Magnus L. Kirø May 18, 2014

Sentiment analysis of Tweets in correlation with financial investments

Work in progress,
to be completed by 1. jun 2014.
https://github.com/magnuskiro/master

Masters Thesis,

Artificial Intelligence Group
Department of Computer and Information Science
Faculty of Information Technology, Mathematics and Electrical Engineering

NTNU – Trondheim
Norwegian University of
Science and Technology

Abstract

Background: As Twitter has become a global microblogging site, its influence in the stock market has become significant. This makes tweets an interesting medium for gathering sentiment. A sentiment that might influence trends in the stock market.

Motivation: If twitter can be used to predict trends in the stock market the casual investor would gain an advantage over the day-trader or the modern trading algorithms.

Another interesting aspect is the role of twitter in sentiment analysis. And how twitters role as a data source influences trends in the stock market.

Methods and experiments: Twitter is used as the data source. It provides easy access, lots of data, and many possibilities to utilise the available metadata.

To improve and verify the sentiment classification and trend comparisons we use a variation of methods. Simple statistical methods, such as counting positive and negative words. More advanced methods such as part of speech and other NLP related magic.We also explore the use of mete data such as location and language tags.

Results: Rough results of my research.

Conclusion: All OK ? No?

Acknowledgements

Many thanks to the people ..... bla bla bla.
TODO Arivd + Pinar TODO add proof readers.

Metadata

Metadata ?

Typically repositories, links to code, file downloads, websites etc. Maybe contact info.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 What

TODO write what this thesis contains and what is the goal of the thesis.

What has been done and what was going to happen. What is this thesis about? What are we doing? What are the goals of this thesis? What is the setting for this thesis, the circumstances and environment of the work.

1.2 Why, Motivation

TODO write why I want to do this and why we want to look at these specific points.

Why we do this and the motivation we have for doing this. Why is this work done? Why do we benefit from this? Why do I want to do this? Why is this relevant for others?

1.3 Research questions

1.3.1 How do we determine the sentiment of a tweet?

Can we extract knowledge from tweets to find a sentiment?

We will look at the usefulness of tweets as a way to extract sentiment.

Which parts of a tweet is useful for the classification of a tweets sentiment?

Which methods are best to classify tweets?

How do we best find the sentiment of tweets?

1.3.2 How can twitter be used to aggregate a trend?

Can we build a trend based on information from tweets?

Can Twitter as a microblogging site be used as a data source in aggregation of trends.

Credibility, what sort of credibility level has to be attained to certify the quality of the trend prediction.

Which parts of twitter are most useful to generate a trend?

1.3.3 How does trends on twitter compare to technical analysis in the stock market?

Technical analysis compared with the tweet trend.

We will look at possible applications for the sentiment in the stock market.

Which twitter sources are most suitable for predicting the stock market trend?

In finance, the moving average is a result of technical analysis. This and other trend defining qualities of financial data is used to compile trends.

Twitter has data such as the amount of tweets posted today, the location where tweets are posted from, and which users has posted. Aggregated, these data become represents a trend.

Previously researchers have managed to predict direction of the market the next few days based on the volume of tweets.

We are interested in the correlation between trends on twitter and the moving average in finance. Hopefully this will give some insight of how the sentiment on Twitter influences the sock market.

1.4 Findings

TODO briefly outline what we have found in this thesis. What we figured out in this thesis.

1.5 Outline

TODO write where stuff are in this thesis. Should be short. The outline of the document and the description of what which part is about.

Chapter 2

Background and Previous Work

TODO write a summary of the newest techniques and inventions in the field of twitter research related to finance.

2.1 Twitter

Twitter is a social and information network. It's a real-time service for sharing and gathering small messages. These messages can represent everything form a persons opinion of ice cream, to the latest changes in the financial market or pictures from a Mars rover.

At the core of Twitter you have the Tweet. The Tweet is the 140 character message. These small pieces of information combined are the life line of Twitter. Tweets lets you communicate with other users, share photos and post all kinds of information. The small size of the tweets are not a hindrance for the flow of information. [1]

The fast growing messaging service handles 1.6 billion search queries every day. As of 2012 the 500 million users would generate 3.2 queries each on any given day. 340 million tweets were posted every day. [2]

Most medium and large companies have a presence on Twitter today. Posts can contain any type of information, from promotional content to service status to financial reports. [Jubbega, 2011, p8] says that 77 of the Fortune 100 companies have a twitter account.

Companies use twitter for feedback and customer relations. Questions can be asked with a specific hashtag. Or with an at sign to target a specific user. This makes it easy to filter the messages, and therefore easier to get in contact with the customer. Best Buy demonstrated the successfulness of twitter in

[1] About Twitter: https://twitter.com/about
[2] Wikipedia: http://en.wikipedia.org/wiki/Twitter

customer relations by answering questions with a specific hashtag. In 2009 they had answered nearly 20 thousand questions using twitter. [Li and Li, 2013, p1] Market Intelligence is also a major aspect of the microbloggin sphere.

Twitter represents one of the largest and most dynamic datasets of user generated content. Along with Facebook twitter data is in real time. This has major implications for anyone who are interested in sentiment, public opinion or customer interaction. [Speriosu et al., 2011]

A typical tweet contains about 11 words and provides an opinion or state of mind or a piece of information. Tweets can contain hashtags: something, user: @username, or other adaptations of prefixes such as $STO which represents a stock. The different prefixes or tags ($, #, @) easily distinguishes the content of the tweet. This also makes it easier to search and classify the content of tweets. Examples of tweets can be found in figure:2.1 and figure:2.2.

The retrieval of tweets seems like a challenge and impractical with a web scraper. But Twitter has made this easy by providing an API [3]. With the API you can write tweets and update the status of a user. But the best part of the API is that it provides search capabilities. To get a certain subset of all tweets, we can use the search function and view only the tweets we want.

On the front page of twitter we have the search function at the top right of the page. The search provides the ability to specify which types of tweets you want. And gives you the opportunity to find the information you are looking for.

Figure 2.1: Typical tweet from Twitter.

2.2 Sentiment

Opinion mining on the web is not a new phenomenon. But in resent years it has become much more attractive to traders in the financial world. The usage of Twitter and other social media platforms is increasing. This means

[3] API: Application programming interface

Figure 2.2: Typical tweet from Twitter.

a surplus of raw data with easy access. Companies all over the world has started to use the social networks to their benefit. The use of information from social media has become part of the trend, although there are some drawbacks and shortcomings. Noise and garbage is one of them. The difficulty of the huge amount of available data is that it's difficult to find only the information relevant for your use. Even if you're right 80% of the time, the last 20% can prove devastating. [Stevenson, 2012]

Sentiment broadly refers to a persons state of mind. Based on the state of mind the person will do optimistic or pessimistic choices. A positive state of mind leads to optimistic judgements of future events, and a negative state of mind leads to pessimistic. [Doukas et al., January 10, 2010, p4]

The users may have different roles and intentions in different communities in the microblogging sphere, [Java et al., 2007]. A users intentions and its reasons for participation might be a factor in the sentiment analysis.

