“What do they say about us on Twitter?”
Hybrid sentiment retrieval for organisations

Master Thesis - august 2013

Pieter Visser

<<This page was left blank intentionally>>

“What do they say about us on Twitter?” Hybrid sentiment retrieval for organisations

MASTER THESIS RESEARCH

Submitted in the partial fulfilment of
the requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE
TRACK INFORMATION ARCHITECTURE

by

Pieter Visser
Born in Rotterdam, the Netherlands

Web Information Systems
Department of Software and Computer Technology
Faculty EEMCS, Delft University of Technology
Delft, the Netherlands
http://wis.ewi.tudelft.nl/

GreenOnline B.V.
Valutaboulevard 27
Amsterdam,
the Netherlands
www.greenonline.nl

“What do they say about us on Twitter?” Hybrid sentiment retrieval for organisations

Author: Pieter Visser
Student ID: 1100599
Email: pieter@pietervisser.nl

Abstract

We conclude this report with a system design and proof-of-concept to show how an adaptable hybrid sentiment classification system is able to improve sentiment analysis for organisations.

GreenOnline, a service company in the field of customer services, wants to be able to quantify sentiment for organisations precisely, to create new services for organisations. To start with, this sentiment analysis will be based on Twitter messages.

The main challenge during this research was that Tweets, short WOM (Word-Of-Mouth) messages that contain only little words, are highly abbreviated and sentiment is expressed in subtle ways with irony, sarcasm, slang and other linguistic shades of grey [9]. Therefore, the focus of this thesis project was to design a system that is able to combine different sentiment analysis techniques to find sentiment. Also, not only existing algorithms were combined, but also information from the message (message attributes) are regarded as a way to determine the sentiment or which algorithm will classify the sentiment of that message best. Overall, it was regarded that all these different elements leave room for optimisation, for what algorithms and attributes to use and for what messages to select from Twitter for an organisation. To support a process of optimisation for a campaign or organisation another goal was to embrace the ability of system optimisation by (GreenOnline) customer service experts.

The result is a design and proof-of-concept implementation of a hybrid and adaptable sentiment analysis system design, which is using implementations of three sub classifier algorithms and message properties, that are combined by a hybrid sentiment classifier in a sentiment value of negative, positive or neutral.

This proof-of-concept implementation showed a performance of 71,2% which is a great improvement with respect to the single sub classifications of which the best performance was only 58,2%. By improvement of customer service experts this performance can even grow further.

Keywords
Sentiment analysis, hybrid classification, Twitter sentiment

Preface

This thesis report describes my final work for the MSc program of Computer Science at Delft University of Science, in which I studied the Information Architecture track. The assignment was performed in association with GreenOnline, a service company in the field of customer service and directed to find out how an adaptable hybrid sentiment classification can improve sentiment analysis for organisations. The research, design and proof-of-concept towards finding answers on this topic can be found in this report. This proof-of-concept showed a great improvement of hybrid sentiment classification of 71,2% over only 58,2% for the best single sentiment classifier and differences between hybrid combining techniques.

The thesis committee exists of prof. Dr. ir. Geert-Jan Houben and Dr. ir. Rafael Bidarra of Delft University of Technology, and Kees van Nuland as founder of GreenOnline.

During my study in computer science I have always had the curiosity of putting to practice my knowledge of computer systems, artificial intelligence and business analysis theory to create new and ready to use solutions for real problems. In this thesis assignment I was challenged by the topic of hybrid sentiment analysis for organisations, in which all my different fields of interest have come together and by which new valuable services can be made. It was a great assignment for me to work on, in part by the support of my mentors. I would like to thank Geert-Jan Houben, who helped me to discover the right approach and direction for myself by asking great questions and giving feedback. I also want to thank Kees van Nuland, for his enormous enthusiasm about the subject and seeing great value in this thesis, before putting it to practice. Finally, I would like to thank my friends and family for their love and support letting me accomplish this great achievement.

Pieter Visser, August 2013

Contents

Part I - Background & context 4

1. Introduction 5

1.1 From attention economics to sentiment analysis 5
1.2 Sentiment analysis in WOM messages: a challenging task 6
1.3 GreenOnline service in sentiment analysis for organisations 7
1.4 Approach & report structure 7

2. Problem situation & analysis 8

2.1 Goal: Sentiment analysis service for organisations 8
2.2 Project challenges 11
2.3 Intended contributions to sentiment analysis 12

3. Related work 13

3.1 Sentiment analysis techniques 13
3.2 Combining sentiment classifiers in hybrid classification 18
3.3 Evaluation of sentiment analysis 23
3.4 Personal interpretation of findings for application in a hybrid sentiment classifier 29

Part II - Design & architecture 32

4. Design of a hybrid sentiment analysis workbench 33

4.1 Strategy to hybrid sentiment classification 33
4.2 System design of an adaptable sentiment analysis system with hybrid classification 37
4.3 Workbench performance 43
4.4 Additional components 47

5. Proof-of-concept implementation of sentiment analysis workbench 49

5.1 Outline of proof-of-concept implementation of workbench 49
5.2 Starting points for implementation 50
5.3 Software development outline 50
5.4 Campaigns and their messages 54
5.5 Extracting attributes from messages 57

5.6 Combiners for Hybrid classification 61
5.7 Combiners using classification learning 63
5.8 Benchmarking 67

Part III - Results & findings 69

6. Findings 70
6.1 Performance of individual classifiers 70
6.2 Performance of hybrid sentiment classification 72
6.3 Noteworthy findings of hybrid classification 74
6.4 Role of message properties 76
6.5 Performance measures as indicators of performance 76

7. Conclusions & recommendations 77
7.1 Answers to research questions 77
7.2 Proof-of-concept evaluation 79
7.3 Future prospects for the sentiment analysis workbench 81

References 82

Appendix A Shortlist of corpus messages 86

Part I - Background & context

1. Introduction

1.1 From attention economics to sentiment analysis

As the principle of Bishop George Berkeley[i] states *"esse est percipi" ("To be is to be perceived")*, something only exists if it is perceived. In marketing terms, a product, brand or organisation is only present when people perceive their presence.

Consumers define their opinions based on marketing actions and opinions of others. A change to the concept of branding a product, brand or organisation is micro-blogging, in which consumers online discuss details and opinions about products or services with other people [28]. These actual word-of-mouth (WOM) referrals have substantially stronger effects than traditional marketing actions [29]. In this new field of WOM branding, consumers are gaining to attach value to opinions from within their social networks, while WOM opinions from outside someone's network such as online reviews are losing interest [28]. The constant connection between social networks and marketing is called the attention economy [3], in which a significant part of the current marketing takes place. Word-of-mouth branding is gaining interest of marketeers as well, focussing on social media where the most personal discussions take place.

Going back to Bishop George Berkeley, things have to be perceived to exist, but how something is perceived is just as important. It can be perceived in a positive, negative or neutral way. Because consumers significantly base their opinions about organisations on their social networks [28], it is important for marketeers to know what the sentiment about organisations is within these social networks and what effect their marketing actions have on that sentiment.

[i] Bishop Berkeley - Irish philosopher and Anglican bishop who opposed the materialism of Thomas Hobbes (1685-1753)

Since the emerging of social media networks, especially Twitter with its mostly public posts, a large amount of sentiment data became available [10]. Organisations want to know about this sentiment. The process of extracting sentiment from data is called sentiment analysis, which in this case can be used to extract sentiment about organisations in WOM messages.

1.2 Sentiment analysis in WOM messages: a challenging task

Sentiment analysis is identifying positive and negative opinions and emotions from text [30]. The outcomes of sentiment analysis in WOM messages have great value. Market sentiment can make or brake a product, service or a brand in the market and therefore sentiment data is a new kind of currency for organisations, but identifying sentiment in text is still a challenging task [9].

Sentiment expressed in subtle ways

Identifying opinions in human language as negative or positive statements will always be imperfect due to cultural factors and linguistic nuances [13]. These linguistics nuances cause simple algorithms to fail on capturing the subtleties that humans use in language: irony, sarcasm, slang and other linguistic shades of grey [9]. Even when no sentiment-bearing keywords are used, like "I bought a Honda", a message could still be negative or positive for Honda. Therefore, a great challenge sentiment analysis research is facing, is the challenge to deal with sentiment expressed in subtle ways (Pang - Opinion mining & sentiment analysis) [13].

WOM messages are short and abbreviated

On top of the subtle ways of expressing sentiment in language, it is even harder to distinguish this in WOM messages that are known to be highly abbreviated and contain even more room for interpretation (by humans and computers).

How to measure and compare performance of a system

Another challenge of sentiment analysis is that there is no standard way to evaluate the performance of a sentiment analysis system, partly because the accuracy is dependent on the task where sentiment analysis is performed at. Seth Grimes [13] states some companies claim to have about 95% accuracy on sentiment analysis for social media monitoring purposes and discusses that this is very dependant on the way it is measured, the task that is performed and furthermore that the commercial value of stating to have a high accuracy is also undermining the credibility of this high accuracy. In order to be able to compare outcomes of different classifications, a standard way to evaluate the outcomes is desired. Comparing different sentiment analysis classifications can then help improve sentiment classification incrementally, because effects of changes in a system can be evaluated.

1.3 GreenOnline service in sentiment analysis for organisations

GreenOnline is a service company in the market of customer contact. In the emerging market of sentiment analysis for social media, it sees great market potential for a future sentiment analysis service, able to provide organisations with sentiment data about their organisation. The first step for GreenOnline towards such a sentiment analysis service is to find a way to gain high accuracy on sentiment analysis, which is also the focus of this thesis project.

1.4 Approach & report structure

This thesis reports on a new approach for sentiment classification of WOM messages, called hybrid sentiment classification. It is structured in three parts. The first part introduces the subject; the second part contains the actual research, design and architecture used to meet the goals of this research; the final part discusses the conclusions from this research.

Part I

To get started, first the goal of this research is determined. After analysis of the problem situation, the main research question is defined accompanied with its subquestions towards answering this main question. This is described in Chapter 2.

Chapter 3 contains two sections describing the state of sentiment analysis in literature. The first section contains a general analysis of sentiment analysis in relation to social media monitoring and WOM messages for insight and understanding. In the second section, different approaches to design sentiment analysis architectures are discussed.

Part II

Chapter 4 answers the previously defined research questions by the design for a hybrid sentiment analysis system workbench, supported by the knowledge found in part I. An implementation of a part of this design was built to proof the design of the workbench. Details about this proof-of-concept implementation can be found in Chapter 5.

Part III

This research is evaluated by discussing the results and findings in Chapter 6. The conclusions that can be drawn from this discussion can be found in Chapter 7, as well as recommendations for future work.

2. Problem situation & analysis

2.1 Goal: Sentiment analysis service for organisations

GreenOnline wants to offer a service for organisations to monitor and analyse sentiment about their organisation, product or service. Accurate sentiment analysis data for organisations is valuable. Key to success of a sentiment analysis service is to be able to classify sentiment of messages influencing sentiment about an organisation accurately into positive, negative and neutral messages. Another strong selling point would be to give organisations insight in these messages through the sentiment analysis service. This insight is secondary to accurate classifications and therefore GreenOnline wants to focus on gaining a high accuracy in sentiment analysis for organisations first. The first step towards extracting sentiment with high accuracy is this research project.

Roadmap towards accurate sentiment analysis of WOM messages for organisations

Regarding sentiment analysis for organisations, GreenOnline is aware of the fact that in general sentiment analysis lies a challenging topic. Many resources can be used to extract sentiment for organisations, but earlier research showed word-of-mouth messages in social networks are mostly influencing sentiment [3] and have therefore the greatest interest of GreenOnline at this point. These WOM messages are micro-blogging messages, that are actually short online messages, mostly posted in social media networks.

Micro-blogging messages in social networks, especially Twitter messages, have their own style of writing using their own language, # (hashtag), RT(retweet) and @ to address someone and due to the limit of 140 characters per message, sentences are highly abbreviated and context is sparse [7]. Current sentiment classification techniques (classifying the sentiment in classes of positive, neutral or negative) however are basically designed to extract meaning of large text corpora that are written in full and neat sentences, like reviews and articles. The effectiveness of current sentiment analysis on micro-blogging messages techniques was reviewed by Blenn [7], showing a accuracy of 50-60% on Twitter messages. To improve the accuracy on micro-blogging messages GreenOnline wants to develop an approach by combining several existing techniques or classifiers in a hybrid classifier.

Scope and scientific challenges in classifying sentiment of Twitter messages

There are two main scientific challenges to extracting sentiment from especially short Twitter messages. At first, due to the typicality of these short messages, current sentiment analysis

techniques have disappointing accuracies of 50-60%. Although these short messages contain less predictive words for sentiment, sentiment is present and therefore the first challenge is to extract this sentiment from WOM messages.

Secondly, it is questionable how representative these messages are for the actual general sentiment about organisations. Posting a message on Twitter has a different threshold for everyone, and there is only a small group of people using Twitter frequently. Also, one could think of messages that are written with a purpose, to gain interest of friends, followers, or of an organisation or for marketing reasons. Altogether, the representation of the actual sentiment in Twitter messages is questionable.

This thesis assignment focuses on the first challenge of sentiment in short Twitter messages. Wether these are representing the actual general sentiment about organisations is out of the scope of this project.

Smart choice based on properties and sub classifications: a hybrid classifier

There are multiple ways in which sentiment can be packed inside small WOM messages and for each a different approach is best in finding this sentiment. The expectation that is driving this thesis work is that the sentiment analysis process can be optimised by making a 'smart' combination of techniques based on information about a message in order to find the right sentiment. In this thesis, information about a message is represented as a collection of attributes: Property values and classifier outcomes. These classifier outcomes are further called sub classifications to distinguish them from the hybrid classification.

Messages with for example more than five capitals might be best classified by sub classifier A, and messages ending with an explanation mark by sub classifier B. Messages with multiple positive emoticons might mostly be positive. In other words, it is expected that each sub classifier is good at classifying sentiment of messages with certain properties and message properties might also give direct clues about the sentiment of a message. This approach is called hybrid sentiment classification.

A challenge of this hybrid approach is to find a way to learn for which messages to use which sub classifier outcomes, and for which a certain property value is key to base classification of sentiment upon. Another challenge is to overall find properties that substantially influence sentiment in order to support hybrid sentiment classification.

The differences between attributes, properties and sub classifications are explained in paragraph 4.1.

Improving micro-blogging sentiment analysis

Sentiment classification techniques should be optimised to improve performance on short WOM messages. GreenOnline states that customer service experts can use their expertise and ability to recognise trends on this topic to optimise a dedicated tool. Although incremental changes on a trial-and-error basis can affect the outcomes and this accuracy positively, changes driven on quantified evaluations and insight can improve performance incrementally better.

This scientific approach to incremental improvements should be able to give deep insight in effects of changes made. Evaluation quantifications like the accuracy rate and other measures should give the customer service experts valid feedback of changes made in order to make changes that count and incrementally improve sentiment analysis for WOM messages.

Besides, this dedicated tool operated by customer service experts should from any location be easy to operate through a web interface, be easily accessible and operable without any software requirements. Incremental changes should be made from within this web interface, rather than changing the underlying software by code, which requires a developer.

2.2 Project challenges

The main challenge in the goals of GreenOnline is to design an approach for hybrid sentiment classification, to be used to perform incremental improvements by customer service experts using a dedicated and easy to use tool. This design challenge can be broken down into a few challenges that this project focusses on:

- Design a hybrid sentiment classification approach to WOM messages, in which outcomes of different sub classifiers and property values of messages are used. The hybrid classifier should learn how to use different message attributes (sub classifications and message properties) to classify sentiment of a message.
- How to implement a hybrid sentiment classification system that customer service experts can control through a web interface?
- How can customer service experts observe effects of changes that enable them to incrementally improve sentiment analysis for WOM messages about organisations?

2.2.1 Main research question

This research targets how an adaptable hybrid sentiment classification tool can improve sentiment analysis for organisations. The main research question is therefore stated as follows:

How can adaptable hybrid sentiment classification improve sentiment analysis for organisations?

2.2.2 Sub questions

To answer the main research question, answers to the following sub questions should be given.

1. *How to design a hybrid sentiment classifier strategy?*
   The strategy for hybrid classification should be able to classify a message's sentiment, using a combination of different sentiment classifiers and values of properties for this message. Both the result of the sentiment classifications and values of properties influence the final sentiment classification.

2. *How to design a system architecture for the hybrid sentiment classification strategy that is adaptable by customer service experts and what are the underlying components of this design?*
   The strategy found in subquestion 1 should be used to design a system in which customer

service experts are able to adapt the sentiment classification strategy and get feedback of the performance for these changes.

3. *How does the chosen design enable customer service experts to improve sentiment analysis for organisations? And how does the design improve performance of sentiment analysis?*
   In the adaptable and hybrid system design it should be regarded that customer service experts can improve sentiment analysis by its adaptability and that this system design improves sentiment analysis performance. A proof-of-concept of this hybrid system design will show how this system enables customer service experts to improve sentiment analysis for organisations.

2.3 Intended contributions to sentiment analysis

By answering the research questions above, the intention is to contribute to sentiment analysis with a system design that combines different sentiment classification techniques, is adaptable and is, by its design, able to incrementally improve sentiment analysis for organisations.

To conclude the contributions with hands-on information, an architecture for implementing the supposed system design is also elaborated in a proof-of-concept implementation in order to show how customer service experts can use this.

3. Related work

This chapter starts with three sections describing the state of the art in sentiment analysis and other related work. In paragraphs 3.1-3.3 we objectively analyse current sentiment analysis techniques and approaches in relation to social media monitoring and WOM messages for insight and understanding.

Afterwards in paragraph 3.4, we discuss how we interpret these insights in relation to the design of a hybrid sentiment classifier, and we discuss how they fit in a hybrid architecture.

3.1 Sentiment analysis techniques

Sentiment analysis is the set of techniques to identify sentiment in text, either a document or a sentence. The sentiment or opinion about the subject can either be a negative or positive polarity that deviates from the neutral state [31]. It is common to first check if a message contains sentiment (subjectivity extraction, also see 3.1.4), whereafter messages with sentiment are classified in two classes of sentiment: positive and negative [12] using classification algorithms, called sentiment classifiers (also see 3.1.3).

Three class classification

If during training of sentiment classifiers only positive and negative messages are used, the classifiers will not be trained to accurately classify neutral messages. Moshe Koppel [32] found that excluding the neutral sentiment class is unfounded and shows that using this neutral class of sentiment as well in training data, improves sentiment classification. Moreover, it also improves the classifiers' ability to distinct positive and negative messages [32].

3.1.2 Natural Language Processing

Natural Language Processing (NLP) is the collective name for several techniques that are used as machine translation, translating human written text into suited forms for computer computations, in order to extract information. In sentiment analysis, different NLP techniques are used. Below, some common NLP techniques are discussed that are relevant to the research question.

Text segmentation

Text segmentation is a process which segments text based on the boundaries between words or phrases into pieces that are one sentence or paragraph. The segmentation involves the differentiation of usage of a character to end a line, or as a an expression. In the phrase *"Mr. Smith is in Paris."* Mr. is not a separate sentence and the capital of Paris also doesn't start a new sentence. Text segmentation can be used by other (NLP) techniques [14].

Stemming

Stemming is a proximate method for grouping words with a similar basic meaning together. Each word is reduced to its root or stem, by removing suffixes. Words with an identical stem usually have a similar meaning [15] and could have similar sentiment. For example, *connect, connected, connecting, connection* and *connections* all have the same stem: *connect*. Using stemming, the number of different words in a text is drastically reduced, which simplifies sentiment classification [17]. Text segmentation and stemming can be used in combination with any sentiment classifier, other NLP techniques are specifically used by classifiers themselves and will be discussed below.

Part-of-speech tagging

Part-of-speech tagging, mostly referred to as POS tagging, is the technique to mark each word corresponding to a particular part-of-speech: nouns, verbs, adverbs etc. [11]. POS tagging is often used in sentiment analysis approaches, where the existence of sentiment is based on particular POS pattern, including an adjective or an adverb.

POS tagging is challenging due to the characteristics of micro-blogging messages: messages together form a conversation, messages do not consistently follow language rules for spelling, punctuation and capitalisation and messages are limited to 140 characters. These characteristics degrade performance of POS tagging [11].

