Radboud University Nijmegen

Master's Thesis

Predicting TV Ratings using Twitter: A correlation between tweets, sentiment and TV ratings

Author:
ing. H.J. van den Brink

Supervisor:
prof. dr. ir. Th.P. (Theo) van der Weide

December 8, 2013

Abstract

The thesis describes the research that has been done on using Twitter to predict TV ratings. The models used in predicting TV ratings are realized through the following phases:

1. Literature study to determine the state of the art in using Twitter in sentiment mining and predicting and statistics in TV ratings.

2. An analysis of the possibilities of Twitter and how to process tweets in a formal form for future use.

3. The development of the model for sentiment mining.

4. The results and accuracy of the sentiment mining model.

5. Multiple models for predicting TV ratings using Twitter and sentiment mining.

6. Results and discussion of the most effective model in predicting TV ratings and the use of the gathered sentiment.

These steps resulted in multiple models for predicting TV ratings with different variables. The used variables are the total tweets, the tweets for each day of the week and the sentiment in these set of tweets. Using these different variables we determine which model, with which variables, result in the most accurate model in predicting TV ratings. We end with a discussion and take a look at the results, compare them and try to distinguish which model performs the most accurate. One models may look promising but comparing the root-mean-square (RMS) error value, a statistical measurement, will give a better way of comparing the methods. The RMS error value also gives a substantiated way of arguing. The conclusion of this research is that using the least square method with positive sentiment is one of the best performing methods we researched. Although future research would be necessary to get it more accurate with less deviation, the methods researched are able to predict TV ratings in more than 66% of the time with a tolerance of 150.000 viewers.

Contents

Chapter 1

Introduction

1.1 Context of the research

Twitter is a free internet service which allows users to send a 140 characters maximum length message called a tweet. It is a social network on which users to interact with each other and share tweets. It was launched in July 2006 by Jack Dorsey and is now in the top 10 most visited internet sites of the world. Twitter has 550 million registered users which send approximately 340 million tweets each day[1].

Twitter is used to send tweets about all things in life and gives a great way to take the laboratory experiments outside the laboratory and to an online world. Text analysis, mood detection and other measurement algorithms are getting more accurate by the day. This allows scientists to use Twitter to do experiments on a large scale. Twitter offers a huge amount of data, allowing researchers to create a technic to automatically learn a machine to process these data and get more accurate in predictions. This is also a part of this research, refining and improving sentiment mining on Twitter and using this data to predict TV ratings.

The people behind TV programs discovered they can increase their reach by using social media. Almost every program has a specific hash tag. For example the Dutch soap "Goede Tijden Slechte Tijden" has the hash tag #GTST, The Voice of Holland has the hash tag #tvoh and Pauw en Witteman has the hash tag #penw.

[1] https://blog.twitter.com/2012/twitter-turns-six

These hashtags are used to refer to a specific TV programs on Twitter. Twitter is used by 3.2 million Dutch people[2], thus it can give a great insight in how the Dutch think about certain topics or how many people watch, for example, a TV program. To extract the interesting facts out of these tweets we need methodologies and algorithms.

The goal of this research is to create a model which can be used to predict the sentiment of tweets about a TV program and the TV ratings of a certain TV program by using history data about the TV ratings and different variables in the tweets about this TV program. We try to derive the mood and other statistics out of the tweets and together with the history TV ratings we try to create a model which can predict the current TV ratings.

1.2 Research question

In order to keep a good view on our topic we created a main research question and several sub questions.

Main research question:

To what extend can Twitter be used in predicting TV ratings of a specific TV program?

Sub questions:

How to mine sentiment of tweets with a automatic model?
How to predict the TV ratings using the gathered sentimental data?
How are tweets and TV ratings correlated?
How are tweets and TV ratings correlated? Can sentiment in the tweets affect the accuracy of this predicting of TV ratings?
To what extend do different models create a better predicting?

1.3 Research method

To answer the main research question we have to answer the sub questions first. Previous studies can be helpful in creating a model for the sentiment mining part. We first do a literature study to the current state of the art methods in sentiment mining and predicting TV ratings.

[2] Newcom Research & Consultancy 2013 http://www.socialmediameetlat.nl/pdf/newcom.pdf

The accuracy of the sentiment mining is important when using this sentiment in the TV rating prediction. If the accuracy of the sentiment mining is very low, it will be useless in the prediction of TV ratings since we keep calculating with errors and create even bigger errors.

In the prediction of TV ratings we use the root-mean-square (RMS) error value as measurement of the performance of the used methods. The RMS is a stastical measure of the magnitude between the real TV ratings and the predicted TV ratings. It calculates the distance between both measurements to get a value which can be used to compare other methods.

Chapter 2

Related work on sentiment mining and the prediction of TV ratings

2.1 Identifying moods and sentiment mining on Twitter

Twitter has been subject of several studies lately concerning mood and emotion detection, also called sentiment mining. The mood on Twitter has been used to predict the stock market [1, 3] that gave a an accuracy of 86.7% in predicting the daily up and down changes in the closing values of the Dow Jones Industrial Average.

Automatic detection of emotion has also been done with a large emotion-labeled dataset in the study of Harnessing Twitter 'Big Data' for Automatic Emotion Identication [14]. That study showed interesting results about emotion and mood detection of the general public using Twitter. The research is about categorizing tweets into 7 different emotions: joy, sadness, anger, love, fear, thankfulness and surprise. The research is a study of how a data set can be categorized just by using the hashtags in tweets instead of manual annotation by a human expert. They were able to gather a much larger training data set, instead of using a manual annotated data set which is time consuming and generally smaller.

The methodology we use to determine the mood of tweets in our research will be derived from previous researches, in a way which enables the system to process a large amount of tweets. A machine learning technic would be a way to process large amount of tweets.

By using emotion words and classifying them in the same emotion-category as used in another research about annotating and detection emotions in tweets[10]. The method is similar to the method when only looking at emoticons in determination of the emotion (three classes; positive, negative and neutral), used by Pak and Paroubek in an article of 2010 [7]. Using emoticon can perform up to 70% accuracy with predicting the sentiment of an article[9].

2.2 TV ratings in the Netherlands

TV Ratings in the Netherlands are provided by "Stichting KijkOnderzoek" (SKO). The research is done by private companies Intomart GfK and The Nielsen Company on behalf of SKO. The panel of viewers consists of 1235 households, they are equipped with special hardware. The hardware registers the programs being watched. This data is then sent directly to SKO [1]. The households are selected randomly.

As soon as the TV turns on the system of SKO starts registering what TV program is currently watched and at what time of the day. The user is also asked who is watching and with how many people they are watching. These statistics are sent to SKO and processed at SKO.

Intomart GfK records the viewing habits of the panel members and provides information on the members, so different groups of viewers can be distinguished. The Nielsen Company provides the information necessary to determine to what programs or advertisements people have looked at. This includes programs titles, the start and end times of programs and ads and the genre of the programs. The ratings are created by combining the data of both agencies.

The highest rate of viewers is often reached at a Sunday, according to SKO. In the Netherlands "Acht uur Journaal" and "Studio Sport" are favorite programs. Another frequently watched program is "Boer zoekt vrouw" which is only at a certain period of the year broadcasted, but gains as much as 4 million viewers.

Stichting KijkOnderzoek is adapting new ways to measure TV ratings, since the developments in delayed viewing using a disk recorder or settopbox. The way people consume TV programs changes by the influence of internet in our society. More people are using the internet to watch their favorite programs when they want to. These ways of viewing TV programs are not included in this research. We only focus on real-time broadcasting. But this study can, when refined, by used to also analyze delayed viewing.

[1] http://www.kijkonderzoek.nl/Methodologische_beschrijving_kijkonderzoek_2013_def.pdf

2.3 Nielsen about Twitter and TV ratings

2.3.1 Correlation between Twitter and TV ratings

A research done by Nielsen[2] and reported in March 2013, shows that there is a correlation between Twitter and TV ratings. Though we can't verify the results, since they didn't publish the paper (only a press article), we still take it into account when looking at Twitter and TV ratings. Twitter is one of the key variables in the TV ratings, according to the research. As the number of tweets grow on a certain program, the TV ratings of that program are also likely to increase. Prior year rating and advertising are the two other variables. Prior year ratings is the biggest one of these two, if a TV program was watched by millions of people last year it will probabily attract millions in a new season or year. The research and research method is not public, but some results are. Though it is not possible to verify them or get to know the method used.

"According to the study, for premiere episodes, an 8.5% increase in Twitter volume is associated with a 1% increase in TV program ratings for 18-34 year olds. Additionally, a 14.0% increase in Twitter volume is associated with a 1% increase in TV program ratings for 35-49 year olds, reflecting a stronger relationship between Twitter and TV for younger audiences."

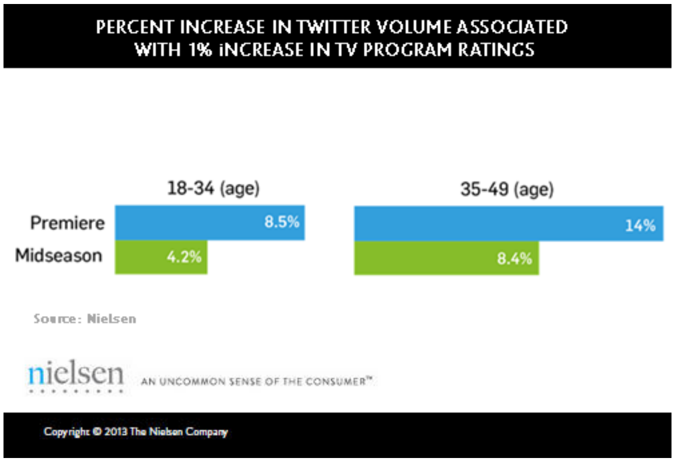


Figure 2.1: Twitter influence on TV Ratings

[2] http://www.nielsen.com/us/en/press-room/2013/new-study-confirms-correlation-between-twitter-and-tv-ratings.html

These percentages can be useful in the model to predict real time tv ratings, using gathered variables like mood, total amount of tweets and the day of the week.

