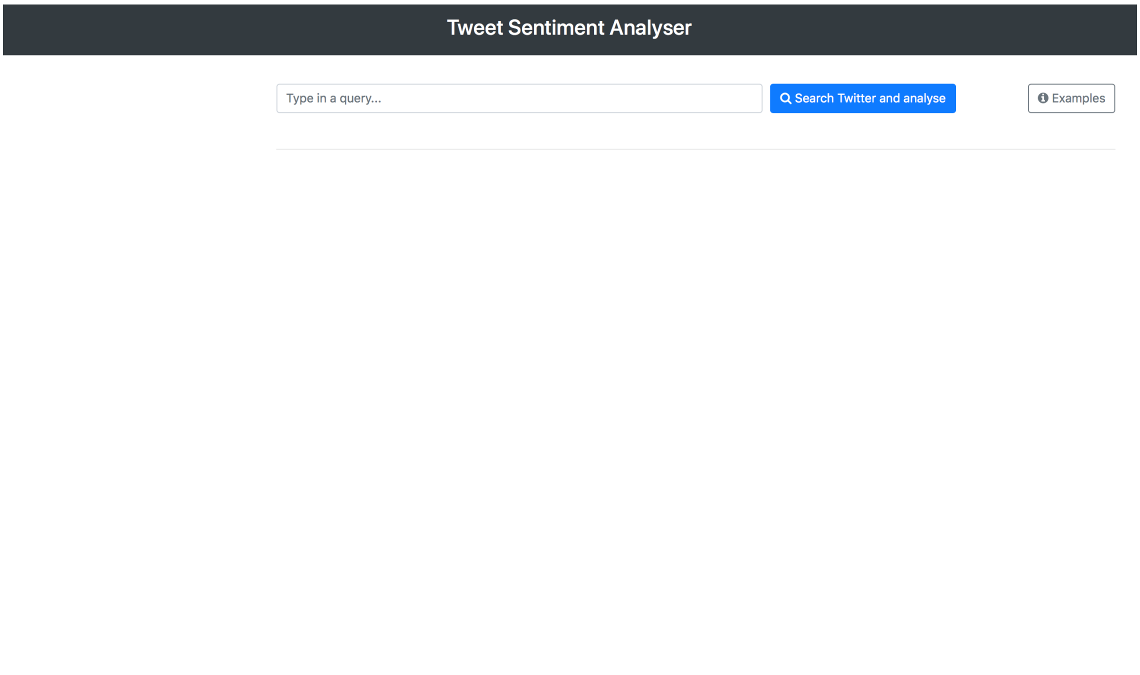
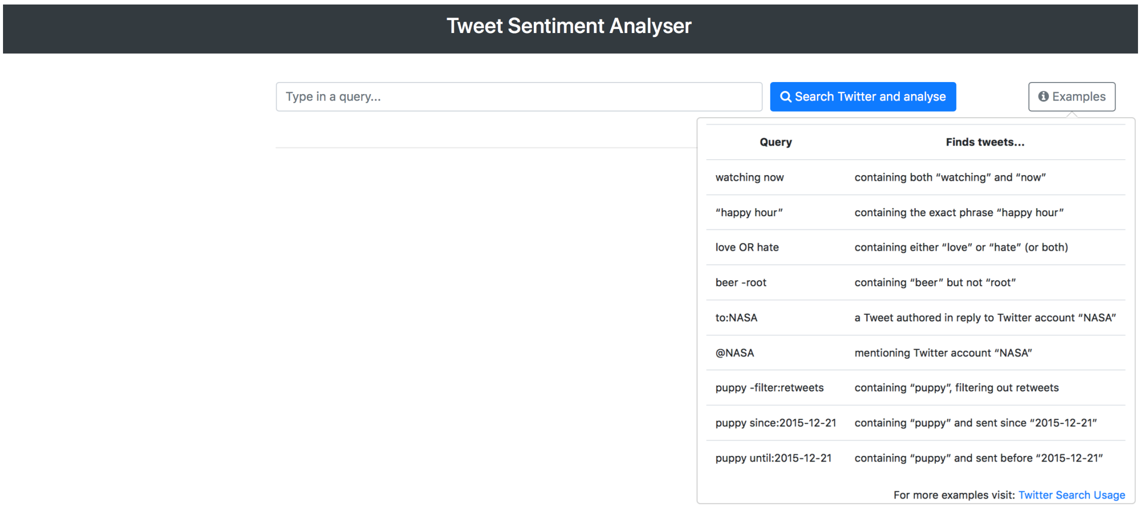


Figure 4.3: The prototype web-interface before a search has been done.

The received tweets will then be classified by our ensemble. Once the results are ready and all tweets have been classified, they will be displayed below the horizontal line, with their twitter user-name and the date and time it was posted. The background of the tweet will have a light-blue colour if the sentiment is positive, or a light-red colour if the sentiment is negative.

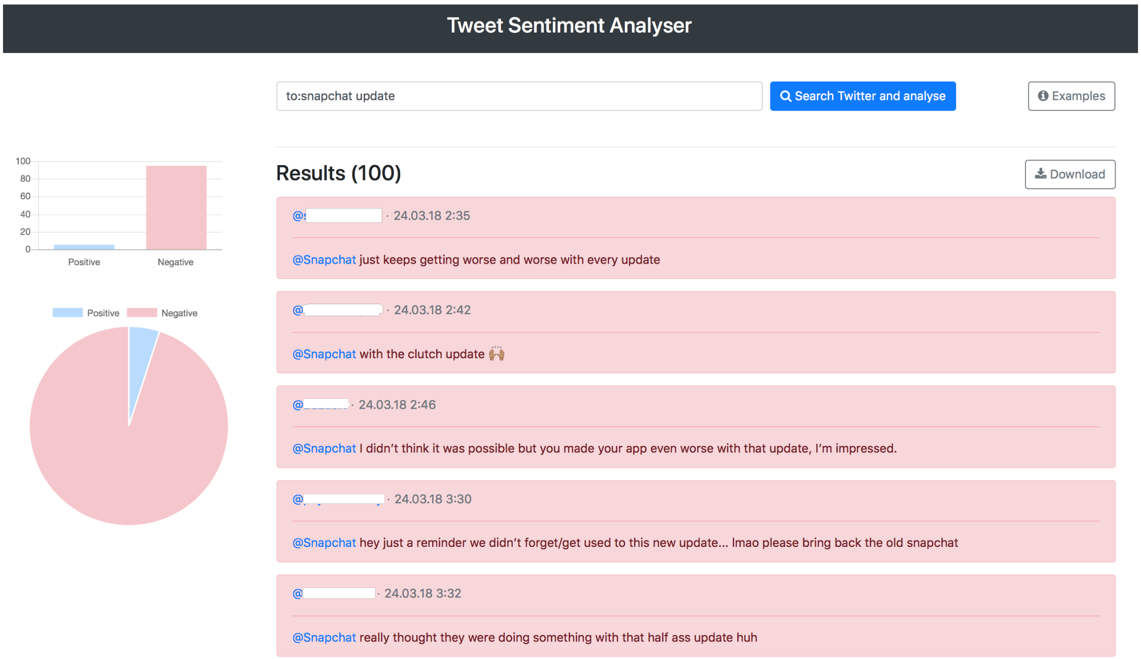
Figure 4.5 shows how the web-interface looks like after the user has performed a search, and the results are displayed on the web-interface with their sentiment. In this example with the query "to:snapchat update", which gave us tweets sent to *snapchat* that mention the word *update*.



Query	Finds tweets...
watching now	containing both "watching" and "now"
"happy hour"	containing the exact phrase "happy hour"
love OR hate	containing either "love" or "hate" (or both)
beer -root	containing "beer" but not "root"
to:NASA	a Tweet authored in reply to Twitter account "NASA"
@NASA	mentioning Twitter account "NASA"
puppy -filter:retweets	containing "puppy", filtering out retweets
puppy since:2015-12-21	containing "puppy" and sent since "2015-12-21"
puppy until:2015-12-21	containing "puppy" and sent before "2015-12-21"

For more examples visit: Twitter Search Usage

Figure 4.4: The prototype web-interface before a search has been done and with the examples provided.



Results (100)

@[redacted] · 24.03.18 2:35

@Snapchat just keeps getting worse and worse with every update

@[redacted] · 24.03.18 2:42

@Snapchat with the clutch update 🙌

@[redacted] · 24.03.18 2:46

@Snapchat I didn't think it was possible but you made your app even worse with that update, I'm impressed.

@[redacted] · 24.03.18 3:30

@Snapchat hey just a reminder we didn't forget/get used to this new update... lmao please bring back the old snapchat

@[redacted] · 24.03.18 3:32

@Snapchat really thought they were doing something with that half ass update huh

Figure 4.5: The prototype web-interface after a search has been done with the query "to:snapchat update" in this example. The user-names in the results has been hidden.