2.2.1 What is Sentiment Analysis

There are two main categories of approaches to sentiment analysis. The first is to use a classifier. The classifier can use methods such as naive Byes, maximum entropy or support vector model [Li and Li, 2013] This is typically a method where it would be natural to use machine learning of evolutionary algorithms to increase the classification correctness over time. The other is to use linguistic resources, such as corpora of negative and positive words. The developed linguistic resources are used to classify the sentiment of the text [Li and Li, 2013].

Li and Li has created a framework for sentiment analysis. The system consists of four main steps and is tested with experiments on twitter. First they do topic detection, identifying and extracting the topics mentioned in the tweet. Secondly opinions are classified. The polarity of the opinion is decided and the users impression is captured. Third. Credibility is assessed. This creates a better summarization of the expresser's credibility. Fourth, step one, two, and three is aggregated to reflect the true opinion and point of view. Combining the first three steps in the fourth results in a truer reflection

of the expresser's opinion. [Li and Li, 2013]

One way of classifying tweets is to use predefined lexicon of positive and negative words. Consumer confidence and fluctuations of voting polls can be tracked in this way [Connor et al., 2010].

The work of [Diakopoulos and Shamma, 2010] describes a methodology for better understanding of temporal dynamics of sentiment. The system uses visual representation to achieve this. This is investigated in the reaction to debate video. Further [Diakopoulos and Shamma, 2010] detects sentiment pulse and controversial topics with the help of visualisation and metrics. [Diakopoulos and Shamma, 2010] used crowdsourcing[4] to classify batches of tweets. This was accomplished with Amazon Mechanical Turk, a crowdsourcing site[5].

[Barbosa and Feng, 2010] explores the problem of noise in biased and noisy data. They focus on noisy labels and add features to the tweets to increase the classification properties of the tweets. To filter out tweets that don't project a sentiment tweets are classified as subjective or objective. The subjective tweets are classified as positive or negative.

Classification of tweets can be generalised by using features. Features are elements such as unigrams, bigrams, and part-of-speech tags. An abstract representation of a tweet would be beneficiary to the classification. In this abstract representation [Barbosa and Feng, 2010] propose to use characteristics about how tweets are written and meta-information about the words in tweets. Meta-features and tweet syntax features are further features that can improve classification. Meta-features are information about the tweet, such as location, language, and number of retweets. The tweet syntax features are things such as hashtags, retweet, reply, links, punctuation and emoticons [Barbosa and Feng, 2010].

Another approach to the sentiment challenges with twitter is explored by [Becker et al., 2013]. They explore techniques for contextual polarity disambiguation and message polarity classification. Constrained and supervised learning is used to create models for classification. They describe a system that solves these tasks with the help of polarity lexicons and dependency parsers. Expanded vocabulary is one of the main aspects of their success, as they say in their findings: "We hypothesize this performance is largely due to the expanded vocabulary obtained via unlabeled data and the richer syntactic context captured with dependency path representations." [Becker

[4]Crowdsourcing is the practice of obtaining needed services, ideas, or content by soliciting contributions from a large group of people, and especially from an online community, rather than from traditional employees or suppliers. http://en.wikipedia.org/wiki/Crowdsourcing

[5]Amazon Mechanical Turk (AMT): https://www.mturk.com/mturk/

et al., 2013]

In contrast to [Becker et al., 2013], [Speriosu et al., 2011] has used distant supervision and labeled propagation on a graph based data structure. The data structure represents users with tweets as nodes. And tweets with bigrams, unigrams, hashtags, etc as subnodes of the tweets. A label propagation approach rivals a model supervised with in-domain annotated tweets and outperforms the noisily supervised classifier and a lexicon-based polarity ratio classifier. [Speriosu et al., 2011]

2.2.2 Sentiment analysis in Finance

[Brown and Cliff, 2004, p2] writes the following on over-reaction of investors: "*He(Siegel (1992)) concludes that shifts in investor sentiment are correlated with market returns around the crash. Intuitively, sentiment represents the expectations of market participants relative to a norm: a bullish (bearish) investor expects returns to be above (below) average, whatever "average" may be.*". In the light of resent changes in the financial world and the use of sentiment from social media, the notion that opinions and sentiment of investors and market actors affect the market is not a new observation.

Use of sentiment can predict changes and momentum in the market. Bad news in an optimistic period creates cognitive dissonance in the small investors. This impacts the market by slowing down the selling rate of loosing stocks. [Doukas et al., January 10, 2010, p29] Further we can see that optimistic sentiment has a 2% monthly average return. While the investor sentiment is pessimistic we see a drastic reduction in returns. Down to 0.34%,[Doukas et al., January 10, 2010, p5]. After optimistic periods it is indicated that the monthly return is reduced to -0.49%. On the contrary there is no equivalent change after a pessimistic period, [Doukas et al., January 10, 2010, p6-7]. Momentum profits are only significant when the sentiment is optimistic, [Doukas et al., January 10, 2010, p29].

Hope and fear is used by [Zhang et al., 2011] to decide the movement of the market. The sentiment is aggregated to be hopeful or fearful. This basically focuses on positivity and negativity of the sentiment of that particular day. The daily sentiment is then compared to the market indicators of the same day to create a prediction of the market. [Zhang et al., 2011] finds that calm times give little hope or other emotions. Little turmoil results in few fluctuations in the market. And opposite, lots of emotions(hope, worry, fear), gives speed to the market.

[Brown and Cliff, 2004, p3] indicates that the sentiment does not cause subsequent market returns. For a short-term marketing timing this is bad news. However with the changes in social media over the last decade how is

the situation today? With the microblogging sphere of today we can easily see the correlation of sentiment and the market indicators, [Jubbega, 2011]. But does the sentiment cause changes in the market-return? [Brown and Cliff, 2004, p3] also says that optimism is associated with overvaluation and subsequent low returns.

[Brown and Cliff, 2004, p] concludes that aggregated sentiment measures has strong co-movement with changes in the market. He also indicates that sentiment doesn't appear to be a good trading strategy. This, in the view of [Zhang et al., 2011], indicates a leap in sentiment research and what is possible with the microblogging of today.

2.3 Finance and Trading

The management of assets or liabilities and the management of funds over a period of time is called Finance. In finance the valuation of assets are time dependant. The same asset is not worth the same now and in a few minutes. Assets are priced based on expected returns and risk level. The three sub categories of finance are: personal, corporate and public. [6]. These categories describes very different parts of the financial world.