2.3.2 Casual influence

Nielsen also showed a two-way casual influence between Twitter and TV ratings[3]. TV ratings influenced the tweets about the program in 48%, while Twitter influenced the TV ratings in 29% of the analyzed 221 primetime shows. The research also shows that tweets causing TV ratings to increase depends on the genre of the program. Reality shows, drama and sports are the genres on which the tweets caused the TV ratings to increase.

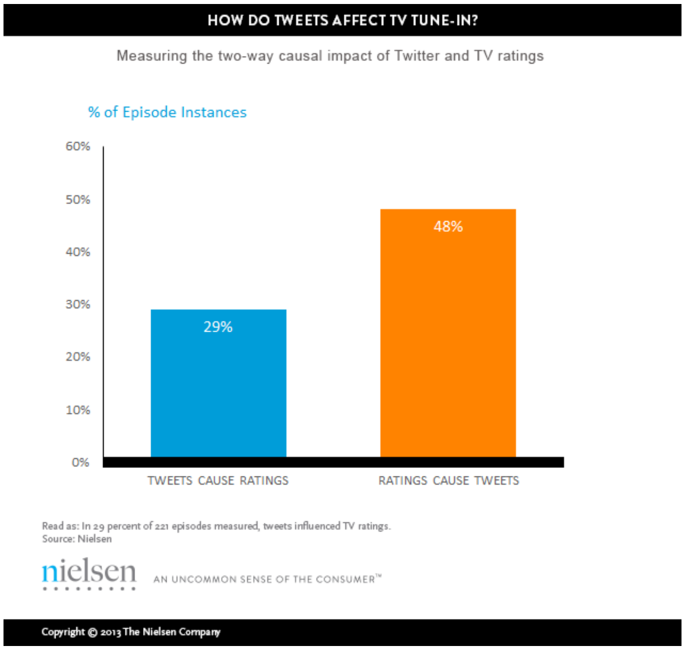
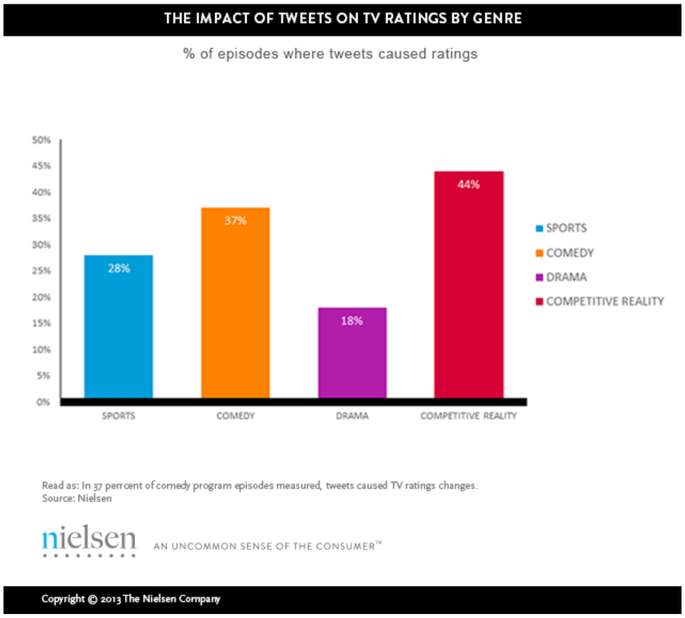


Figure 2.2: The impact of Tweets on TV ratings

[3] http://www.nielsen.com/us/en/newswire/2013/the-follow-back–understanding-the-two-way-causal-influence-betw.html



Read as: In 37 perrcent of comedy program episodes measured, tweets caused TV ratings changes.
Source: Nielsen

Copyright © 2013 The Nielsen Company

Figure 2.3: How do Tweets affect TV tune-in

2.4 Emoticons

In this study we include use of emoticons for our self-learning system in detecting the mood on a particular TV program. Emoticons are a way of expressing your mood or feelings by only using a few characters. Emoticons are founded in the 20th century. In 19 September 1982 it was presented in a message sent by Scott Fahlman.

The actual message:

```
19-Sep-82 11:44    Scott E  Fahlman             :-)
From: Scott E  Fahlman <Fahlman at Cmu-20c>

I propose that the following character sequence for joke markers:
:-)
Read it sideways.  Actually, it is probably more economical to mark
things that are NOT jokes, given current trends.  For this, use
:-(
```

There are many different emoticons in use. The top 20 used emoticons, as described in 2.4, account for 90% of the total use of emoticons.

In our study, however, we focus only on the use of :) and :(. We do this because these emoticons have a clear, common understanding. :) is positive, and : (is negative, we use these emoticons to learn the ‘value’ of a word. We therefore want to know what words are often used in combination with a positive emoticon and which words are often used with a negative emoticon. We than use this information to predict what the mood of a tweet without an emoticon would be.

:)	Happy face
:D	Laugh
:(	Sad face
;)	Wink
:-)	Happy face (with nose)
:P	Tongue out
=)	Happy face
(:	Happy face (mirror)
;-)	Wink (with nose)
:/	Uneasy, undecided, skeptical, annoyed
XD	Big grin
=D	Laugh
:o	Shock, Yawn
=]	Happy face
D:	Grin (mirror)
;D	Wink and grin
:]	Happy face
:-(	Unhappy
=/	Uneasy, undecided, skeptical, annoyed
#=(	Unhappy

Chapter 3

Case study on Twitter and TV ratings

3.1 TV ratings

As mentioned before, Stichting KijkOnderzoek (SKO) is responsible for gathering TV ratings for all TV stations in the Netherlands. The ratings are daily presented on the website of SKO[1], besides a top 25 of the most watched programs you can also check the ratings of each TV station or TV program for the past two weeks. Intomart GfK and The Nielsen Company are assigned by SKO to do this job.

3.2 Twitter

Twitter is a free Internet service that allows users to send a 140 characters maximum length message called a tweet. It is a social network on which users to interact with each other and share tweets. It was launched in July 2006 by Jack Dorsey and is now in the top 10 most visited Internet sites. Twitter has 550 million registered users. These users send approximately 340 million tweets each day[2].

Twitter is used to send tweets about all things in life and gives a great way to take the laboratory experiments outside the laboratory and to an online world. Text analysis, mood detection and other measurement algorithms are getting more accurate by the day. This allows scientists to use

[1] http://www.kijkonderzoek.nl/

[2] https://blog.twitter.com/2012/twitter-turns-six

Twitter to do experiments on a large scale. Twitter offers a huge amount of data, allowing researchers to create a technic to automatically learn a machine to process these data and get more accurate in predictions. This is also a part of this research, refining and improving sentiment mining on Twitter.

A research done by Pear Analytics [8] shows that 40.55% of the total tweets they captured are pointless babble, 37.55% of the tweets were conversational, 8.7% were Pass-Along Value (also called retweets), 6% were 'self-promotion', 3.75% was classified as spam and 3.6% as news tweets.

3.3 Dutch TV series Goede tijden, slechte tijden

In our research we focus on the Dutch TV soap Goede tijden, slechte tijden (GTST). GTST is the longest-running Dutch soap, which started on the 1st of October 1990. The reason why we choose this particular series is because it is broadcasted from Monday till Friday. It has an average of 1.5 million viewers per episode. Because this continuity we get a constant flow of TV ratings and tweets. RTL Nederland, the TV station which broadcasts GTST, made the TV ratings of GTST over the last year available for our research. We have also collected tweets about GTST since March 2013. GTST is on hold in the summer because of the holidays. This means from July to September we were not able to collect tweets. Though we were still able to collect tweets over more than 5 months.

3.4 Why research Twitter and TV ratings

Watching TV has changed in recent years . People are increasingly using the opting for delayed viewing for example by recording a program or looking back over the Internet.

This way of consuming TV also requires a change of measuring ratings. Stichting KijkOnderzoek has been focusing increasingly on investigating delayed viewing, to get a global view of each TV program.

Twitter can be an important medium in gathering viewing habits of people. To investigate whether Twitter is suitable we look in our study if there is correlation between TV broadcast and tweets around the TV broadcast. When it can be used for the measurement of ratings about regular broadcasts, Twitter could possible also be used for the measurement of delayed viewing.

To investigate the effects on Twitter with another approach, we also investigate whether the intensity of emotion in terms of tweets effect on viewing behavior. If this is the case, then this can be used as part of the calculation for the final TV ratings number.

Nielsen already showed a correlation between TV ratings and tweets. In our research we want to verify thus probable correlation and use this correlation in predicting TV ratings. Since the normal prediction is mainly done by the 1235 households, it relies mostly on statistics. Twitter can provide a lot more information and statistics that can be used in predicting and classifying TV ratings. It can show at what time of the day people conversate about a certain topic, it can provide information about age groups and sex. Twitter can also be useful in looking at delayed viewing and tweets about a program as a whole.

If we are able to use the tweets to predict, with some uncertainty, TV ratings, the model shows it has also the potential to be used in further use in statistics about TV programs.

Chapter 4

The collection and processing of tweets in sentiment mining

4.1 Gathering Tweets about a certain object

Twitter allows developers to get access to the so called set of streaming APIs (application programming interface). This is a set of APIs which allows a developer to get messages pushed directly from Twitter. The APIs are divided into three different endpoints; public streams, user streams and site streams.

Public streams are about gathering public data through Twitter, like specific users, topics and data mining. User streams are about gathering data from one particular user, it contains roughly all the data corresponding with a single user of Twitter. Site streams are multi-user version of user streams. It allows a developer to connect to Twitter on behalf of many users.

In our system we will be using public streams. The public stream will be used to gather all tweets about a certain topic, or certain topics. In particular we will be using POST statuses/filter to filter tweets with certain topics. In one single connection to the Streaming API we can specify multiple parameters. The default access level allows up to 400 track keywords.

4.2 Pre-processing the Tweets

Tweets sent by users can contain #hashtags, replies or mentions to @users, URLs like http://google.com and other things which may influence the way we can process tweets. To get a clean dataset without hashtags, references to user profiles or URLs we will use an algorithm to strip these things out of the tweet. A reply, or user mention, like justinbieber will be automatically changed to the token USER. This way we do not only anonymous our dataset, but we can also later on count how many user mentions has been done by a particular user.