4.6.2 Tools

We have used the following extensions and tools for usability and design purposes:

Bootstrap v4.0 is free and open source front-end framework for developing with HTML, CSS, and JavaScript. Bootstrap offers a responsive grid system, prebuilt components and plugins built on JavaScript [8]. Bootstrap was used to create the design of the prototype, such as the buttons, navigation bar, input form, and for displaying each tweet.

Chart JS offers an easy way to include animated, interactive graphs on websites [12]. The tool was used to create the charts displayed on the interface.

Moment JS offer an easy solution to parse and display dates and times in JavaScript [44]. The tool was used to format the date and time received from the Twitter API of each tweet to be displayed in a Norwegian format.

Linkify is a JavaScript plugin for finding links in plain-text and converting them to HTML <a> tags [35]. This tool was very efficient for formatting URLs in the tweet text into clickable links.

Font Awesome is one of the largest icon toolkit and offers a large library with icons to be used in websites [21]. Icons were used in the prototype to inform the user about the functionality behind each button.

4.7 Source code

To keep our application organised and structured, we separated the source code into three python files, *main.py*, *twitter_api.py*, and *classifier.py*, and one html file *index.html*. In this section we present some of the source code from the python files.

We do not present the full source code of the prototype in this thesis. We have therefore uploaded it to GitHub under the repository *TweetSentimentAnalyser*[7].

Listing 4.1 presents some part of the source code in the *classifier.py* file. In this file, we first import our extensions for the classification process, *pandas* and *scikit-learn*. We then load and prepare the datasets for training and initialise our classifiers. We then create a function, *train*, which will train all of the classifiers with the datasets. We also created another function, *classify*, which takes one parameter, the tweet to be classified. The tweet is represented using the bag-of-words model, before all nine classifiers predict the sentiment of the tweet. All predictions are then stored in an array, before it is looped through to perform a simple majority vote to decide whether to return positive or negative.

```python
# Filename: classifier.py

from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn import svm
from sklearn.naive_bayes import MultinomialNB
from sklearn.ensemble import RandomForestClassifier
from sklearn import metrics
import pandas as pd

# Load and prepare datasets using pandas
...
```

[7] https://github.com/Mohamad93/TweetSentimentAnalyser

```
# Initialise the nine classifiers
...

def train():
    # Train all nine classifiers
    ...

# Classify tweet based on the majority of predictions
def classify(tweet):
    # Nine classifiers predict the sentiment of the tweet
    ...

    # Majority Voting process
    predictions = [svm1, svm2, svm3, nb1, nb2, nb3, rf1, rf2, rf3]
    pos_count = 0
    neg_count = 0
    for x in predictions:
        if(x == 'positive'):
            pos_count = pos_count + 1
        if(x == 'negative'):
            neg_count = neg_count + 1

    if(pos_count>neg_count):
        return 'positive'
    else:
        return 'negative'
```

Listing 4.1: The prototype source code in python from the *classifier.py* file. Only the majority voting process source code has been included in this listing, but comments for the remaining code has been included for understanding the processes in the file.

Listing 4.2 presents our *twitter_api* file. We import our *classifier* file and *tweepy* for authentication and requesting the tweets. The authentication must include four keys that are generated by creating a Twitter application on their website. We have not included the four keys in this listing but show how the authentication was done.

In our *search_tweets* function is where we authenticate our request to receive a response from the Twitter API. We then send our request with the parameter, the query specified by the user. The Twitter API response consist of all tweets found based on the query with meta-data for each tweet. We loop through all tweets and extract the tweet text, author of the tweet, and the date. We call our *classify* function which we import from our *classifier.py* file, to get the sentiment of the tweet. We insert all the tweets with its data and sentiment in an array which is then returned once the loop is completed.

```
# Filename: twitter_api.py

import classifier
from tweepy import OAuthHandler

# Access and search Twitter API with a query
# Return tweets with text, user name, date and the sentiment
def search_tweets(query):
    tweets = []
    # Authentication for accessing Twitter API
    ckey = "..."
    csecret="..."
    atoken=”...”
    asecret=”...”

    auth = OAuthHandler(ckey, csecret)
    auth.set_access_token(atoken, asecret)
    api = tweepy.API(auth)

    try:
        # Search Twitter using the query written in the interface
        # 'query' is received as a parameter from the interface
        public_tweets = api.search(q=query, lang='en', count=100)

        # We only need the text, user name and date of each tweet
        # Insert these into tweets array with the sentiment
        for tweet in public_tweets:
            parsed_tweet = {}
            tweet_text = tweet.text
            parsed_tweet['text'] = tweet
            parsed_tweet['user'] = tweet.user.screen_name
            parsed_tweet['created'] = str(tweet.created_at)
            parsed_tweet['sentiment'] = classifier.classify(tweet_text)
            tweets.insert(0, parsed_tweet)

        # Return the tweets
        return tweets

    except tweepy.TweepError as e:
        print(”Error:␣” + str(e))
```

Listing 4.2: The prototype source code in python from the *twitter_api.py* file

In our *main.py* file, listing 4.3, we first import *flask* to create our web-application, and our two files *twitter_api* and *classifier*. We then train our classifiers by calling our *train* function as soon as the web-application is launched. The web-application is now waiting for the user to access the web-interface in a browser where it will display the *index.html* file. Once the user writes a query clicks on the search button, the query will be sent as a parameter in our *search_tweets* function. The returned tweets will then be displayed on the web-interface by our JavaScripts.

```

# Filename: main.py

from flask import Flask, render_template, request, jsonify
import twitter_api
import classifier

app = Flask(__name__)

# Train our classifiers
classifier.train()

@app.route(”/”)
# Display the user web-interface
def index():
    return render_template(”index.html”)

# Request query from web-interface and run search_tweets function
# with the query as a parameter and return results
```

```python
@app.route('/query/')
def query():
    query = request.args.get('query')
    tweets = twitter_api.search_tweets(query)
    return jsonify({'data': tweets})


if __name__ == "__main__":
    app.run()
```

Listing 4.3: The prototype source code in python from the *main.py* file.

Chapter 5

The Study

The study was performed in two parts, prototype *testing* and *interviews*, and it was conducted on five software developers. The subjects were informed about their anonymity and about how the study would be conducted. The subjects were told to think-aloud during the testing, and that audio recording would be used only during the interview. Before starting the study, every test subject signed a consent and anonymity form. The study was conducted in Norwegian, but the questions and test-cases presented in this chapter has been translated to English.

5.1 Testing

The prototype testing part of the study was conducted by asking the test subjects to perform three test-cases while using the think-aloud protocol. The following test-cases were used in the testing:

Task 1: Search for messages sent to snapchat and read some of these.

Task 2: How many negative messages are sent to snapchat related to their update?

Task 3: Download messages sent to Microsoft related to Skype.

After completing the test-cases, the subject was allowed to further test the prototype freely. Notes of our observations were taken during the testing.

With these test-cases we would be able to observe if the prototype provides enough information for solving the tasks. We did not present or give the subject any information before the study. We want to know if the subjects can through experimentation and the information provided in the prototype learn how to use the prototype.

5.2 Interview

The interview is meant to allow the test subjects to express their experience of using the prototype, and to get their opinions and insights on using the prototype and sentiment analysis to find customer feedback in social media. An Olympus digital recorder was used for audio recording. The recordings were transcribed and deleted after the study.

We wanted to find out if the information given in the prototype is good enough for understanding how to use it, and the subjects opinion on the classifications. We also wanted to find out how the subjects usually handle customer feedback and if a tool such as the prototype could be of any use for them. Lastly, we hoped to get some ideas for improvements of the prototype.

The following main questions were used in the interview, and we also followed up with some sub-question if the response was short or if we wanted the subject to clarify more:

Q1. What information was available in the prototype?

Q2. Was the information good enough for you to understand how to use the prototype?

Q3. Which classifications are used in the prototype and were these good enough in regards of the results?

Q4. Compare how it is to use such a tool against how you typically handle customer feedback.

Q5. Is there anything you would like to add or change in the prototype?

5.3 Consent Form

A consent form was prepared and signed by each of the subjects, before we started the study. The consent form informed the subject about us, about this thesis, anonymity and voluntary participation. The consent form can be found in Appendix A.

5.4 Study Plan

The following plan was used during the study with each of the subject.

Consent Form (5 minutes)

Inform the subject about their anonymity and that audio recording will be used during the interview. Testing part will begin as soon as the subject reads and signs the consent form.

Testing (10 minutes)

Think-aloud is used while solving these tasks.

Task 1: Search for messages sent to snapchat and read some of these.

Task 2: How many negative messages are sent to snapchat related to their update?

Task 3: Download messages sent to Microsoft related to Skype.

Subject allowed to test the prototype freely.

Interview (10-15 minutes)

Q1. What information was available in the prototype?

Q1 a. Was the information clear enough or did you want more information?

Q2. Was the information good enough for you to understand how to use the prototype?

Q2 a. Was there anything that was misunderstanding?

Q3. Which classifications are used in the prototype and were these good enough in regards of the results?

Q3 a. Are there any other classifications you could think of which could be useful in such a tool?