Trading is the action of buying or selling financial instruments. Financial instruments can be stocks, bonds, derivatives or commodities [7]. Trades takes place in markets, stock markets, derivatives markets or commodity markets.

Technical analysis in finance.

2.4 The Trend

The trend is the general opinion of the masses. As defined by the Free Dictionary: "The direction and momentum of a market, price, economy, or other measure. For example, if the price of a security is going mainly downward with only a few gains, it is said to be on a downward trend. Identifying and predicting trends is important finding the right moment to buy and sell securities. Trends are especially important in technical analysis, which recommends buying at the bottom of a downward trend and selling at the top of an upward trend." [8]

[6] Wikipedia:http://en.wikipedia.org/wiki/Finance

[7] Wikipedia:http://en.wikipedia.org/wiki/Trader_(finance)

[8] Dictionary description of trend: http://financial-dictionary.thefreedictionary.com/Trend

It's often talk about the fashion trend or the music trend when regular people talk about the trend. Or just the general direction of which a subject or subculture are moving.

Trends work in much the same way as opinions. An opinion is uttered then others start to think the same thing or feel the same way. The first group of people that move in the same direction are called trend setters. They are the people that show others how this trend works and what this trend is about.

On twitter we have lots of subcultures that all express themselves on their specific topic. Whether it's technology, art, finance or any other thing. In the sense of twitter we can take a step back and look at the content of messages and from there see if we can find common topics that people talk about, this being the topic of a subculture or a subspace of twitter. To get the trend we have to look at the content of the messages in a subspace. Given that the trend is the collective general collective opinion of the subspace we can look into this an see if we can find certain topics or areas of interest that aggregates to a trend.

When looking for twitter and trends there are few of far between those who work on it. No material or indication is found to suggest that trending on twitter is researched in regards to sentiment analysis of tweets.

Chapter 3

Data, retrieval and structure

We aim to describe the structure, the characteristics, the metadata and usage for the data used.

In this data chapter we have three sections.

The section abut twitter and tweets, 3.1. Where we describe how we use tweets, what tweets to use and how we acquire them.

Then we have a section describing the dictionaries, 3.2. How we compile the dictionaries from tweets and how we use them. Their shortcomings and the possible improvements.

And last we have a section that describes the financial data that we use. Where we get them, the structure and how the data is used.

3.1 Tweets

A tweet is a massage posted on twitter. The message can be up to 140 characters long and in many ways it resembles the well known SMS[1].

Tweets are posted to the users profile. When other people posts a previously posted tweet again it is called a retweet.

All users can follow other users on Twitter. Tweets from users you follow will appear in you stream of tweets on the main page of twitter.

3.1.1 Tweet Structure

Structure

There are a lot of metedata in the tweets. In fact most of the data in a tweet object is metadata.

[1]Wikipedia on Short Messaging Service:https://en.wikipedia.org/wiki/Short_Message_Service

The data we acquire from Twitter is in the JSON data format. JSON or *JavaScript Object Notation* is an open standard format that uses human-readable text to transmit data objects consisting of attribute–value pairs[2].

A positive thing about the JSON data format is that we can directly evaluate it in python. By using literal evaluation in python the tweet object is interpreted as a dict. This makes the use of a tweet easy.

For an example of the data structure of a tweet, see appendix: D.

Content

A tweet is an astonishing compilation of data about who, where and when a tweet was posted.

As for the content we have the text, the message itself. With the content we have fields for all the links, all the emoticons, and all hashtags that are present in a tweet.

Every tweet is posted by a user. All the data of a user is also present for each tweet. Here we have data on follower count, profile images, friend count, time zone, and many other profile related items.

For the sharing of a single tweet we have data fields such as favorite_count and user_mentions. We also have favorited, retweet_count.

In addition we have the location of a given tweet. Where the tweet was posted, the name of the place, the coordinates of the tweet, the country, and the id of this place.

See all the different metadata types in appendix: D.

3.1.2 Twitter API

The twitter API is a convenient way for lots of people to access data from twitter. Tweets, streams, timelines, profiles and more are available through the api.

To provide easy easy access and conformity to industry standards, the api provides data in the JSON format.

While the api does not give access to 100% of the data from twitter, it gives a good representation of the tweets from the last 7 days.

Setup

To get access to the api there are a few requirements. You have to have a twitter account. You have to register the application you are going to use the api with, thereby getting authentication keys. Then you have to used the keys authenticate with twitter before you access the api.

[2]Wikipedia on JSON:https://en.wikipedia.org/wiki/Json

For a simple guide to this we have http://datascienceandprogramming.wordpress.com/2013/05/14/twitter-api/ as a good example.

The 4 authentication tokens you get from Twitter, app_key, app_secret, oauth_token,a nd oauth_token_secret, is used with the Twython library[3] as described below.

The simplest example of use of the api and the twython library can be described as follows:

- Authenticate towards twitter.
- Execute search query on twitter.
- Print ID's of all retrieved tweets.

The code of the example is as follows:

```
twitter = Twython(APP_KEY, APP_SECRET, OAUTH_TOKEN, OAUTH_TOKEN_SECRET)
results = twitter.search(q='Search query', count='15')

for status in results['statuses']:
    print status['id']
```

For more advanced use we have generators, and lots of parameters and api endpoints to use. Endpoints and search parameters will be described under section 3.1.2, *API endpoints options*. The twyhon framework and its advanced usage can be explored more in the code and in the documentation of the framework [4].

Restrictions

The api has some access restrictions. Or rather rate limitations. This is to be expected, as unlimited access would cripple the api.

Twitter limits request of a particular kind at 180 requests per 15 minutes. Which means that we can do 180 searches spread evenly over the time interval. Or we could do 180 requests as fast as possible and then wait. The rate limits can be explored thoroughly in the twitter documentation[5]

As for the practical implications of the limitations. They are not a problem in our case. We get more then enough data. By using a generator and pour out tweets we get 1000 tweets in about 60 seconds. But then we have to wait 15 minutes. This is suboptimal. As we will crash the program at access denied from the api.

[3] https://pypi.python.org/pypi/twython/

[4] https://twython.readthedocs.org/en/latest/usage/advanced_usage.html

[5] Twitter: https://dev.twitter.com/docs/rate-limiting/1.1

API endpoints
Twitter has made a lot of different endpoints available. The endpoints are divided into these categories: *Timelines*, *Tweets*, *Search*, *Streaming*, *Direct Messages*, *Friends & Followers*, *Users*, *Suggested Users*, *Favorites*, *Lists*, *Saved Searches*, *Places & Geo*, *Trends*, *Spam Reporting*, *OAuth*, and *Help*.