The # of the hashtags will be stripped, for example #work will become "work". This way it will become a regular word, which again makes the word usable for statistics. If a TV program is mentioned in the hash tag it will also be used to categorize the tweets. Next thing thats being done is to strip all the punctuation marks and other marks like: , ; . ! ? - = % &̂ * ( )

Exclamation points can be possible interesting when looking at mood or emotion, so we count them in each tweets and store this number in the database for possible later use.

4.3 What is a tweet in our model

We consider a tweet T as a subsequence of words, hashtags and emoticons called p.

$$T =< p_1, ...p_k > \tag{4.1}$$

With $em(T)$ as a subsequence with only emoticons and $txt(T)$ as a subsequence of words and hashtags, whereas hashtags are considered as words. The output of $txt(T) =< w_1, ...w_n >$. W_i are all words and hashtags found in a tweet, but without the emoticons and other punctuations.

Chapter 5

The prediction of sentiment out of tweets using emoticons

We want to know characteristics about a tweet. To get these characteristics we use certain algorithms. By creating a theoretical framework using these algorithms we try to create a uniform framework that can be applied to most languages. In this study we focus on the Dutch language, by looking at tweets only containing the word "het". The word "het" can be used as a definite article or pronoun in Dutch. This allows us to mostly only collect Dutch tweets and eliminate a certain bias.

5.1 Used symbols and abbreviations in equations

Symbol	Meaning
T	Tweet
pos(t)	Positive tweet
neg(t)	Negative tweet
pem	Positive emoticon
nem	Negative emoticon
ug	Unigram
bg	Bigram
pos(ug)	Number of counted unigrams in a positive tweet
neg(ug)	Number of counted unigrams in a negative tweet
pos(bg)	Number of counted bigrams in a positive tweet
neg(bg)	Number of counted bigrams in a negative tweet
TUGp	Total positive unigrams
TUGn	Total negative unigrams
TBGp	Total positive bigrams
TBGn	Total negative bigrams

5.2 Noise reduction on the training set

Tweets are free expressions by people, this means it has no predefined format and can contain almost every symbol or letter. Anyone can construct a tweet in any way, which may also contain grammatical errors, misspellings or 'slang'. These errors may create noise in the data set. With the use of emoticons we still are able to use these errors in a constructive way. If a word is deliberately misspelled, like for example "pwned" instead of "owned", we can still use this data. Since our system gathers data by using emoticons, also these misspelled words are of value.

Retweets are a real threat in our way of using tweets in the training set. Retweets are tweets re-send by other users. In most cases a retweet starts with the symbols RT. Because some tweets can be retweeted like 1000 times, this would blur the training set if such a tweets contains an emoticon. Words in the tweet, which are retweeted 1000 times, would get a higher weight on either positive or negative sentiment. To counter this possible noise in the training set we ignore all tweets starting with RT or all tweets, which are retweeted according to the meta-information send along with the tweet.

Delayed viewing of TV programs can also cause to influence the noise in the data set. Since we can't detect tweets that are sent about delayed viewed TV watching, we can still focus on a particular time of the day to be surer the tweets are about the TV program going on in real time. This can be done by extracting the start and stop time of each TV program out of the TV guide and only use the tweets that are sent 30 minutes before and 30 minutes after the TV program was scheduled.

We consider a tweet either positive or negative. Tweets can also be sarcastic, which can blur our training set. Since they only occur occasionally, we ignore them. Although mixed sentiment wouldn't create a real noise, since the words get both a positive and negative add up in value, we still want to avoid these kind of tweets. To avoid mixed sentiment in our training set we assume

$$pos(T) \cap neg(T) = \varnothing \tag{5.1}$$

5.3 Different approaches in sentiment mining

5.3.1 The use of unigrams in predicting sentiment

In the unigrams table, we take each word out of the tweet and depending on the emoticon found in the tweet, it is either a positive or negative emoticon. Unigrams are used a lot in text analysis. It is an easy way of learning in a machine-learning environment. By extracting each word out of a tweet T we gather all the unigrams. Given a tweet $txt(T) =< w_1, ...w_n >$, the associated set of unigrams is $\{w_1, w_2, ...w_k\}$.

For example, given the Dutch tweet $T =$ *"Ik studeer de opleiding Informatiekunde aan de Radboud Universiteit"*, the associated set of unigrams is: $ug(T) = \{$*Ik, studeer, de, opleiding, Informatiekunde, aan, de, Radboud, Universiteit*$\}$

5.3.2 The use of bigrams in predicting sentiment

Bigrams are combinations of words. Given $txt(T) =< w_1, ..., w_n >$, the associated set of bigrams is: $bg(T) = \{(w1, w2),(w2, w3), ...,(wn - 1, wn)\}$. For example, the Dutch tweet $T =$ *"Ik studeer aan de Radboud Universiteit* will lead to: $bg(T) =\{$*(Ik,studeer), (studeer,aan),(aan,de), (de,Radboud), (Radboud, Universiteit)*$\}$

It's better to use bigrams in order to detect negated phrases like "not good" or "not bad" [5]. However, bigrams combined with unigrams give a much higher accuracy.

5.3.3 The training set used in the sentiment mining model

Our intention is to estimate the mood of a tweet from the words used in the tweet. In order to learn from our training set how words contribute to the mood, we make the assumption that the mood of a tweet is derived from its usage of emoticons. Let the emotional value pem (positive value) or nem (negative value) denote whether emoticon is assumed to be a positive or negative emoticon. Consider tweet T, the tweet T will only be used in the training set if $em(T) \neq \varnothing$.

The tweets used in the training set are gathered using the algorithm described in section 4.2 Pre-processing the tweets. The training set has two different tablets, one with unigrams and one with bigrams. The tweets used for the training set are selected by looking for emoticons by selecting $em(T)$ tweets, either positive or negative. Given a positive or negative tweet, unigrams and bigrams are created and stored in the database along with the value of the emoticon. If an emoticon is negative, the counter for negative linked to the unigram will be incremented. This way a set of unigrams and bigrams will be created with positive and negative values. If $em(T)$ contains only positive emoticons, all the words in $txt(T)$ will get a positive value in the database. The negative values are determined similar.

We create our learning set of unigrams and bigrams with the following abbreviations:

$$P(pos(t)|pem \in t) \approx 1 \tag{5.2}$$

$$P(neg(t)|nem \in t) \approx 1 \tag{5.3}$$

Given a positive or a negative emoticon, in the most cases the tweet is also positive or negative. There are some exceptions, like sarcasm, but they only occur occasionally. Since we don't focus on these exceptions, we choose to use the assumptions above.

5.4 Limitation of the programming language used

We use a PHP script to calculate all these probabilities. PHP is a considered a scripting language, mostly used for create dynamical websites. PHP was founded in 1994 by Rasmus Lerdorf. The scripts run server side and are compiled in real time when a PHP page is requested.

Because of the numerous calculations we do, numerical stability is something we need to be aware of. A trivial example is when you are limited to only use 3 digits, and need to add one millions times one to the number 1234. Since you are limited to 3 digits, 1234 will become $123 * 10^1$, which can be rewritten as 1230. If you then add 1 to this number, it will be 1231, but since the 3 digits limitation it will again be $123 * 10^1$. Though adding one millions times one to this number seems to make a huge difference, it is not.

Another example is when you divide 1 by 2, and the result again by 2 and keep doing that. Theoretically this should get close to 0, but never become 0. If you create a simple script in PHP, which does exactly that, it returns 0 after 1075 times dividing the result. In our PHP implementation, we found that if it tries to divide $4.9406564584125 * 10^{-324}$ by 2, the result is 0. Although we are not limited to 3 digits in our programming or keep dividing a number multiple times, we still need to take into account that numerical stability can become an issue when adding or subtracting very small numbers. The rationale is that the floating point arithmetic is based on a finite (though very large) set $\mathcal{F}$ of numbers. Therefore floating point calculations cannot have the exact value when this result is not a number from $\mathcal{F}$. In that case the result is a number from $\mathcal{F}$ that is close to this result. So in the above 3-digit arithmetic, $123 * 10^1$ is the best approximation of 1230. When doing calculations, errors may propagate as we see in the computation above. In the case above we see that the error propagation is such that the final answer is not even realistic anymore.

This can be explained because PHP, but also i.e. Java, Python and Ruby, use IEEE 754 double precision format. In this format the significand (or mantissa) has 52 bits and the exponent has 11 bits. The exponent can be in a range of -1022 and 1023, which means either all bits are 0 or 1. The value of the exponent defines the position of the 'floating point' in the siginificand, which means it defines how many digits are available for the integer part and how many digits are available for the fraction part. If the exponent part is depleted, only 52 bits are left for the fraction part. This results in 1023 + 52, which explains the maximum of 1075 times dividing.

The lowest number we were able to get by keep dividing the result of 1 divided by 2 was $4.9406564584125 * 10^{-}324$, this number is represented in the 64-bits floating point as $5 * 10^{-}324$:

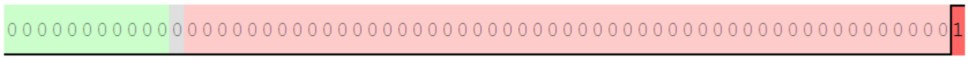


Figure 5.1: The smallest number in floating points as a visual representation

This immediately shows why we can't have a smaller number than that. We're out of bits to get any smaller. This also shows why 0.1+0.2 is not equal to 0.3 according to PHP (but also Ruby, Python and Java). Because fraction numbers are calculated by starting from 2^{-1} it means, when a number is smaller then 1, with some few exceptions, they need to sum multiple 2^{-i} to approximate the number. This means the fraction part can only be calculated using 2^{-1} or smaller exponents, which causes the calculated value never to be precise 0.1 but an approximation.

Chapter 6

The determination of the sentiment using naive Bayesian Classifier

To predict if a tweet is positive or negative we need to know $P(pos|T)$ and $P(neg|T)$. We calculate $P(pos|T)$ with the assumption that $P(pos) = P(pos|txt(T))$. We can make this asumption because we only strip out all punctuations and hashtags are considered as normal words.