Q4. Compare how it is to use such a tool against how you usually handle customer feedback.

Q4 a. Is something done better or worse?

Q5. Is there anything you would like to add or change in the prototype?

Chapter 6

Study Results

The subjects had no knowledge about the prototype and none were familiar with the Search API provided by Twitter. The study results presented in this section has been translated to English. The transcribed interviews and our notes, in Norwegian, from the study can be found in appendix B.

This chapter presents our observations from the prototype testing and the transcribed interviews. We end this chapter with summaries of the results.

Subject A

Testing

Task 1: The subject took some time before starting on the first task. The subject was not sure what the prototype actually did. The subject wrote "snapchat" without checking the examples offered in the prototype.

Task 2: The subject read the examples given in the prototype and managed to solve the task on the second try.

Task 3: The subject solved the task on first try.

Overall: The subject took some time before starting and wanted some information on what the prototype did. The subject was not sure on how the prototype worked when first seeing it. but went great and understood it after completing the first task.

Interview

INTERVIEWER : What information was available in the prototype and was the information clear enough or did you want more information?

RESPONDENT : I did struggle a bit getting started on the first task, so there was a little barrier there for me. But it is clear that it is a prototype that would be used by a twitter user since you are searching twitter.

INTERVIEWER : Was the information good enough for you to understand how to use the prototype?

RESPONDENT : Yes, after the first task, it became easier.

INTERVIEWER : Was there anything in the prototype that was easy to misunderstand?

Respondent : No, again the start, but then you find out this is the prototype that was going to be tested. But it went fine after that.

Interviewer : Which classifications are used in the prototype and were these good enough regarding the result?

Respondent : Yes, so classification as in positive and negative. I have not studied how they are defined here.

Interviewer : Are there any other classifications you could think of which could be useful in such a tool?

Respondent : No, I do not use anything like this.

Interviewer : You have now fetched feedback from users related to snapchat. Can you compare how it was utilising such a tool against how you usually handle feedback from user.

Respondent : Usually we get feedback on email or through a case management system. Clearly the prototype does a good job of giving an overview if you have many users and active with lots of feedback, which would also give you an impression on how your product is doing on a positive/negative scale. I can see a benefit using this.

Interviewer : Is there anything you would like to add or change?

Respondent : Maybe take out some statistics on how many positive and negative. And another classification could be a possibility. Maybe you would want to compare how many had that hashtag or that hashtag. Something like that.

Subject B

Testing

Task 1: The subject misunderstood the task and searched for something else. The subject managed to solve the task after reading it again and seeing the examples.

Task 2: The subject managed to solve the task on the second try.

Task 3: The subject solved the task on the first try.

Overall: The subject tested out different keywords because the first task was misunderstood. But the subject then understood very fast what the prototype did after seeing the results. Solving the task went good after the first task.

Interview

Interviewer : What information was available in the prototype and was the information clear enough or did you want more information?

Respondent : It was missing some information on how to use it but I think it still went great with the examples, then you will easily figure it out.

Interviewer : Was the information good enough for you to understand how to use the prototype?

Respondent : No, I have to say that I did use some time on it.

Interviewer : Was there anything in the prototype that was easy to misunderstand?

RESPONDENT : I thought it would search both Twitter and Snapchat, I do not use any of them so much.

INTERVIEWER : Which classifications are used in the prototype and were these good enough regarding the result?

RESPONDENT : Yes, they were easy to understand.

INTERVIEWER : Are there any other classifications you could think of which could be useful in such a tool?

RESPONDENT : Yes, I think neutral could have been there, because there is a lot of information which someone just shares and could be hard to tell if they are directly positive, unless they contain something like "This was an amazing product", or "I did not like this".

INTERVIEWER : Compare how it is to use such a tool against how you usually handle customer feedback.

RESPONDENT : This is like being on a fishing trip, where you do not exactly know what you will get or find, until you get something interesting. It is a good addition to what we currently have. Systems we are using are e-mails, phone, and other ticketing systems, where we get more specific information about things. But with the prototype we could catch additional feedback. Because if there is a bug, then it will be registered, and we have rules how to handle it, but we also see it is becoming more important to handle things more pro-active and that is when a system like this could come in. Because a user has alternatives, and if they are not happy with one product they give their feedback and just move on to another. That is why it could be important to use something like this.

INTERVIEWER : Is there anything you would like to add or change in the prototype?

RESPONDENT : No, not so much, I thought it was a good prototype, when I first understood how to use it.

Subject C

Testing

Task 1: The subject was confused and did not understand the task. The subject searched for "Snapchat".

Task 2: The subject did not solve the task and required some assistance.

Task 3: The subject did not solve the task and required some assistance.

Overall: The subject seemed confused on the first task as to what the prototype did or how it worked. But after some assistance and seeing the results, we believe the subject understood more.

Interview

INTERVIEWER : What information was available in the prototype and was the information clear enough or did you want more information?

RESPONDENT : Yes, I wanted some explanation on what the prototype did, but I do understand it now that it goes through twitter and checks for positive and negative feedback.

INTERVIEWER : Was the information good enough for you to understand how to use the prototype?

RESPONDENT : It was easy to understand, even though the search method was a little different.

INTERVIEWER : Which classifications are used in the prototype and were these good enough regarding the result? Are there any other classifications you could think of which could be useful in such a tool?

RESPONDENT : You mean the positive and negative. They did miss on a few but they were still very accurate.

INTERVIEWER : Compare how it is to use such a tool against how you usually handle customer feedback.

RESPONDENT : If we are talking about customer feedback, it is so much more than just positive and negative. We have some kind of a form for customer satisfaction, and there it is based more on what the customer themselves want to highlight. And twitter would be a little un-serious, for instance the messages we found here. But I think those customer satisfaction surveys are more thorough.

INTERVIEWER : Is something done better or worse?

RESPONDENT : The whole point with having those surveys are to always improve. You would always want to improve, and you must have that focus towards the customer as well.

INTERVIEWER : Is there anything you would like to add or change in the prototype?

RESPONDENT : These kinds of prototypes are always nice basis to develop something new. Maybe track the activity on a tweet to check how much people tweet to a specific user, could be interesting.

Subject D

Testing

Task 1: The subject wrote “snapchat” without looking at the examples. The subject also clicked on some of the tweet’s user name (which redirects to the user’s Twitter page).

Task 2: The subject read the examples and solved the task.

Task 3: The subject solved the task on first try.

Overall: The subject solved the tasks very quickly. The subject understood the prototype and how it worked. The subject also explored the prototype by trying other functions, such as visiting their Twitter user profiles.

Interview

INTERVIEWER : What information was available in the prototype?

Respondent : Looks like you can search messages with positive and negative. I am a little unsure on what the positive and negative actually is and how they are measured.

Interviewer : Was the information clear enough or did you want more information?

Respondent : Yes, since I was unsure about the positive and negative, then I would like to know what they actually were.

Interviewer : Was the information good enough for you to understand how to use the prototype?

Respondent : Yes, it was. I was a little fast, but I figured out later that you could write more advanced search queries.

Interviewer : Was there anything in the prototype that was misunderstanding?

Respondent : No, maybe not, but wanted the ability to press "Enter" on search, instead of clicking on the button.

Interviewer : Which classifications are used in the prototype and were these good enough in regards of the results?

Respondent : Well that is positive and negative, and as mentioned I was a little unsure about them.

Interviewer : Are there any other classifications you could think of which could be useful in such a tool?

Respondent : Yes, I am a little unsure again on these, since I did not completely understand what they were. No, I do not know, I cannot say anything.

Interviewer : Compare how it is to use such a tool against how you usually handle customer feedback.

Respondent : Yes, that is on email, so it's probably faster in the prototype then searching through your email accounts and finding what you are looking for.

Interviewer : Is something done better or worse?

Respondent : It would probably be better; the problem is that you would get another tool to deal with. But it seemed, very fast and useful if you know what you are looking for.

Interviewer : Is there anything you would like to add or change in the prototype?

Respondent : Clicking on "Enter" button on keyboard, and when you click open the examples, that it would close automatically after searching. And maybe some simple description on what the positive and negative is.

Subject E

Testing

Task 1: The subject wrote "Snapchat" without reading the examples.

Task 2: The subject read the examples given in the prototype, and took some time reading through these. Wrote "to:snapchat update".

Task 3: The subject wrote "Microsoft skype", but then changed "to:Microsoft skype".

Overall: The subject understood the prototype and its functionality and solved the task easily. The subject mentioned that the prototype is clearly using Twitter API. Solving the tasks went good after reading the examples.

Interview

Interviewer : What information was available in the prototype?

Respondent : It is obviously a search engine, connected to Twitter API, and contains search examples to make it easier to use the application, but I miss maybe the possibility to use buttons to put together a search query.

Interviewer : Was the information clear enough or did you want more information?

Respondent : Examples were clear, wanted maybe checkboxes or something to specify that I wanted to search for only this or only that, even though it is a search field being filled in with text that I could have written myself, but that it does it for me.