Of these categories we mainly use *OAuth*, *Help* and *Search*. Listing the endpoints used with parameters we get this list:

Searching
To perform a search we use the parameters described in the *API endpoints* table in section 3.1.2.

Mainly we need a query. The query is a normal search string where twitter will find tweets containing all words in the string.

```
query = "Finance Increase"
```

Further we can expand the search by using logic.

```
query = "Finance OR Investment AND Economy OR Growth"
```

The specific search used to compile the tweetset the dictionaries are based on is the following. Where we get all tweets containing one of the words: *Finance*, *Investment*, *Economy*, and *Growth*.

```
query = "Finance OR Investment OR Economy OR Growth"
```

Adding other parameters such as count and language we can further improve our search. To execute such a search we get python code like the following, where 'results' is a data structure containing tweets.

```
query = "Finance OR Investment OR Economy OR Growth"
results = twitter.search(q=query, count='15', language='no')
```

Mining optimization
The acquisition of tweets, or the mining operation went quite well. We got lots of tweets. But we ran into a problem. Retweets. In some cases we got up to 90% retweets in a mining session. Many of the tweets being essentially the same one, only retweeted multiple times.

When we are using tweets to create dictionaries, we do not want duplicate data that we do not need. Retweets are after all mostly duplicate data. So we removed the retweets and got a lot more different tweets.

As nearly all retweets start with 'RT' we can easily sort them out. Then we get a query like this.

Table 3.1: Used Twitter API endpoints table

Category	Parameter	Description
search	-	Used for acquisition of tweets. A query can take the following parameters:
	q	A UTF-8, URL-encoded search query of 1,000 characters maximum, including operators. Queries may additionally be limited by complexity.
	count	The amount of tweets acquired in each request. Standard = 15, max = 100.
	geocode	Get tweets close to these coordinates.
	lang	The language we want tweets in. Very limited by the number of people that speaks that language.
	locale	Language specific. If the query is in Norwegian we get tweets in Norwegian.
	result_type	mixed, recent or popular. This is the general mix of tweets returned in a search. Recent is the newest tweets, while popular are tweets that are retweeted a lot and tweets from users with many followers.
	until	Gets tweets before the given time.
	since_id	We get tweets posted after the given time.
	max_id	We get tweets with ids lower then the one given.
	include_entities	This parameter has no practical application to us.
	callback	An optional place where twitter can post tweets back to us.
authenticate	-	How we login and get access to the api.
	oauth_token	It is the authentication code or password to access the api.
rate_limit_status	-	The way we know if we are still inside the request limitations.
	resources	The elements that we want the status of. Currently that is search and help.

```
query = "Finance OR Investment OR Economy OR Growth AND -RT"
```

When using the twython framework and its cursor function we get a continuous stream of tweets. A problem with this is that twython's cursor basically executes multiple searches. Thus yielding the same tweets multiple times, unless you change the search. So we change the search to accommodate this and use the *max_id* parameter.

```
query = "Finance OR Investment OR Economy OR Growth"
results = twitter.cursor(
                twitter.search,
                q=query,
                count="100",
                lauage=language,
                max_id='the id of the last tweet')

for result in results:
    print result
```

API Caveats

The main caveats of the twitter API is the request limitations and the limitations of the search engine.

As twitter says themselves: '*Please note that Twitter's search service and, by extension, the Search API is not meant to be an exhaustive source of Tweets. Not all Tweets will be indexed or made available via the search interface.*' and '*The limitations of the search engine of twitter indexes only about a weeks worth of tweets.*'[6]

Although this is not a big problem, coding and data acquisition could have been simpler. The solution for the week limitation was to broaden the search to include more words. This resulted in a more varied dataset, but also more tweets. We initially wanted to analyse tweets related to finance, so this was a bit of a negative point.

Further caveats of the request limitations. This means that we have to mine tweets over time. We should really have set this up on a server and mined tweets with a cron job every 15 minutesWikipedia on Cron: https://en.wikipedia.org/wiki/Cron.

[6]Twitter: https://dev.twitter.com/docs/faq#8650

3.1.3 Tweet sets

We ended up with two datasets to be used. The obama tweetset[7], which is a set of tweets containing around 1300 tweets about obama and the election of 2008/2009. And a self compile dataset, referenced as the kiro dataset, based on the words: *Finance*, *Investment*, *Economy*, and *Growth*.

Search terms
The kiro dataset is based on only four words. This is a very limited part of twitter and in no way representative for it's full content. Neither does the search terms represent a full range of finance words.

An improvement would use a wide variety of finance words to mine tweets. This would improve the relation to finance and the relevance of the dictionaries afterwords.

Structure The self compiled datsets has one tweet per line. This is the JSON data object that is automatically imported into python.

As for the obama tweet set we only have the text, where we have one tweet text per line.

Caveats
Obama tweets is not ideal for sentiment analysis. To much political nonsense to comprehend. A political statement is not positive or negative. It is positive in the eyes on some and negative in the eyes of others. And there are people that think it is neutral because they do not care. Retweets are also present, so the actual data we get out of the tweet set is limited.

The kiro dataset has a lot retweets and neutral tweets in it. Therefore we have only used positive and negative tweets later, ignoring all the neutral ones. This gives us more relevant data to work with and less noise. Although we should have used the neutral tweets to improve the dictionaries.

3.1.4 Trend Data

When mining larger sets of tweets, the rate limitations and week limitation is a challenge. But solved by broadening the mining. The mining in itself is quite easy. Just execute a search and store all the new unique tweets.

To get a broad search we have a list of search terms. The terms are mostly usernames, but also some hashtags and other words. A drawback with the

[7]Neal Caren of University of North Carolina. Tweet file: http://www.unc.edu/~ncaren/haphazard/obama_tweets.txt

search terms are that they might not resemble the area of tweets we want to get. That are the finance related terms.

The trend tweets are stored in files named with the search term. Each term having it's own file with tweets. Then we sort the tweets by date and get files containing tweets for an individual date.

All the used search terms a stored in the file: '_search-terms' [8]. Most of the search terms are based on an article from 'Teknisk Ukeblad' where they list the twitter handles that are most significant in the oil industry. Since Norway has a lot of oil, the financial market and the trend is greatly dependent on it. And we can see if there are any relations between the compiled trend and the value of the stock exchange.

3.1.5 Problems, Shortcomings, and Possible Improvements

Retweets are a source of concern. The retweets does not give much in the sense of new and unique data. But they can provide a significance in sharing and importance of a given tweet. Retweets should be investigated more thoroughly in the future.

A shortcoming of the data mining is the search terms. Are the terms representative? Do we get good data or not? Are there other terms that are better suited to get accurate results? There are to many questions to ask about the data to rely on them to much. A wide array of tests and analysis should be done to remove this factor as a problem.