By using the Multinomial naive Bayesian Classifier[6, 15] we are able to get the probability for the positive and negative value. The Multinomial Nave Bayes classifier is one of the two classic nave Bayes variants used in text classification. The multinomial variant calculates the likelihood for a word or token, in our case unigrams and bigrams. The other variant is the Bernoulli nave Bayes classifier. The Bernoulli variant can be used for multiple features but each one is assumed to be a binary-valued. Bayes' Theorem is often used in text analysis and classification[12]. $P(pos|T)$ is $P(pos|txt(T))$. With $txt(T) =< x_1, ...x_n >$. If, like in this case, we have multiple instances of x where x can be an unigram or a bigram, the equation becomes:

$$P(pos|x_1, ...x_n) = \frac{P(pos)P(x_1, ..., x_n|pos)}{P(x_1, ...x_n)} \tag{6.1}$$

When we take a better look at part $P(x_1, ...x_n|pos)$ we can rewrite this because of the the "naive" conditional independence assumptions. This means that each feature x_i is conditionally indepedent of every other feature x_j for the same category pos or neg.

$$\begin{aligned}
& P(x_1, ..., x_n|pos) \\
&= P(x_1|pos)P(x_2, ..., x_n|pos, x_1) \\
&= P(x_1|pos)P(x_2, ..., x_n|pos) \\
&= \prod_{i=1}^{n} P(x_i|pos) \qquad (6.2)
\end{aligned}$$

Using the naive independency we can rewrite the first equation with this new knowledge. Below we show the equations for both positive and negative probability.

$$P(pos|x_1, ..., x_n) = \frac{P(pos)}{P(x_1, ...x_n)} \prod_{i=1}^{n} P(x_i|pos) \qquad (6.3)$$

$$P(neg|x_1, ..., x_n) = \frac{P(neg)}{P(x_1, ...x_n)} \prod_{i=1}^{n} P(x_i|neg) \qquad (6.4)$$

Because the denominator is the same for both the positive and negative calculation, since it only is the chance on finding these particular features in our training set, we can strip this part of the equation. This results in the equation we can use in determing the probability if a tweet T is positive or negative.

$$P(pos|x_1, ..., x_n) = P(pos) \prod_{i=1}^{n} P(x_i|pos) \qquad (6.5)$$

$$P(neg|x_1, ..., x_n) = P(neg) \prod_{i=1}^{n} P(x_i|neg) \qquad (6.6)$$

6.1 Using multinomial naive Bayesian classifier in predicting sentiment

6.1.1 Abbreviations explained

$$\begin{aligned}
P(pos) &= \frac{|TBGp|}{|TBGp| + |TBGn|} \\
P(neg) &= \frac{|TBGn|}{|TBGn| + |TBGp|} \\
P(ug|pos) &= \frac{pos(ug)}{|TUGp|} \\
P(ug|neg) &= \frac{neg(ug)}{|TUGn|} \\
P(bg|pos) &= \frac{pos(bg)}{|TBGp|} \\
P(bg|neg) &= \frac{neg(bg)}{|TBGn|} \qquad (6.7)
\end{aligned}$$

6.1.2 The determination of the probability using the naive basyes classifier

We use the Multinomial nave Bayesian Classifier [6] in predicting both unigrams and bigrams. Although it could be more elegant to use another formula for the bigrams. In our case we consider the bigram (consisting of two words) as one feature. Using a bigram formula which considers both words separately, so the chance of finding both words together can be predicted, could in theory be more accurate.

Considering the equation $P(pos|T) = P(pos) \prod_{i=1}^{n} P(x_i|pos)$.

$P(pos)$ is called the *a priori*. $P(pos)$ can be calculated by only looking at the prior change of the total set. If a set of words is counted 100 times as positive and 50 times as negative, the chance of randomly selecting a positive word is 2/3. The a priori is used to put a weight on the outcome of the next step in the formula.

The next step in the formula is $P(pos)\prod_{i=1}^{n} P(x_i|pos)$. This means that the chance of finding $x_i, ..., x_n$ given it is positive is multiplied with each other. If you, for example, have the tweet $T = Information\ retrieval\ is\ very\ interesting$ than the equation would become:

$$\begin{aligned} &P(\mathit{information}|\mathit{pos}) \cdot P(\mathit{retrieval}|\mathit{pos}) \cdot P(\mathit{is}|\mathit{pos}) \\ &\quad \cdot P(\mathit{very}|\mathit{pos}) \cdot P(\mathit{interesting}|\mathit{pos}) \end{aligned} \tag{6.8}$$

$$P(ug|pos) \quad = \quad \frac{PosValue(information)}{TotalNumberOfPositiveValues} \tag{6.9}$$

If a word x_i does not occur in our training set for positive sentiment, then obviously we have $P(x_j|pos) = 0$. Note that word w still may occur in the training set for negative sentiment. The effect of a zero probability is that it leads to loss of information, since in that case we will have:

$$\prod_{i=1}^{n} P(x_i|pos) = 0 \tag{6.10}$$

This may be overcome by assuming that each word has some basic occurence probability. For example, each Dutch word has some basic probability of being used by some Dutch speaker. By mixing this basic probability with the probability obtained from the training set, we avoid these zero probabilities. In practice, this basic probability may not be known of too complex to obtain. In such cases, a rough approximation is taken for basic probability. For example, we give a low value of 0.01 divided by the total number of negative values to the occurence probability to words that are not encountered in the training set.

6.2 Using multinomial naive Bayesian classifier by classifying a tweet

If we consider the following values (and no other values in the unigram table): The first thing we need to calculated if we want to know if the tweet is positive, is the a priori, or the $P(pos)$ value. $P(pos)$ can be calculated by dividing the sum of positive by the sum of both positive and negative.

Unigram	**Positive**	**Negative**
information	15	10
retrieval	11	9
is	10	10
very	10	12
interesting	35	2
	81	**43**

$$\begin{aligned} P(pos) &= \frac{\sum PositiveValues}{\sum PositiveValues + \sum NegativeValues} \\ &= \frac{81}{81+43} \approx 0.6532 \end{aligned} \tag{6.11}$$

We then need to calculate the $\prod_{i=1}^{n} P(x_i|pos)$. $P(x_i|pos)$ can be calculated using $pos(ug_i)/|TUGP|$, so for each word we need to know positive value and divide that value by the total positive values.

$$\begin{aligned} \prod_{i=1}^{n} P(x_i|pos) &= \frac{15}{81} * \frac{11}{81} * \frac{10}{81} * \frac{10}{81} * \frac{35}{81} \\ &= \frac{192500}{1162261467} \approx 0.00016562538 \end{aligned} \tag{6.12}$$

We now know both values of the equation and we only need to multiply them.

$$\begin{aligned} P(pos) \prod_{i=1}^{n} P(x_i|pos) &= 0.6532 * 0.00016562538 \\ &\approx 0.00010818649 \end{aligned} \tag{6.13}$$

Knowing the value for the positive probability doesn't mean anything if we don't know the value for the probability that it's a negative tweet. So we also need to calculate the value for the negative probability.

$$\begin{aligned} P(neg) = 1 - P(pos) &= \frac{\sum \mathit{Negative Values}}{\sum \mathit{Negative Values} + \sum \mathit{Positive Values}} \\ &= \frac{43}{81+43} \\ &\approx 0.3467 \end{aligned} \tag{6.14}$$

$$\begin{aligned} \prod_{i=1}^{n} P(x_i|neg) &= \frac{10}{43} * \frac{9}{43} * \frac{10}{43} * \frac{12}{43} * \frac{2}{43} \\ &= \frac{21600}{147008443} \\ &\approx 0.000146930332 \end{aligned} \tag{6.15}$$

$$\begin{aligned} P(neg)\prod_{i=1}^{n} P(x_i|neg) &= 0.3467 * 0.000146930332 \\ &\approx 0.00005094074 \end{aligned} \tag{6.16}$$

If we then compare both values for positive and negative, we find out that positive is greater than negative. This means that the probability the tweet is positive is higher than the probability that the tweet is negative.

6.3 Validation of used model to determine sentiment

The training set is a set of tweets written in Dutch. The approach in gathering data to analyze and setup a training set is to collect random tweets which used the word “het”, a typical Dutch word and not very common in other languages. This way we are able to gather Dutch tweets only, on which we will focus. The control set is gathered the same way, but in a different time period.

Statistics training set		Statistics control set	
Total tweets	10.000.000	Total tweets	4.500.000
Emoticons positive	176.255	Emoticons positive	25.000 (limited)
Emoticons negative	95.820	Emoticons negative	25.000 (limited)

Gathered between	Gathered between
2013-09-17 00:00:20	2013-10-30 00:49:13
2013-10-30 00:49:12	2013-11-20 14:14:47

6.4 Test method and accuracy of the sentiment mining model

To measure the accuracy of unigrams we take the training set of tweets to train our system and another set to test our setup, called the control set. To test the accuracy without any distortion or overlap of the two sets we enforced that tweets that are used in the training set are different than tweets in the control set:

$$\textit{Training set} \cap \textit{Control set} = \varnothing$$

The control set consists is limited to the first of 25000 tweets with a positive emoticon and 25000 tweets with a negative emoticon. This limitation is because using more would probably not give another view on the accuracy, but would increase the computational time. To compute the accuracy of these 25000 tweets for both negative and positive takes around 30 min.

The emoticon is stripped out of the tweet, the system will than define the probable sentiment of the tweet, so it determines if a tweet should have contained a positive or negative emoticon. We measure the times the system is wrong. To select 25000 tweets containing a negative emoticon, we use the same way as the tweets are selected in the trainings model. Each tweet will than go through the system to determine whether it is a positive or negative tweet, according to our learning set. If a tweet is called positive, while we only selected negative tweets, we count these false hits.

To show the accuracy stabilizes while adding more tweets to the training set, we did check for accuracy for different amount of tweets in the training set.