Interviewer : Was the information good enough for you to understand how to use the prototype?

Respondent : Yes, I did understand how to use it. Seemed relative easy to use, only thing is that I do not have knowledge to the queries.

Interviewer : Was there anything that was misunderstanding?

Respondent : Not really. After using it for two seconds I felt that it was relatively easy to understand.

Interviewer : Which classifications are used in the prototype and were these good enough in regards of the results?

Respondent : You are thinking about the positive and negative for example. Feel that they fit very well, even though they missed on a few that were positive.

Interviewer : Are there any other classifications you could think of which could be useful in such a tool?

Respondent : Maybe a form of neutral categorisation, that not everything is necessarily black and white in relation to positive and negative. And of course, be able to filtrate on only positive or only negative.

Interviewer : Compare how it is to use such a tool against how you usually handle customer feedback.

Respondent : Usually we use some kind of case management system tool and compared to this, I believe that this would be able to catch things that are not reported in to us. Not necessarily a replacement tool, but as an addition to our case management system.

Interviewer : Is there anything you would like to add or change in the prototype?

Respondent : No, nothing else then being able to filter based on the classification and maybe even some form of calendar functionality, so I can see last month or last week for example.

6.1 Testing Results Summary

The subjects did not have any knowledge or any information on the prototype and were not familiar with the Search API provided by Twitter. Table 6.1 presents our overall observation for each of the subjects from the prototype testing.

	Testing
A	The subject took some time before starting and wanted some information on what the prototype did. The subject was not sure on how the prototype worked when first seeing it. but went great and understood it after completing the first task.
B	The subject tested out different keywords because the first task was misunderstood. But the subject then understood very fast what the prototype did after seeing the results. Solving the task went good after the first task.
C	The subject seemed confused on the first task as to what the prototype did or how it worked. But after some assistance and seeing the results, we believe the subject understood more.
D	The subject solved the tasks very quickly. The subject understood the prototype and how it worked. The subject also explored the prototype by trying other functions, such as visiting their Twitter user profiles.
E	The subject understood the prototype and its functionality and solved the task easily. The subject mentioned that the prototype is clearly using Twitter API. Solving the tasks went good after reading the examples.

Table 6.1: Our overall observation of each subject in the prototype testing.

6.2 Interview Results Summary

Table 6.2 presents a summary of key findings for each of the subjects on each of our main questions.

	Q1	**Q2**	**Q3**	**Q4**	**Q5**
A	Struggled a bit, but understood that it searches for tweets in Twitter	Became easier after first task, misunderstood the first question	Recognised the classifications, but did not understand how tweets were classified	Feedback on email or case management system, and mentioned that the prototype can give a good overview of your product or application	Statistics and another classification, also ability to compare
B	Wanted information on how to use it, but was easy to figure it out with the examples	Thought it would search both Snapchat and Twitter	Recognised the classifications, and that neutral class could be added	Compared the prototype as catching fish. Mentioned that it would be a good addition to their current systems to catch additional feedback.	Nothing to add
C	Wanted information on the prototype, but understood that it searches for feedback in Twitter	Easy to understand, even though the search method was different	Recognised the classifications, and that it was very accurate	Was happy with their current systems, and thought that searching Twitter for feedback would be un-serious	Track activity
D	Wanted information about the classifications and how they were measured	Easy to understand and figured out that one could write more advanced queries	Recognised the classifications but wanted some information on how they were classified	Mentioned that it would be faster and better than their current systems, but also that you would have another tool to deal with.	Usability improvements
E	Mentioned that it was a search engine using Twitter API, and has some examples to make it easier to use	Easy to understand and use	Recognised the classifications, and that neutral class could be added	Mentioned that the prototype would be able to catch feedback which are not reported them, not as a replacement but as an additional tool	Filtering options and some calendar functionality

Table 6.2: A short summary of the replies from each subject on the main questions.

Chapter 7

Discussion

The results from the prototype testing in the study show that the subjects seemed more confident in solving the tasks after reading and trying some of the examples. We also saw that some of the subjects were confused at first, but that they understood the purpose of the prototype after completing the first task and seeing the results. This suggests that we should have given the subjects a proper introduction to the prototype before the tests, but it also shows that the prototype was easy to understand through experimentation. Four out of five subjects completed all the tasks by themselves. Only one subject required some assistance to finish all three. This shows that the prototype was fairly easy to use.

The results of the interviews confirm that the prototype was easy to understand and use. All subjects were able to understand the functionality of the prototype and have a rough understanding of how it works.

Four of the subjects thought the prototype could be a valuable addition to the normal channels which consist of email, phone and a case management system. Several subjects mentioned that the prototype could help identify new feedback, and two subjects mentioned that such a tool could provide a good overview of customer and user satisfaction. One subject pointed out that mining from social media requires your product or application to have a relatively large user base. Subject A mentioned that the prototype clearly does a good job of giving an overview:

> *“Usually we get feedback on email or through a case management system. Clearly the prototype does a good job of giving an overview...”*

Subjects mentioned that neutral could be added to the classification, because users may sometimes share information which could be hard to classify as positive or negative. Subject B mentioned the following when asked about other classifications which could be useful in such a tool:

> *“I think neutral could have been there, because there is a lot of information which someone just shares...”*

One subject also mentioned that user feedback is so much more than positive and negative.

Some of the subjects wanted a more helpful user interface, such as a calendar functionality for filtering results based on *from* date and *until* date, an option to filter results based on their class, and the use of buttons instead of the standard operators the Twitter API offers when writing in specific queries. Subject E wanted to filter the results based on classification, and mentioned:

> *"..filter based on the classification and maybe even some form of calendar functionality.."*

Subject D mentioned that such a tool would be better then their current systems, but that a problem would be that they would get another tool to deal with:

> *"It would probably be better; the problem is that you would get another tool to deal with. But it seemed, very fast and useful if you know what you are looking for."*

The tool does not have to be independent and can easily be implemented as part of a larger system, for example a customer support system. The prototype can be used by anyone working towards a better customer satisfaction, and is not restricted to only product and application developers.

The prototype offers a simple web-interface with an input field, where the user can write in a query to search for. Subjects mention that it was easy to understand the prototype even though the search method was different. Subject B also mentioned that you can easily figure out how to use it with the examples provided:

> *"It was easy to understand, even though the search method was a little different."*

The results from the comparison of the classifiers and the ensemble shows that an ensemble of classifiers achieves higher accuracy, which confirms what we found related work. We compared the accuracies from each classifier and the ensemble on the three datasets. The ensemble achieved almost 4% better accuracy then Random Forest on Dataset I and over 5% better accuracy on Dataset III. The average accuracy was also calculated, and the ensemble achieved an 83.9% average accuracy on the datasets, while SVM achieved 81.7%, MNB achieved 83.1% and RF achieved 80.3%.

The prototype design is simple and has some of the design features from some of the related tools. Our prototype differs from the available tools primarily because of the classification process implemented. We classify on document-level whereas some of them analyse and classify on sentence-level. We also implement a more advanced search field for getting better results from the API.

Future Improvements

The results from the study show that improvements can be made to our prototype. We suggest the following improvements and modifications in an improved and extended system:

- **Keywords** The improved system would only require the user to manage a list of keywords. The system will search and track for feedback based on the list.

- **Automatic search** The improved system would automatically search and track the keywords managed by the user. We suggest implementing more APIs, we talked about some of these in chapter 4, such as News API, Facebook API, and maybe even crawl and mine app stores reviews.

- **Reply** We suggest implementing a reply feature so that the user may easily reply back to the author of the feedback that was found by the system.

- **Results** The improved system would have the option to filtrate the results based on either sentiment or keywords.

- **Notifications** The improved system would notify the user when it finds feedback that needs immediate attention, such as bugs or application crashes.

- **History** The improved system would save the results for each day. The user may use a calendar functionality when searching for a specific day.

- **Language** We suggest the ability to search in more languages.

The extended and improved system would only require the user to manage the list of keywords. The system will then start its search for feedback based on the keywords in social media, app stores, news and forums. All the findings by the system will be displayed in the new web-interface, which we will come back to. The user may reply back to the author of the feedback on the application it was posted in (e.g. Facebook, Twitter). If the system finds any feedback that includes anything that needs immediate attention (such as bugs, application crash, etc.), a notification will be sent to the user. Figure 7.1 shows an overview of the processes in the improved system.