3.1.3 Sentiment classifiers

There are several different classifiers that can be used for sentiment analysis. In general, a classifier takes a message as its input and it will determine the sentiment of that message as the output, shown in figure 1 below.

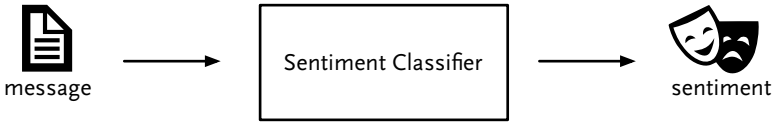

Figure 1 - Sentiment classifier

Different classifiers use different algorithms to determine the sentiment. To give a short insight in how different these algorithms are, a few techniques are discussed here. Some classifiers just count positive and negative words (according to a pre-defined lexicon of words and their related sentiment), others are able to construct a lexicon themselves by learning and use that to count negative and positive words. In doing so, messages can be seen like a bag of words (without any order or relation of words within the message), or the relation of words (grammar) can be used for determination of sentiment. Vector based sentiment classifiers predict sentiment by comparing the vector of a message to earlier known analysis.

Examples of often used sentiment classifiers are discussed below.

Word count classifier

The word count classifier classifies sentiment (positive, neutral, negative) based on the total number of words that express sentiment in the message. It uses a specific lexicon with hand-picked words, that are commonly used to express sentiment. Each word in the lexicon is identified positive or negative, which can be used to calculate the total number of positive and negative words for a given message. Words in the message that are not found in the lexicon are automatically regarded as neutral. When the total number of positive words exceeds the negative ones, the message is classified as positive. When there are more negative words then positive ones, the message is classified as negative [18].

Naïve Bayes classifier (NB)

Naïve Bayes is a statistical classification method that calculates the probability of classification for each class based on the features of the subject. In the case of sentiment analysis the subject is the message, and each word in it is a feature $f_1..f_n$. Thus, it uses the presence of each word to predict wether a message is more likely to be in class A than class

B based on statistics. Naïve Bayes assumes each feature is independent of other features. The classifier need training with messages and their according classes in order to learn what words appear in positive or negative messages [12, 17].

The probability of classification c_i for a given message d is calculated as follows [12, 17]:

$$P(c|d) = \frac{P(c)P(d|c)}{P(d)} = \frac{P(c)P\left(\prod_{i=1}^{n} P(f_i|c)^{n_i(d)}\right)}{P(d)}$$

_K_-nearest neighbour (KNN)

KNN is a machine learning algorithm classifying a message according to the classes of messages that are similar. During training of a KNN classifier it renders each message in a virtual space of features (words in a message). New messages are also rendered in this space, whereafter a majority vote of the classes of the nearest neighbours in the feature space will determine their classification, where *k* is the amount of neighbours that are included in voting [35].

Support vector machine (SVM)

SVM is a machine learning model used to predict the class of an object, based on pattern recognition of input data. The examples are represented as points in space, forming two clearly separated groups: Positive and negative messages (and also neutral for a three class classification approach). New samples are mapped in that space as well and according to their location (in respect to the groups of sample message that represent different sentiments) it is predicted to which group they belong, and thus what the sentiment of the message is.

Emoticon classifier

Research by Yuasa et al. [19] has shown that emoticons in text serve as emotional indicators similarly to how our brain processes other nonverbal means. Emoticons can therefore be good indicators for sentiment polarity in a message. This dedicated classifier to extract sentiment from the emoticons in the message is, like other classifiers, challenged with irony and sarcasm as well as ambiguity of meaning by using multiple conflicting emoticons [19].

Sentiment classifier challenges

Absence of subjective words

Sentiment classifiers are algorithms that come from topic categorisation and actually do not perform as well on sentiment classification [12]. Extracting a topic from a message is a different task than extracting sentiment, where topics can often be identified by keywords

alone and sentiment is expressed in a more subtle manner. Consider the following movie review: *"How could anyone sit through this movie?"* contains no single word that is clearly subjective, still the review is [12]. Using current sentiment classification techniques it is important to take into consideration that they are mostly built to extract obvious indicating words for a classification.

Ambiguity of sentiment indicators
When analysing a message it can return conflicting indicators for sentiment, some negative, some positive. Classifiers decide what to do with this information in order to classify the message's sentiment.

3.1.4 Subjectivity extracts

A sentence is either subjective (with sentiment) or objective (describing a topic). For sentiment analysis only subjective messages should be taken into account, others should be classified as neutral. Pang [20] proposes a method to first create an extract with subjective sentences only, before classifying sentiment. Below a figure of this strategy is shown (figure 2), showing first on the left the separate sentences s_n of a review, then to the right detection of subjectivity for each sentence s from which only subjective sentences pass through, being a selection of the messages s_m that is the input for the polarity classifier.

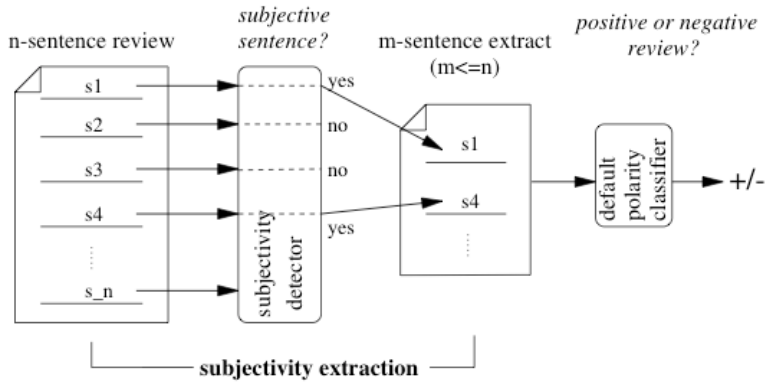

Figure 2: Polarity classification via subjectivity detection [20]

Using this subjectivity extraction Pang [20] showed significant improvement (from 82,8% to 86,4%) on sentiment classification from documents using only 60% of the words as the subjectivity extract. Although the actual percentage of performance does not tell really much, as it is mostly dependent on the performance definition, the improvement of 4,4% in the last 17,2% to perfection is significant enough to take a closer look.

Messages with clear and consistent evidence of positive or negative sentiment will be extracted correctly for a high percentage. The sentiment analysis difficulty lies in messages that have evidence for both polarities of sentiment and messages in which sentiment is expressed in subtle way. Clearly, leaving this last type of messages out for sentiment analysis would increase performance of the messages that are analysed, but it is questionable wether it is also a better reflection of the actual sentiment. Further research should be done in order to find out what effect leaving out messages with questionable sentiment has on sentiment analysis, but that is out of the scope for this thesis.

WOM messages are packed with subtle expressions of sentiment, it is hard to simply extract subjectivity. Therefore, extracting subjectivity should be as well accurate as sentiment classification. To do so, a similar set of techniques should be used to extract subjectivity accurately from WOM messages like the actual sentiment classification techniques.

3.2 Combining sentiment classifiers in hybrid classification

Individual sentiment analysis techniques have shown minimal results on three class sentiment classification for short WOM messages like Tweets: only 50-60% was the best accuracy Blenn [7] found. Combining different techniques might result in better performance.

3.2.1 Combining by voting or mean class

Combining by majority voting

Das [18] uses five sub classifiers in isolation and applies a simple voting system thereafter to reduce false positives. This voting system defines whether a message is given the same classification by the majority of the sub classifiers (in this case of 5 classifiers 3 or more) and assigns that class to the message. If no majority is found, the message is assigned to be neutral.

Combining by mean class

The overall sentiment can also be represented as the mean of all sentiment values. Positive messages are 1, negative -1 and neutral score a 0. Some errors in classification occur, resulting in some incorrect sentiment values that influence the overall sentiment incorrectly. The higher the performance of sentiment classification, the more precise the overall sentiment value will be.

Errors that have high discrepancy from the actual sentiment cause higher deviation of the overall sentiment to the actual current sentiment and are therefore more costly. In classifying

a message as positive (1) instead of negative (-1) the discrepancy is 2, where classifying messages neutral (0) instead of positive (1) only creates a discrepancy of 1.

3.2.2 From majority voting to hybrid classification

Looking at the majority voting technique to combine sentiment classifications shows that this technique is also a classifier. Its input is not the message itself, but the outcomes of sentiment classifiers. The output is the sentiment of the message.

To differentiate between the sentiment classifiers discussed in 3.1.3 and a classifier that uses not the message itself, but the messages's attributes, here I introduce the term 'combiner' for this new type of classifier. Also see figure 3 below.

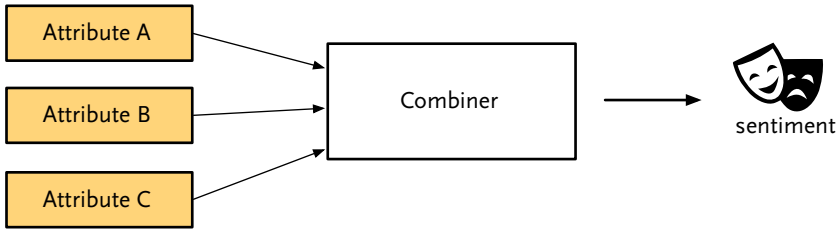

figure 3 - Combiner: combining different attributes into sentiment

Combiner: A classifier that uses a message's attributes as input to classify sentiment.

The combiner is not a 'real' sentiment classifier, in a sense that its input is not a message, but a collection of attributes. To create a sentiment classifier for which its input is a message and uses a combiner to classify sentiment I introduce the term 'hybrid sentiment classifier'. This hybrid classifier also consists of a component to extract attributes from the message which the combiner needs. See the figure 4 below.

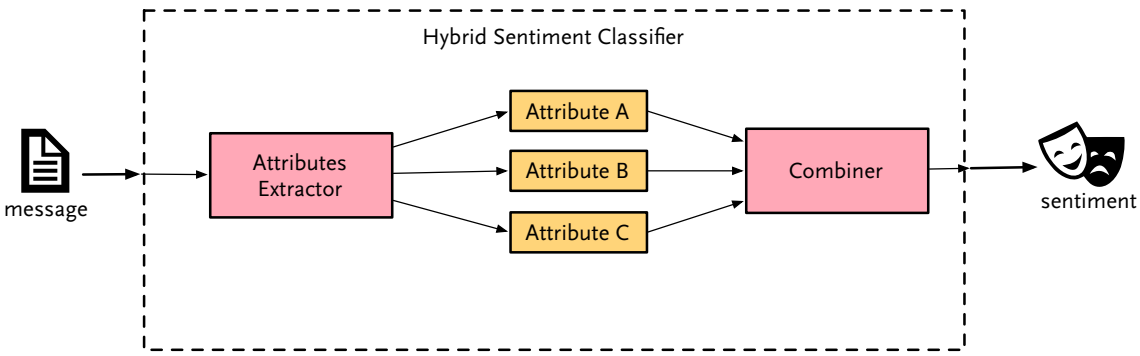

figure 4 - Hybrid sentiment classifier

Hybrid sentiment classifier: A sentiment classifier that uses a message as input and a combiner to classify the sentiment of that message.

Attributes, property values and sub classifications defined

From a message different indicators for sentiment can be extracted to be used by the combining classifier to classify sentiment of a message. All different indicators are called message **attributes**. Throughout the rest of this thesis two types of messages attributes are used: **sub classifications** and **property values**, also shown in figure 5 below.

An example of a property value is the number of words in message or the gender of the author. A sub classification is the sentiment that is extracted by a sentiment classifier as discussed in 3.1.x, for example the outcome of the Naïve Bayes classifier for the message. Sentiment classification outcomes that are used as message attributes are called **sub classifications**, to distinguish them from hybrid classifications throughout this thesis.

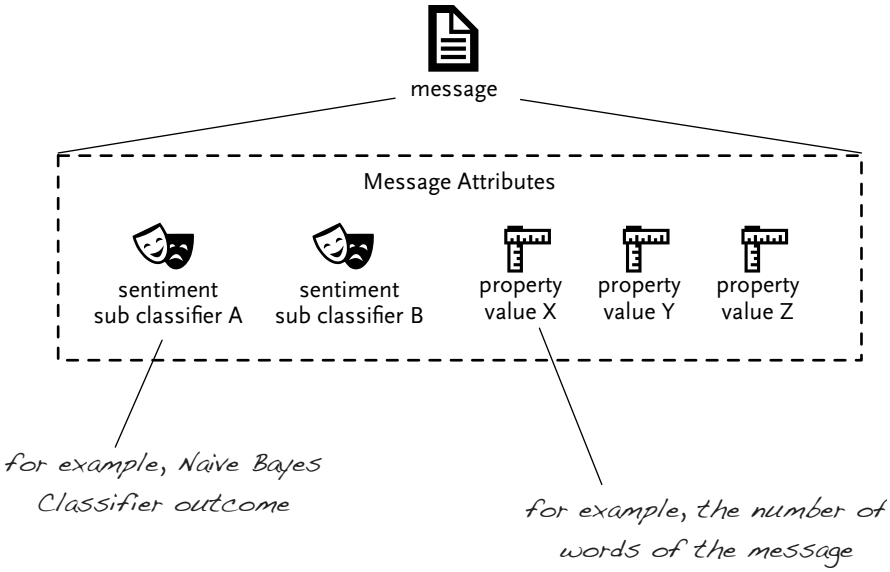

Figure 5: The relation of a message to its attributes: From a message different attributes can be extracted resulting in different sub classifications and property values. An example of a sub classification is the sentiment classified by the Naive Bayes classifier (sub classifier A), an example of a property value is the number of words of the message.

3.2.3 Classification learning to create a hybrid classifier

By trial-and-error new hybrid classifiers can be designed, where each new classifier uses a 'special' combination of rules to classify the sentiment based on the message's attributes. Another way of creating a classifier is by learning from a set of sample data. This is called classification learning. This technique can be used to create hybrid sentiment classifiers. The combiner of the hybrid classifier predicts sentiment of messages according to a model it has

learned from a set of sample messages for which the sentiment is already known. Different learning schemes can make a model of a set of sample messages by machine learning, for example Decision Trees and Decision Rules. These models describe relations of sample data attributes to the sentiment in a message. For new, unknown messages the sentiment can then be predicted by applying the model to the new message's attributes.

Decision trees

A basic decision tree is determining an outcome on conditions, that are split in multiple levels and arranges samples among different groups. Decision trees are created by the divide-and-conquer principle, splitting the set of items on an attribute. Below an example in figure 6:

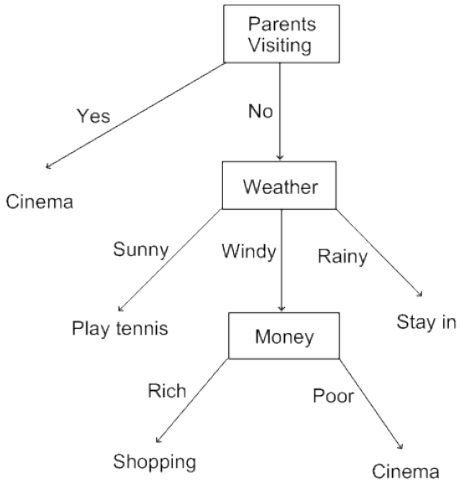

Figure 6: A basic decision tree [26]

In the above decision tree (figure 6) the conditions (square boxes) could be each message's attributes, from a property value to the outcome of a sub classifier.

C4.5 - J48 decision trees

In the description of decision trees in the WEKA data mining book of Witten 2011 [8], C4.5 is named "*a landmark decision tree program that is probably the machine learning workhorse most widely used in practice to date*" [Witten 2011]. It is based on the first model tree described by Ross Quinlan in 1993, called ID3 algorithm, working top-down by choosing a variable at each step that is the next best variable to use in splitting the set of items [22]. When all items have the same class, a leaf node for the decision tree is created, giving that pass that node that class.

C4.5 is written in the programming language C, and J48 is an implementation in Java that is also available in the WEKA data mining tool.

One-Rule decision trees

One-Rule, also called OneR, is actually a decision tree with only one level. For each attribute of the messages in the training data one rule is created based on the most frequent class for an attribute value. From all single rules, the rule with the lowest error rate is chosen as the one rule to use for classification [33].

Although OneR seems simple, results are only slightly less accurate then state of the art classification learning.

Decision rules

Decision rules are another way to make a model of the sample data. A set of rules describes the rules to follow to identify the decision (in this case classification) to make. The difference with decision trees is that it is based on the separate-and-conquer principle, that identifies a rule to cover a large group of samples. After each rule formed, the covered messages are taken out of the sample set, after which new rules are formed to cover the remainders.

Prism

Prism is a covering algorithm that generates a set of decision rules by trying to cover all instances of a class by a rule. This way all instances are covered by a rule and a set of decision rules is created [8].

Comparison of decision rules and decision trees

Decision trees and decision rules can represent the same model, but in many occasions the decision rules are more compact and therefore easier to understand. Below an example of a small set of decision rules replicated in a decision tree in figure 7.

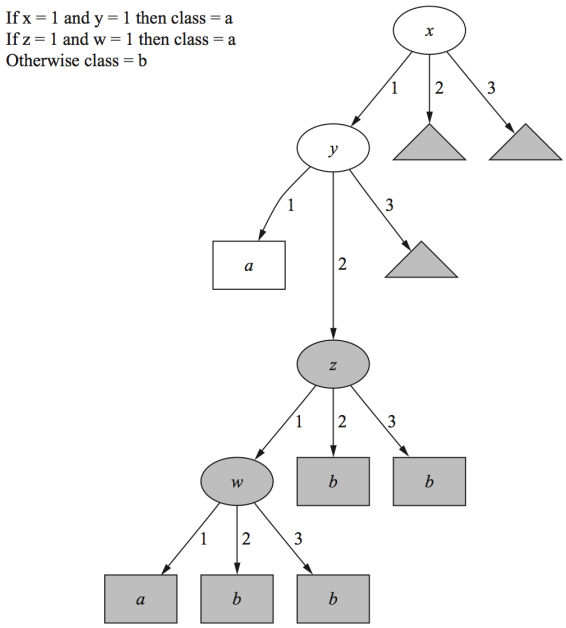

Figure 7: Decision rules with a replicated decision tree

PART

The PART algorithm creates a set of decision rules by using parts of J48 decision trees. PART generates a number of rules, selected before running the algorithm, from each best leaf in the tree.

When using J48 decision trees already, the PART decision rules add little differentiation to combining knowledge, because it uses the same J48 algorithm for rule generation [8].

3.3 Evaluation of sentiment analysis

The performance of sentiment analysis expresses how accurate the output of a classification is, what the rate of correctly classified messages is, called **accuracy** throughout this thesis. Other measures are also regarded, discussed in 3.3.2. Manual classified messages are regarded to be messages of which the sentiment is known. After classifying these messages by a classifier, results can be compared to the actual sentiment and this way the performance can be measured.

While measuring performance, some issues should be regarded, which are discussed further in this paragraph.

3.3.1 Test & training data sets

Messages of which the sentiment is known (by manual classification for example) are needed to train classifiers and to measure performance. According to Bifet [1,2], it is challenging to evaluate data streams (like Twitter messages) in real time, where literature mostly considers how to build a picture of accuracy afterwards. He shows there are two main approaches to use this data for training and testing:

- **Holdout**: Performance is measured using on single holdout set. A set of messages is once classified manually in order to serve as training data for classifiers as well as test data for performance measuring.
- **Interleaved Test-Then-Train or prequential**: Each individual example is used to test the model before it is used for training, and accuracy is incrementally updated.

The first holdout model is a basic approach that does not consider the actuality of messages over time and the effect of using the same messages for training and testing. In the second model Bifet shows new sets of test messages are made in order to be able to always test with messages that are not used for learning and that are a new reflection of the actual types of messages. After using the new test messages for testing these can be used for training, because a new test will use yet another set of new test messages.

When data for test and training is limited, it is common to use one third of the data for testing and two thirds for training [8].

3.3.2 Selecting representative and balanced test & training data sets

Stratification

In most classification problems the amount of test and training data is limited and therefore it might easily occur they are not balanced in such way that each class is represented proportionally in both the test set and training set. Stratification is the process towards balanced training and test sets [8].

Stratified cross-validation

Another way to balance test and training sets is to use stratified cross-validation, it is not only covering the challenge of balanced classes, but also a balanced distribution of errors. The whole test is divided in different folds, for example three folds, and then each fold is subsequently used for training, while using the remainder for testing. To errors are averaged, representing the overall error rate. Using 10 folds has become the standard, but 5 or 20 folds are likely to perform similar [8].