Another interesting point to consider is whether or not the choice of finance as the are of focus was smart. Is this area more or less difficult then other areas to navigate? We think that finance related tweets and news are more objective and has a firmer answer. And in total have less emotion and less room for interpretation.

3.2 Dictionaries

The dictionaries are lists of words used in the classification process. For each set we have a list of positive words and a list of negative words.

The dictionaries are self compiled or downloaded. The downloaded dictionaries are very specific in their area. The LoughranMcDonald dictionaries only contain financial terms, while the self compiled dictionaries contains everything.

[8]Search term file: https://github.com/magnuskiro/master/blob/master/code/trend/_search-terms

The purpose of the dictionaries are to provide a way to separate words. And further to give a quantitative way to distinguish positive and negative tweets. We also want to look at the quality of the different dictionaries. And the method for dictionary compilation.

We use the dictionaries to count mono-, bi-, and tri-grams. And we can say something about the quality of the dictionaries and the method for compilation.

The dictionaries are used in both classification methods. The simple word_count(4.2) classification and the more advanced classification using SVM(4.3.1) and Naive bayes(4.3.2) classifiers.

3.2.1 Downloaded Dictionaries

The downloaded dictionaries are dictionaries found on the Internet. They are compiled by others, and their quality is questioned. The most significant feature of the downloaded dictionaries are that they contain words form a certain domain. The obama dictionary contains more words linked to politics, while the LoughranMcDonald dictionary only has words from the financial domain.

Obama

The obama dictionary was created in relation to the obama tweet set [9], and the us presidential election of 2008.

In the positive list [10] there are 2230 words. The frequency of political words over other types words are uncertain. We have not done any analysis on this part. *wisdom*, *truthful*, *profit*, and *intact* are words found in the positive list. This in itself speaks for the incorrectness of the dictionary. Neither *intact* nor *wisdom* are words that in themselves can be described as positive or negative.

There are 3905 negative words in the list of negative words[11]. The negative list include words such as *decrease*, *worried* and *tricky*. Duplicate words are also present, so some sort of improvement of this dictionary should be done.

[9]Neal Caren of University of North Carolina. Tweet file: http://www.unc.edu/~ncaren/haphazard/obama_tweets.txt

[10]Neal Caren of University of North Carolina. Positive words: http://www.unc.edu/~ncaren/haphazard/positive.txt

[11]Neal Caren of University of North Carolina. Negative words: http://www.unc.edu/~ncaren/haphazard/negative.txt

Table 3.2: LoughranMcDonald available dictionaries

Negative words	General list of negative words.
Positive words	General list of positive words.
Uncertainty words	Words like *may*, *maybe*, and *nearly*. Words that flag content to have no concrete sentiment.
Litigious words	Law related words, not much use for us.
Modal words strong	Strong descriptive words, such as *Always*, and *Strongly*
Modal words weak	Weak words on moods, such as Somewhat, and *Depends*.

Loughran McDonald
Tim Loughran and Bill McDonald has a set of dictionaries available from the websites of University of Notre Dame [12].

The lists of words:

The potential of these dictionaries are big, and much unused. We only use the positive and negative lists of words. The other lists could be used for a better measure of polarity or weighting of bigrams. The best use of this dictionary is for comparison of the different dictionaries.

3.2.2 Compiled Dictionaries

The compiled dictionaries are based on the two manually labeled tweet sets. The kiro tweet set, and the obama tweet set.

Details about the process of manually classifying tweets can be found in section 4.1.

Dictionary Compilation
The compilation of a dictionary is quite simple. We take a set of manually labeled tweets, and extract words, then sort them into positive and negative.

- 1: We import all the tweets to create dictionaries from.
- 2: We take all the positive tweets and generate words from them. Mono-, bi- or tri-grams.

[12]Bill McDonald, University of Notre Dame: http://www3.nd.edu/~mcdonald/Word_Lists.html

- 3: We repeat step 2 with the negative tweets.
- 4: We remove words that are present in both the positive and negative list. Removal of duplicate words. Resulting in two lists of unique words either positive or negative.

Characteristics For both datasets we compiled mono-, bi-, and trigrams. Giving us 6 sets of compiled dictionaries to test. All dictionaries has the drawback of personal bias. The mindset of the person labeling the dataset also effects the dictionaries. We also do not consider stemming and stop words, because we remove duplicate words.

3.2.3 List of dictionaries

Table, listing all the different sets dictionaries used. There are a list of negative and a list of positive words in each set.

3.2.4 Error analysis, removal of duplicate words

When creating the different dictionaries we remove duplicates from the positive and negative dictionary set. Words that are present in both the positive and negative dictionary is removed. By doing this we remove words that has no significance in the classification. But we also risk removing words with significance.

When looking at the duplicate words from the monogram dictionary based on the kiro dataset we found some errors. As a selection of words found, we have *dangerous, bad, go, inc, let, up, or, need, good, if, no, are, and, of, on, the, is, as.* Here we can see that the words *good* and *bad* are represented. Which is not good. By removing the words from the dictionaries we have removed significant words in further classification, thus reducing correctness of the algorithm. This is one of the drawbacks of the monogram dictionaries.

When looking at the removed duplicate words for bigram and trigrams we found no indication of the same problem. As the uniqueness of bigrams and trigrams are a lot greater we end up with very few duplicates and only duplicates that has no significance to the over all classification. Although we might have other unknown problems.

Most stop words and other insignificant words are removed with the removal of duplicate words. The same thing cannot be said about the bigram and trigram dictionaries. There we have no stop words present in themselves, but they are frequently part of other terms. For further improvements of classification with word counting and dictionary quality we should remove stop

Table 3.3: Dictionary table

Name of dictionary	Description
Downloaded:	
Obama original	Monograms, in relation to the obama tweet set.
LoughranMcDonald	Monograms, acquired from Bill McDonald's webpage.
Combined Obama original and LoughranMcDonald	Monogram. A combination of the previous two dictionaries.
Compiled:	
Kiro, Monogram	Compiled from the kiro dataset. Containing monograms.
Kiro, Bigram	Containing terms consisting of two separate words. (*an example*)
Kiro, Trigram	Containing terms consisting of three separate trigrams. (*example of trigram*)
Obama, Monogram	Compile from the obama tweet set, monograms.
Obama, Bigram	Containing terms consisting of two separate words. (*another example*)
Obama, Trigram	Containing terms consisting of three separate trigrams. (*also an example*)

words, such as *as, is, on, off, and, or* etc, from the tweet/sentence before creating bi- and trigrams.

3.3 Finance Data

Data acquisition
Obtaining the financial data is easy. Point and click to receive a csv file containing the data we need. Netfonds.no provides the content we need.