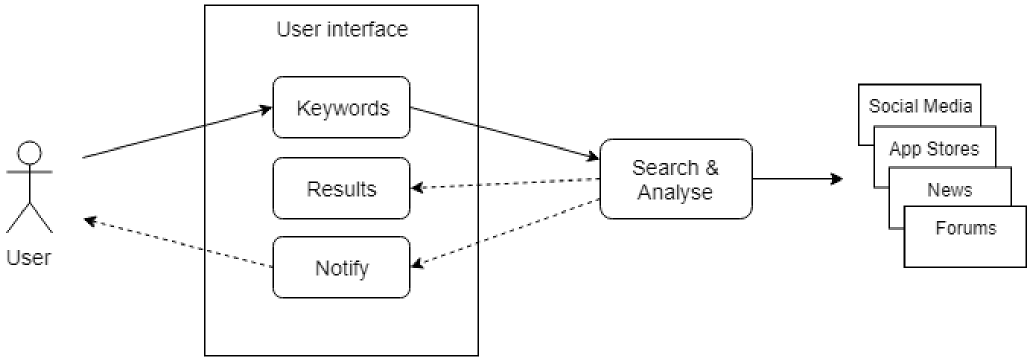
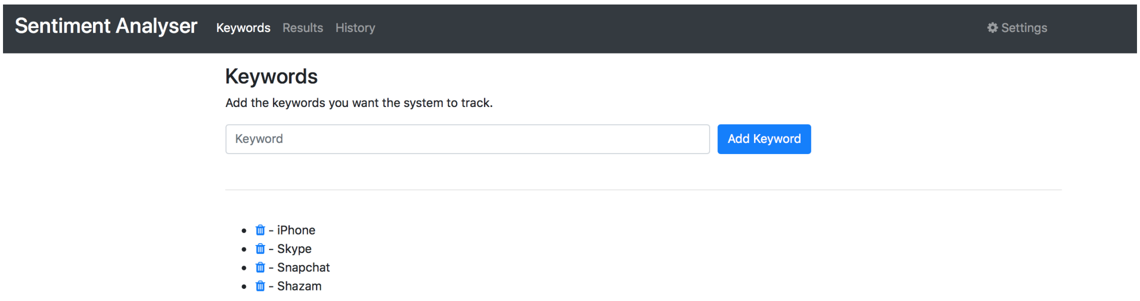


Figure 7.1: An overview of the processes in the improved system

The user can add several keywords in the *Keywords* page that they would like the system to track and search for. Keywords can at any time be added or deleted. Figure 7.2 shows the new web-interface page *Keywords*, where the user may add or delete keywords. The system will automatically track and search for the keywords in the list and display all the findings with statistics for the current day in the new *Results* page.



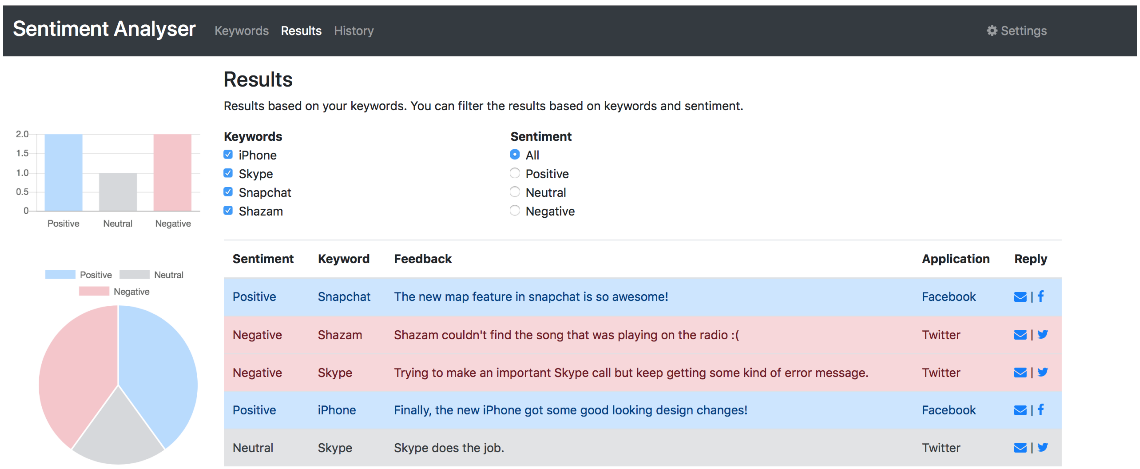
Keywords

Add the keywords you want the system to track.

- 🗑 - iPhone
- 🗑 - Skype
- 🗑 - Snapchat
- 🗑 - Shazam

Figure 7.2: The new web-interface design in the improved system of the *Keywords* page.

The user can see an overview of the results and findings for the current day in the *Results* page. The results can be filtered based on *keywords* and *sentiment*. Figure 7.3 shows the new web-interface page *Results*, where the user can filter current day's results. The user may also reply to the feedback through the application it was found in, or simply contact the author of the feedback on email.



Results

Results based on your keywords. You can filter the results based on keywords and sentiment.

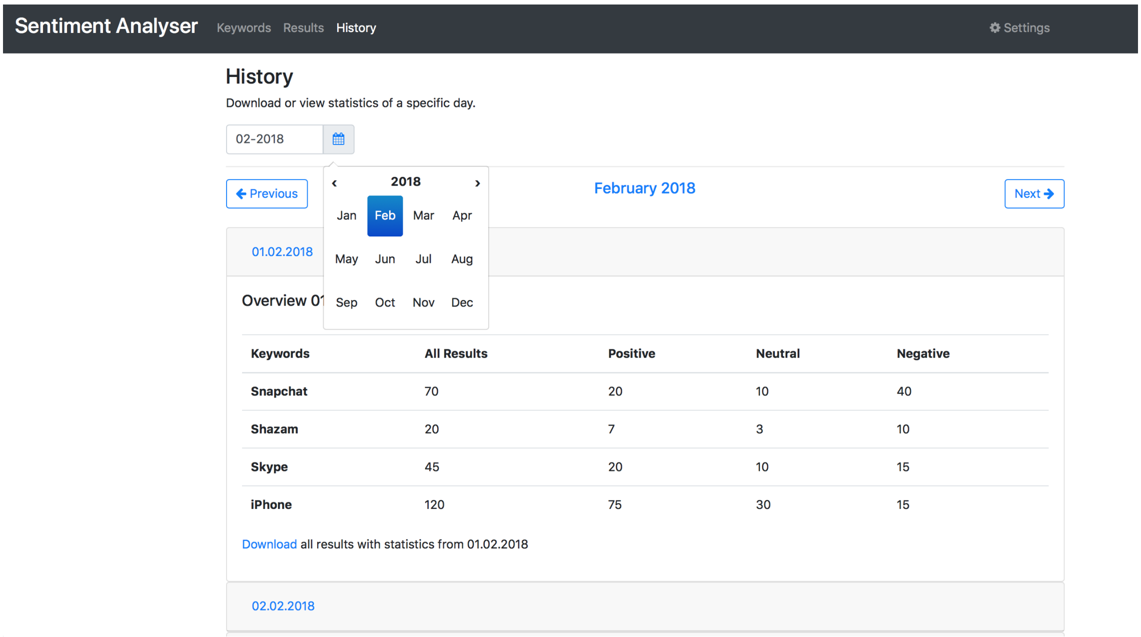
Keywords
☑ iPhone
☑ Skype
☑ Snapchat
☑ Shazam

Sentiment
● All
○ Positive
○ Neutral
○ Negative

Sentiment	Keyword	Feedback	Application	Reply
Positive	Snapchat	The new map feature in snapchat is so awesome!	Facebook	✉ | f
Negative	Shazam	Shazam couldn't find the song that was playing on the radio :(	Twitter	✉ | 🐦
Negative	Skype	Trying to make an important Skype call but keep getting some kind of error message.	Twitter	✉ | 🐦
Positive	iPhone	Finally, the new iPhone got some good looking design changes!	Facebook	✉ | f
Neutral	Skype	Skype does the job.	Twitter	✉ | 🐦

Figure 7.3: The new web-interface design in the improved system of the *Results* page.

The user can view or download all results with statistics of any past day in the *History* page. Figure 7.4 shows the new web-interface page *History*, where the user may select a year and month using the calendar to get a list of all the days in that month. Each day will present statistics for each keyword it was tracking on that specific day. The user will also be able to download the findings and results with the statistics of that day.



History

Download or view statistics of a specific day.

02-2018

01.02.2018

Overview 01

Keywords	All Results	Positive	Neutral	Negative
Snapchat	70	20	10	40
Shazam	20	7	3	10
Skype	45	20	10	15
iPhone	120	75	30	15

Download all results with statistics from 01.02.2018

02.02.2018

Figure 7.4: The new web-interface design in the improved system of the *History* page.

Chapter 8

Conclusions

The goal in this thesis was to find the value, if any, of using sentiment analysis in the domain customer and user feedback in social media. We assessed the research question by reviewing related work and tools utilising sentiment analysis. We developed a sentiment analysis web-application for mining social media. The prototype was based on the research and techniques that were found in related work. The prototype was used in a qualitative study conducted on five software developers.

The results from the study show two main benefits of using a sentiment analysis tool to gather customer feedback from social media. It allows the identification of customer feedback that may never reach the developers through traditional channels. The second main benefit is that it can be helpful for producing an overview of user experiences of a product, application or other services. This suggests that sentiment analysis can be a valuable tool for software developers and other companies and institutes dealing with customers or users.

The prototype design can be improved by adding more features, which was mentioned by some of the participants in our study. We have described and designed an improved prototype with new features, such as replying to user feedback, notification when the system finds feedback that require immediate attention (e.g. application crashes and bugs), and an overview of previous results.