3.3.3 Measuring performance of sentiment analysis

In depth information about performance to improve system

In depth information about performance gives customer service experts insight in how changes effect the hybrid classifier and how the overall performance can be improved. Evaluation can show for example a high error rate in negative messages classifying as neutral, or that of all wrong classification most are classified positive. Being able to improve these errors without giving in on other subjects will be a great opportunity for this hybrid sentiment classifier to be developed into a good performing system.

Below an overview of different ways to show performance of (sentiment) classification, that corresponds to what is used regularly in literature of comparable research subjects. In most sentiment classification problems, only two states of sentiment are regarded, positive or negative, and therefore performance is mostly presented in binary measures. All the measures discussed below can also be used to represent a three-class classification problem.

Confusion matrix

The confusion matrix is a way to represent the data found in different classes, from which all performance measures can be calculated [25]. Below, in figure 8, the model of a confusion matrix, in which the columns represent the different classes that are predicted and the horizontal lines represent the classes that the messages actually belong to (according to manual classification). From the positive predicted messages, the "true positives" or TP are the ones that are actually positive, otherwise they are "false positives" or FP. From the negative predicted messages the ones that are actually positive are "false negatives" (FN) and when the negative prediction corresponds to a actual negative classification the results are "true negatives" or TN.

		Predicted class	
		positive	negative
Actual class	positive	TP	FN
	negative	FP	TN

Figure 8: Abstract binary confusion matrix

Accuracy (a)

Accuracy is an overall proportion of correct classifications from all classifications. A high accuracy represents accurate sentiment classification.

$$a = \frac{TP + TN}{TP + TN + FP + FN}$$

In each application of sentiment analysis a high accuracy is desirable, but it is also important to know in what context the errors occur. Some errors are more critical than others and knowledge about these errors can be used to improve performance. The distribution of classifications amongst the different types of predictions in a binary classification are TP, TN, FP and FN.

Precision (p)

The precision is the fraction of true positives from all messages that are predicted as positive, also called the "positive predictive value (PPV)".

$$p = \frac{TP}{TP + FP}$$

Recall/sensitivity (r)

Recall is representing the rate of true positive classified messages from all actual positive messages, also called "true positive rate" or sensitivity.

$$r = \frac{TP}{TP + FN}$$

Specificity (s)

Also called "true negative rate", represents the rate of negative predictions from all negative messages.

$$s = \frac{TN}{TN + FP}$$

F-Measure

Where precision, recall and specificity evaluates the classifications on a class level of different types of errors, F-measure is a measure for the distribution of different types of errors. For overall insight in classification performance accuracy and F-measure are good indicators. For deeper insight in types of errors precision, recall and specificity can be used.

$$f = 2 \cdot \frac{p \cdot r}{p + r}$$

3.3.4 Three-class classification performance

In 3.1 it was discussed that including a third class of sentiment, neutral, suits sentiment analysis better and brings with it three-class classification. The above methods to represent performance of sentiment classification is based on a two-class sentiment analysis, where messages are either positive or negative.

Three-class measures for performance

Defining the performance measures in a three-class classification differs a bit from the standard binary classifications. As shown in figure 8 above, the confusion matrix shows other outcomes than the three-class confusion matrix in figure 9 below. When a prediction is equal to the actual class, the prediction is true (T). In other cases the outcomes are errors (E). Below the calculations of the performance measures for three-class classification. See figure 9 below for the confusion matrix and performance measures for a three-class classification:

		Predicted class		
		A	B	C
Actual class	A	T_A	E_{AB}	E_{AC}
	B	E_{BA}	T_B	E_{BC}
	C	E_{CA}	E_{CB}	T_C

Figure 9: Confusion matrix for three-class classifications

Accuracy $$a = \frac{T}{T + E}$$

Precision $$p_A = \frac{T_A}{T_A + E_{BA} + E_{CA}}$$

Recall $$r_A = \frac{T_A}{T_A + E_{AB} + E_{AC}}$$

Specificity $$s_A = \frac{TN_A}{TN_A + E_{BA} + E_{CA}}, \text{ where } TN_A = T_B + E_{BC} + E_{CB} + T_C$$

[24]

Sentiment classification of short WOM messages is difficult due to the little information the messages contain, which counts for subjectivity extraction as well. When using a two-class

classification, both systems for extracting information from messages should be optimised. Using a three-class classification for sentiment analysis decreases the need for subjectivity extraction where neutral messages are classified correctly as well.

3.3.5 Cost-sensitive learning

In almost each classification problem some errors cost more than others. "*The cost of sending junk mail to a household that doesn't respond is far less than the lost-business cost of not sending it to a household that would have responded*" [27]. In this case, false positives might have a different cost than false negatives.

Two classifier outcomes with the same overall accuracy, as discussed in 3.3.3, can have different distributions on error types and therefore their real performance could be different. Including a cost matrix in the learning process of a (hybrid) classifier will improve accuracy, regarding the costs of errors. Figure 10 below shows an example of a cost matrix for a three-class classification.

		Predicted class		
		A	B	C
Actual class	A	**0**	1	1
	B	1	**0**	1
	C	1	1	**0**

Figure 10: Cost matrix for three-class classifications

When certain errors would cost more than others, the above cost matrix can be adjusted. An example of how to set up costs is that classifying an actual positive message as negative is more faulty than classifying it is neutral. Also, missing out sentiment containing messages could be less expensive than classifying neutral messages as positive or negative. Another way to use cost-sensitive learning is to make financial models of costs of a classification problem, where error costs are related to a value of profit or loss to an organisation.

3.3.6 Sentiment classifiers in respect to active learning systems

Active learning systems are systems that keep on learning the model of a continuously/ frequently offered new sample of messages. A sentiment analysis workbench should frequently receive new messages to keep up with the latest types of messages it can expect to classify and should therefore contain an active learning system.

In the study of Brew 2010 [1,2], *"Using Crowdsourcing and Active Learning to Track Sentiment in Online Media"*, different sentiment classifiers have been supervised on tracking sentiment in online media. They used three previously proven classifiers: Naïve Bayes, SVM and κ – nearest neighbour. The performance of SVM is marginally better than naïve Bayes, and KNN did not perform well at all. More importantly for the purpose of using a classifier in an active learning system, another difference of these classifiers was found in algorithm time complexity, which is important when considering active learning (frequently repeated learning) in a system. It was found that Naïve Bayes has a linear learning time and SVM has cubic learning time. For the use in a system that involves learning frequently, it is preferable to use Naïve Bayes over SVM [1,2].

3.4 Personal interpretation of findings for application in a hybrid sentiment classifier

This paragraph describes the interpretation of the knowledge and insights found in the above paragraphs for application in this thesis. Topics are discussed in order that they appeared in the above paragraphs.

3.4.1 Personal interpretations for WOM classification

Three class sentiment classification

In the design of a hybrid sentiment classifier in chapter 4, the findings of Koppel [32] are taken as a basis to use three classes of sentiment in classification for this thesis work. Another advantage of using a three-class classification is that it decreases the need for a subjectivity extractor in front of classification. Sentiment classification and subjectivity extraction are much alike and therefore it would be doubtful wether to make two separate systems for their tasks. Overall, it makes implementation and optimisation of the hybrid sentiment classification more dense and straightforward, which is also a benefit in performance evaluation.

Text segmentation

Short WOM messages, typically contain only one (partly) sentence and when done this should also be done very carefully, otherwise segmented sentences will not represent the messages and sentiment classification performance will decrease. In this thesis segmentation is not further taken into account.

Part-of-speech tagging

POS tagging is known to have a low performance on short WOM messages [11] and it is not further considered in this thesis.

3.4.2 Measuring performance of sentiment analysis

In comparable research, performance of sentiment classifications is mostly expressed in a confusion matrix with the accuracy- and F-measure. Although these are good indicators of how the performance of sentiment analysis changes, they are limited and too superficial to show effects of small changes to the hybrid sentiment classification system. Classifications with similar accuracy and F-measures can still have very different performance on the other measures.

Therefore, to expose these differences it was chosen to return all different performance measures to the user in a way it can understand the internal changes they made to the system. The hybrid classification system can be evaluated by comparing performance measures of two states of this system and this way the effects of changes can be evaluated. By gaining insight in how changes affect the system, it can be incrementally improved into a high performing sentiment analysis system. For best insight not only the confusion matrix, accuracy and F-measure will be given, but also the precision, recall and specificity.

3.4.3 Subjectivity extracts

Extracting subjectivity from WOM messages has proven to be difficult, but necessary in a two-class classification system. An alternative to this is to perform three class sentiment classification.

Subjectivity as neutral sentiment class

A two-class classification system without a subjectivity extractor will classify all neutral messages incorrectly as polarised messages, resulting in a bad performance. Introducing a third sentiment class, neutral, to the classification system is another way to deal with neutral messages. This way, sentiment classification and subjectivity extraction can be performed by the same components of the classification system.

In a three-class hybrid classification system a dedicated classifier for subjectivity extraction can be used as well. The outcome of this subjectivity classifier can be used by the hybrid classifier in order to predict sentiment of the message. For example, when the subjectivity classifier would predict there is no sentiment and on top of that the outcomes of other classifiers is mainly neutral, good chance that there is no sentiment in the message.

3.4.5 Three-class classification

For this thesis assignment a three class classification is preferred. It brings along a little more complex calculation of performance measures, but the user will still be able to compare these values similarly. Also, evaluation of performance of sentiment analysis might even be better to understand, because the effect of neutral messages is considered in the central (hybrid) classification rather than using two types of extraction.

Part II - Design & architecture

4. Design of a hybrid sentiment analysis workbench

4.1 Strategy to hybrid sentiment classification

To answer the first sub question, "*How to design a hybrid sentiment classifier?*", a strategy for hybrid sentiment classification needs to be formed to use as an abstract approach to sentiment classification. From this basic and abstract approach to the full functionality of the workbench, the overall design for the workbench will be made in 4.2.

Das and Chen [18] developed a method to extract sentiment from Twitter messages. In this method they used five different classifiers and combined their outcomes by majority voting.

The method of Das and Chen is a simple form of the strategy that is needed for this thesis and was taken as a starting point. Not only multiple classifiers can be combined, but other information extracted from a message can be used as well, like number of words in a message or wether it ends with an explanation mark. Combining classifications into an overall sentiment classification can be an average or majority of the different classifications found. When also other inputs than classifications are combined a calculation of outcomes is no longer valid, but the different attributes can be inputs for the combiner.

The strategy to hybrid sentiment classification is formed with the following principles and is shown below in figure 11:

- The strategy is classifying sentiment of messages, therefore the input is a message and the output is a sentiment class
- It uses the result of different attributes (sub classifications and property values) to classify the sentiment of the message

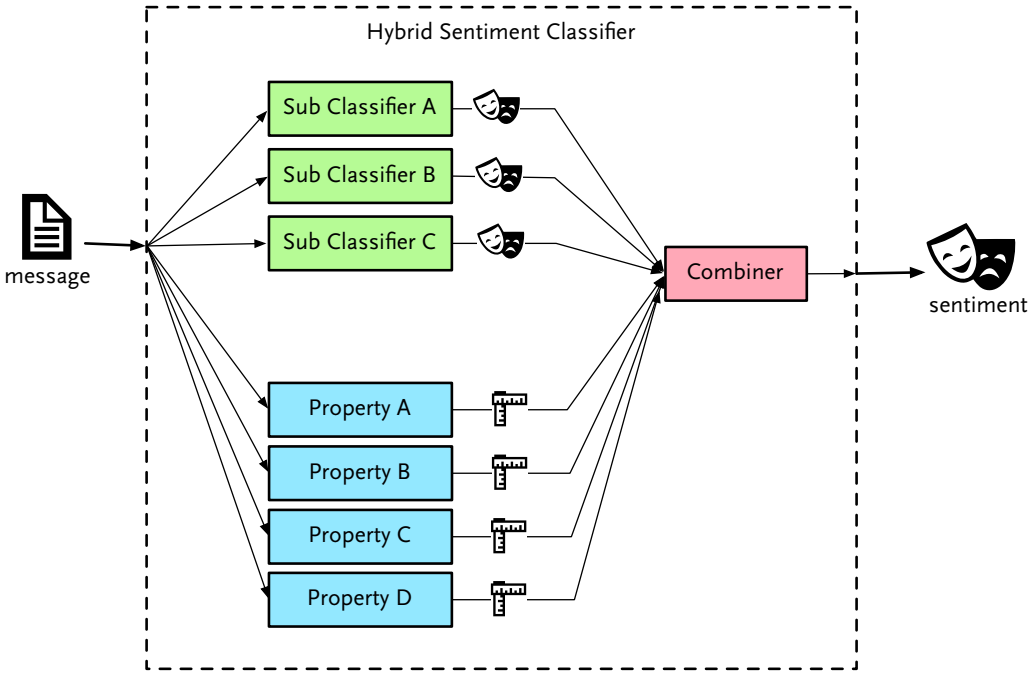

Figure 11: The strategy to hybrid sentiment classification

As shown in figure 11 above, the marked box contains the strategy for the hybrid sentiment classifier, with a message as input and sentiment as output. Within the hybrid sentiment classifier the strategy is to classify a message based on multiple sub classifiers and its extract ed properties values. Both these property and sentiment values are then used as input for the combiner, which is also a classifier, to predict the message's sentiment. In order to make accurate predictions, this combiner should be able to find relations of property values and sub classifications in a message. This way it will for example be able to select sub classification A when property value C is greater than 5. For more information about how the combiner is integrated in the overall design, see 4.2.

The challenge of filling in the above strategy is to find ways for different components to communicate in such a way that changes made from the workbench can affect the hybrid classifier on many levels. The design of an adaptable hybrid sentiment classification workbench is further explained in paragraph 4.2. First, the different components of the strategy are discussed.

Message (input)

The input for the hybrid classifier is a message. This message contains the message's text, but also its meta-data. Examples of meta-data for a message are the author, the source of the message, the time the message was created and its URI (a unique identifier of the message) [23]. Properties can be extracted from the actual message as well as from the meta-data. Examples of message properties are the time of day it was written, the author's gender or its location that can each lead towards accurate sentiment classification.

Sentiment (output)

The output of the classifier is one of the following sentiment classes, positive, negative or neutral. As discussed in chapter 3 a three class sentiment classification fits the subject of this thesis best because it improves classification of neutral messages as well as a classifiers' ability to distinct positive and negative messages [32].

What literature shows is that a combination of a subjectivity extractor in front of classification together with a two class classification is used.

In this thesis a subjectivity extractor in front of the rest of the process does not fit in the hybrid sentiment classification strategy, and therefore the three class classification is chosen.

Sub classifiers

The message is classified by multiple sentiment classifiers, called sub classifiers. Classifiers are used on several levels of the architecture and therefore they are named to be able to direct them unambiguously. Sub classifiers can be any sentiment classifier which input is a message. Its output must be sentiment using one of the following classes: positive, negative or neutral. For classifiers, see 3.1.3.

Properties

The hybrid strategy also uses property values of a message to combine the sentiments, which enables the combiner to make better decisions about which sub classifier to use. A property value is a value for a particular property. A property can have numeric or nominal types. Examples of properties, its type and a possible value are shown below in figure 12:

Property	Possible values	Example value
Number of words	any numeric value	12
Is a question	{yes, no}	yes
Written in part of day	{morning, afternoon, evening, night}	{night}

Figure 12: Example of properties and property values

Other message attributes

The combiner classifier does not use the actual message as input, but attributes extracted from it. In the strategy these are limited to sub classifications and property values, but other attributes could be added as well.

A subjectivity extractor could be used to predict wether or not sentiment is actually present, of which the overall outcome can also be used as a message attribute as input for the combiner. Sub classifiers extract sentiment parallel to subjectivity extraction, after which the combiner can choose not to use any of the sub classifications because the message property with the subjectivity extract has strong evidence the message has no sentiment at all.

This subjectivity extraction could be a new attribute, but there is no actual reason to not use this extraction in a property value.

Combiner classifier

The combiner is responsible for generating the output sentiment, based on the message's attributes (sub classifications and property values). This combiner can use different strategies, as long as its input is a collection of attributes and its output is sentiment (positive, negative, neutral).

A strategy could be to only use the result of the sub classifiers and use a majority vote to determine the overall sentiment [18]. This way the properties are ignored. A different combine strategy would be that the combiner is actually a classifier, where its inputs are a list of attributes, its output one of the three sentiment classes.

This also shows the big advantage of generating the collection of attributes, instead of letting the combiner determine which classifier to use. The combiner is now not required to have any knowledge about the internals of a sub classifier or a property.

4.2 System design of an adaptable sentiment analysis system with hybrid classification

Now that the conceptual strategy is available for hybrid sentiment classification, a software system can be designed that is able to perform this strategy. This system design will answer the second sub question:

How design a system architecture for the hybrid sentiment classification strategy that is adaptable by customer service experts and what are the underlying components of this design?

From this second sub question, the design should at least comply with the following three principles:

- The design must enable customer service experts to change the sentiment analysis.
- The design should give feedback about how these changes affect the performance of the sentiment analysis. This will be discussed further in 4.3.
- The design includes supporting components in order to be a complete system. This means it includes a way to get new messages, filter these messages, store messages and other data generated by the sentiment analysis system, is able to be operated from a web interface and can be trained and evaluated by train and test data.

4.2.1 System design: Workbench architecture

A system which is able to analyse sentiment, in which the process of sentiment analysis can be adapted and improved, is entitled by its own name. This is called a sentiment analysis workbench, or workbench for short. The system design for hybrid sentiment analysis for this thesis is therefore the workbench architecture and is shown below in figure 13. It is designed with the strategy as a basis and includes the necessary components and relations between them. All components are discussed in paragraphs below.

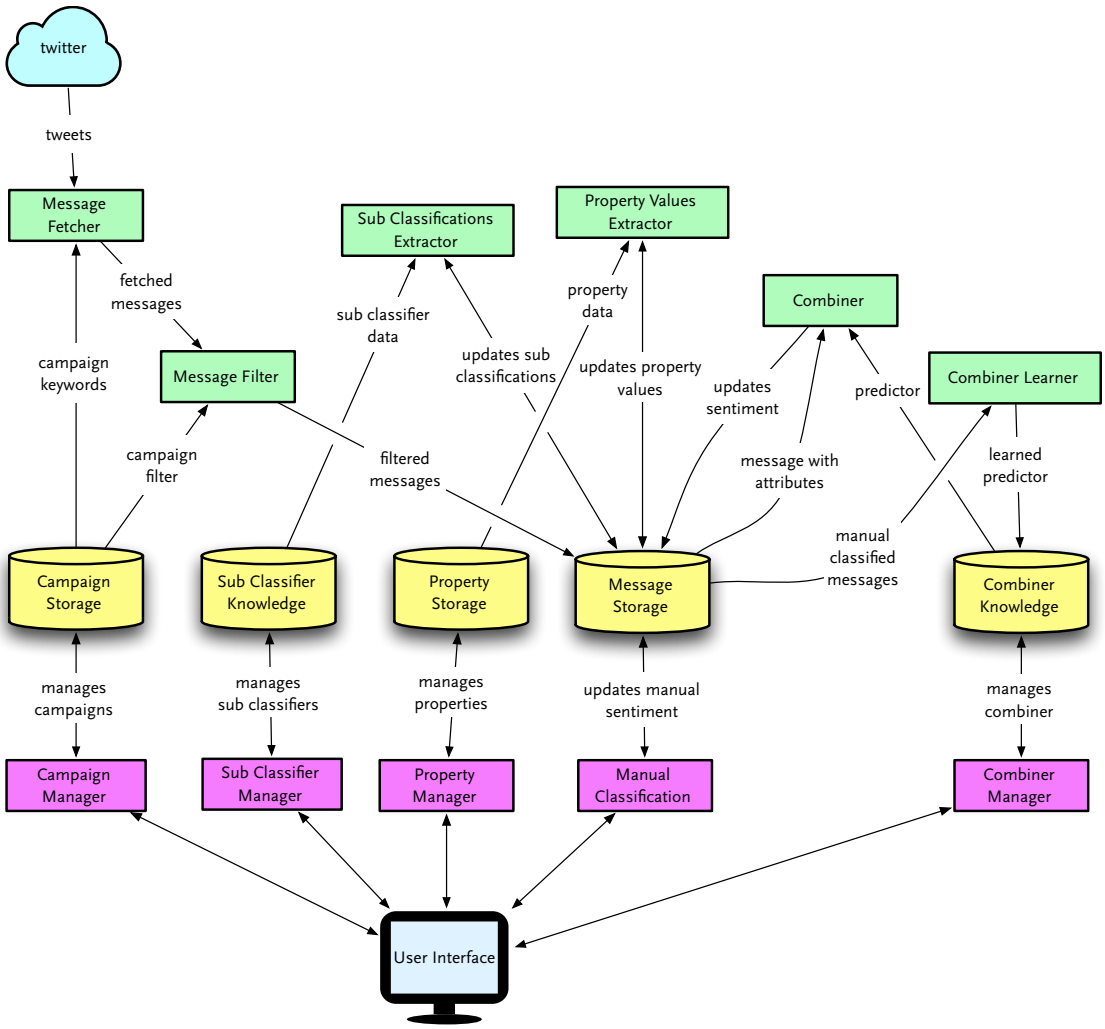

Figure 13: Architecture of hybrid sentiment analysis workbench

Architecture components layout

Starting from the bottom up, the user interface is the connecting component for managing the workbench. Different actions can be done in the workbench by the user; manage campaigns, sub classifiers, properties, the combiner and manual classifications. All these actions need a "manager", a component that supports performing these actions and storing the information for use by the workbench. The managers are drawn in purple boxes.