More specifically we get the data for the Oslo Stock exchange[13].

By creating a little script we get fresh data every time we run the script.

```
stock_exchange_history = urllib.urlopen(urli_of_datafile).readlines()
    for record in stock_exchange_history:
        # do tomething with the data
        print record
```

Structure
The data file contains data back to 1997. Data for one day on each line. Only the days the exchange have been open are present. Fields in the csv file are: *quote_date,paper*, *exch*, *open*, *high*, *low*, *close*, *volume*, and *value*.

Usage
The values we use are: *quote_date*, *close*, and *volume*. A potential drawback of the data are the days that the exchange is closed. This might complicate the coding of the trend comparison.

[13]Netfonds data on OSEBX http://www.netfonds.no/quotes/paperhistory.php?paper=OSEBX.OSE&csv_format=csv

Chapter 4

Sentiment Classification

This chapter describes the experiments done. High level description and execution of experiments. Detailed descriptions of execution and technical details in appendix.

Sections in this chapter are as follows. Manual classifications, , where we see how people classify tweets. Then we count words with 'Word count classification', . Expanding with the use of classifiers in section . A comparison of the classifiers and associated results can be found in section . And a brief discussion an comments come last, .

Sentiment is described as "an attitude toward something; regard; opinion."[1]. The sentiment is the perceived positivity of the message that the user tries to communicate. Sentiment is in many cases a personal thing, and can change from person to person or from setting to setting. We think of the sentiment as conveyed meaning of a message.

Some of the motivation for acquiring the sentiment of a tweet or a sentence, is that we can say something about a persons state of mind and from that predict behaviour. We want to use the sentiment to make smart decisions later. As an example of usage it would be ideal to find a correlation between sentiment and stock exchange, thus making us able to increase revenue with decisions based on the sentiment.

In this thesis we have two main ways of classifying tweets. Word counting and training a classifier. Both methods require dictionaries of positive and negative words, 3.2. In the classifier we use the dictionary to extract features from a tweet. And with the word counting we count the number of positive and negative words.

[1] Dictionary.com on Sentiment: http://dictionary.reference.com/browse/sentiment?s=t

4.1 Manual Classification

When labeling tweets manually there are a number of factors the complicates the process. Among them are the quality of the tweet, state of mind, language, and political affiliation.

The quality of the tweet describes the content in many ways. Does the tweet contain links or hashtags? Are users mentioned?

State of mind for a person dictates that persons actions in many cases. This also has an effect on labeling tweets. A positive state of mind classifies more tweets negative as positive then others.

Political affiliation plays a huge role when labeling the obama tweet set. Do you like obama? Do you like Romney? Neither? This matters. The tweetset in itself is pro obama. So while labeling this tweet set we put aside political influence and looked at the core of the sentiment.

Note that all this happens in the brain during 3 to 60 seconds while reading the tweet text and is a very rough description of the thoughts happening. Labeling a tweet follows this algorithm in many ways:

Step	Thought	Description
1	Have we seen this tweet before?	Skip it or use previous classification.
2	#Hashtags or links present?	Some hashtags are automatically positive or negative. Also remove noise such as users and links.
3	Sarcasm?	If sarcasm is present, put up a warning flag saying that it is the opposite of step 4.
4	Special words?	Find a word that triggers positive or negative impression.
5	Done	Label tweet as positive or negative.

Result files
When classifying with manually we create files with the results. These files are comma separated files with three fields.

- Sentiment: Positive, neutral or negative. Represented by 1, 0 or -1.
- Tweet id, if it exists, else 'id'. It is a long number.
- The tweet text. In some cases sanitized.

Obama tweet set
When labeling the obama tweet set we found that the tweets present was rather rubbish. Whether or not the data is representative for twitter in general is difficult to say. But we observed lots of retweets and political related content. Tweets favoring obama are positive for poeple who support obama and negative for supporters of Romney at the same time. So while labeling the tweets we tried to remove personal political views from the equation, but that is very difficult in the long run.

Kiro tweet set
The kiro tweet set also has a lot of retweets, but they are not use later, thereby removing the retweet reduced data uniqueness. Also the search terms the dataset is based on could be broader to increase the spectrum of tweets.

4.2 Word count classification

We describe the classification process and the different parts of it, then we show the results and discuss the drawbacks of this method. The algorithm does, simply put, count the positive and negative words. More positive words than negative means the tweet is positive and vice versa.

4.2.1 Classification

Polarity
The polarity of a given tweet is based on the difference in the amount of positive verses negative words.

First we calculate the amount of positive words. Then divide that on the total amount of words. Giving us the percentage of positive words. Then we do the same thing for negative words.

Then we look at the difference between the negative and the positive word percentage. If the difference is positive we have a positive tweet. And if the difference is negative we have a negative tweet.

Code wise we get something like this:

```
var positive_words_count
var negative_words_count

positivity = positive_words_count / total_number_words
negativity = negative_words_count / total_number_words
```

```
polarity = positivity - negativity

if polarity > 0:
    tweet is positive
else:
    tweet is negative
```

Threshold
Threshold is the ratio of positive vs negative words that has to be present for a tweet to be either positive of negative.

The percentage of positive words minus the percentage of negative words gives the polarity value, or the positivity(how positive a tweet is) of a tweet. When actually deciding if a tweet is positive or negative we look at the polarity value. If the polarity value is above the threshold (polarity ¿ threshold) the tweet is classified as positive.

Examples of classification follows:
Example tweets:

- t1 = "good that he was decreasing badly"
- t2 = "he was good for increase"
- t3 = "good or bad"

Classification of t1:

- pos = 1 / 6 = 0.16666
- neg = 2 / 6 = 0.33333
- polarity = pos - neg = -0.1667
- threshold of 0 gives negative classification
- threshold of 0.1 gives negative classification
- threshold of -0.2 gives positive classification

Classification of t2:

- pos = 2 / 5 (to av fem ord) = 0.4
- neg = 0 / 5 = 0

Table 4.1: Average threshold accuracy table.

Threshold	Accuracy	Threshold	Accuracy
-	-	0.0	0.6479
-0.1	0.6316	0.1	0.6516
-0.2	0.6161	0.2	0.6511
-0.3	0.6059	0.3	0.6430
-0.4	0.5988	0.4	0.6305
-0.5	0.5888	0.5	0.6122
-0.6	0.5711	0.6	0.5934
-0.7	0.5423	0.7	0.5712
-0.8	0.5083	0.8	0.5457
-0.9	0.4881	0.9	0.5307

- polarity = pos - neg = 0.4 - 0.0 = 0.4
- threshold = 0.4 $\Rightarrow$ positive
- threshold = 0.5 $\Rightarrow$ negative
- threshold = -0.1 $\Rightarrow$ positive

Classification of t3:

- pos = 1 / 3 = 0.3333
- neg = 1 / 3 = 0.3333
- polarity = pos - neg = 0
- threshold = 0 $\Rightarrow$ positive
- threshold = 0.1 $\Rightarrow$ negative
- threshold = -0.1 $\Rightarrow$ positive

Further we found the best average threshold value to be 0.1. From the table under we have the threshold value, and the average classification accuracy among the 18 entries for each threshold value.