6.5 Accuracy unigrams of bigrams in sentiment mining

6.5.1 Accuracy of unigrams in sentiment mining with emoticons

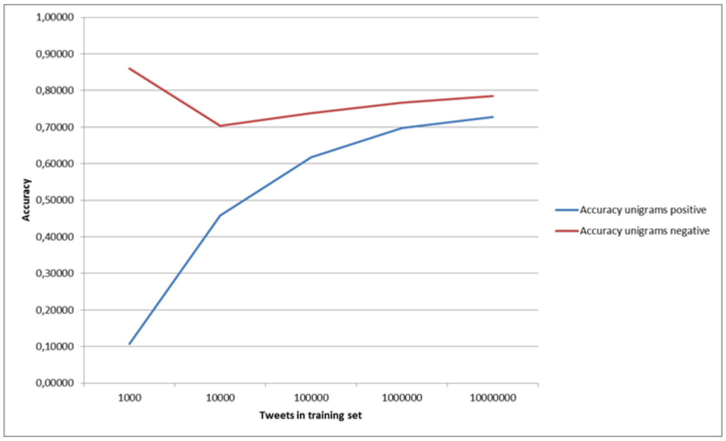


Figure 6.1: Accuracy of unigrams with an increasing training set of tweets

The descend of the line in the graph between 1000 and 10000 tweets training set can be explained because of the probability a word is only known for one sentiment, like positive or negative. If a word is not known for a particular sentiment a small value is taken, 0.01, and this value is divided by the total number of either positive or negative. This means with a small training set it has a relatively large impact.

6.5.2 Accuracy of bigrams in sentiment mining with emoticons

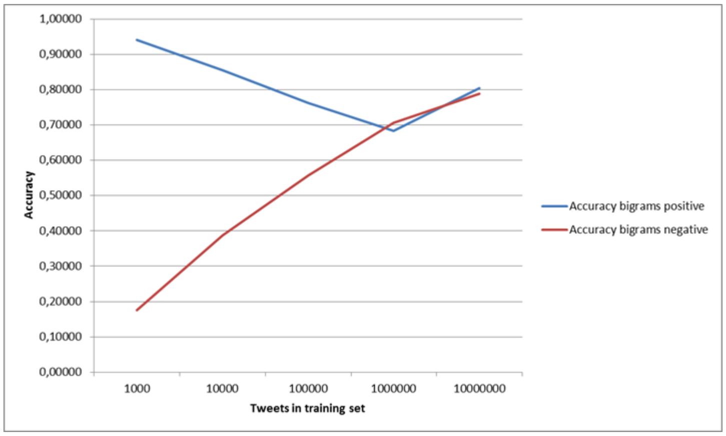


Figure 6.2: Accuracy of bigrams with an increasing training set of tweets

Bigrams seem to increase in accuracy from around 1.000.000 tweets and start to flatten when more than 10.000.000 tweets are used. Bigrams show the same way of decreasing accuracy for positive sentiment. This is due the same reason as discussed with unigrams.

6.5.3 Comparison between unigrams and bigrams

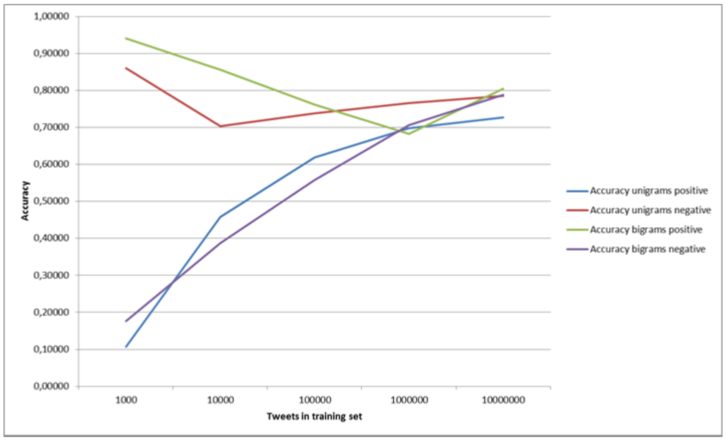


Figure 6.3: Accuracy of unigrams and bigrams in one graph

Chapter 7

The prediction of TV ratings using Twitter and sentiment

7.1 The training set and the meaning of the data in predicting TV ratings

To predict the TV ratings we use the collected data about the program which we want to calculate. The tweets we use to calculate the ratings are gathered from a 2-hour period surrounding the broadcast of the program. In this case the GTST is broadcasted from 20:00 to about 20:30. We use the tweets in our calculation sent in the period from 19:30 to 21:30. In order to counteract noise in the data, all retweets are removed. Probably retweets will also have a share related to the TV ratings, but in our case we remove them to create less noise. When we plot the gathered data as a scatter chart, we can roughly see if there is any connection between the tweets and the ratings. Even though there is a significant outlier in terms of ratings and tweets, there does seem to be a relation between the two values.

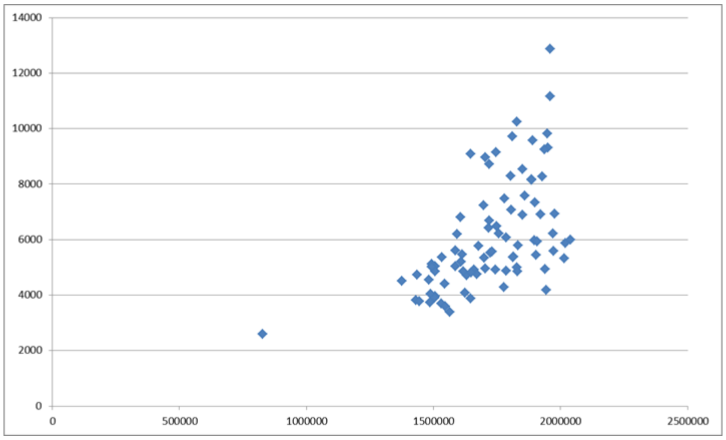


Figure 7.1: A scatter plot of the number of tweets and the TV ratings (on GTST)

This chart shows the number of tweets (y-axis) and the TV ratings (x-axis).

7.2 Test method and accuracy of the TV rating predictions

To measure how well our method performs, we use the root mean square error (RMS) formula . RMS is a statistical measure of the magnitude of a varying quantity. The RMS error is a frequently used measure of the differences between values predicted by a model and the values actually observed. The lower the value of RMS, the better the method performs. In the formula the x is the real TV ratings and the y are the predicted TV ratings.

$$x_{rms} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - y_i)^2}{n}} \tag{7.1}$$

7.3 Training set and control set of the TV ratings prediction

The training set consists of 86 measurements of TV ratings and tweets about GTST. Although RTL Nederland provided us the TV ratings of one year, we were only able to gather tweets about 107 broadcastings. We use these data in our different methods to determine which method is the most accurate. We need to take into account that the 'actual ratings' are also gathered using statistics as described before. This could, theoretically, mean that our prediction is more accurate. Because of the limitation of the measurements we use only 21 measurements in our control to check the accuracy of our method. This means that the variables from the training set are extracted from 86 measurements and the control set has a total of 21 measurements.

7.4 Used abbreviations in predicting TV ratings

twts	= Total number of tweets
tvr	= Total number of TV ratings
twtsp	= Number of tweets for the predicted date p
tvrmin	= The minimum number of TV ratings
tvrmax	= The maximum number of TV ratings
twtsmin	= The minimum number of tweets
twtsMax	= The maximum number of tweets
tvrdMin	= The minimum TV ratings of that day in the week in history day by p
tvrdMax	= The maximum TV ratings of that day in the week in history day by p
twtsdMin	= The minimum tweets of that day in the week in history day by p
twtsdMax	= The maximum tweets of that day in the week in history day by p

7.5 Used formulas in our TV rating prediction methods

We test two different formulas in predicting TV ratings. Both formulas have a different approach in the way they predict the TV ratings using the tweets. The first one uses a weighted growth and the other one uses a linear growth. We test both methods to get to know which one is the best in predicting TV ratings.

7.5.1 Weighted growth using history of TV ratings and tweets

The first formula is using the lowest amount of tweets and TV ratings and the highest amount of tweets and TV ratings in history. The graph below shows the formula as it works. On the x-axis are the tweets, on the y-axis are the TV ratings. The formula uses the principle that as TV ratings increase, tweets increase faster. This is a kind of weighted growth, but with a maximum that was derived from the maximum number of tweets and TV ratings.

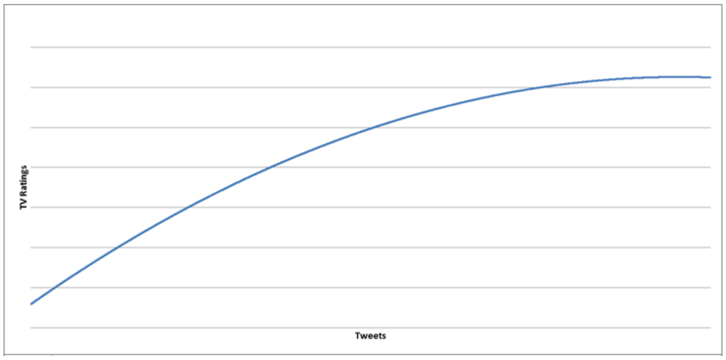


Figure 7.2: Weighted growth using history data (x-axis tweets, y-axis TV ratings)

7.5.2 Least squares method using history of TV ratings and tweets

The least squares method is a common method in predicting. The method can be used to predict a trend in a set of values. See the example below, where the trend is calculated using the least squares method and plotted into the chart.