The storage these managers need are all stored in a database, in the architecture drawn as yellow barrels.

Another layer of components is added on top of the storage layer, handling functions that run in the background (in the perspective of the user), like fetching and filtering messages, performing sub classifications and extracting property values for stored messages, learning the combiner and performing the combining classification by the combiner. All these

components, drawn as green boxes in the architecture, use one or more types of information stored in the database and put new information back into the database.

The individual components of the workbench architecture are discussed below, starting with the general message fetcher and thereafter arranged around the five different actions the workbench supports: Campaigns, sub classifiers, properties, manual classification and combining classification.

Message fetcher

The message fetcher is importing messages with sentiment. Many sources for sentiment messages can be used together, but this would add another dimension to the workbench in which classifier, properties and knowledge would also need to know about the differences in source and take that into account. As Twitter is the leading source for WOM sentiment messages, this architecture is based on using Twitter messages only.

Campaigns

The workbench is not supposed to serve one subject, like “customer service of KPN”, but to be used in different campaigns. In each campaign other messages should be selected and all knowledge about classifying these messages should be focussed on the campaign the classification is about.

Adaptable message filter

Figure 14 below shows the adaptable message filter.

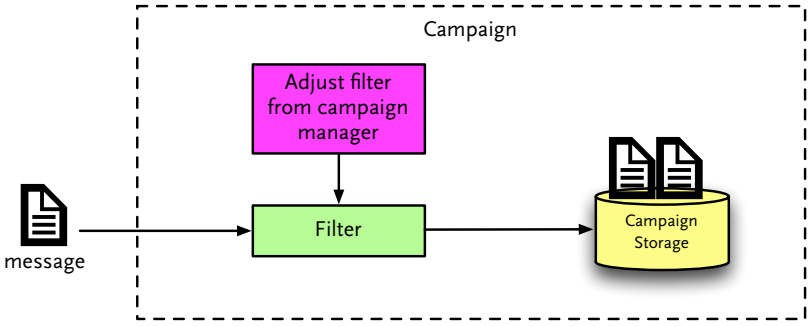

Figure 14: Adaptable message filter. From the campaign manager the message filter can be adjusted, resulting in a different set of campaign messages in the campaign storage.

Each campaign has its own message filter, based on a set of rules, to select messages that are of interest for the sentiment about a certain subject the campaign targets. This message filter rules should be adaptable by the customer service expert from the user interface, in order to improve the workbench for a campaign. For example, when a customer service

experts suspects there are certain type of messages that should not be taken into account for the campaign, he can exclude these messages by changing the message filter.

The rules in the message filter can then look like this:

```
CONTAINS “Ziggo” OR “Alles-in-een” and NOT “Ziggo Dome”
```

This way, messages about the subjects Ziggo are selected but Ziggo Dome is excluded, because this is a different subject and sentiment connected to Ziggo Dome as the subject are not wanted to be taken into account.

Managing campaigns
The campaign manager includes an adaptable message filter, and other settings for a campaign can be managed from the campaign manager in the user interface, for example the name of the campaign.

Campaign storage
Storage of campaigns is constrained to storing the settings of each campaign that give the message fetcher the information needed to fetch new messages for the campaigns. Messages fetched for each campaign are stored in the message storage.

Sub classifiers
Sub classifiers are each making a classification for messages of a campaign, of which the outcomes (sub classifications) are used as input for the combiner. Sub classifiers can have a (manageable) set of rules, but can also learn how to classify messages for a campaign according to training data. This sub classifier knowledge is stored for each classifier for each campaign.

In this design of a hybrid sentiment classification workbench it is not taken into account to be able to add new sub classifiers from within the workbench. Depending on the type of sub classifier, some need to be able to learn themselves with training data in order to build a scheme of the messages. For learning purposes as well, sub classifiers might need to use elements of the WEKA machine learning framework, requiring correct communication in order to gain the right results.

Al of these issues in order to be able to introduce new sub classifiers from the interface of the workbench can be tackled, but are for the purpose of this project of secondary interest, and therefore this functionality is not further elaborated in the design.

The workbench can exist with only one basic sub classifier up to an infinite number. What needs to be considered is wether adding a classifier is also valuable by adding new information to improve sentiment classification and actually improves the system. Although sentiment classification will not be harmed by numerous sub classifiers, the workbench will get harder to manage. Performing new sub classification runs will take much more time and finding key elements that need improvement will be harder to find due to dispersal.

Properties

Properties are characteristics of messages, that can be obtained from the actual message or its meta data. A property can have numeric or nominal types (see 4.1).

Properties are defined in the property manager and these definitions are stored in the property storage. From there, the property value extractor component uses these definitions to extract property values from messages, which are then stored together with the message in the message storage.

Property definitions can be managed and added from the user interface, whereafter the property value can be extracted all over for the messages of the campaign.

Combiner

The combiner is the classifier that combines all message attributes into a sentiment classification. This classifier can predict sentiment either based on the average of the sub classifications (simple voting), or based on a model of sample data that it has learned prior to classifying new messages.

Based on literature discussed in 3.2.3 there are several techniques, actually called learning schemes, according to which a hybrid classifier can learn the model of the sample data. A classifier like Naive Bayes can be used for this purpose as well as machine learning techniques like decision trees and decision rules. Multiple hybrid classifiers can be available in the workbench that can be chosen in the user interface. When a different hybrid classifier is selected, it will be trained by sample data first when needed, whereafter the combining hybrid classification will be performed (again).

Combiner Learning

The combiner learning is performed as follows. A training set of messages is manually classified, property values are extracted and sub classifications are performed. All these message attributes are stored in the message database in a training set for a campaign and can be used to train a combiner. Figure 15 below shows how a learning scheme uses the message attributes and manual classification of training data to make a model of the data.

Different hybrid classifiers make different models of the sample data, for example a decision tree or decision rules.

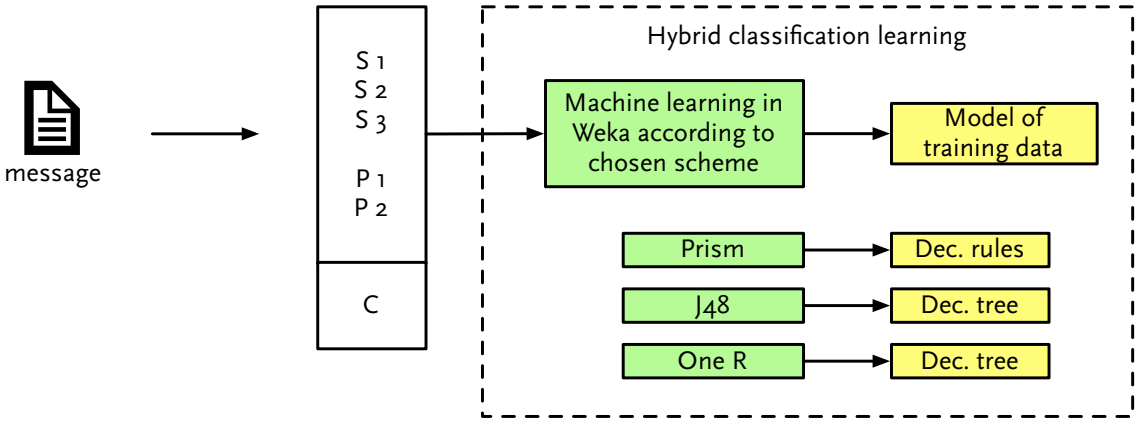

Figure 15: Combiner learning strategy, with sub classification S1..S3, properties P1..P2 and manual classifiction C as input. According to the chosen scheme Weka is used to perform machine learning in order to make a model of the training data. The output of Weka is a model, consisting of decision rules or a decision tree.

Hybrid sentiment classification

To classify sentiment in new messages a data model, for example a decision tree, can be used to predict sentiment. Other techniques for hybrid sentiment classification can also perform this classification, for which the input is only the set of message attributes and no data model. Also see figure 16 below.

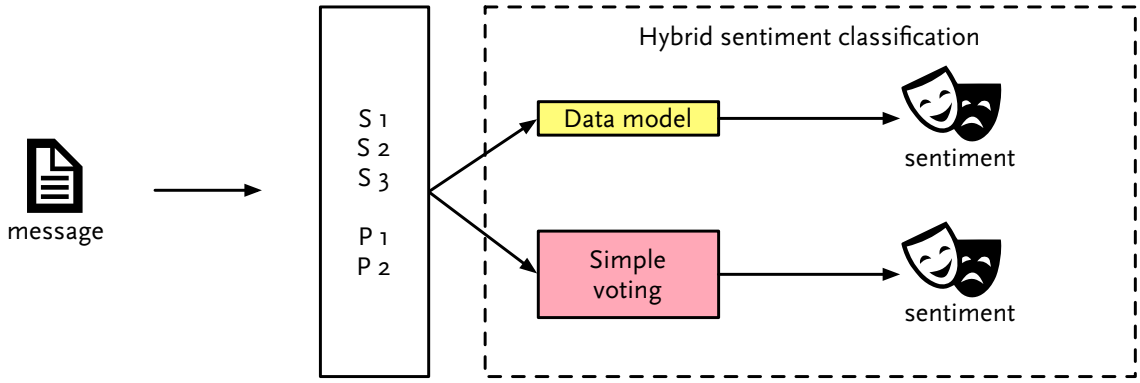

Figure 16: Hybrid sentiment classification can be performed by using a data model of learning data built with machine learning or a voting technique. The input is the sub classification S1..S3 and properties P1..P2 and the output is a sentiment class.

New hybrid classifiers

New algorithms for hybrid classification can be added and could use WEKA for machine learning when classification learning is needed. From the workbench user interface each

hybrid classifier can be chosen, but always one at a time in order to find out different performances of different approaches.

Data storage

A relational database seems to have great potential in this system design, but its implementation is less conventional. As the focus of this thesis is not on database systems, it was chosen to use a PostgreSql database.

4.3 Workbench performance

The second point of the second sub question "*And how does the chosen design improve the performance of sentiment analysis?*" will be discussed in this paragraph.

The performance of the workbench is the score of how good it is able to classify sentiment of messages. The ultimate performance of 100% would be reached when all messages of a test set are classified the same as their manual classification. The workbench performance is evaluated through the discrepancy between the sentiment it classifies and manual classifications for sets of test data. Also see figure 17 below, where the blue box represents the workbench performance by evaluating the discrepancy between manual and hybrid classified sentiment.

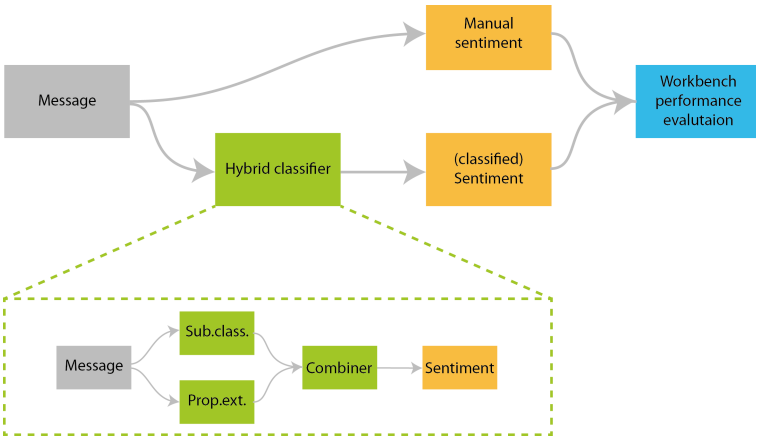

Figure 17: Workbench performance as discrepancy between manual and hybrid classified sentiment

4.3.1 Performance quantifications

As discussed in chapter 3.3.4 a common representation of classification performance for three-class classifications is the confusion matrix with the accuracy rate and F-measure. Evaluating and comparing different classifications this way does not reveal thoroughly what

actually happens in the system. Making small changes might have little effect on the accuracy level, but what really counts is how this accuracy is built up. The different performance measures discussed in chapter 3.3.3 (precision, recall, specificity) are able to show the distribution of incorrect classifications over different types, that can be totally different even when accuracy and f-measure are the same. Precision, recall and specificity are the measures that show how good the distribution of true positives from all messages that are predicted as positive (precision), the rate of true positive classified messages from all actual positive messages (recall) and the rate of negative predictions from all negative messages (specificity). Further details about these measures can be found in 3.3.3 and 3.3.4.

4.3.2 Cost sensitive learning

In chapter 3.3.5 cost-sensitive-learning was discussed, as a solution to anticipate on different error costs. In the case of sentiment analysis, there are no errors that are extremely more costly other than influencing the overall sentiment more incorrectly. Although there is a good reason to introduce cost-sensitive-learning to the workbench, it is hard to determine the right cost values, that might also differ from campaign to campaign. Within the boundaries of this thesis it was chosen to not study this subject further.

4.3.3 Insight in performance

Comparing different classifications: Benchmarking

Being able to compare performance of different classifications enables users to address new changes to specific type of errors and also regard the effect of such an improvement on all other types of errors. The workbench is designed to support incremental improvements on sentiment classification for organisations and comparing different classifications is an essential part of this process. Therefore comparing classifications is also included in the design. For each point in time, the performance of a classification can be stored and later two classifications can be compared to gain insight in how changes effected performance. Each of these stored classifications is called a benchmark, representing the state of the workbench, sub classifiers, properties and property values, learning messages (manual classified messages) and the hybrid sentiment classification outcomes quantified in the extended performance measures discussed in 4.3.1. Besides, also meta-data of the benchmark will be available: name, settings, data, user, etc.

Insight in errors

Using the performance measures and the confusion matrix a user can get insight in the distribution of errors: false positives, false negatives and false neutrals. To improve performance of the workbench on one of these type of errors it might be useful to see what

actual messages were classified in these error groups. For this reason the workbench also provides a way to browse through the classified messages while seeing the error made on the message. To support users to find ways to improve the workbench a search option in this message list is available to find a set of messages with specific characteristics.

This way, it can be found out messages containing “Ziggo Dome” should not contain any sentiment about Ziggo and how well this is incorporated in the workbench.

Visualising classifier knowledge (model) in workbench

For deeper insight in how a change in the workbench has effect on data models of the hybrid classifier, decision trees and decision rules, these can also be evaluated through visualised model representations. Workbench users can use these visualised decision trees or decision rules to see how each property or sub classification is influencing choices in the model, that is used for classifications. It can for example be seen what properties and/or sub classifications are of great importance and which are not. Also, comparing models of different benchmarks gives insight in the change of model according to the changes made to the workbench.

In short, insight in the model of the hybrid classifier gives workbench users hands-on tools for benchmark comparison and improvements.

4.3.4 Sample set: Training & test data

The sample set contains manually classified messages for test and training purposes. Messages that are used for training will not be used to test the outcomes of the classification.

Training data

Training data is used by classifiers to learn how to classify new messages in a campaign. Each time the user changes settings to a classifier, properties or campaign, classifiers should be re-trained. Using the updated models of the data, new benchmarks of the workbench can be made in order to evaluate the changes made.

Test data

Test data is a set of messages that are classified manually, as interpretation of a person, and are regarded as the actual sentiment of a message. These are compared to outcomes of the workbench to calculate the performance of the classification.

Sample size

The number of sample messages is mainly influencing accuracy in classifying new messages. For the best trained classifiers not only the distribution of classes should be balanced, but also the types of messages about an organisation should be balanced. Not only messages about a specific event on one date, but a representation of messages it can expect through time.

To start with, the sample set should contain enough messages to even be able to learn how to classify the expected messages. But a larger sample size is not always better, regarding overfitting the classifiers for a moment in time what weakens its ability to classify many different types of messages accurately. There are no clear guidelines about how to define a sample size, but it should at least contain more than a thousand messages.

Balancing test and training data sets - stratification

Paragraph 3.3.5 discusses stratification and cross validation are credibility checks on test and training sets. Stratification is the act of balancing the distribution of classes over the two sets of sample data (training and testing), in order to make both sets representative.

Application of stratification

From the available sample data the workbench divides random messages in the test or training set, proportional to their sizes. If for example a classification uses one third of the sample data for learning and two thirds for testing, from all negative messages also a third is randomly put in the learning set and two thirds in the test set.

Stratified-cross-validation

These techniques do not make a strong division between test and training data. The sample data is divided in a number of folds (mostly 10 [8]), whereafter each fold is used for learning and the remainder for testing subsequently until all folds have been the learning set. Using this technique, the performance of the hybrid sentiment classifier of the workbench could be evaluated with much more precision and is therefore interesting.

Some characteristics of the workbench do not cooperate with using stratified-cross-validation. Re-training classifiers will be performed repetitively and it is also important to be able to give clear insight in the built-up knowledge of classifiers. When stratified-cross-validation would be implemented, there would be 10 versions of the model of which the outcomes are combined. For the above two reasons it is not valuable to implement this technique in the workbench, but in a future commercial sentiment classification service it is advised to do so.

4.3.5 Workbench providing performance improvement suggestions

Although customer service experts using the workbench already have a great tool in their hands, it still is a challenge to find the right subjects to address improvements on. The workbench can use its knowledge to provide the main subjects for improved performance.

For example, the workbench can use messages of a specific type of error and find out what their correlation is. Maybe a specific word in the messages, usage of emoticons in combination with another property or that is was written in English. In short, all message information can be used to find correlation to a certain type of error and given back to the user to try to improve it in a new benchmark.

The design of such suggesting functionality adds another layer to the architecture and is outside of scope for this thesis.

4.4 Additional components

There are some additional components that could be added to the design that have been proven to be useful in other research. The subjects that have most appeal to this thesis are subjectivity extraction, part-of-speech tagging and topic relevance. The added value of these components to the workbench is questionable and therefore they are not included in the workbench architecture. Below each of these components are discussed in relation to this thesis and how they could be included in the architecture.

Subjectivity extractor

In paragraph 3.1.4 subjectivity extraction was discussed. Currently it is a popular component to use in front of two-class-classification, as Pang described [20]. In the architecture of the workbench the hybrid classifier is the central combiner of knowledge about messages and therefore it was chosen not the pre-select messages with subjectivity. Proven its contribution to improved classification performance it is interesting to use this as an extra way to enrich messages' attributes in order to give the hybrid classifier more information to build a data model.

Building the subjectivity extractor as a separate component (like the property value extractor), other components could also make use of its outcomes.

Being an extra component that connects with other components throughout the workbench adds another dimension to the design, which is out of the scope for this thesis, and is therefore currently not incorporated in the workbench architecture.

Part-Of-Speech tagger

In chapter 3.1.2 the technique of tagging each word to a part-of-speech (POS), noun, verb, adberb, etc, was discussed in order to use patterns in POS help to extract sentiment from messages. As Gimpel [11] states, the use of these POS pattern in micro-blogging messages is questionable, because these short abbreviated messages follow far from strictly the language rules for spelling, punctuation and capitalisation.

Although these common language rules do not add much value in understanding the sentiment of a micro blogging message, it might very well be that there are actually rules that can be composed from the POS tagging information.