Future Work

The two class sentiment analysis setup can likely be improved by introducing other classes such as a neutral class, as suggested by participants in our study. This restriction is an imposition caused by the annotations in the datasets available for training a sentiment classifier today. For future work it is worth looking into generating new data which do not have this limitation.

There are also more advanced methods of natural language processing that are worth investigating. These methods can provide an information-based classification of text, which could prove more valuable than a simple sentiment classification. It would also allow us to gather information from more advanced forms of text such as forum and blog posts.

Bibliography

[1] N. Altrabsheh, M. Cocea, and S. Fallahkhair. Sentiment analysis: towards a tool for analysing real-time students feedback. In *Tools with Artificial Intelligence (ICTAI), 2014 IEEE 26th International Conference on*, pages 419–423. IEEE, 2014.

[2] Application programming interface. `https://en.wikipedia.org/wiki/Application_programming_interface`. Retrieved: 2018-05-03.

[3] N. Archak, A. Ghose, and P. G. Ipeirotis. Show me the money!: deriving the pricing power of product features by mining consumer reviews. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 56–65. ACM, 2007.

[4] S. Asur and B. A. Huberman. Predicting the future with social media. In *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 01*, pages 492–499. IEEE Computer Society, 2010.

[5] S. Baccianella, A. Esuli, and F. Sebastiani. Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 2200–2204, 2010.

[6] J. Barlow and C. Møller. *A complaint is a gift: using customer feedback as a strategic tool.* Berrett-koehler publishers, 1996.

[7] S. Bird, E. Klein, and E. Loper. *Natural language processing with Python: analyzing text with the natural language toolkit.* O'Reilly Media, Inc., 2009.

[8] Bootstrap. `https://getbootstrap.com/`. Retrieved: 2018-05-06.

[9] M. M. Bradley and P. J. Lang. Affective norms for english words (anew): Instruction manual and affective ratings. Technical report, Citeseer, 1999.

[10] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[11] E. Cambria, B. Schuller, Y. Xia, and C. Havasi. New avenues in opinion mining and sentiment analysis. *IEEE Intelligent Systems*, 28(2):15–21, 2013.

[12] ChartJS. `http://www.chartjs.org/`. Retrieved: 2018-05-06.

[13] N. F. Da Silva, E. R. Hruschka, and E. R. Hruschka Jr. Tweet sentiment analysis with classifier ensembles. *Decision Support Systems*, 66:170–179, 2014.

[14] S. R. Das and M. Y. Chen. Yahoo! for amazon: Sentiment extraction from small talk on the web. *Management science*, 53(9):1375–1388, 2007.

[15] K. Dave, S. Lawrence, and D. M. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web*, pages 519–528. ACM, 2003.

[16] N. B. Ellison et al. Social network sites: Definition, history, and scholarship. *Journal of computer-mediated Communication*, 13(1):210–230, 2007.

[17] Facebook About. `https://www.facebook.com/pg/facebook/about/`. Retrieved: 2018-05-08.

[18] Facebook Reports First Quarter 2018 Results. `https://s21.q4cdn.com/399680738/files/doc_financials/2018/Q1/Q1-2018-Press-Release.pdf`. Retrieved: 2018-05-05.

[19] X. Fang and J. Zhan. Sentiment analysis using product review data. *Journal of Big Data*, 2(1):5, 2015.

[20] Flask. `http://flask.pocoo.org/`. Retrieved: 2018-05-07.

[21] Font Awesome. `https://fontawesome.com/`. Retrieved: 2018-05-06.

[22] J. Gallaugher and S. Ransbotham. Social media and customer dialog management at starbucks. *MIS Quarterly Executive*, 9(4), 2010.

[23] A. Ghose, P. Ipeirotis, and A. Sundararajan. Opinion mining using econometrics: A case study on reputation systems. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 416–423, 2007.

[24] A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 1(12), 2009.

[25] C. Healey and S. Ramaswamy. Visualizing twitter sentiment. *Sentiment Viz, Available at: https://www. csc. ncsu. edu/faculty/healey/tweet_viz/tweet_app*, 2011.

[26] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant. *Applied logistic regression*, volume 398. John Wiley & Sons, 2013.

[27] C.-W. Hsu, C.-C. Chang, C.-J. Lin, et al. A practical guide to support vector classification. 2003.

[28] M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM, 2004.

[29] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer, 1998.

[30] A. M. Kaplan and M. Haenlein. Users of the world, unite! the challenges and opportunities of social media. *Business horizons*, 53(1):59–68, 2010.

[31] A. Z. Khan, M. Atique, and V. Thakare. Combining lexicon-based and learning-based methods for twitter sentiment analysis. *International Journal of Electronics, Communication and Soft Computing Science & Engineering (IJECSCSE)*, page 89, 2015.

[32] S. Kiritchenko, X. Zhu, and S. M. Mohammad. Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*, 50:723–762, 2014.

[33] W. Koehrsen. Random Forest Simple Explanation. `https://medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d`, 2017. Retrieved: 2018-04-12.

[34] D. Lee, O.-R. Jeong, and S.-g. Lee. Opinion mining of customer feedback data on the web. In *Proceedings of the 2nd international conference on Ubiquitous information management and communication*, pages 230–235. ACM, 2008.

[35] Linkify. `http://soapbox.github.io/linkifyjs/`. Retrieved: 2018-05-06.

[36] A. Lipsman. Online consumer-generated reviews have significant impact on offline purchase behavior." comscore. *Inc. Industry Analysis*, pages 2–28, 2007.

[37] B. Liu. Sentiment analysis and subjectivity. *Handbook of natural language processing*, 2:627–666, 2010.

[38] B. Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167, 2012.

[39] W. Maalej and H. Nabil. Bug report, feature request, or simply praise? on automatically classifying app reviews. In *Requirements Engineering Conference (RE), 2015 IEEE 23rd International*, pages 116–125. IEEE, 2015.

[40] D. M. Magerman. Statistical decision-tree models for parsing. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, pages 276–283. Association for Computational Linguistics, 1995.

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

[42] P. Melville, W. Gryc, and R. D. Lawrence. Sentiment analysis of blogs by combining lexical knowledge with text classification. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1275–1284. ACM, 2009.

[43] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[44] Moment. `https://momentjs.com/`. Retrieved: 2018-05-06.

[45] S. M. Mudambi and D. Schuff. Research note: What makes a helpful online review? a study of customer reviews on amazon. com. *MIS quarterly*, pages 185–200, 2010.

[46] K. P. Murphy. Naive bayes classifiers. *University of British Columbia*, 18, 2006.

[47] Pandas. `https://pandas.pydata.org/`. Retrieved: 2018-05-07.

[48] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.

[49] R. Prabowo and M. Thelwall. Sentiment analysis: A combined approach. *Journal of Informetrics*, 3(2):143–157, 2009.

[50] Python. `https://www.python.org`. Retrieved: 2018-05-07.

[51] Scikit-Learn. `http://scikit-learn.org`. Retrieved: 2018-05-07.

[52] F. Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.

[53] SentiWordNet. `http://sentiwordnet.isti.cnr.it/`. Retrieved: 2018-05-07.

[54] Standard Search - Twitter Developers. `https://developer.twitter.com/en/docs/tweets/search/overview/standard`. Retrieved: 2018-03-29.

[55] Statista. Leading global social network sites 2018. `https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/`, 2018. Retrieved: 2018-03-25.

[56] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede. Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307, 2011.

[57] Tweepy. `http://www.tweepy.org/`. Retrieved: 2018-05-07.

[58] Twitter: Getting Started. `https://help.twitter.com/en/twitter-guide`. Retrieved: 2018-05-05.

[59] N. Vyrva. Sentiment analysis in social media. Master's thesis, Ostfold University College, 2016.

[60] H. Wang, D. Can, A. Kazemzadeh, F. Bar, and S. Narayanan. A system for real-time twitter sentiment analysis of 2012 us presidential election cycle. In *Proceedings of the ACL 2012 System Demonstrations*, pages 115–120. Association for Computational Linguistics, 2012.

[61] K. Weil. Measuring Tweets. `https://blog.twitter.com/official/en_us/a/2010/measuring-tweets.html`, 2010. Retrieved: 2018-03-27.

[62] K. Weil. Coming soon to Twitter. `https://blog.twitter.com/official/en_us/a/2014/coming-soon-to-twitter.html`, 2014. Retrieved: 2018-05-08.

[63] T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 347–354. Association for Computational Linguistics, 2005.

[64] H. Zhang. The optimality of naive bayes. *AA*, 1(2):3, 2004.

Appendix A

Consent Form

Description of the Master Thesis A student from the Faculty of Computer Sciences, University College of Østfold. The master thesis will contribute to application and product developers searching and finding customer feedback to improve the customer satisfaction. A prototype has been developed in this thesis and as part of this study the student wish to have a qualitative interview with you to investigate how a tool like this could help achieve a better customer satisfaction.