Table 4.2: Word Count classification results table

id	Dictionary	Failed	Correct	Accuracy
	– Kiro compiled dataset –	a	b	b/(a+b)
1	Monogram, obama	578	419	0.4203
2	Monogram LoughranMcDonald	491	506	0.5075
3	Monogram, combined Obama and LoughranMcDonald	416	581	0.5827
4	Kiro, Monogram, self compiled	115	882	0.8847
5	Kiro, Bigram, self compiled	17	980	0.9829
6	Kiro, Trigram, self compiled	18	979	0.9819
7	Obama, Monogram, self compiled	567	430	0.4313
8	Obama Bigram, self compiled	534	463	0.4644
9	Obama Trigram, self compiled	567	430	0.4313
	– Obama tweet set –	a	b	b/(a+b)
10	Monogram, obama	855	510	0.3736
11	Monogram LoughranMcDonald	508	857	0.6278
12	Monogram, combined Obama and LoughranMcDonald	544	821	0.6015
13	Kiro, Monogram, self compiled	632	733	0.5370
14	Kiro, Bigram, self compiled	521	844	0.6183
15	Kiro, Trigram, self compiled	498	867	0.6352
16	Obama, Monogram, self compiled	493	872	0.6388
17	Obama Bigram, self compiled	37	1328	0.9729
18	Obama Trigram, self compiled	39	1326	0.9714

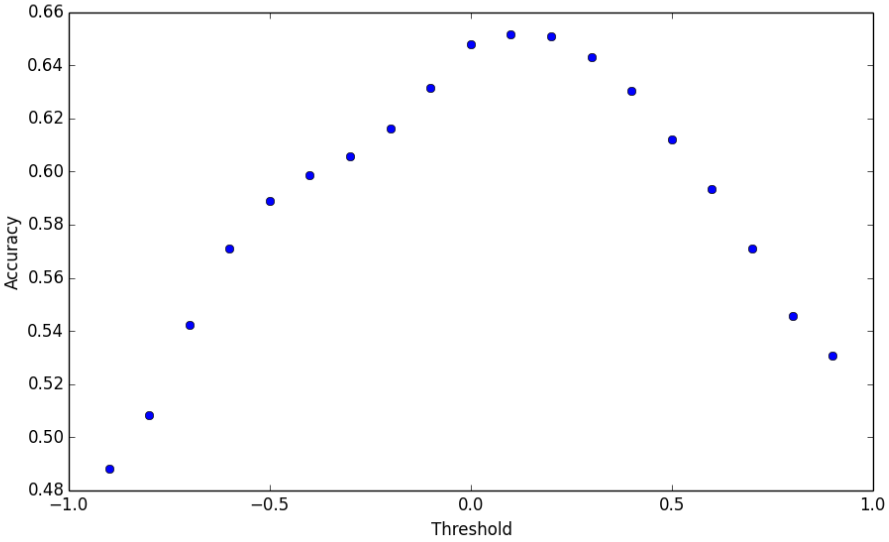

Figure 4.1: Average threshold accuracy
Graph plots of the 'Average threshold accuracy' table.

4.2.2 Results

The results from the word count classification can be plotted in the tabled 4.2.2 and in the graph 4.2.2.

5, 6, 17, 18 are the dictionaries with best accuracy. Which is to be expected for the classification of the dataset the dictionary was created from.

But more interestingly, if we take the dictionaries created form one dataset, and look at the results of the classification on the other dataset. We compare 7, 8, 9 with each other and 13, 14, 15 with each other. Then we can see that the bigram and trigram dictionaries perform overall better than the monogram dictionaries.

Further we see indications that the quality of the dictionary plays an important role. The LoguhranMcDonald monogram dictionary performs quite good with both datasets. Even on par with the compiled dictionaries on opposite dataset. This indicates that a well crafted dictionary can perform as well as compiled dictionaries.

When comparing the results from one dataset to the other we can see indications of how the content of the dataset also plays a huge role in the results. Some tweets are more difficult to classify then others.

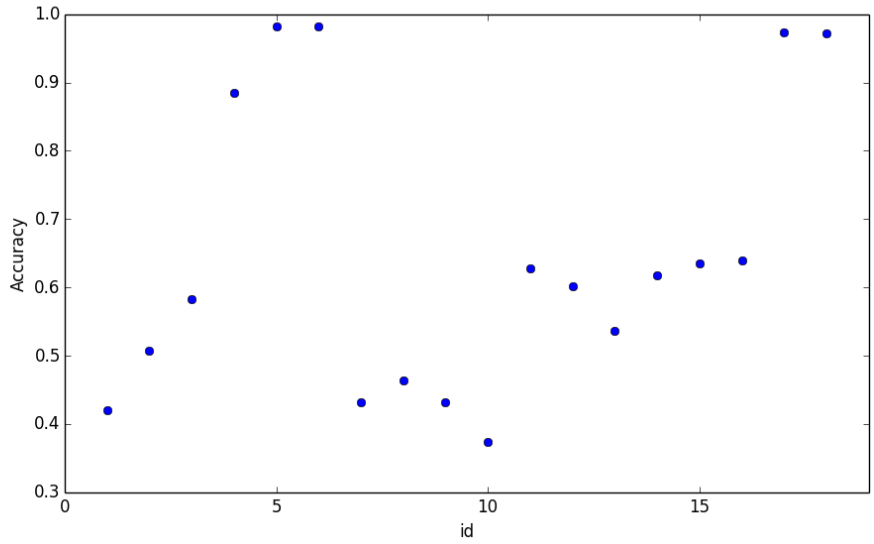

Figure 4.2: Dictionary Accuracy plot
The graph shows the plotted accuracies from the word count classification. The ids represents the ids from the table above, 4.2.2

Threshold variations
By varying the threshold we hoped to find an optimal point of which we could separate tweets based on polarity. From the following graphs, figure 4.2.2, we can see no clear distinction of one value being better than the other ones.

In figure 4.2.2 we list the results of the experimentation with the threshold. Table 4.2.2 lists the dictionaries and dataset used for which graphs in figure 4.2.2. 'kiro dataset' and 'obama dataset' columns tells which dataset that was classified in which graph.