In this architecture design for the workbench a POS tagger is not taken into account because it is out of the scope of this thesis. Actually, POS tagging is alike classification and the outcomes should be able to be used by classifiers that classify sentiment on topics. For adding this to a future architecture, a POS tagger could be an extra extractor, like the property value extractor and sub classifications extractor. The outcomes of the POS tagging can be stored together as meta-data of the messages in the message storage and can than in turn be used by a sub classifier to better understand the message.

Topic relevance

As 4.4.2 describes, POS tagging can also be used by topic-sensitive classifiers in order to understand sentiment in messages. Other algorithms also target the extraction of topics of messages, and are used mostly in front of classification for selecting relevant messages.

Of course, the message extractor should desirably only collect relevant messages to the topic, which are in turn extracted on sentiment. For now, the focus is not on this element of the workbench and therefore it is not elaborated in the architecture. It would be useful to improve the message extractor with topic relevance selection.

5. Proof-of-concept implementation of sentiment analysis workbench

The designed architecture of chapter 4 is an overall design of elements that together form the workbench which customer service experts can use to incrementally improve sentiment analysis for micro-blogging messages, and meanwhile give them insight in the process of this analysis. This chapter describes a proof-of-concept implementation of the workbench design of chapter 4.

5.1 Outline of proof-of-concept implementation of workbench

To make a proof-of-concept of this architecture, an implementation is made in which the most important functions can be performed and the usage can be shown. This chapter proceeds with the starting points for implementation in 5.2, whereafter the outline for the software development is discussed in 5.3.

The implementation of the functions of the workbench starts with managing campaigns and how messages are fetched for each campaign in 5.4. How message attributes (message properties and sub classifications) are extracted for each message is discussed in 5.5 and how these are thereafter combined into an overall sentiment classification by a hybrid sentiment classifier is the subject of paragraph 5.6. Some hybrid sentiment classifiers are trained using machine learning in WEKA. Paragraph 5.7 shows the implementation of the use of WEKA for training purposes and how WEKA communicates with the app.

Another important function of the workbench is to make and compare benchmarks of the workbench at a specific time and corresponding workbench settings, of which the implementation can be found in paragraph 5.8.

The implementation of the above functions is accompanied by a user interface from which all these functions can be performed by customer service experts.

5.2 Starting points for implementation

Below the source and subject for the campaign in the proof-of-concept implementation is discussed. The source of messages is of importance because the workbench should import the messages from Twitter somehow, and the subject is described to show it is a typical subject for future use of a sentiment analysis system for organisations.

Twitter
In chapters 3 and 4 the importance of Twitter in the word-of-mouth micro-blogging messages has been emphasised already, and from there it follows that this first implementation uses Twitter messages as the source for WOM messages about an organisation.

Topic: Ziggo
The organisation to use in the campaign for this proof-of-concept is of secondary interest, because the main goal is to show the contributions of the workbench architecture to sentiment analysis.

GreenOnline suggested that creating a campaign for a future customer of their sentiment analysis tool could once be useful and therefore they suggested to use Ziggo as the subjected organisation in the proof-of-concept. This means that real data will be used and therefore will better reflect the reality. In chapter 6 the findings of this test campaign are discussed.

5.3 Software development outline

5.3.1 Web app in Ruby on Rails

The main goal of this implementation is to show functionality and there are no bounds towards using a specific programming language. The workbench will be a web based application written in the object-oriented scripting language Ruby, well-suited for web development. It is a powerful language that is readable and concise, making it not only easy to understand the code, but by using less code it quickens development as well.

Ruby can be used with different web frameworks to speed up development even more. Examples of Ruby web frameworks are Padrino, Sinatra, and Rails.

Especially the open source web framework Rails is well suited for this type of web applications because it embraces the MVC model, has a built in ORM, routing and session control. Following the MVC model, code is divided up in business logic, the (UI) view-related code and the controllers between these. Its Object Relational Mapping (ORM)

accommodates mapping of data from a database to objects in code and memory. Rails' routing provides handling of web requests by the associated controller, which is based on the URL the request comes from. Also session control is a feature of the Rails framework, which provides all handling around the storage of session data in for example cookies. For all the above reasons it was chosen to use Ruby together with the Rails web framework.

5.3.1 Database Storage

Common relational databases are not always the best option to store data. In this project, data storage is mostly about messages and message attributes like property values, sub classifications and manual classifications. Performing re-classifications and learning sub/ hybrid classifiers stresses the data storage system, resulting in a slow workbench when stressed too much.

Another issue the data storage system has to be prepared for is the scalability. Typical of the workbench is the infinite number of message properties that can be added by the workbench user, new sub classifiers and other message extractors can be added in order to gain even more information about the message.

Therefore alternative storage systems were explored. A great alternative can be the document-oriented file system, of which Mongo-DB is an example, storing data in records. One record can contain the message itself, the sub classification outcomes, the manual classification and all property values. A query to a message and all its related data is then quick and easy by addressing the message. Other queries that are performed across messages will contrarily be more complex.

Another alternative that is used in the field of classifications is to use the RDF (Resource Description Framework) document format, that was originally found in order to describe data for the semantic web. It stores all data as triples containing *subject*, *predicate* and *object*. An example of storing the positive manual sentiment of message xx: (message xx, manual sentiment, positive). New entities that should be stored can easily be added and therefore using RDF is easily scalable.

Although both alternatives have potential to enhance the workbench, it is not further looked into in this proof-of-concept. At first, the overall design of an adaptable hybrid classification system needs to be made and this project ends with a proof-of-concept of that design. Performance improvements on speed of classifications or scalability are therefore no issues during the scope of this project and are further undiscussed. Therefore a relational database, PostgreSQL is chosen as the data storage engine.

5.3.2 Basic Architecture

The application is split into three layers following the Models, Views and Controller (MVC) design pattern [34]. This makes the application easier to understand and develop. It also makes business logic unaware and independent of the user-interface/view implementation. Adding a different view, like a mobile device, or an API does not need any changes to the business logic.

Models
The models in this application will represent the business objects like campaigns, messages and classifiers. There are also models to communicate with WEKA or Twitter.

Views
The role of the views are to render html given some related data. This can for example be the list of campaigns.

Controllers
The role of the controller is to handle the web requests of the user, get the relevant data of the models and makes the views render themselves with that data. The result is returned to the user.

5.3.3 WEKA

Sentiment classification techniques partially use machine learning techniques, for which WEKA (Waikato Environment for Knowledge Analysis [8]) is the most common suite of software that supports many different techniques for data analysis, data modelling, data visualisation and also supplies its own GUI to access the functionalities. Some of the functions WEKA offers can also be implemented from scratch, but using the WEKA suite speeds up development and therefore it was chosen to use WEKA. It is used for knowledge analysis, such as (machine) learning of combiners into models of the data, for example J48 decision trees or PRISM decision rules.

In our architecture WEKA is not used for all purposes it supports. For example learning of sub classifiers is done within the Ruby on Rails app. For further explanation please see paragraph 5.7.

Data communications between WEKA and workbench
To make use of the functionalities of WEKA communication between the Ruby on Rails app and WEKA was setup. One element of the business logic is a component which role is to

communicate with WEKA. Besides a GUI in which WEKA can be used, there is also a command line interface (CLI). This makes it very easy to use WEKA with all kind of programming languages, including Ruby. When the application needs to use some functionality of WEKA, it uses its CLI with the right parameters. Because most of WEKA's functionality needs data (e.g. classification learning), WEKA requires its input to be in the ARFF format. The application is able to export (part) of its data to the ARFF format, which is used as input for WEKA. Also see figure 18 below.

Figure 18: Communication between Ruby on Rails application and WEKA is performed via command line interface, data input for WEKA is formatted in .arff file format.

ARFF files

ARFF stands for Attribute-Relation File Format and is developed for WEKA. An ARFF file contains two parts, a header and data. The header section defines an ordered sequence of all the attributes used in the data. An attribute has a name and a datatype. The data section represents the data, where each instance is defined on a single line and each attribute-value is delimited by a comma. Below is a sample ARFF file shown. It contains 12 attributes and 7 data instances.

Sample .arff file

```
@RELATION "TRAIN SUB"

@ATTRIBUTE manual_sentiment {positive,negative,neutral,unknown,undefinable}

@ATTRIBUTE "P:Word Count" numeric

@ATTRIBUTE "P:Day Section" {morning,afternoon,evening,night}

@ATTRIBUTE "P:Retweet?" {yes,no}

@ATTRIBUTE "P:Contains !" {yes,no}

@ATTRIBUTE "P:Contains ?" {yes,no}

@ATTRIBUTE "P:contains Hashtag" {yes,no}

@ATTRIBUTE "P:Contains url" {yes,no}

@ATTRIBUTE "P:#FAIL" {yes,no}

@ATTRIBUTE "C:Naive Bayes" {positive,negative,neutral,unknown,undefinable}

@ATTRIBUTE "C:Word Count" {positive,negative,neutral,unknown,undefinable}

@ATTRIBUTE "C:Emoticons" {positive,negative,neutral,unknown,undefinable}

@DATA

neutral, 12, afternoon, no, no, no, no, yes, no, neutral, neutral, neutral
```

```
negative, 28, afternoon, no, no, no, no, no, no, negative, neutral, neutral
negative, 24, afternoon, no, no, no, yes, no, no, negative, neutral, neutral
negative, 18, afternoon, no, no, no, yes, no, yes, negative, negative, neutral
negative, 9, afternoon, no, no, no, yes, no, yes, negative, negative, neutral
neutral, 24, afternoon, yes, yes, yes, yes, no, no, neutral, neutral, neutral
positive, 16, morning, no, no, no, no, no, no, positive, positive, neutral
```

5.3.4 Asynchronous execution

In this application, some functionality takes some time to complete. Fetching new messages from Twitter or classifying sentiment for a large collection of messages could take longer than just a few seconds. While executing these resource intensive tasks, the web application would 'block' other incoming requests. This means that the web server needs to finish these tasks before serving other requests from the user.

In this application, these kind of tasks are executed in the background using Resque, a ruby gem (plugin) which uses redis, an in-memory store. For each of these tasks a Job is created (e.g. a MessageClassificationJob). These jobs are executed on another process, while the webserver, thereby also the user, can continue its work.

5.4 Campaigns and their messages

As discussed in 4.2.1 the workbench can analyse sentiment for different campaigns. This paragraph will discuss how campaigns are implemented and how the specific messages for those campaigns are fetched.

5.4.1 Campaigns

For each organisation a campaign can be created to show market sentiment for that corporation. An example of such a campaign could be 'Ziggo'. This campaign is interested in the sentiment about Ziggo. Figure 19 below shows the messages view of the Ziggo campaign in the workbench.

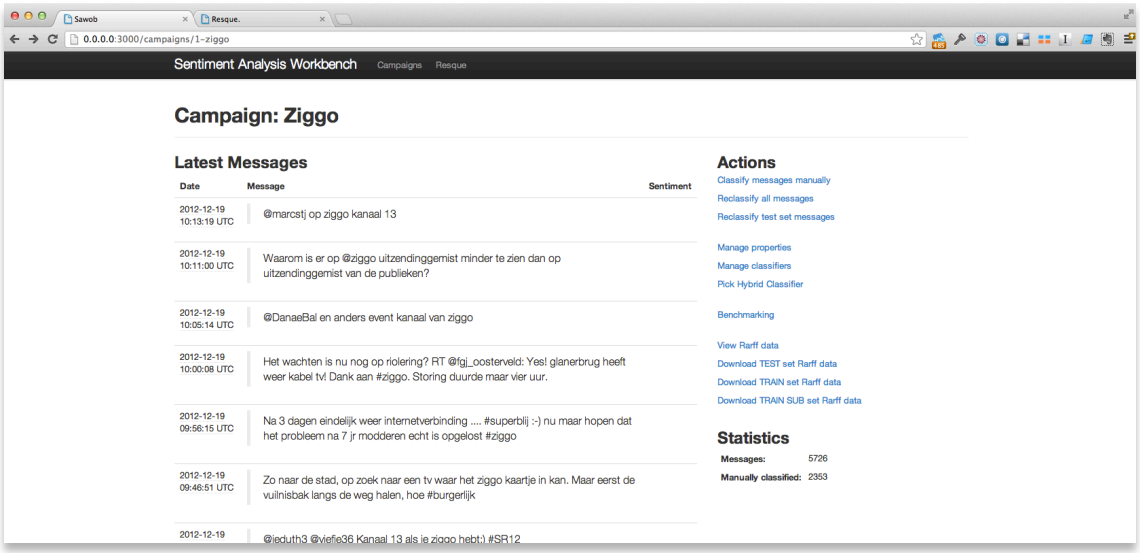

Figure 19: Campaign overview and actions in the workbench

5.4.2 Messages

To make it able for the workbench to analyse sentiment for a specific campaign, messages for this campaign need to be fetched. Each campaign has its own message filter, that can be managed in the workbench by adding and ignoring keywords. Also see figure 20 below.

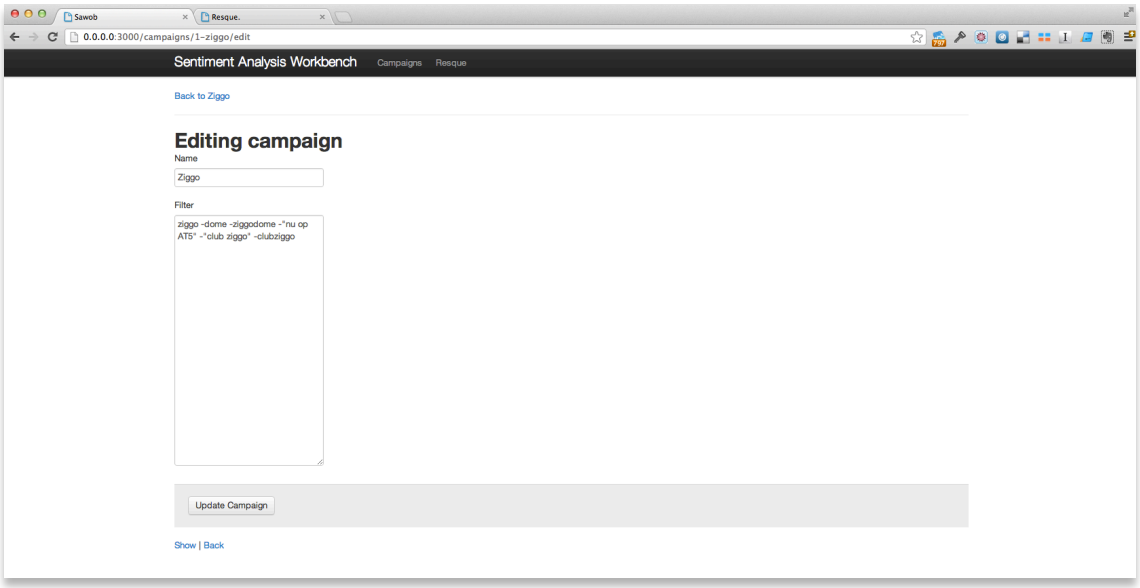

Figure 20: Edit message filter for a campaign

The better the messages are relevant to the campaign, the better the workbench can extract sentiment from these messages. Changes to the message filter for the campaign Ziggo in the workbench will change the performance. As shown in figure 20, the keyword clubziggo is

ignored, in order to target the campaign on the telecommunications company Ziggo and not clubziggo.

The syntax of the filter is the same as the syntax for twitter search. Exclusion of keywords is done by prepending the term with a '-'.

Naturally, the message itself is very important for sentiment classification, but other meta data could be relevant as well. Therefore, the author of the message, the time on which it is posted and the location are also stored next to the message itself.

A shortlist of messages used in the workbench as test and training data, with manual classified sentiment, can be found in Appendix A.

5.4.3 Fetching Twitter messages

In the proof-of-concept, Twitter messages are fetched using the Twitter REST API. Using this Api, messages which contain certain keywords are fetched as json data. This data is parsed and translated to 'Message' objects in the application, which are then stored in the database. Fetching new messages is done periodically, or can be performed manually. An alternative to the Twitter REST API is the streaming API. This allows for a continuous stream of new messages, without checking manually. The REST API is better suited for this proof-of-concept, because a continuous stream of new messages is unwanted and this application is not about real-time sentiment analysis. Making small increments to the hybrid classification strategy and afterwards evaluating the performance of the new strategy requires the same messages as input for the classification to make a good comparison.

After the messages are fetched from Twitter and are stored in the database, the messages are immediately classified using the current hybrid sentiment classification strategy. This process is also performed in the background using Resque (see 5.3.4 for more information about background jobs).

5.4.3 Showing messages to the user

To give insight in all the messages for a campaign, the message overview page is implemented as show in figure 21 below. This view shows the messages in chronological order. The message filter at the top and other parts will be discussed later.

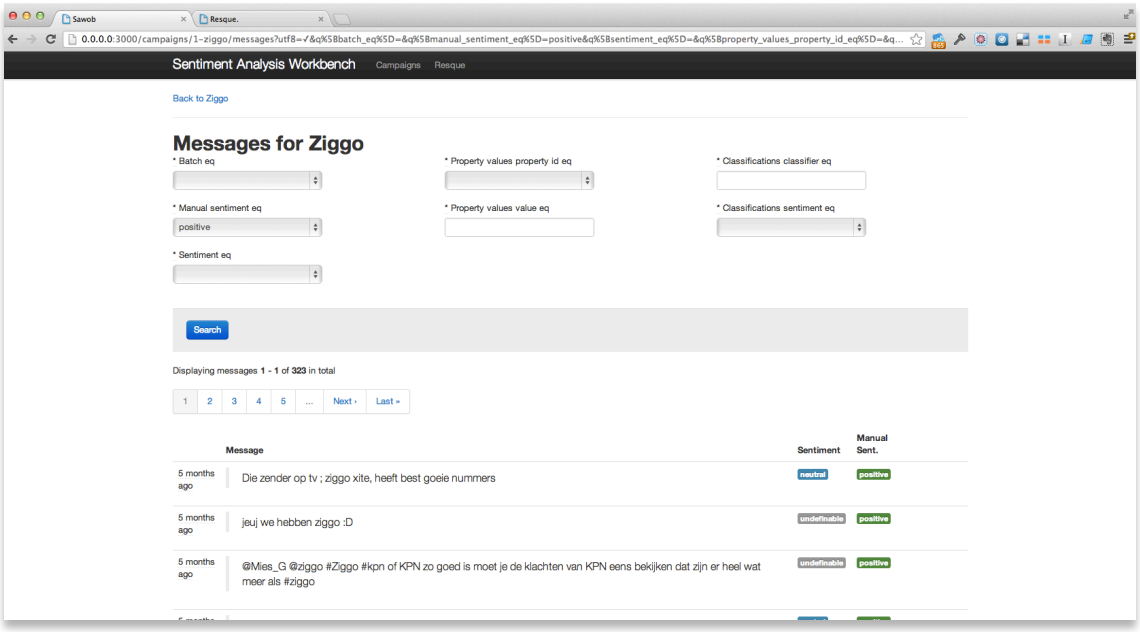

Figure 21: Campaign messages overview

Clicking on a specific message will show its details as shown in figure 22 below. This view shows all the attributes of the message and its sentiment. The extraction of these attributes will be discussed in 5.5.

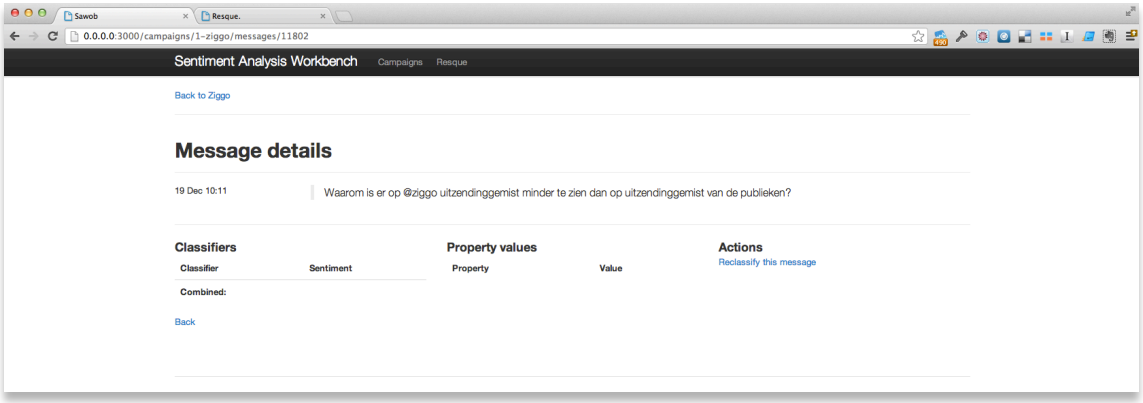

Figure 22: Message details

5.5 Extracting attributes from messages

In the architecture of chapter 4, two different attribute types were taken into account in the basic architecture: sub classifications and properties.