Student: Mohamad H. Jalloul – mohamad.h.jalloul@hiof.no

Supervisor: Lars Vidar Magnusson – lars.v.magnusson@hiof.no

Voluntary Participation All participation is voluntary, and the respondent may stop the testing or the interview anytime. There will be used audio recording only during the interview if the respondent consent for this. The student will take notes if the respondent does not feel comfortable with audio recording.

Anonymity Audio recordings or notes and all other information from the interview and testing will not trace back to the respondent. Only the student will have knowledge of who the respondent is.

Consent I have read this form and consent to participate in the testing and interview.

Interviewer Date

Respondent Date

Appendix B

Transcribed Interviews

B.1 Subject A

Testing

Oppgave 1: Tok sin tid før start, var ikke sikker på hva prototypen gjore. Skriver inn "snapchat" uten å sjekke eksempler i prototypen.

Oppgave 2: Leste eksempler gitt i prototypen, og klarte oppgaven på andre forsøk.

Oppgave 3: Klarte oppgaven på første forsøk.

Oppsummering: Tok sin tid før start, var usikker på hvordan prototypen fungerte. Kunne trengt mer informasjon om prototypen. Gikk greit på oppgave 2 og forstod prototypen etter oppgave 2.

Intervju

INTERVJUER : Hvilken informasjon finnes i prototypen og var informasjonen klar eller savnet du noe mer informasjon?

RESPONDENT : Jeg slet jo litt med å komme i gang på første oppgaven i hvertfall, så var jo en liten barriere der for meg som ikke er en twitter bruker hvertfall. Men det er klart at det er jo en prototype/verktøy typisk vil brukes av en twitter bruker, altså du vil jo leite i twitter, så det er meg som har mangel og.. (ler litt)

INTERVJUER : Var informasjonen tydelig, forstod du hvordan du skulle bruke prototypen?

RESPONDENT : Ja etter den første oppgaven, gikk det lettere.

INTERVJUER : Var det noe i prototypen som var lett å misforstå?

RESPONDENT : Nei, det var litt igjen den starten, men så finner du ut av at det var dette som var prototypen som skulle testes. Men så gikk det fint.

INTERVJUER : Hvilke klassifiseringer er brukt i prototypen og passet disse til resultatene?

RESPONDENT : Ja, så klassifiseringer som i positive og negative tenker du. Ja jeg har jo ikke studert hvordan denne her definerer positive som positive og negative.

INTERVJUER : Er det noen andre klassifiseringer du kan tenke på som kunne vært nyttig i et slikt verktøy?

RESPONDENT : Nei, jeg bruker jo ikke sånne ting, så tror ikke det er noe forutsetninger på dette her.

INTERVJUER : Du har nå henta inn tilbakemeldinger fra brukere relatert til snapchat og sånt. Så kan du sammenligne hvordan det er å benytte et slikt verktøy mot hvordan dere vanligvis håndterer tilbakemeldinger fra brukere.

RESPONDENT : Nei har ikke en vanligvis måte, nei sånn som jeg jobber så er jo vanlig tilbakemelding å få en mail eller saksbehandlingssystem eller en sånn type ting da. Klar dette her gjør jo en bra oversikt hvis du har mange brukere og veldig aktive med mye tilbakemeldinger, så vil man jo fort få et inntrykk på hvordan du ligger ann, sånn positiv/negativ fordeling, der den burde vær. Kan jo se en nytte av det.

INTERVJUER : Var det noe du savnet ved prototypen eller noe du vil endre?

RESPONDENT : Nei, det er jo litt tilbake til det spørsmålet du hadde i stad da om andre ting man ønsker å ha med, det tror jeg har hvertfall gitt innsyn i hva en bruker som bruker en sånn løsning. Men helt sånn fall inn kanskje tatt ut en statistikk på antall positiv og negativ. Ja altså hatt en annen klassifisering da, kanskje vært en mulighet. Kanskje du ønsker å gjøre en opptelling på antall den emneknaggen og den. Sammenligna hvor mange som hadde den og den knaggen. Noe i den retningen.

B.2 Subject B

Testing

Oppgave 1: Missforstå oppgaven og søkte på noe annet. Klare å løse oppgaven etter at oppgaven ble lest opp på nytt og lese eksemplene.

Oppgave 2: Klarte å løse oppgaven på andre forsøk.

Oppgave 3: Løste oppgaven på første forsøk

Oppsummering: Testet litt forskjellig først, ble kjent med prototypen, før brukeren startet på selve oppgaven. Brukeren ble litt forvirret med snapchat oppgaven. Men skjønte fort etter første oppgave hvordan prototypen fungerte.

Intervju

INTERVJUER : Hvilken informasjon finnes i prototypen og var informasjonen klar eller savnet du noe mer informasjon?

RESPONDENT : Det mangla jo litt informasjon for hvordan man skulle bruke den, men jeg tenker det var jo forsovet greit med de eksemplene, så finner man jo alltid fort ut av det

INTERVJUER : Var informasjonen tydelig, forstod du hvordan du skulle bruke prototypen

RESPONDENT : Nei, jeg brukte litt tid på det, det må jeg si jeg gjorde.

INTERVJUER : Var det noe i prototypen som var lett å misforstå?

RESPONDENT : Kanskje det med at den søkte.. Ja jeg missa at den søkte twitter, bare twitter og snapchat eksempelet kanskje det var som da, “Søker den snapchat?”

tenkte jeg, jeg tenkte ikke bare at det bare var en snapchat bruker kun på twitter. Jeg bruker ikke snapchat og twitter så mye i det daglige.

Intervjuer : Hvilke klassifiseringer er brukt i prototypen og passet disse til resultatene?

Respondent : Jeg vet ikke noe om klassifiseringer… Ja sånn ja det var lett å forstå.

Intervjuer : Er det noen andre klassifiseringer du kan tenke på som kunne vært nyttig i et slikt verktøy?

Respondent : Ja jeg tenker Neutral kunne vært, fordi det er mye informasjon som du bare deler, som er vanskelig å si er direkte positiv, hvis det ikke inneholder noe som dette her var et fantastisk produkt eller det like jeg eller noe sånt, at det bare er nå har det kommet noe nytt fra noen, kunne vært en egen klassifisering.

Intervjuer : Sammenlign hvordan det er å benytte et slikt verktøy mot hvordan dere vanligvis håndterer tilbakemeldinger fra brukere.

Respondent : Det her blir jo litt mer som å være på fisketur, du vet jo ikke helt hva du får, du bare ser om du finner noe interessant, det er jo veldig nyttig tillegg til de vi har allerede, så de systemene vi fanger opp tilbakemelding er jo e-post, telefon og de ticketing systemene vi har, der informasjon kommer på helt konkrete ting. Mens det her vil jo være ting som vi kan, mer fange opp i tillegg til det. Fordi finnes det en feil så blir feilen registrert og da har vi klare regler for hvordan det skal håndteres, men vi ser at det blir viktigere og viktigere å håndtere ting pro-aktivt, og da vi trenger et tillegg som det her fordi vår hypotese er jo at sluttbrukere har alternativer, så hvis de er missfornøyd med noe så slenger de det ut på sosiale medier så finner de seg en annen istedet. Derfor er det viktig å ta i bruk det her.

Intervjuer : Var det noe du savnet ved prototypen eller noe du vil endre?

Respondent : Nei ikke så mye, tenkte det var en veldig god prototype, når jeg først skjønte hvordan jeg skulle bruke den. Så tenkte jeg denne her var kjempe fin prototype.

B.3 Subject C

Testing

Oppgave 1: Forvirret, skjønte ikke oppgaven, tok litt tid, men klarte til slutt å søke på "snapchat".

Oppgave 2: Klarte ikke løse oppgaven. Fikk hjelp til slutt

Oppgave 3: Klarte ikke løse oppgaven. Fikk hjelp til slutt

Oppsummering: Ble forvirret på første oppgave. Skjønte ikke hva prototypen gjorde eller hvordan den fungerte.

Intervju

Intervjuer : Hvilken informasjon finnes i prototypen og var informasjonen klar eller savnet du noe mer informasjon?

Respondent : Ja, så jeg savna litt forklaring på hva prototypen gjorde, det vil si, jeg skjønner jo det nå etter som du forklarte det, at den går gjennom twitter rett og slett og sjekker positive og negative responser egentlig eller negative tweets.

Intervjuer : Var informasjonen tydelig, forstod du hvordan du skulle bruke prototypen eller var noe lett å misforstå?

Respondent : Neida, det var i grunn greit å forstå, selve den søkemetoden var litt uvant synes jeg, måten å søke på ting, men nå er ikke jeg sånn veldig vandt til å bruke twitter heller da, så det kan godt hende det er litt i tråd med hvordan det brukes der.

Intervjuer : Hvilke klassifiseringer er brukt i prototypen og passet disse til resultatene? Er det noen andre klassifiseringer du kan tenke på som kunne vært nyttig i et slikt verktøy?