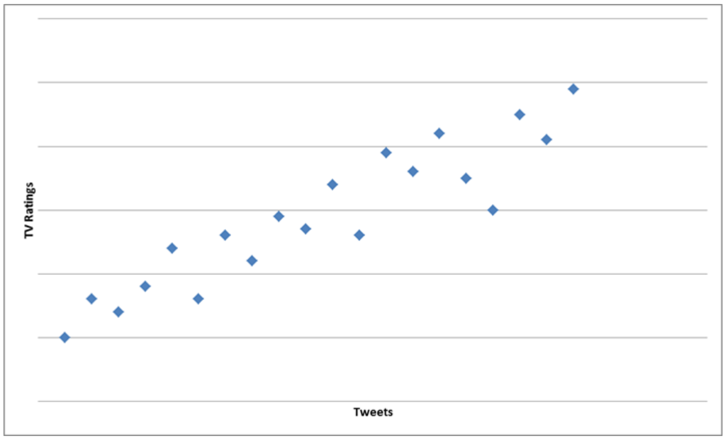


Figure 7.3: Linear growth using history data (example)

The formula used in determining the least squares for the prediction of TV Ratings: $y = \alpha + x\beta$

$$\beta = \frac{\sum(x - \overline{x})(y - \overline{y})}{\sum(x - \overline{x})^2} \tag{7.2}$$

$$\alpha = \overline{y} - \beta\overline{x} \tag{7.3}$$

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_n \tag{7.4}$$

$$\overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_n \tag{7.5}$$

7.6 Method 1: All tweets about GTST together

In our first method, we look at the results of a simple but seemly effective approach. We draw a line between the largest values of ratings and tweets and the smallest values of ratings and tweets over the measured period. It is noteworthy that when you share the highest ratings by the highest number of tweets this yields a smaller number than if you share the lowest number of audience by the lowest number of tweets. This difference we call the KT-factor. Note that the highest values do not need to share the same day but are purely an estimator for the maximum number of values. The KT-factors lowers if the number of tweets increases.

In order to make use of the KT-factor for optimal use, we have established a formula that has a weighted growth, on the basis of the number of tweets.

$$Predicted\,Ratings_p = twts_p(\frac{tvr^{Min}}{twts^{Min}} - (twts_p - twts^{Min}\frac{\frac{tvr^{Max}}{twts^{Max}}}{twts^{Max} - twts^{Min}})) \tag{7.6}$$

7.6.1 Method 1: Result using all tweets about GTST

The predicted values (red) and the real TV ratings (blue) are plotted in the same graph to show the difference. The graph looks like this:

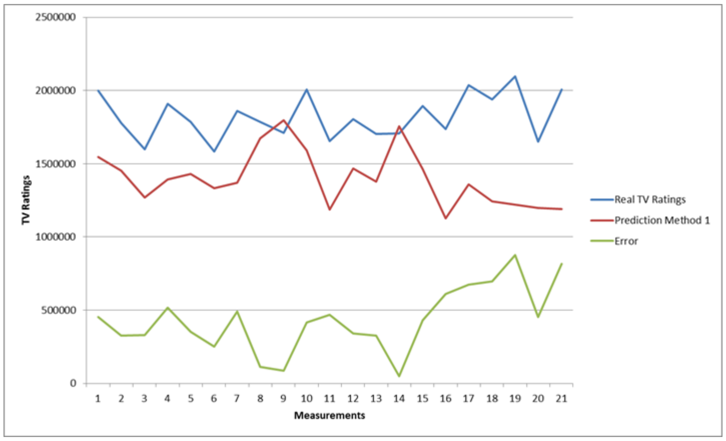


Figure 7.4: Result using all tweets about GTST (weighted growth)

Root-mean-square error: 482586

The RMS error value is clear about the performance of this method. It has a very high value. That means it does not approache the real TV ratings. This could be related to the fact that the viewing ratings in the training set were overall higher than in the control set.

7.7 Method 2: Counting number tweets about GTST per day of the week

In the first method we took the entire data set for the calculation of the values. In this method we use the day of the week in our calculation. Namely the relationship between the tweets and ratings vary by day of the week. On Mondays, the KT-factor is ≈ 352, while this is on a Friday ≈ 461. In other words, for a tweet on Monday there is an average of about 352 viewers while there is an average of about 461 viewers per tweet on Friday.

This difference about tweets and TV Ratings per day of the week we take into account in the next formula. This means that we do not calculate the minimum and maximum of the total, but the minima and maxima of the day. The table shows the differences in statistics per day measured in our data set.

Day	minTweets	maxTweets	minTvr/minTwts	maxTvr/maxTwts
Monday	4269	8925	352.7758	228.4593
Tuesday	2294	8116	360.9415	242.9768
Wednesday	3378	10335	427.4718	189.6468
Thursday	3299	7689	435.2834	234.6208
Friday	2983	6372	461.2805	317.0119

The formula will slightly change with respect to the one used in method 1, where the day of the week is used in the calculation of the min and max of the tweets and TV ratings.

$$Predicted\,TV Ratings_p = twts_p(\frac{tvr_d^{Min}}{twts_d^{Min}} - (twts_p - twts_d^{Min}\frac{\frac{tvr_d^{Max}}{twts_d^{Max}}}{twts_d^{Max} - twts_d^{Min}})) \tag{7.7}$$

7.7.1 Method 2: Result using tweets about GTST per day of the week

The predicted values (red) and the real TV ratings (blue) are plotted in the same graph to show the difference. The graph looks like this:

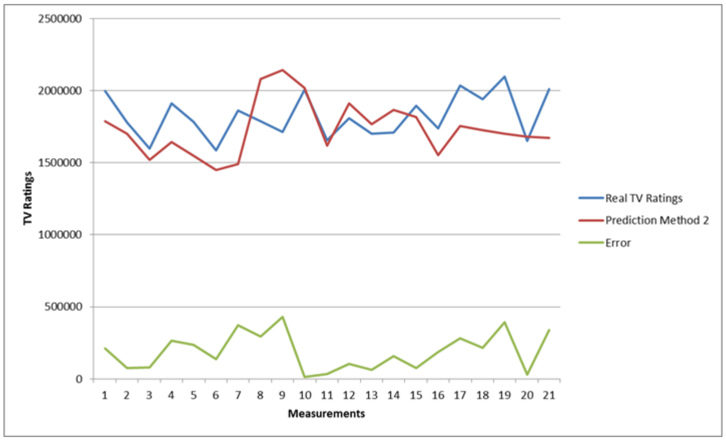


Figure 7.5: Result using tweets about GTST per day of the week (weighted growth)

Root-mean-square error: 227823

Where method 1 seemed to perform worse with the formula used, method 2 shows a better result. With the use of the day of the week the RMS error is decreased by a factor two, which means you could say this method performs twice as good as the previous one.

7.8 Method 3: Least squares regression on the total set of data

In the first next method we try to get the most accurate prediction of the TV ratings is using the tweets sent by users to create a formula by using the least squares method. In this case we only count the amount of users who tweeted about GTST, in later test method we will find out that this method generates an almost equal Pearson correlation as with using just the total amount of tweets sent.

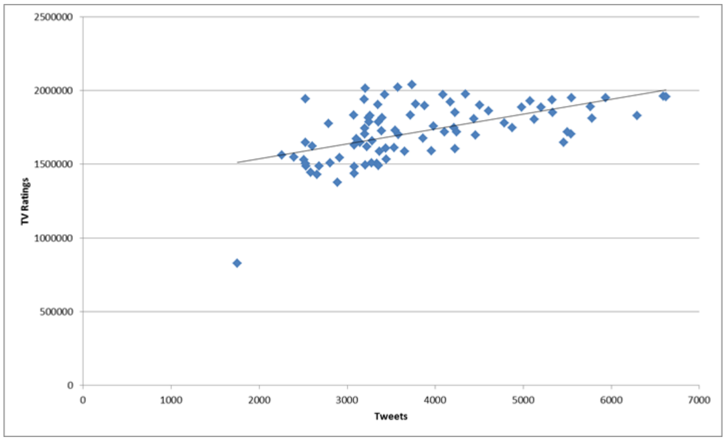


Figure 7.6: Visual presentation of the used least squares method

The outlier is 30th of April 2013, the day the king of the Netherlands was crowned.

To calculate the linear formula that can be used to predict the TV Ratings we use the standard formula for calculating the least squares:

$$\Delta t_k = twts_k - \overline{twts} \tag{7.8}$$

$$\Delta r_k = tvr_k - \overline{tvr} \tag{7.9}$$

$$\beta = \frac{\sum_{k=1}^{n} \Delta t_k * \Delta r_k}{\sum_{k=1}^{n} \Delta t_k^2} \tag{7.10}$$

$$\alpha = \overline{tvr} - \beta * \overline{tvr} \tag{7.11}$$

$$Predicted\ TV\,Ratings_p = \alpha * twts_p + \beta \tag{7.12}$$

7.8.1 Method 3: Result using least square method on total set

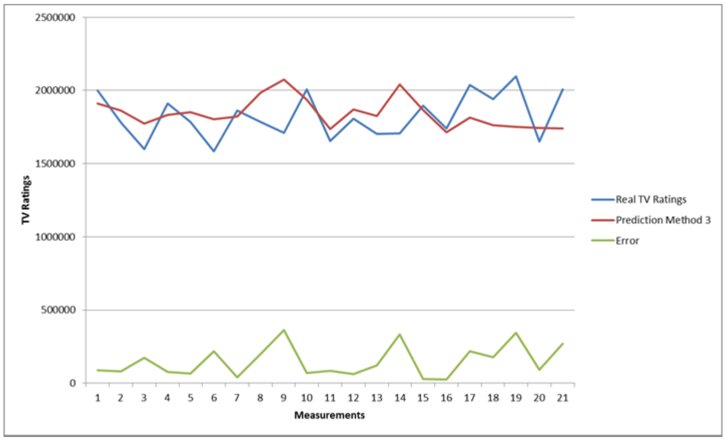


Figure 7.7: Result using least square method on total set

Root-mean-square error: 182362

The graph is smoother than the ones before with the other method. This can be explained since this method uses the least squares, while the other method uses the minimum and maximum of both the tweets and TV ratings as a starting point. Although the Pearson coefficient is not as high in method 1, the method looks very promising. In method 4 we will use the same approach as in method 2, by calculating the α and β for a particular day.