The extraction of sub classifications and properties will be discussed in 5.5.1 and 5.5.3 respectively.

5.5.1 Extracting sub classifications from messages

The hybrid classifier uses different sub classifiers. The implementation of these classifiers can be extended to an infinite depth, meaning optimising the sub classifiers itself. In this thesis the focus is not on making the best quality implementation of each sub classifier, but to the hybrid combination of different message attributes. Therefore, basic implementations of three sub classifiers were made, to be able to use three different approaches in the hybrid classifier. In the future, new sub classifiers can also be added by implementation as well as by use of external (black box) libraries from which only the outcomes are used in the workbench.

In this proof-of-concept the following sentiment classifiers as sub classifiers are chosen:

- Naive Bayes sentiment classifier
- Emoticon sentiment classifier
- Word count sentiment classifier

As discussed in 3.1 these three sentiment classifiers all work differently: Naive Bayes uses probabilities based on messages for which the sentiment was known. Emoticon classifier uses emotions expressed as icons to base the sentiment analysis on and the word count classifier uses a predefined lexicon of words which indicate sentiment.

Each sub classifier in the workbench has its own component and is a subclass of SubClassifier. Each sub classifier responds to the 'classify' method, which takes a message as its input and returns the sentiment as output. This makes it very easy to introduce new sub classifiers. They just need to conform to the SubClassifier interface.

5.5.2 Showing the sub classifiers to the user

All the sub classifiers for a campaign can be viewed as shown in figure 23 below. It is also possible to disable (or enable) specific sub classifiers for the campaign. This makes it possible to evaluate the effect of a sub classifier on the overall sentiment analysis of the workbench.

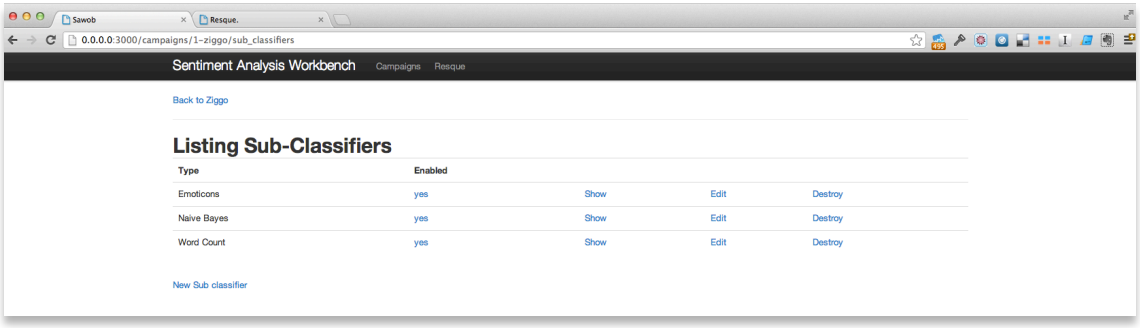

Figure 23: Editing a property in sawob

5.5.3 Extracting properties from messages

Where the sub classifiers need their own component in the workbench, extracting properties in the workbench works differently. New message properties can be added by specifying the property extraction script for that property. This script should be written in javascript. This makes it easy for customer service experts to create their own properties. Details of editing such a property is shown in figure 24 below.

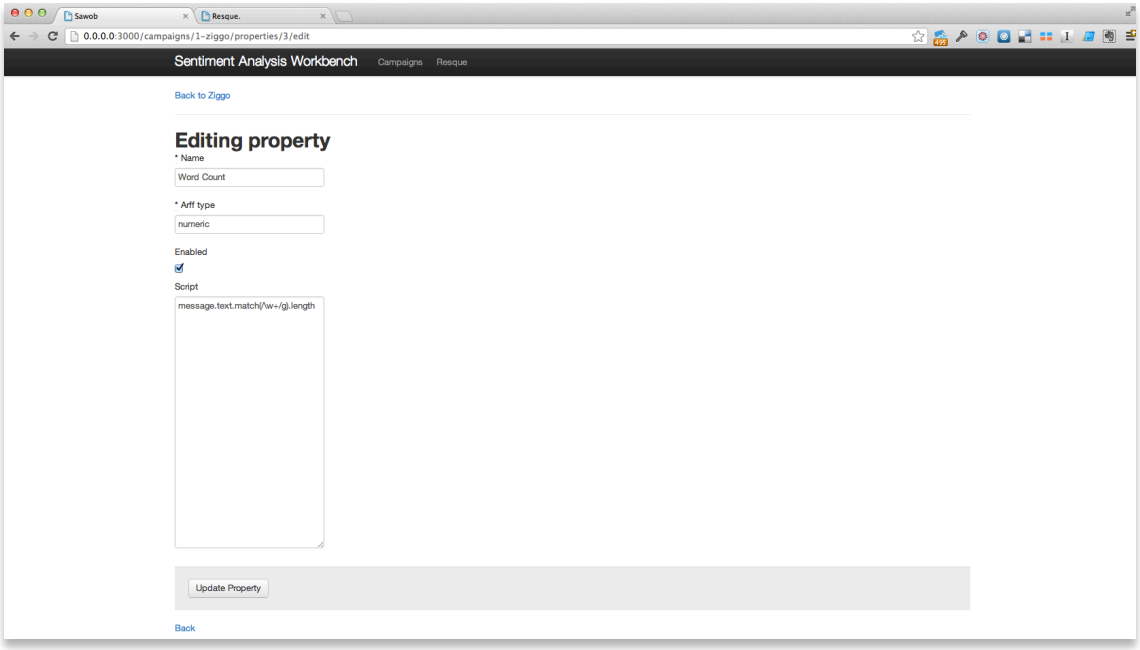

Figure 24: Editing a property

In the proof-of-concept the following properties are extracted. Their script is also provided:

Property	Description	Script
Word Count	The message's number of words	`message.text.match(/\w+/g).length`

Day Section	Section of the day the message was created	`var result='evening', hour = new Date(message.created_at.to_i*1000).getHours();`
		
		`if(hour < 6) {`
		`  result = 'night';`
		`} else if(hour < 12) {`
		`  result = 'morning';`
		`} else if(hour < 18) {`
		`  result = 'afternoon';`
		`}`
		
		`result;`

Retweet?	Is the message a retweet?	`bts(/^\s*RT/.test(message.text))`
Contains !	Does the message contain a '!'	`bts(/!/.test(message.text))`
Contains ?	Does the message contain a '?'	`bts(/\?/.test(message.text))`
Contains Hashtag	Does the message contain a hashtag	`bts(/#\w+/.test(message.text))`
Contains url	Does the message contain a url	`bts(/https?\:\/\//.test(message.text))`
#FAIL	Does the message contain #FAIL	`bts(/#fail/i.test(message.text))`
Bevat Kut	Does the message contain a 'kut' or 'Kut' (a Dutch invective)	`bts(/kut/i.test(message.text))`

Property	Description	Script
Laughing	Does the message contain laughing expressed by 'haha'	`bts(/haha/i.test(message.text))`

5.5.4 Showing properties to the user

All the properties of a campaign can be viewed as shown in figure 25 below.

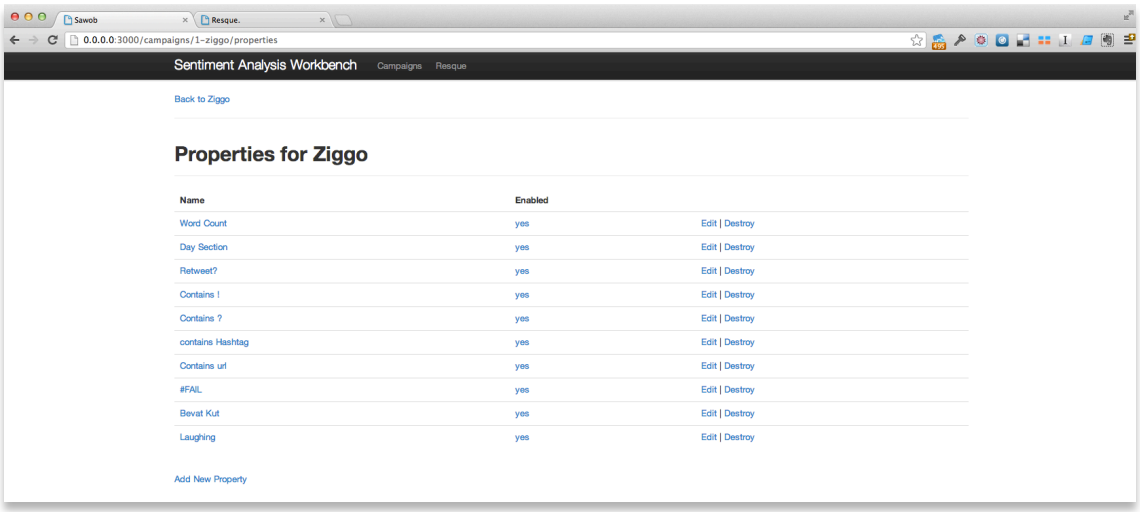

Figure 25: Properties in Ziggo campaign

To see how properties influence hybrid classification of sentiment, they can be disabled from the property overview. When a property is no longer needed it can also be deleted.

5.6 Combiners for Hybrid classification

After the attributes of a message are extracted, these can be used to determine sentiment. As discussed in 4.2 this component is called the combiner. To let the user experiment with different types of combiners, as discussed in 3.2, the user can select a combiner from the available combiners that are implemented in the proof-of-concept. The figure 26 below shows this selection view.

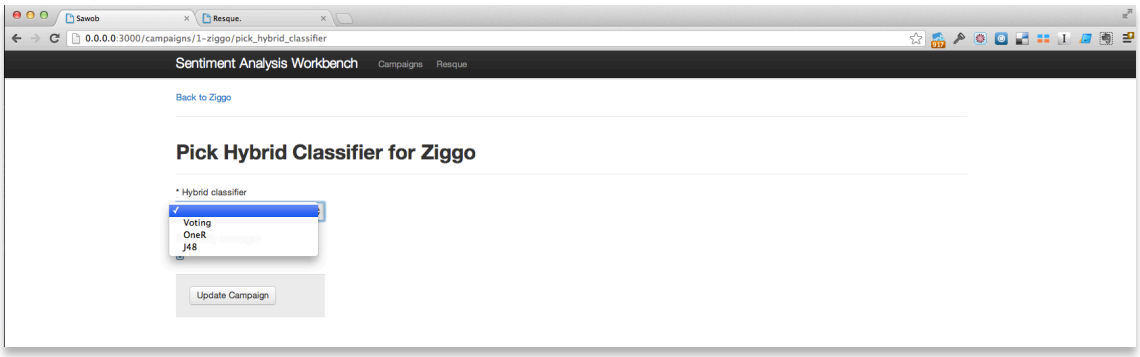

Figure 26: Picking a hybrid classifier in the workbench

Each combiner is a subclass of 'Combiner' and needs to implement the combine method. This method takes the attributes of a message as input and outputs a sentiment class.

To show the basic functionality of a combiner, first the simple voting technique discussed in 3.2.1 was implemented. This implementation will be discussed in 5.6.1. Two other combiners, that will use the properties and sub classifications together will be discussed in 5.7.

5.6.1 Combiner: simple voting

The simple voting combiner only takes the sub classifications into account, and discards the property-data. This technique bases its sentiment classification on the majority of sub classifications. The process is as follows. First, al sub classifications are converted to -1 if it's negative, 0 when it's neutral and +1 when it's positive. Then these numbers are added and the total is divided by the number of classifications. This is shown by the following formula where A is the average sentiment and C_i is the classification for each classifier n .

$$A = \frac{\sum_{i=1}^{n} C_i}{n}$$

The combined sentiment S is defined according to these rules:

S = **Positive** when $A \geq 0{,}5$

S = **Neutral** when $-0{,}5 > A < 0{,}5$

S = **Negative** when $A \leq -0{,}5$

5.7 Combiners using classification learning

Where the simple voting combiner discussed in 5.6.1 determines the sentiment on a calculation of the sub classifications, the techniques discussed below are more advanced. They take all the attributes into account, so properties as well as sub classifications. These techniques will base their sentiment classification on a classification scheme, that is learned from existing data as discussed in 3.2.3.

The creation of training and test data will be discussed in 5.7.1 including manual classification. Creating a classification scheme from that training data will be discussed in 5.7.2. To make the result of this learning process visible to the user, a special view is implemented and discussed in 5.7.3. The classification process in which the combiner uses the classification schema to determine the sentiment will be discussed in 5.7.4. In 5.7.5 an alternative classifier is discussed.

5.7.1 Training & test data for classification learning

Classification learning creates a classification scheme based on historical data for which the classification is known. To create a training set, the application has a dedicated section to manually classify sentiment for messages. This process can be performed from within the workbench, where each message can be classified positive, negative, neutral and undefinable as well. This last option is added to not force someone to choose one of the three classes of sentiment when he or she is uncertain. What actually happens is that this message is seen as unclassified still, but makes sure the user is not asked for a manual classification for this message again.

Some messages are indirect expressions of sentiment about the campaign. These messages can for example be reports of trouble using Ziggo services, or people transferring to a competitive service. These messages are important for the overall sentiment for Ziggo and should for certain be taken into account for determining sentiment.

Other messages are plain reports of an event on which Ziggo's sentiment will not be accounted. These messages should not be taken as sentimental, because they are neutral to the sentiment. An example is the hourly message of what is on TV Channel AT5, including the channel for Ziggo cable users. In the test campaign for Ziggo these messages were finally even filtered out by the campaign message filter.

Figure 27 below show the screen in which users can manually classify messages for a campaign. The user will see one message at a time and four options, represented by four

buttons. After the user classifies the sentiment of the message, the next message will be shown, creating a quick process of manual classification.

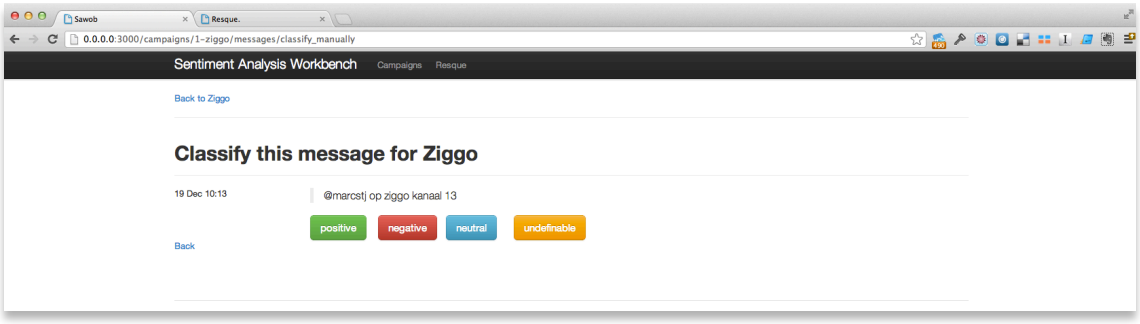

Figure 27: Manual classification in sawob

In the proof-of-concept, the holdout strategy was chosen because of the low number of manually classified messages. In the proof-of-concept a total of 2200 messages were manually classified. From the 2200 messages, two fifths of them are marked as a the training set for the combiner. Another two fifths are marked as the training set for the sub classifiers (like Naive Bayes). The last fifth of the messages are marked as test data, to evaluate the combiner and workbench.

5.7.2 Classification learning

In this proof-of-concept the J48 decision tree is chosen as the classification strategy in the combiner. The outcome of a decision tree is a set of rules which allow the user to see which attributes are relevant (and which are not).

To create a classification scheme for J48, WEKA is used. The WEKA CLI-API supports a 'weka.classifiers.trees.J48' command in which a training set is needed as input. Its output is text which describes al the learned rules. An example of this output is shown below:

```
C:Naive Bayes = positive
|   C:Word Count = positive
|   |   P:Word Count <= 20: positive (13.0/3.0)
|   |   P:Word Count > 20: neutral (10.0/4.0)
|   C:Word Count = negative: negative (11.0/1.0)
|   C:Word Count = neutral
|   |   P:Retweet? = yes: neutral (34.0/6.0)
|   |   P:Retweet? = no
|   |   |   P:Contains url = yes: neutral (43.0/13.0)
```

```
|   |   |   P:Contains url = no
|   |   |   |   P:Day Section = morning: neutral (14.0/6.0)
|   |   |   |   P:Day Section = afternoon
|   |   |   |   |   P:Contains ! = yes: positive (11.0/7.0)
|   |   |   |   |   P:Contains ! = no
|   |   |   |   |   |   P:contains Hashtag = yes: positive (13.0/7.0)
|   |   |   |   |   |   P:contains Hashtag = no: neutral (19.0/9.0)
|   |   |   |   P:Day Section = evening: negative (75.0/44.0)
|   |   |   |   P:Day Section = night: positive (2.0)
|   C:Word Count = unknown: neutral (0.0)
|   C:Word Count = undefinable: neutral (0.0)
C:Naive Bayes = negative: negative (159.0/35.0)
C:Naive Bayes = neutral: neutral (96.0/2.0)
C:Naive Bayes = unknown: neutral (0.0)
C:Naive Bayes = undefinable: neutral (0.0)
```

Each line represents a rule and optionally the outcome of the classification. At the end of each classification result, the two numbers represent the number of false-positives. This text-output is parsed by a J48TreeParser component which translated the text into a real J48 Tree object in ruby.

5.7.3 Visualising the decision tree

The J48 tree, which is the output of the classification learning process, is not only necessary for the sentiment classification in the combiner, but this result can also give very interesting insight in which attributes contribute to a correct sentiment analysis and overall insight in how the learning system in WEKA reacts on different settings of the workbench.

To make it easy for the customer service expert to see these rules, a tree visualizer is implemented. Visualising the decision tree data in the web interface was performed by Javascript. Together with D3 javascript library, which is a valuable javascript library for visualising data, this tree visualizer was build as shown in figure 28 below.

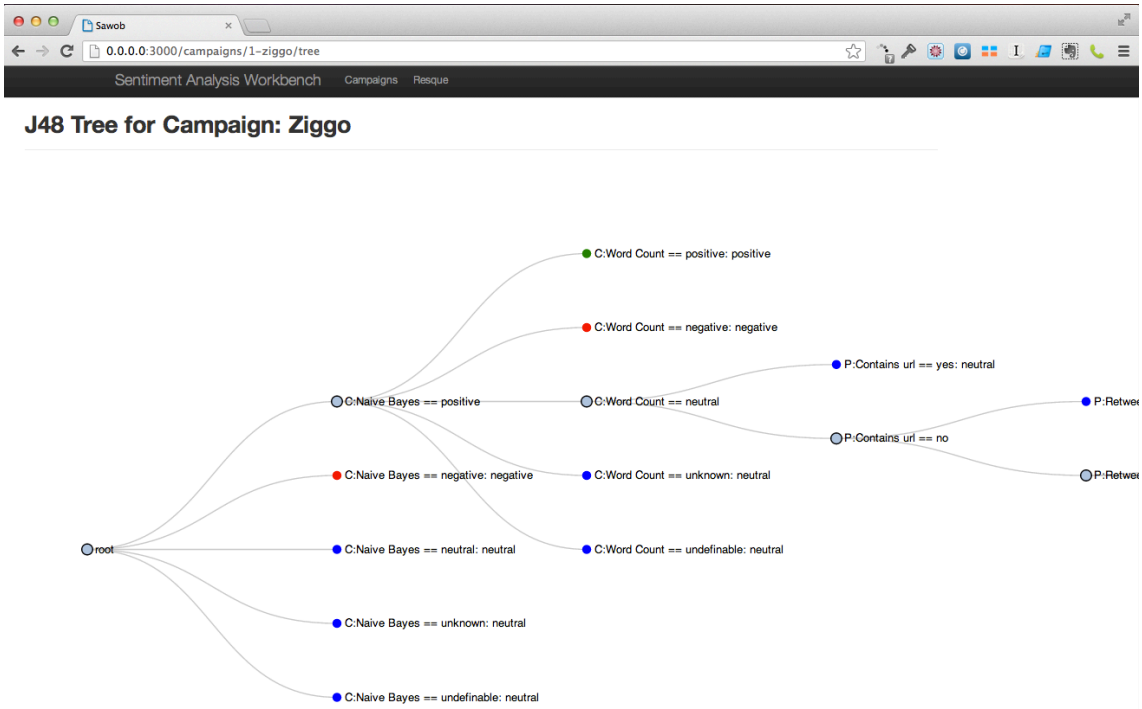

Figure 28: Visualising the J48 Tree

In this example, it is shown that the Naive Bayes sub classification is directly used as outcome, except for when it classified the sentiment of the message as positive. Then the outcome of the Word Count sub classifier is used, etc.