Respondent : Tenker du på klassifisering da positivt eller negativt, de bomma jo innimellom, men var jo stort sett ganske nøyaktig.

Intervjuer : Er det noen andre klassifiseringer du kan tenke på som kunne vært nyttig i et slikt verktøy?

Respondent : Det kunne vært nyttig kanskje bare, sånn... Ting som inneholder et bestemt ord, eller inneholder den teksten som går til en bestemt mottakker, men det er jo sånn sikkert mulig å gjøre allerede vil jeg tro.

Intervjuer : Sammenligne hvordan det er å benytte et slikt verktøy mot hvordan dere vanligvis håndterer tilbakemeldinger fra brukere.

Respondent : Ja, hvordan håndterer vi tilbakemeldinger fra brukere a tro? Hvis vi snakker om kunetilbakemeldinger, så har vi jo gjerne, det går jo på så mye mer enn bare positivt og negativt. Vi har jo gjerne en sånn form for customer satisfaction inquiry løsning, og da er det jo basert på hva kunden selv legger vekt på. Så for en tilbakemelding fra en kunde så vet jeg ikke, sånn twitter, det blir litt sånn useriøst. Hvis du skal måtte basert det på sånn som dette, for som vi går gjennom de tinga vi finner her så er det mye useriøst. Jeg vet ikke. Men er nok mye mer grundig de kundetilfredshet undersøkelsene vi har, er mye mer grundig vil jeg tro.

Intervjuer : Er det noe som gjøres bedre elller verre?

Respondent : Hele formålet med å ha en type undersøkelse er jo at man hele tiden skal forbedre seg og det er jo noe vi har fokus på hele tiden. I alle type arbeidsprosesser så har vi også det. I sånne agile prosesser er det jo alltid sånn retrospekter, hvor man ønsker å forbedre seg hele tiden og det er klart at man må jo ha det fokuset ut mot kunden og.

Intervjuer : Var det noe du savnet ved prototypen eller noe du vil endre?

Respondent : Sånne prototyper er alltid fine utgangspunkt for å gjøre nye ting, finne på nye ting etterpå og sånne nye forslag vil alltid komme tenker jeg. Kanskje bare måle aktivitet og på en måte, på en tweet for å sjekke liksom hvor mye folk tweeter til en bestemt motakker, det kan være interessant.

B.4 Subject D

Testing

Oppgave 1: Skriver inn “snapchat” uten å sjekke eksempler. Trykker på brukernavn, i stedet for å lese opp tweeten.

Oppgave 2: Sjekker eksempler og la til “update” og endret “snapchat” til “to:snapchat”

Oppgave 3: Løste oppgaven på første forsøk.

Oppsummering: Veldig rask utførelse. Men forstod prototypen og hvordan den fungerte ganske raskt.

Intervju

INTERVJUER : Hvilken informasjon finnes i prototypen?

RESPONDENT : Ser vell ut som du kan søke på meldinger.. twitter meldinger da. Med positive og negative. Litt usikker på hva de positive og negative egentlig er og hvordan de måles. Om det er positive meninger eller hva det er for noe er jeg litt usikker på.

INTERVJUER : Var informasjonen klar eller savnet du noe mer informasjon?

RESPONDENT : Ja siden jeg ikke er helt sikker på positive/negative, så savner jeg kanskje litt om hva det faktisk var.

INTERVJUER : Var informasjonen tydelig, forstod du hvordan du skulle bruke prototypen?

RESPONDENT : Ja den gjorde det. Jeg var jo litt kjapp der så, men det gikk vell fram etter hvert at jeg kunne skrive “To:” og sånt noe. Så det, måtte bare tenke meg om litt.

INTERVJUER : Var det noe i prototypen som var lett å misforstå?

RESPONDENT : Nei, kanskje ikke, men skulle ønske at det gikk ann å trykke “Enter” på søk, i stedet for å klikke på søke knappen

INTERVJUER : Hvilke klassifiseringer er brukt i prototypen og passet disse til resultatene?

RESPONDENT : Ja det var vell positive og negative og det er som sagt litt usikker på hva som ligger i det.

INTERVJUER : Er det noen andre klassifiseringer du kan tenke på som kunne vært nyttig i et slikt verktøy?

RESPONDENT : Ja, jeg er jo litt usikker igjen da, siden jeg ikke helt vet skjønte hva det var. Nei jeg vet ikke. Jeg kan ikke si noe.

INTERVJUER : Sammenlign hvordan det er å benytte et slikt verktøy mot hvordan dere vanligvis håndterer tilbakemeldinger fra brukere.

RESPONDENT : Ja det er jo på mail, så det går mye fortere her antakeligvis, enn det det vil gjøre på å lete gjennom mailbokser og finne fram.

INTERVJUER : Er det noe som gjøres bedre eller verre?

RESPONDENT : Det vil vell sikkert gjøre det bedre, problemet er vell at du får enda et nytt verktøy å forholde deg til i forhold til det man gjør i dag. Spesielt ute hos kunder så er de veldig flinke til å mye verktøy der og verktøy her. Men det her virka jo, uten at jeg vet hva som ligger bak, så virka det jo veldig raskt og nyttig å bruke, hvis man vet hva man leter etter.

Intervjuer : Var det noe du savnet ved prototypen eller noe du vil endre?

Respondent : Det er som sagt den “enter” knappen og når du trykker eksempler, når man da trykker søk så skal den forsvinne igjen, den eksempel, ikke at det skal ligge oppe. Og så er det som sagt litt, kanskje om det er prototypen eller hva det er, litt mer beskrivelse hva er det positive og hva er det negative, enkel forklaring. Det kan godt hende det ligger en hjelpe knapp der som jeg ikke så. Så jeg vet ikke helt hva om det bare er positive svar eller negative svar, det er det det går på.

B.5 Subject E

Testing

Oppgave 1: Skriver inn “snapchat” uten å sjekke eksempler.

Oppgave 2: Sjekker eksempler, tar litt tid på å lese gjennom. Skriver så inn “to:snapchat update”

Oppgave 3: Skriver inn “Microsoft skype” og søker, derretter retter opp til “to:microsoft skype”.

Oppsummering: Gikk veldig bra. Brukeren forstod hvordan bruke prototypen og klarte oppgavene fint.

Intervju

Intervjuer : Hvilken informasjon finnes i prototypen?

Respondent : Det er åpenbart at det er en søkemotor, som bruker twitter sitt API, den inneholder også søke eksempler som gjør enklere å bruke applikasjonen, men savner vell kanskje litt mulighet til å bruke knapper istedenfor, for å kunne sette sammen ett søk.

Intervjuer : Var informasjonen klar eller savnet du noe mer informasjon?

Respondent : Forslag å sånt ting er klart, savner kanskje checkbokser eller noe sånt for å spesifisere at jeg ønsker å søke etter bare dette eller bare dette, selv om det bare egentlig er at søkefeltet blir fylt med informasjon som jeg kunne ha skrevet selv, men at det gjøres for meg.

Intervjuer : Var informasjonen tydelig, forstod du hvordan du skulle bruke prototypen?

Respondent : Ja det gjorde det. Den virker relativt enkel å bruke, det eneste er at jeg ikke kjenner spørresetningene godt nok.

Intervjuer : Var det noe i prototypen som var lett å misforstå?

Respondent : Ikke egentlig. Etter å ha brukt den i 2 sekunder følte jeg at den var relativ enkel å forstå.

Intervjuer : Hvilke klassifiseringer er brukt i prototypen og passet disse til resultatene?

RESPONDENT : Du tenker på positive og negative for eksempel. Føler at de passa veldig godt, sånn bortsett fra den bommer på noen ting som er positivt, men det er vell bare en algoritme som må justeres regner jeg med.

INTERVJUER : Er det noen andre klassifiseringer du kan tenke på som kunne vært nyttig i et slikt verktøy?

RESPONDENT : Kanskje en form for neutral kategorisering, at ikke alt nødvendigvis er så sort og hvitt i forhold til positivt og negativt. Og selvfølgelig kunne filtrere på bare se positive eller bare se negative.

INTERVJUER : Sammenlign hvordan det er å benytte et slikt verktøy mot hvordan dere vanligvis håndterer tilbakemeldinger fra brukere.

RESPONDENT : Vanligvis så bruker vi et eller annet form for saksbehandling verktøy og sammenligna med det så vil jeg tro denne her kan fange opp, nå er jo denne her twitter-spesifikk, men det trenger jo ikke nødvendigvis være så relevant, denne her kan sikkert bruke flere andre og da tenker jeg at den kan være med på å fange opp ting som ikke blir meldt inn. Men ikke nødvendigvis er en erstatning, at den kan være mer et supplement til saksbehandlingsverktøy.

INTERVJUER : Var det noe du savnet ved prototypen eller noe du vil endre?

RESPONDENT : Nei ikke noe mer enn mulighet for å filtrere og kanskje også, jeg ser det jo nå i søke eksemplene at det er mulig å legge til dato, så en eller annen form for kalender funksjonalitet kanskje, så jeg kan finne siste måneden eller siste uka.