7.9 Method 4: Least squares analysis per day of the week

When using the same formula as in method 4, but focusing on the days of the week, we get the a even better result. If we take a look at that scatter plot for the tweets and TV ratings on Tuesday we see (the X-axis are the amount of tweets) and draw a line in it using the least squares method we see:

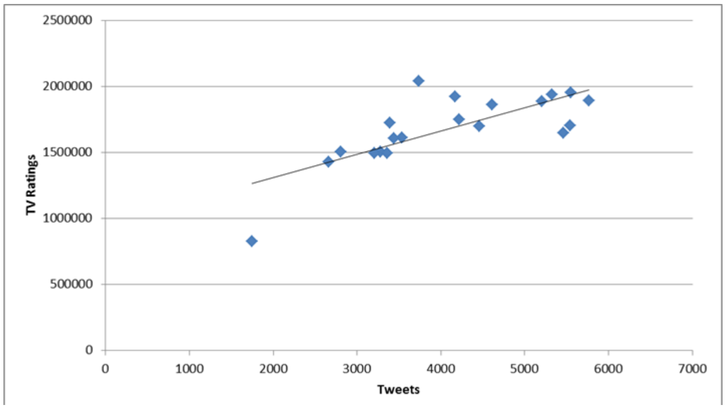


Figure 7.8: Visual representation of least squares analysis per day of the week

To calculate the linear formula that can be used to predict the TV Ratings for a particular day d we use:

$$\Delta t_{d,k} = twts_{d,k} - \overline{twts_d} \tag{7.13}$$

$$\Delta r_{d,k} = tvr_{d,k} - \overline{tvr_d} \tag{7.14}$$

$$\beta_d = \frac{\sum_{k=1}^{n} \Delta t_{d,k} * \Delta r_{d,k}}{\sum_{k=1}^{n} \Delta t_{d,k}^2} \tag{7.15}$$

$$\alpha_d = \overline{tvr_d} - \beta_d * \overline{twts_d} \tag{7.16}$$

$$Predicted\ TV Ratings_p = \alpha_d * twts_p + \beta_d \tag{7.17}$$

7.9.1 Method 4: Result using least square method per day of the week

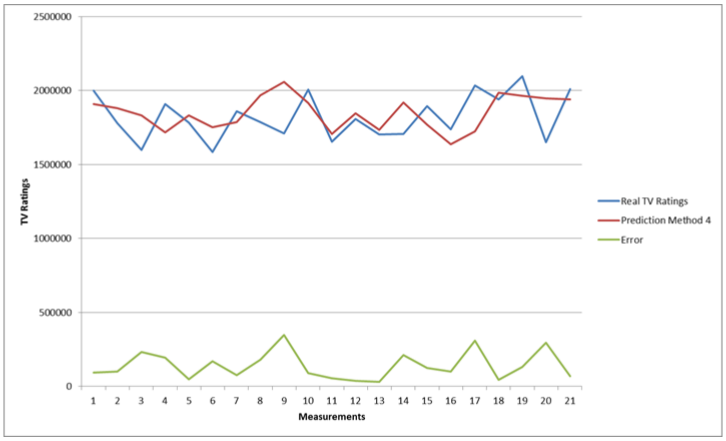


Figure 7.9: Result using least square method calculated per day of the week

Root-mean-square error: 167912

The RMS error slightly lowered when using the day of the week in the prediction of the TV ratings. Although it is not a significant improvement, it still is better than the use of the total set.

7.10 Method 5: Using sentiment data of each day of the week

In method 5 we continue using the least squares method, since its performance are already quite good, but with the addition of the number of positive and negative tweets. Since unigrams and bigrams perform almost the same we used unigrams to determine the sentiment. Using both unigrams and bigrams would lead to a computational problem, since we then would need to examine the sentiment for all tweets we use for our prediction of TV ratings. The reason to choose for unigrams is because its perfomance is equal to the perfomance of the bigrams. The fact that the unigrams table used for the prediction is much smaller than the one of the bigrams, is the reason to finally choose for unigrams. Using bigrams would probably significantly improve the time it takes for predicting the TV ratings.

Using our sentiment mining system to determine whether a tweet is positive or negative we get three columns with either positive, negative or undefined. The undefined tweets are left out of the data that is used to determine the predicted TV ratings. Because the approach to use the day of the week showed a better result, we use the same approach in this method.

The formula to predict the TV ratings particular day d and the particular sentiment negative n:

$$\Delta t^-_{d,k} = twts^-_{d,k} - \overline{twts^-_d} \tag{7.18}$$

$$\Delta r^-_{d,k} = tvr^-_{d,k} - \overline{tvr^-_d} \tag{7.19}$$

$$\beta^-_d = \frac{\sum_{k=1}^{n} \Delta t^-_{d,k} * \Delta^- r_{d,k}}{\sum_{k=1}^{n} \Delta t^-_{d,k} 2} \tag{7.20}$$

$$\alpha^-_d = \overline{tvr^-_d} - \beta^-_d * \overline{twts^-_d} \tag{7.21}$$

$$Predicted\ TV Ratings_p = \alpha^-_d * twts^-_p + \beta^-_d \tag{7.22}$$

In this approach the tweets first go through the system of analyzing the sentiment. All tweets are than separated using the system into positive, negative and undefined. Using either positive or negative in our least square method, we get the values β_{dn} and α_{dn} for the particular day we are looking for. We can then use this historical data to predict the TV Ratings but multiplying the amount of tweets with α_{dn} and then add β_{dn} to it.

7.10.1 Method 5: Result using positive sentiment to predict TV Ratings

When we use the positive values to determine the TV ratings using the least square method, we get the following result:

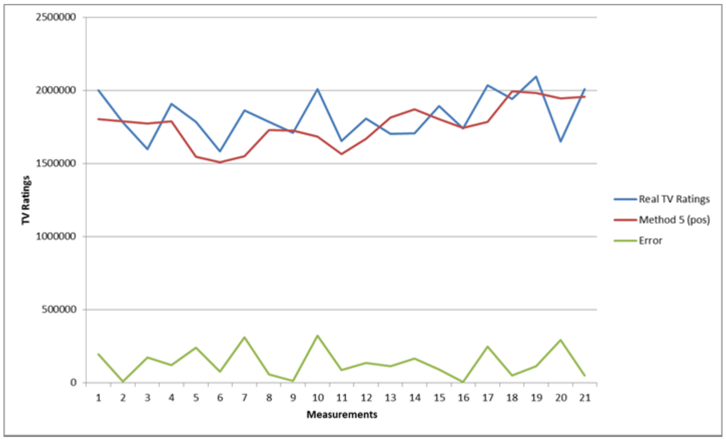


Figure 7.10: Result using positive sentiment with least squares method

Root-mean-square error: 165599

This method shows a low RMS error value, which means it is close to the actual measurement. If you take a look at the graph it also shows that it follows the trend of the TV ratings

7.10.2 Method 5: Result using negative sentiment to predict TV Ratings

When we use the negative values to determine the TV ratings using the least square method, we get the following result:

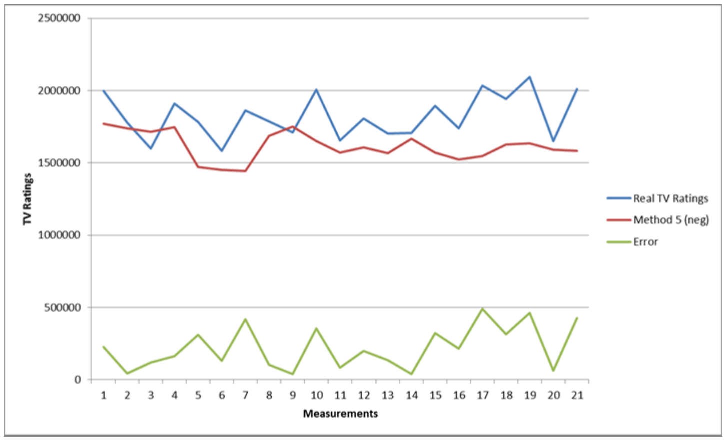


Figure 7.11: Result using negative sentiment with least squares method

Root-mean-square error: 264472

This graph and RMS error value shows that negative sentiment in comparison with positive sentiment does not perform better, but positive sentimented tweets do perform better in predicting TV ratings.

Chapter 8

Discussion

8.1 Sentiment mining using emoticons

We used emoticons in our method for sentiment mining. The downside of using emoticons is that it is binary . Binary because it only gives the possibility to distinguish positive and negative tweets. Tweets that are neutral cannot, or not good, be extracted using emoticons.

The advantage of the use of emoticons is that it can be used for almost every language. The only important thing is that the collected tweets used for sentiment mining are specific for that language and only apply to the language that you want to use them for. In our case we have chosen the word "het" because this is a specific Dutch word that is rare in other languages.

The results show that the increase of the data set (training set) provides a significant improvement in the accuracy of the prediction . This shows that when better results want to be achieved this can not only be achieved by improving the method of the probability model, but also by increasing the data set used for training.

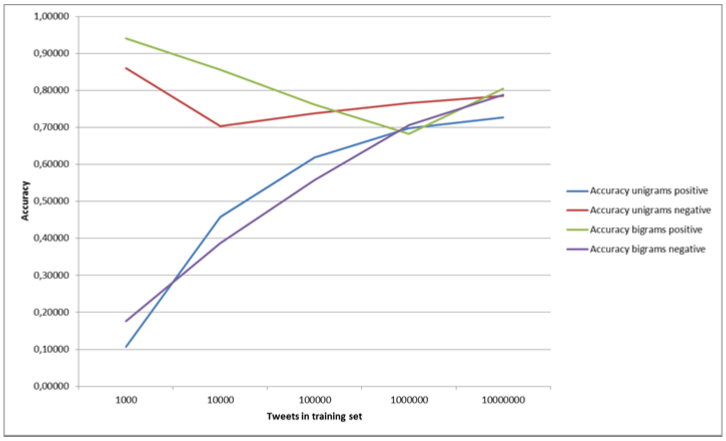


Figure 8.1: Accuracy of unigrams and bigrams

Our system seems to compete with other researches like Using Emoticons to reduce Dependency in Machine Learning Techniques for Sentiment Classication[9]. That study got a 70% performance in the predicting of the sentiment using the emoticon-trained classifier. With an even larger training set the accuracy would probably pass the 80%, which means it is a promosing technic in classification.

8.2 Predicting TV ratings using twitter

Predicting TV ratings by the use of tweets appears feasible. The intensity of the peaks is a problem, but the overall trend is being followed in the prediction. Improving the prediction for this peaks would considerably improve the overall performance of the method. Nevertheless, the prediction values follow the real number very accurately. However, it should be noted that the 'real' TV ratings are also a statistical prediction.