5.7.4 Classifying sentiment using the decision tree

With the J48 tree object as the result of the classification learning process discussed in 5.7.2 sentiment classifications can be made. WEKA is not involved here. The ruby J48 tree object consists of nodes which are rules that can be evaluated given a set of attributes. The rules are evaluated one by one and when a leave-node is encountered, the sentiment for that rule is returned.

5.7.5 One R as alternative for combining attributes

In 3.2.3 another classification strategy, named One R was also discussed and it was said that is has a good performance, although it only consists of a few rules. To evaluate this, the One R decision rules classifier was also implemented. The classification learning is performed with the use of WEKA and its 'weka.classifiers.rules.OneR' command. The result is also a text which describes a set of rules just as the J48 output. The real difference here is that the OneR output is not nested, so it's not a tree. However, the same rules parser can be used as for the

J48 classification output, because its root node still is a list of rules. The outcome of this process is therefore also a ruby J48 tree object, which can be visualised and used for classification.

5.8 Benchmarking

Making changes to the campaign trying to improve the sentiment analysis is only useful when performance of the sentiment analysis can be benchmarked.

Benchmarks are reports of the workbench hybrid classification for a certain campaign at a certain moment. Benchmarks can be made and named after the state of the workbench, a date or to the user of the workbench. From the benchmarks overview, two benchmarks can be chosen to compare to each other.

It is this comparison that makes the workbench a system that gives customer service experts insight in changes they make to the workbench in relation to the performance of the workbench in multiple performance measures.

5.8.1 Benchmark storage

Each benchmark has a date, on which the benchmark was performed. A name, for future reference and of course the results of the benchmark. The test set, as discussed in 5.7.1 is used to calculate these measurements. These include the accuracy, confusion matrix and performance measures of the overall hybrid classification and the sub classifiers as well. In the future, the current settings for properties, sub classifiers and combiner can also be stored, so the workbench can be reverted to the specific settings for the selected benchmark.

To make it easy to compare benchmarks, two benchmarks can be shown simultaneously as show in figure 29 below.

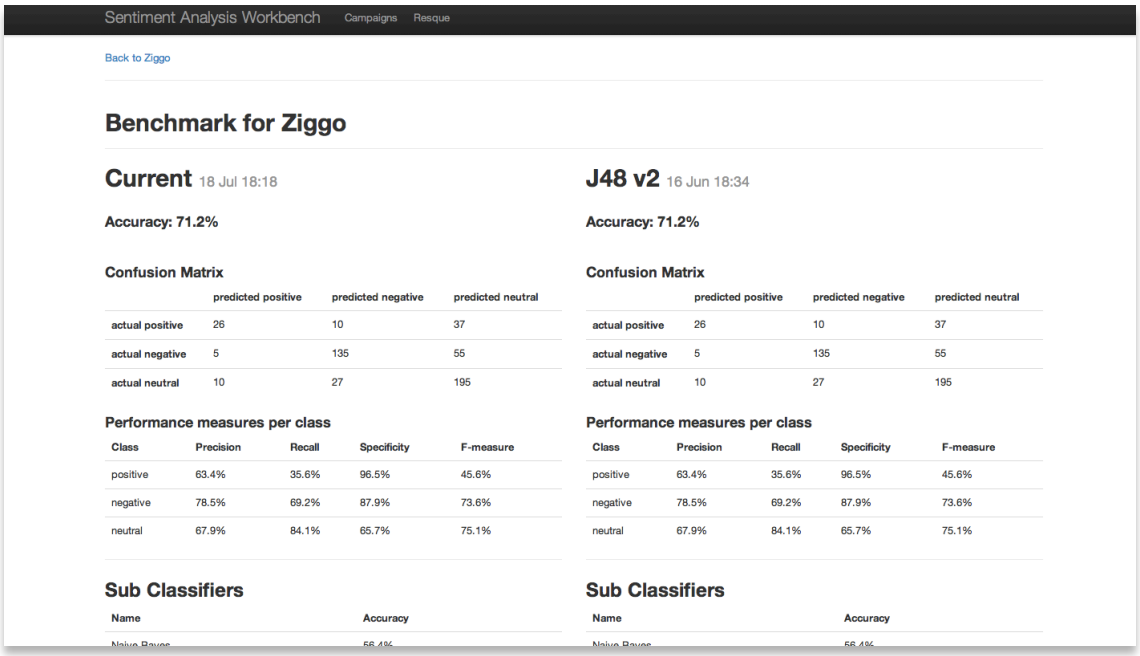

Figure 29: Comparing two benchmarks for the Ziggo campaign in the workbench

In the next chapter, the results and findings will be discussed based on the benchmarks and other indicators.

Part III - Results & findings

6. Findings

This chapter describes all findings of the proof-of-concept implementation of the so-called workbench that contribute to the overall conclusion of chapter 7. First, the performance of the individual sub classifiers is given as a point of reference and remarkable findings in their performance is discussed. Thereafter, the performance of the hybrid classifier using simple voting, j48 and OneR respectively is shown and discussed. The role of message properties is discussed for a case that uses the j48 decision tree and finally the findings of the adaptability of the system is elaborated.

6.1 Performance of individual classifiers

In the proof-of-concept implementation three sub classifiers (Naive Bayes, Wordcount, emoticons) were used and the individual performance is discussed here. The performance of each individual classifier implemented in this proof-of-concept is discussed first, of which the outcomes are used by the combiner. Hereafter, the performance of hybrid classifiers is discussed in the next paragraph. Note, the performance of these sub classifiers discussed here only applies to the implementation in this proof-of-concept and does not indicate the classifiers' performance in general.

Below the three confusion matrices with performance measures are given for each individual sub classifier. From these confusion matrices the first conclusion is that the accuracies of the wordcount and Bayes classifier are much better than the emoticon classifier. The next indication for the performance is to look at the F-measures for the different classes.

6.1.1 Naive Bayes classifier individual performance

See figure 30 below for the confusion matrix for the Naive Bayes classifier. Starting with Bayes, the lowest F-measure is found on the positive class, of 40%. A closer look at the other performance measures shows a low precision on positive messages of only 26%, meaning

that 26% of the messages that have been classified positive are actual positive messages. Also, the recall of neutral messages is not so good with only 41% of the actual neutral messages that it classifies as neutral and most of the neutral messages are classified positive falsely here as well. The same problem occurs for negative messages, but in a smaller amount. In other words, the Bayes classifier is **predicting too many messages as positive** messages resulting in wrong classifications.

Accuracy	Actual class	Predicted class			Performance measures per class			
		Positive	Negative	Neutral	Precision	Recall	Specificity	F-measure
56.4%	Positive	**64**	9	0	26%	88%	57%	**40%**
	Negative	69	**124**	2	78%	64%	89%	**70%**
	Neutral	112	26	**94**	98%	41%	99%	**57%**

Figure 30: Confusion Matrix of Naive Bayes classifier

6.1.2 Word Count classifier individual performance

Figure 31 below shows the confusion matrix for the Word Count classifier. The accuracy of the word count classifier is the best of the three, but still, an accuracy of 58.2% leaves room for improvement. The F-measure of the positive class looks dramatically, just 24%, lets take a closer look. Its precision is not good, but the recall of 16% is the type of error that should be taken into account mostly. It represents that of all positive messages, only 16% is classified correctly and the rest is classified incorrectly as neutral or negative. This error also occurs for negative messages, of which only 30% is correctly classified and further mostly as neutral. Both these errors cause a low specificity for the neutral class. What actually happens is that this classifier **misses a lot of subjectivity in messages**, and therefore classifies them as neutral.

Accuracy	Actual class	Predicted class			Performance measures per class			
		Positive	Negative	Neutral	Precision	Recall	Specificity	F-measure
58.2%	Positive	**12**	1	60	46%	16%	97%	**24%**
	Negative	5	**59**	131	94%	30%	99%	**46%**
	Neutral	9	3	**220**	54%	95%	29%	**68%**

Figure 31: Confusion Matrix of Word Count classifier

6.1.3 Emoticon classifier individual performance

See figure 32 below for the confusion matrix for the Emoticon classifier. The emoticon classifier is a typical classifier that only detects subjectivity according to emoticons in

messages. In most messages, no emoticons are present and therefore it was expected the performance would be bad because it should classify all messages without an emoticon as neutral. Remarkable is the very high specificity for the positive and negative classes, meaning that this classifier is **very well able not to falsely classify messages as positive or negative**. Another important issue with this emoticon classifier is the precision of the positive and negative classifications. The confusion matrix shows us a precision of 33% for the positive class and 82% for the negative class. In other words, **it is very well able to identify negative messages**, with only little messages that are falsely classified as negative, but **for positive messages the classifier is not reliable**. From 15 messages it classifies as positive, only 5 messages actually are positive.

Accuracy	Actual class	Predicted class			Performance measures per class			
		Positive	Negative	Neutral	Precision	Recall	Specificity	F-measure
47.2%	Positive	**5**	0	68	33%	7%	98%	**11%**
	Negative	2	**9**	184	82%	5%	99%	**9%**
	Neutral	8	2	**222**	47%	96%	6%	**63%**

Figure 32: Confusion Matrix of Emoticon classifier

6.2 Performance of hybrid sentiment classification

In the proof-of-concept implementation there are three hybrid classifiers that can be chosen. Below the performance is shown in figure 33.

Hybrid: Simple voting

Accuracy	Actual class	Predicted class			Performance measures per class			
		Positive	Negative	Neutral	Precision	Recall	Specificity	F-measure
58.2%	Positive	**16**	0	57	52%	22%	97%	**31%**
	Negative	5	**53**	137	100%	27%	100%	**43%**
	Neutral	10	0	**222**	53%	96%	28%	**69%**

Hybrid: J48 decision tree

71.2%	Positive	**26**	10	37	63%	36%	97%	**39%**
	Negative	5	**135**	55	79%	69%	88%	**73%**
	Neutral	10	27	**195**	68%	84%	66%	**75%**

Hybrid: One-R decision rules

	Positive	**0**	9	64	0%	0%	100%	**0%**
66.0%	Negative	0	**124**	71	78%	64%	89%	**70%**

	Neutral	0	26	**206**	60%	89%	50%	**72%**

Figure 33: Hybrid classification performance of simple voting, J48 decision tree and One-R decision rules.

6.2.1 Findings of hybrid classification

The different hybrid classifiers show different accuracies: From 58.2% for the simple voting classifier and 66.0% for the One-R classifier, to 71.2% for the J48 classifier. Overall a big improvement to the individual sub classifiers that showed accuracies of 47.2–58.2%, showing that combining different sub classifications and properties using a hybrid classifier is actually able to perform significantly better than the individual sub classifiers.

6.2.2 Performance of different hybrid classifiers

This paragraph describes the findings of each different hybrid classifier.

Performance of hybrid classification by simple voting classifier

The simple voting classifier performs the worst of the three hybrid classifiers, showing an accuracy of 58.2%, which is the same as the wordcount sub classifier did on itself. Overall not an improvement to single sub classifier usage. The F-measure classes of the positive and negative class are low, caused by their low recall values. Both classes show a high number of incorrect neutral classifications, where there actual sentiment of the message is positive or negative. This hybrid classifier combines the three sub classifications we discussed in last paragraph by voting. The sub classifications of both the wordcount classifier as well as the emoticon classifier showed us to miss sentiment (positive and negative) in much messages and therefore incorrectly classify them as neutral. By using a simple voting technique where two out of three classifiers are showing similar errors, the errors will show up in the overall result as well.

Performance of hybrid classification by J48-decision-tree classifier

With an accuracy score of 71.2%, the hybrid classification with the J-48 decision tree is a very good improvement. It contains two classes that show good F-measures, but for the positive class the F-measure is only 46%. What it shows is that this J48-decision tree classifier is not very good in classifying positive messages correctly and from all messages that are classified as positive, only 36% were actually positive messages and therefore classified correctly. In short, this is the best hybrid classifier of the three, but its weakness lies in identifying positive messages. We need to point out that all sub classifiers have lowers scores for F-measure for the positive class, and this effects the hybrid classification. This hybrid classifier performs

best out of the three we implemented, with an accuracy of 71.2% in relation to 47.2–58.2% for the individual sub classifications.

Performance of hybrid classification by One-R-decision-rules classifier

The One-R classifier shows an interesting result: it did not classify any message to be positive. This is also the reason the F-measure for the positive class is 0%. The other two classes show good F-measures and the other performance measures show no weak results as well.

6.3 Noteworthy findings of hybrid classification

Simple voting as a hybrid classifier

The simple voting technique as a hybrid classifier showed no improvements in accuracy over the Naive Bayes classifier. In the concept of hybrid classification the idea is that any sub classifier that is good at classifying a certain type of messages or class, can improve overall classification. The hybrid classifier should be able to use sub classifications conditionally to only use sub classifications that are accurate. Simple voting however takes all sub classifications of a sub classifier into account, also the neutral classifications of the emoticon classifier that this classifier was not built for.

One-R rules

The decision rules of the One-R hybrid classifier, shown below in figure 34, give insight in what the best attributes are to classify sentiment. In the One-R hybrid classification shown, the best attribute is apparently the Naive Bayes positive classification, but then classified as neutral: *"When sub classification of Naive Bayes = positive, classify this message as neutral"*.

Another remark to the One-R hybrid classification is that in the decision rules there is no rule that results in a positive classification. This means, based on the available information there are no common rules that describe the positive messages that are accurate enough to not cause even more errors that the rules now contain. This rule is not wrong, the overall accuracy was not as bad as a single sub classifier, but it indicates that the Naive Bayes sub classifier needs some attention to be more accurate in positive classifications.

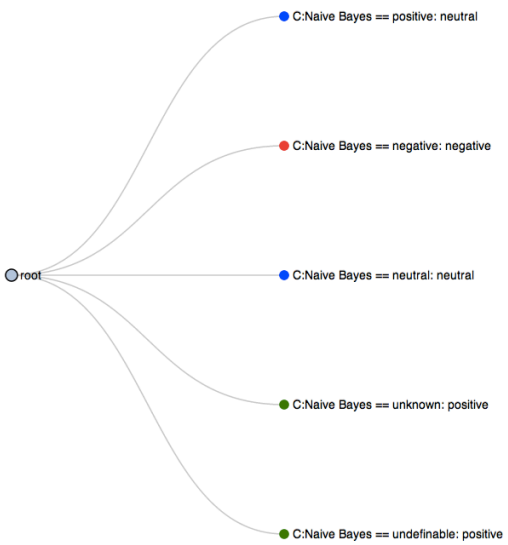

Figure 34: One-R hybrid classification rules

J-48 decision tree

In fact, the J-48 decision tree is an extended version of decision rules, where multiple levels of decisions can be formed. Figure 35 below shows the J-48 decision tree that was used for the hybrid classification described in paragraph 6.2. It shows the same branches in the first level as the rules of One-R (that are also based on the J-48 algorithm), but from the first branch, 3 other levels of branches are formed. All messages that Naive Bayes classified as positive are first sorted on their wordcount sub classification and for the neutral outcome even further for the emoticon sub classification and finally on that neutral class the property "laughing" is used to classify the rest of the messages. Remarkable is the fact that only one branch of the first level has deeper branches, that speaks for the power of the other first level "rules".

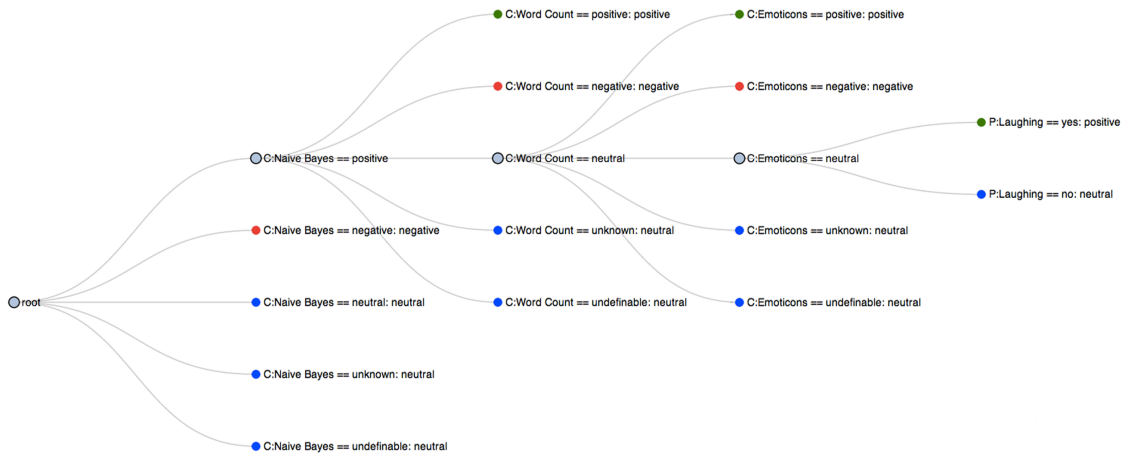

Figure 35: J48 decision tree for hybrid classification.

Weakness on positive messages

All sub classifications and hybrid classifications show weakness in identifying positive messages. For this proof-of-concept implementation we chose not to stratify the test and train data in such extend that each class would be represented the same number of times in each set. Doing so might improve all classifications, but for the purpose of this thesis that was not to get the best classification results but to show the value of hybrid sentiment classification for organisations.

6.4 Role of message properties

In the system design there are two types of message attributes that hybrid classifiers can use to classify a message: Message properties and sub classifications. Simple voting only uses sub classifications, but J48 and One-R also have property values as input. The One-R decision rules we have seen in this chapter uses only sub classifications, of which we can conclude are in this case better indicators for sentiment than the properties. In the decision tree one property is used for a classification decision, being of added-value to the classification. Although properties that are used in this are not in the primary level of the decision tree or rules, they already did contribute in the classification and with larger training test sets properties can be of even bigger effect.

6.5 Performance measures as indicators of performance

While evaluating the performance and interesting findings of the three different sub classifications and the three hybrid classifiers, the used performance measures have proven their usefulness to us. Comparing and reviewing performance is best done topdown, from overall accuracy, to finding out what type of errors occur on a single class. To do this, at first the accuracy and F-measures are regarded. From F-measures, the classes for which the classifier is under-performing can be found and through the other performance measures (recall, precision and specificity) the actual types of errors can be narrowed down to.

Similar to how we used the performance measures in the confusion matrices for the different classifications, customers service experts can use these measures to evaluate changes to the workbench and how to improve sentiment analysis further.

7. Conclusions & recommendations

This thesis report concludes with answers to the research questions and conclusions that can be drawn from research, design and proof-of-concept implementation of the system design for the hybrid approach to sentiment analysis.

7.1 Answers to research questions

How can adaptable hybrid sentiment classification improve sentiment analysis for organisations?

1. *How to design a hybrid sentiment classifier?*

2. *How to design a system architecture for the hybrid sentiment classification strategy that is adaptable by customer service experts and what are the underlying components of this design?*

3. *How does the chosen design enable customer service experts to improve sentiment analysis for organisations? And how does the design improve performance of sentiment analysis?*

7.1.1 How to design a hybrid sentiment classifier?

The design of the hybrid sentiment classifier, as discussed in chapter 4, is the strategy to combine different sub classifications performed by classification algorithms and other message attributes, like message properties, by a combining classifier into a sentiment classification. The combining classifier is trained with a set of messages' attributes (sub classification and properties) from which it learns how to classify new messages.

7.1.2 How to design a system architecture for the hybrid sentiment classification strategy that is adaptable by customer service experts and what are the underlying components of this design?

This second sub question is answered by the functionality of the workbench implementation of chapter 5.

The hybrid system design in short

The design of the hybrid sentiment classifier contains functionalities that supports customer service experts to work autonomously with the workbench to optimise sentiment classification for a given campaign. Manual classifications can be performed, that are used for sub and combiner classification learning as well as performance evaluation of classification results. New campaigns can be made and the message filter for a campaign

can be defined and changed by simple scripting. Message properties are extracted according to a simple script the customer service expert can define. These properties are used as input for the combining classifier next to sub classifications. New combining and sub classifiers can also be implemented to find new and better ways for sentiment classification.

7.1.3 How does the chosen design enable customer service experts to improve sentiment analysis for organisations? And how does the design improve performance of sentiment analysis?

This last sub question directs to how the hybrid system design enables customer service experts to improve sentiment analysis for organisations, by making incremental changes throughout the system based on insight in the classification. Chapter 5 concludes with explaining the implementation of benchmarking in the workbench. By making a change to the classification in the workbench on any level, the change in performance can be observed by comparing different benchmarks through the performance measures. This way it can be found out on a trial and error basis what changes have positive effects and in what way they changed performance: how performance measures change, how a decision tree changed, what attributes influence classification (sub classification and/or properties) or what classifiers performed better or worse. Chapter 6 describes that the performance measures are helpful in finding differences in performance of different benchmarks. Using the workbench, customer service experts will be able to improve sentiment analysis for an organisation successfully, based on insight in the performance and classification rules.

7.1.4 Main research question

Overall, this thesis answers the main research question by the design and proof-of-concept implementation of the hybrid sentiment analysis workbench:

"How can an adaptable hybrid sentiment classifier improve sentiment analysis for organisations?"

A hybrid sentiment classifier improves sentiment analysis for organisations by combining different techniques to extract sentiment or indicators for sentiment from a message. The inputs and techniques used can be managed from within the workbench to optimise sentiment classification for a campaign.

The improvement to sentiment analysis for organisations lies in different aspects that come together in the hybrid architecture. It makes use of different algorithms for different purposes, for sub classifiers and combining classifiers. It enables the customer service experts to define campaigns, for which messages are selected of which the content could contain sentiment about the campaign. The customer service expert can also define new message properties to optimise classification of messages about this campaign. By

comparing different benchmarks of the workbench for a campaign by performance measures the customer service expert is able to incrementally improve sentiment classification.

7.1.5 Contributions of workbench to GreenOnline

The intention was to design a system for hybrid sentiment analysis and a proof-of-concept for use by customer service experts to improve sentiment analysis for organisations based on WOM messages. With this workbench, customer service experts can start doing so right away. During this thesis project multiple parties have shown big interest in the hybrid system design for sentiment analysis and to built further on this workbench.

It is now up to the customer service experts to get to understand the underlying principles of how WOM messages respond to different techniques of the hybrid sentiment analysis workbench, rather than just building upon improvements on performance measures of benchmarks. By optimising a number of very different campaigns it is possible they have similarities like the number of properties to use, what combiner classifier to choose and how specific to specify a campaign to get mostly relevant messages.

7.2 Proof-of-concept evaluation

The workbench was built as a proof-of-concept implementation of the hybrid architecture proposed in chapter 4. During implementation and first use of this workbench some new questions, improvements and remarks were found. For future work on this topic these should be considered. However interesting and valuable, it was out of the scope of this thesis to elaborate further on these topics.

7.2.1 Findings, remarks & recommendations

Relation of properties to classification

In the current campaign of Ziggo, one property that showed good results was the 'laughing' property. Even though the hashtag #fail would imply a negative message, the J-48 did not use it. This might partly be due to the little amount of messages in the training data set that contain a hashtag. When much more training data would be collected and used, more specific decision rules can be formed, leading to even better sentiment classification results.

Limits of WOM messages

The design that is proposed in this thesis report is built to use WOM messages. Although it was found these messages are a good source of sentiment expressions by consumers, it should always be taken into account these messages reflect sentiment of consumers that use the Twitter medium.

Therefore, a next step could also be to find out wether the sentiment found in Twitter messages should be extended with sentiment analysis on other types of messages and what messages this could be. For example, discussions on topic relevant forums, renown bloggers posts, product reviews. These can for each campaign or organisation be different.

Scalability
During this implementation scalability was no issue, but towards a real life implementation or even a larger scale workbench scalability is a factor to take into account.

7.2.3 Further research

Performance for other organisations
During this thesis there was no purpose to significantly try to improve performance of the hybrid sentiment classification of the workbench for Ziggo and therefore the performance for any other topic is expected to be similar.

Sample size
Sample data are messages of which the sentiment is classified by hand and therefore is known. From this sample data three sets are formed: one for learning the combiner classifier, one for learning the sub classifiers and one for testing the performance of the overall hybrid classification by the workbench. In the workbench implementation the sample size was 2200 messages, of which the first two sets contained both two fifths of the messages and the last fifth was left for testing.

The sample size was chosen based on some examples and common sense. Each of the sets should contain enough messages to be a reflection of the messages that will come after. For a sample size there will be an optimum. Too small and classifiers will not learn enough to gain enough knowledge and the test set will not be a representation of all messages the workbench will classify. Too large, on the other hand, will result in overfitting the classifiers for a given moment, for example for a single day, and will give weak learnt classifiers as well.

Further research should be considered in this field, in particular in the field of sentiment analysis for organisations. Finding the right sample size will balance between being too specific and being too superficial. Using the workbench with real-life campaigns for real-life organisations it can be found out what sample size answers to this specific application of sentiment analysis. Also, it is not excluded different sample sizes can be best for general versus specific campaign topics.

Would it be desirable for workbench users to (temporarily) change the decision tree by hand?

Shortly discussed in chapter 5, it could be made possible to alter a decision tree or decision rules by hand to see how the hybrid classifier would perform using the altered model. Doing so, steps towards better fitting of sample data or settings of the workbench can be made to gain better classification performance.

7.3 Future prospects for the sentiment analysis workbench

The workbench can be further developed to be an even more helpful tool to optimise sentiment analysis for organisations. By seeing the current workbench at work some nice-to-haves came up that would be powerful additions.

- Some settings of the workbench can be optimised by the workbench itself, by running different benchmarks and taking the best performing option. This can be for example choosing which combiner classifier to use.
- Even more helpful would be that the workbench would be able to spot types of errors in its classification and to present these to the user in such way that it is possible to optimise on the given data. An example of such an internal evaluation can be that it gives a set of messages of which the workbench expects they do not belong in the campaign, a set of messages that have the same manual classification and are alike.

References

1. **Bifet 2010** A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer (2010). MOA: Massive Online Analysis, http://moa.cs.waikato.ac.nz. Journal of Machine Learning Research (JMLR), 11(May) 2010: 1601–1604.

2. **Bifet 2010-2** Albert Bifet and Eibe Frank, University of Waikato, Hamilton, New Zealand (2010). Sentiment Knowledge Discovery in Twitter Streaming Data, DS'10 Proceedings of the 13th international conference on Discovery science, Pages 1-15

3. **Davenport 2002** Davenport, T. H. and Beck, J. C. (2002), The attention economy: Understanding the new currency of business. Harvard Business Press, Cambridge, MA., 2002.

4. **Bohn 2009** Bohn, R and Short, J (2009) How Much Information? 2009, Report on American Consumers, San Diego, CA: Global Information Industry Center, University of California, San Diego

5. **Olcott 2011**, Olcott, A. (2011) Needing to Be Noticed: Understanding the Market in an Attention Economy, ISD Working Papers In New Diplomacy, Institute for the Study of Diplomacy Edmund A. Walsh School of Foreign Service Georgetown University

6. **Duan 2008** Duan, W., Gub, B. and Whinston, A. B. (2008), Do online reviews matter? - An empirical investigation of panel data, Decision Support Systems, v.45 n.4, p. 1007-1016, November, 2008

7. **Blenn 2012** - N. Blenn, K. Charalampidou, and C. Doerr (2012). Context-sensitive sentiment classification of short colloquial text. In R. Bestak, L. Kencl, L. E. Li, J. Widmer, and H. Yin, editors, Networking (1), volume 7289 of Lecture Notes in Computer Science, pages 97–108. Springer, 2012

8. **Witten 2011** - Witten, I.H., Frank, E., Mark A. Hall (2011), Data mining, practical machine learning tools and techniques - Third Edition, Hall; Morgan Kaufmann Publishers.

9. **Wright 2009** - Wright, A (2009), Mining the Web for Feelings, Not Facts - http://www.nytimes.com/2009/08/24/technology/internet/24emotion.html, Published: August 23, 2009 found may 25, 2012

10. **Helweh 2012** - Helweh, A (2012), Social media & sentiment: 5 takeaways froms the sentiment analysis symposium - http://www.socialmediaexplorer.com/social-media-monitoring/social-and-sentiment-takeaways-from-the-sentiment-analysis-symposium/, Published January 4, 2012, found august 11, 2012

11. **Gimpel 2011** - Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills,Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith (2011), Part-of-Speech Tagging for Twitter: Annotation, Features, and Experiments, Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers, June 19-24, 2011, Portland, Oregon

12. **Pang 2002** - Bo Pang, Lillian Lee, Shivakumar Vaithyanathan (2002), Thumbs up? Sentiment Classification using Machine Learning Techniques, Proceedings of EMNLP 2002, pp. 79–86

13. **Grimes 2012** - Seth Grimes (2012), Never trust sentiment accuracy claims - http://www.socialmediaexplorer.com/social-media-monitoring/never-trust-sentiment-accuracy-claims/, found august 20, 2012

14. **Haoda Huang** - Haoda Huang, Benyu Zhang, Definition of Text Segmentation, http://www.springerreference.com/docs/html/chapterdbid/63535.html, Retrieved october 12, 2012.

15. **Porter 1997** - Porter, M.F. (1997), An algorithm for suffix stripping, Readings in information retrieval - Pages 313-316, Morgan Kaufmann Publishers Inc. San Francisco, CA, USA

16. **Brew 2010** - Anthony Brew, Derek Greene, Pádraig Cunningham (2010), Using Crowdsourcing and Active Learning to Track Sentiment in Online Media, Proceedings of the 2010 conference on ECAI 2010: 19th European Conference on Artificial Intelligence, Pages 145-150

17. **Gaustad 2001** - Tanja Gaustad and Gosse Bouma - Alfa-Informatica Rijksuniversiteit Groningen (2001), Accurate Stemming of Dutch for Text Classification, Selected Papers from the Twelfth CLIN Meeting. Edited by Mariet Theune, Anton Nijholt and Hendri Hondorp , pp. 104-117(14)

18. **Das 2007** - Sanjiv R. Das and Mike Y. Chen (2007), Yahoo! for Amazon: Sentiment Extraction from Small Talk on the Web, Management Science Vol. 53, No. 9, September 2007, pp. 1375–1388

19. **Yuasa 2011** - Masahide Yuasa, Keiichi Saito, Naoki Mukawa (2011), Brain Activity When Reading Sentences and Emoticons: An fMRI Study of Verbal and Nonverbal Communication, Electronics and Communications in Japan Volume 94, Issue 5, pages 17–24, May 2011

20. **Pang 2004** - Bo Pang and Lillian Lee - Department of Computer Science - Cornell University, Ithaca, NY (2004), A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts, Proceeding ACL '04 Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, Article No. 271

21. **Quinlan 1993** - Quinlan, J. R. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.

22. **Rokach 2005** - Rokach, L.; Maimon, O. (2005). Top-down induction of decision trees classifiers-a survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C 35: 476–487.

23. **Berners-Lee 1998** - T. Berners-Lee, R. Fielding, U.C. Irvine, L. Masinter, Xerox Corporation (1998), Uniform Resource Identifiers (URI): Generic Syntax. http://www.ietf.org/rfc/rfc2396.txt, retrieved november 2012

24. **Compumine** - Evaluating a classification model – What does precision and recall tell me?, http://www.compumine.com/web/public/newsletter/20071/precision-recall retrieved november 13, 2012

25. **Goutte 2005** - Cyril Goutte and Eric Gaussier, Xerox Research Centre Europe 6 (2005), A Probabilistic Interpretation of Precision, Recall and F-score, with Implication for Evaluation, Advances in Information Retrieval Lecture Notes in Computer Science Volume 3408, 2005, pp 345-359

26. **Colton 2007** - Simon Colton (2012), Lecture notes of artificial intelligence course of Department of Computing, Imperial College, London http://www.doc.ic.ac.uk/~sgc/teaching/pre2012/v231/dt1.gif, retrieved february 2013

27. **Fuhrer 1993** - Jeffrey C. Fuhrer (1993), What role does consumer sentiment play in the U.S. macroeconomy?, New England Economic Review, January/February 1993 pages 32-44.

28. **Jansen 2009** - Bernard J. Jansen, Mimi Zhang, Kate Sobel, Abdur Chowdury (2009), Micro-blogging as Online Word of Mouth Branding, CHI 2009 ~ Spotlight on Works in Progress ~ Session 1, April 4-9, 2009 ~ Boston, MA, USA

29. **Trusov 2009** - Michael Trusov, Randolph E. Bucklin, & Koen Pauwels (2009), Effects of Word-of-Mouth Versus Traditional Marketing: Findings from an Internet Social Networking Site, Journal of Marketing, Vol. 73 (September 2009), 90–102

30. **Wilson 2005** - Theresa Wilson, Janyce Wiebe, Paul Hoffmann (2005), Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis, roceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), pages 347–354, Vancouver, October 2005

31. **Yi 2003** - Jeonghee Yi, Tetsuya Nasukawa, Razvan Bunescu, Wayne Niblack (2003), Sentiment Analyzer: Extracting Sentiments about a Given Topic using Natural Language Processing Techniques, In IEEE Intl. Conf. on Data Mining (ICDM) 2003, pages 427-434

32. **Koppel 2006** - Moshe Koppel, Jonathan Schler, The importance of neutral examples for learning sentiment, Computational Intelligence, Volume 22, Issue 2, pages 100–109, May 2006

33. **Buddhinath 2006** - Gaya Buddhinath and Damien Derry (2006), A Simple Enhancement to One Rule Classification, Technique Report at 2006.

34. **Gamma 1993** - Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides often referred to as the Gang of Four, Addison-Wesley, October 21, 1994

35. **Erk 2010** - Katrin Erk and Sebastian Pado (2010), Exemplar- based models for word meaning in context, In Proceedings of the ACL 2010 Conference Short Papers, pages 92–97.

36. **Go 2009** - Alec Go, Lei Huang, Richa Bhayani (2009), Twitter Sentiment Analysis, Final Project Report for CS224N course at Stanford NLP.

Appendix A
Shortlist of corpus messages

Below a shortlist of corpus messages in the workbench implementation for the Ziggo campaign and the manual classified sentiment, used for training and testing.

Message	Manual sentiment
Het wachten is nu nog op riolering? RT @fgj_oosterveld: Yes! glanerbrug heeft weer kabel tv! Dank aan #ziggo. Storing duurde maar vier uur.	negative
Na 3 dagen eindelijk weer internetverbinding .... #superblij :-) nu maar hopen dat het probleem na 7 jr modderen echt is opgelost #ziggo	positive
Zo naar de stad, op zoek naar een tv waar het ziggo kaartje in kan. Maar eerst de vuilnisbak langs de weg halen, hoe #burgerlijk	positive
@jeduth3 @viefje36 Kanaal 13 als je ziggo hebt;) #SR12	positive
@jarrnooo Als je digitale TV van ziggo hebt dan op 13.	neutral
@zoekusx ik heb het op een ziggo zender maar ik weet niet of jy dat ook hebt mopp xx	neutral
@ZiggoWebcare hoe zit het met werkzaamheden en een zakelijk Ziggo abonnement? Zou ik dan ook tot 16u geen verbinding hebben?	negative
Waarom heeft de Ziggo klantenservice muziek in de wachtrij als er om de 10 sec iemand doorheen praat?	negative
Dacht dat ik sinds gisteren weer digitale televisie zou hebben, want dan zou mijn smartcard geactiveerd worden op mijn adres. @ziggo	negative
@DirkdeBoer3 Is ook continue op 101tv Ziggo kanaaltje 133 of daar in de buurt. #glazenhuis	positive
Ziggo mannetje is er, hopelijk zo internet voor me leven #woehoe	positive
RT @tvefm: Enschede FM is vanaf vandaag ook digitaal bij #Ziggo op kanaal 780 ! TV Enschede is al langer digitaal op 40 #rmctwente	neutral
@strijb http://t.co/c46yLPji hier staat bij veelgestelde vragen dat je 3 devices kan aanmelden maar niet tegelijk kunt gebruiken ^ED	neutral
@ZiggoWebcaremaak gebruik van een HD recorder geïnstalleerd door ziggo monteur	neutral
Dat waren dus 40 minuten van werken via Ziggo op de server. Vodafone biedt me mijn werkmail en BB. Dat lukt dus nog wel. #werkzaamheden	positive

Message	Manual sentiment
@ZiggoWebcare Waarom krijg je als vaste klant geewn CI-module? En waarom zijn ze 2x zo duur op ziggo.nl als bij derden?	negative
WTF ??? Ziggo is de enige provider in Nederland waar je per maand kan opzeggen ? #maffia	negative
RT @tvefm: Enschede FM is vanaf vandaag ook digitaal te ontvangen bij #Ziggo op kanaal 780 ! TV Enschede is al langer digitaal op 40 #tvefm #rmctwente	neutral
@leeell1 als je inlogt op mijn ziggo, zie je alle smartcard nummers er staan? ^ED	neutral
@ZiggoWebcare ik gebruik geen versterker en installatie is gedaan door een ziggo monteur	negative
@KPNwebcare het is al opgepakt, door Ziggo	neutral
Enschede FM is vanaf vandaag ook digitaal te ontvangen bij #Ziggo op kanaal 780 ! TV Enschede is al langer digitaal op 40 #tvefm #rmctwente	positive
@sleddens ik zie de mail nu ook op m'n ziggo adres. Die gebruikte ik dus niet! Heb 3 mailtjes van afgelopen nacht. Niet voor vanavond	negative
@sleddens wtf... Het hele jaar kijk ik amper televisie, en nu tijdens #sr12 gaan ze me 's nachts afsluiten! #haat #ziggo	negative
@ess_ester jawel op digitale kanaal 101 tv. Ziggo net 133. Staat bij ons de hele dag aan! #SR12	positive
@ZiggoWebcare 2/2 gebruik van een router van Cisco (geleverd door Ziggo).	negative
@sleddens ook gecheckt, niks gehad... Doe eens, ziggo apendingetje johnvh punt enel :)	neutral
@johnvanhulsen Onderhoud, bij Ziggo. Ik heb al 10 mailtjes gehad de afgelopen weken.	negative
@xallisonk Bij mij 13, eventkanaal heet dat, maar ik heb ziggo	neutral
@saskiavanviegen al sturen ze een heel bos... Ziggo gaat de deur uit :-)	negative
Had ik nu maar een Nokia ..... dan kon ik mijn eigen hotspot bouwen. Zonder internet, kom ik het netwerk niet op. #thuiswerkdag #ziggo	negative
Iets met internet en kabel dat uitvalt. #ziggo. Modem maar eens resetten; zo wordt thuiswerken wat lastig.	negative
@arvid ziggo	neutral
Vraag even gratis editie aan! vgoossens@ziggo.nl http://t.co/jFEqCNBW	neutral
Verwarmingsketel wordt vervangen én ziggo is een storing aan het verhelpen. #alswedantochbezigzijn	negative
@AGH74 wij hebben rooibos gekregen...#ziggo	positive
Nu hebben we internet via XS4all, tv via Ziggo en telefoon via KPN. De ISDN-lijn bromt dus er moet wat gebeuren. Vandaag maar eens uitzoeken	neutral
Welke all-in-combi is het voordeligst, inclusief 2 telefoonlijnen? XS4all, KPN en Ziggo zijn in de race in huize Brouwers/Ietswaart.	neutral
Wachten op #ziggo monteur..	negative
ugh #sr2012 heeft steeds storingen met ziggo #grrr	negative
Ziggo doet ook alles om niet nog meer klanten kwijt te raken... Too little too late :-P. #glasvezel http://t.co/gX90Vg1k	negative