Because of the limitation of measurements, in particular the tweets collected about the broadcastings, we are limited to only 107 measurements in total. This limitation means we cannot test our method on a evenly large set of TV ratings and tweets that not have been used in our training set. Though the results we found do give a view of the capabilities of our methods, but we need a larger set to get an even better view.

8.3 Predicting TV ratings using the sentiment on twitter

Using the sentiment in the prediction of the TV ratings does gives better results, according to the methods we use. With even a larger training set, we could even get better results in predicting TV ratings using also the sentiment of the tweets. The reason why positive sentimental tweets perform better when predicting the TV ratings is unclear. It could be that positive tweets are sent more easily when tweeting about a certain subject, but this would only be guessing at the moment.

8.4 Comparison of root-mean-square error values of the different methods

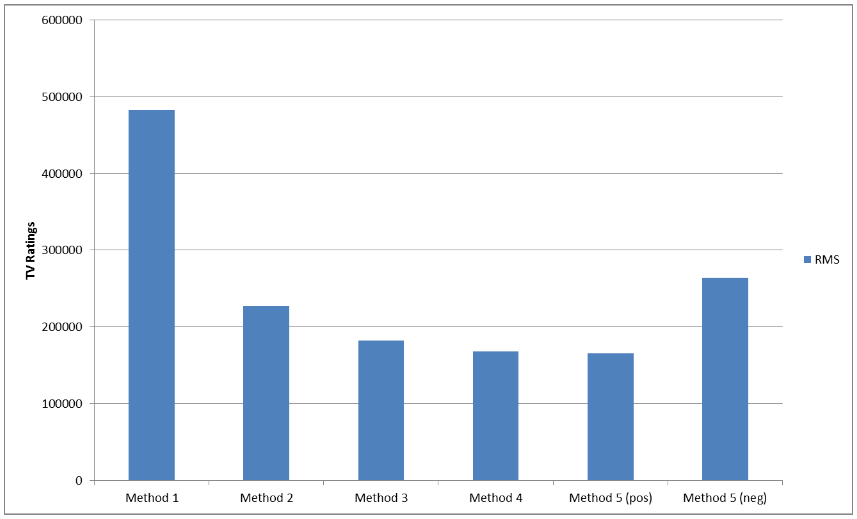


Figure 8.2: RMS error values for each method

Method 5 with positive sentiment gives the lowest root-mean-square error value, this means it has the closest prediction to the real TV ratings. Thus using the least square method with positive sentimental tweets and the historical data of Twitter and TV ratings for that day of the week, gives the best result in predicting TV ratings using Twitter.

8.5 Performance of prediction with different tolerance

If we take a look at the graph that shows the different accuracy values in predicting the TV ratings with different tolerances, we see that method 3 and 5 positive looks like the overall best performers. Method 5 with positive sentimental tweets has the lowest RMS error value and it's performance with different tolerances are the best at low tolerance values. We also added the prediction when we just use the average of the total TV ratings in our prediction without using any tweets, this means that the prediction for a day is always the same as the average of the training set. This shows that 3, 4 and 5 (positive) perform better than using just the average of the training set as a prediction.

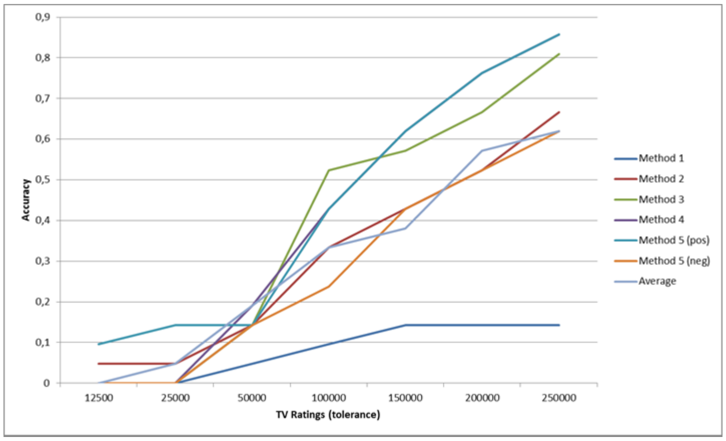


Figure 8.3: Prediction accuracy of TV ratings

Chapter 9

Future work

9.1 Statistical relevance of tweets

As watching TV evolves, new technics need to come into place to get a good overview of the market. Delayed viewing, either via the computer, table or smart TV, is growing by the day, creating the necessity of adopting new technics in the determination of the TV viewers and statistics. Twitter can be a good medium in getting a good overview of the programs being watch, either broadcasted live or viewed delayed. Using statistical data done in earlier research about the Twitter population can create a model which enables to use tweets in the determination of the viewers group in terms of age, sex and interests.

The use of GTST in our research may be determinative in the way the methods perform. Using more data of more TV programs with different viewing groups in age, sex and education could create a uniform method for predicting TV ratings using Twitter. In our research we only showed that with some deviation, TV ratings can be predicted.

9.2 Better prediction in following the spikes

The prediction model is able to predict the TV ratings with a some uncertainty. In most cases it follows the actual TV ratings, but in some cases there are spikes that are not followed in the prediction model. These spikes can be the result external factors like interfering TV programs or football games, but also the weather could be influencing the TV ratings.

9.3 Influence of external factors on tweets and TV ratings

External factors can influence the way people watch a program or tweet about their the program. A good starting point would be to take the weather forecast of each day and research if there is any correlation between the weather and TV ratings. The predicted amount of rainfall, together with the chance of rainfall, could possibly be used to find a correlation between the TV ratings and twitter. Research done by Nielsen, as discussed in chapter 2.3, already shows that bad weather or storms affect TV ratings , it could possible also correlate to tweets and TV ratings. Using this information in the prediction model can be useful in predicting the spikes that are currently hard to predict.

Football matches also influences viewing rating on other programs. Important matches like the Champion Leagues show to significantly influence the TV ratings of other programs. By using this information in the model, the prediction can become more accurate.

9.4 Using Twitter in expanding territory

Twitter itself showed already it is interested in using it's medium as a part of expanding other territories. In May of 2013 they released the tool Twitter Amplify, which can be used to target specific users with a specific tweet when a TV ad is broad casted[1].

Using the sentiment mining model in predicting the current sentiment and than using tools like Twitter Amplify could benefit in the right way of targeting users at a specific time when they are in a certain mood.

[1] http://advertising.twitter.com/2013/05/Amplify-TV-commercials-on-Twitter-Premiering-TV-ad-targeting.html

Chapter 10

Conclusion

10.1 Conclusion on sentiment mining

The results of the sentiment mining model shows a significant improvement with an increasing training set. Increasing the training set with another logarithmic factor would probably flatten the growth and stabilize the accuracy, it would probably not significantly improve the accuracy anymore.

The results with an accuracy of around 80% in sentiment mining can compete with previous studies [4] [2] which managed to get a 71.35% accuracy rate with a slightly different approach. Using other variables would probably improve the accuracy, but could also lead to a computational problem since tweets must be classified one by one.

Using emoticons is a way to classify tweets and build a large training set. Building such a large training set by hand would require a lot of hours. The use of emoticons simplifies this process and enabled us to build a very large training set. Downside of the use of emoticons that is it close to binary. Emoticons are either positive or negative, neutral emoticons are rare. This means the model can only be used to classify positive or negative tweets.

10.2 Conclusion on predicting TV Ratings

Predicting the exact TV ratings are difficult because of substantial differences in spikes of TV ratings and tweets, thereby the changing difference in sentiment in tweets. However, prediction is possible when you take a bit of uncertainty into account. The trend of TV ratings is followed in our prediction model, but the intensity may vary. Using more variables, like weather or competing TV programs, could possibly positively influence the prediction.

The most accurate prediction is done by method 5 with positive sentiment. It also shows the lowest root-mean-square error rate and therefor also the highest accuracy in predicting the TV ratings with different tolerances.

List of Figures

Bibliography

[1] Chen, R and Lazer, M (2011) *Sentiment Analysis of Twitter Feeds for the Prediction of Stock Market Movement.*

[2] Agarwal, A., Xie, B., Vovsha, I., Rambow, O., and Passonneau, R. (2011) *Sentiment Analysis of Twitter Data.*

[3] Bollena, J., Mao, H., and Zeng, X. (2011). *Twitter mood predicts the stock market. Journal of Computational Science.*

[4] Davidov, D., Tsur, O., and Rappoport, A. (2010). *Enhanced Sentiment Learning Using Twitter Hashtags and Smileys.*

[5] Go, A., Bhayani, R., and Huang, L. (2009). *Twitter Sentiment Classication using Distant Supervision.*

[6] McCallum, A., and Nigamy, K. (1998). *A Comparison of Event Models for Naive Bayes Text Classification.*

[7] Pak, A., and Paroubek, P. (2010). *Twitter as a Corpus for Sentiment Analysis and Opinion Mining.*

[8] Pear Analytics. (2009). Retrieved 09 16, 2013, from *http://www.pearanalytics.com/wp-content/uploads/2012/12/Twitter-Study-August-2009.pdf*

[9] Read, J. (2005). *Using Emoticons to reduce Dependency in Machine Learning Techniques for Sentiment Classication.*

[10] Roberts, K., Roach, M., Johnson, J., Guthrie, J., and Harabagiu, S. (2012). *EmpaTweet: Annotating and Detecting Emotions on Twitter.*

[11] Stone and Pennebaker. (2003). *Words of Wisdom: Language Use Over the Life Span.*

[12] Tan, S., Cheng, X., Wang, Y., and Xu, H. (2009). *Adapting Naive Bayes to Domain Adaptation for Sentiment Analysis.*

[13] Twitter Statistics. (2013, May 7). Retrieved June 3, 2013, from StatisticBrain: *http://www.statisticbrain.com/twitter-statistics/*

[14] Wang, W., Chen, L., Thirunarayan, K., and Sheth, A. (2012). *Harnessing Twitter 'Big Data' for Automatic.*

[15] Zhang, H. (2004). *The Optimality of Naive Bayes.*