4.2.3 Drawbacks

There are some drawbacks to the word count classification. The dictionaries could be better and the threshold could be improved. The threshold results depends on the dictionaries, so if we improve the dictionaries we would reduce the threshold errors.

Dictionaries
The main drawbacks with the dictionaries are that they are based on the

Table 4.3: Dictionary to threshold graph plot table

Dictionary name and description	kiro dataset	obama dataset
Obama original, Monogram	1	10
LoughranMcDonald, Monogram	2	11
Combined Obama original and LoughranMcDonald, Monogram	3	12
Kiro, Monogram, self compiled	4	13
Obama, Monogram, self compiled	5	14
Kiro, Bigram, self compiled	6	15
Obama, Bigram, self compiled	7	16
Kiro, Trigram, self compiled	8	17
Obama, Trigram, self compiled	9	18

datasets we classified manually. Although we cross classify so we still get valid results. To improve the dictionaries we should expand the original datasets and combine the dictionaries. We should also remove stop words before creating bi and monograms.

Threshold
When classifying with the word count classifier a potential problem is when we get an equal amount of positive and negative words. Then the polarity value becomes 0. This results in a situation where we have no indication of a tweet being positive or negative. This is an area where we can improve the classification.

These numbers are significantly reduced if the threshold is set to 0.1. We can also see indications that the diversity and quality of the dictionary plays a role in the classification correctness. The more words in the dictionary the more likely it is that we do not get an equal amount of positive and negative words.

4.3 With Classifiers

To test and compare to other classification methods we used Naive Bayes and Support Vector Machine classifiers.

Comparing the two classifiers we found that SVM proved to be the better classifier in this case. Although we see quite a span within the SVM results. Details of the results and classifiers are described below.

Table 4.4: Word Count results where Threshold value=0 table

id	Dictionary	Polarity=0	Tweets
	– Kiro compiled dataset –		
1	Monogram, obama	234	997
2	Monogram LoughranMcDonald	543	997
3	Monogram, combined Obama and LoughranMcDonald	178	997
4	Kiro, Monogram, self compiled	53	997
5	Kiro, Bigram, self compiled	7	997
6	Kiro, Trigram, self compiled	28	997
7	Obama, Monogram, self compiled	14	997
8	Obama Bigram, self compiled	446	997
9	Obama Trigram, self compiled	931	997
	– Obama tweet set –		
10	Monogram, obama	335	1365
11	Monogram LoughranMcDonald	854	1365
12	Monogram, combined Obama and LoughranMcDonald	345	1365
13	Kiro, Monogram, self compiled	233	1365
14	Kiro, Bigram, self compiled	462	1365
15	Kiro, Trigram, self compiled	1221	1365
16	Obama, Monogram, self compiled	37	1365
17	Obama Bigram, self compiled	52	1365
18	Obama Trigram, self compiled	92	1365

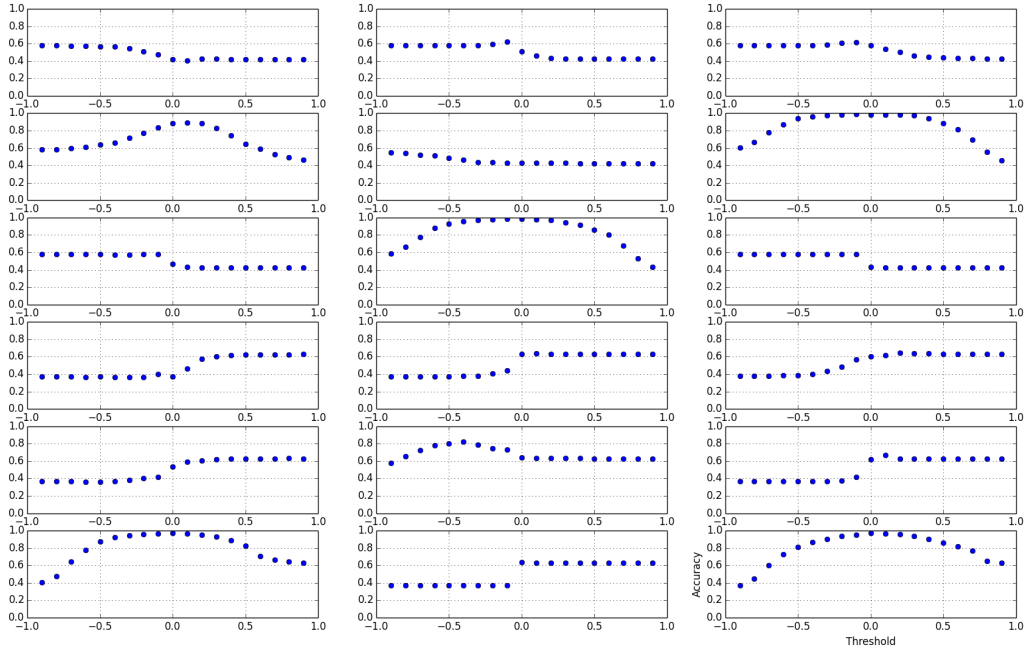

Figure 4.3: Threshold variation accuracy plot
The graphs plot the different variations of threshold. Counting is columns first; top left is 1, top mid is 7, top right is 13.

The self compiled dictionaries based on the kiro dataset was used to compare the two types of classifiers.

Each test with a classifier uses the kiro monogram dictionary for feature extraction, and the described dataset as training set and the dataset to be classified.

4.3.1 SVM

Choice of kernel

When using SVM there are quite a few different kernel modules that can be used to improve results. The kernel is often chosen based on knowledge about the dataset. In the nltk[2] library for python there are a number of different kernels. We tested all of them.

While testing which kernel was best we used the kiro dataset for training, and the kiro monogram dictionary for feature extraction.

[2]Natural Language Toolkt http://www.nltk.org/

Table 4.5: SVM kernel test results table

Kernel	Failed	Correct	Accuracy
LinearSVC	7	990	0.9930
NuSVC	29	968	0.9709
NuSVR	422	575	0.5767
OneClassSVM	575	422	0.4233
SVC	422	575	0.5767
SVR	422	575	0.5767

Table 4.6: SVM classifier results table

Dataset	Type	Failed	Correct	Accuracy
Kiro	Monogram	7	990	0.9930
Kiro	Bigram	422	575	0.5767
Obama	Monogram	35	1330	0.9744
Obama	Bigram	507	858	0.6286

Using the self compile monogram dictionaries and all the different SVM kernels we get these results:

Naturally we used the LinearSVC kernel later.

Results

Results from testing SVM with different dictionaries:

From this we can draw some conclusions. The monogram dictionary performed better than the bigram dictionary. Probably due to the number of features we can extract from each tweet. Also classifiers are more accurate than the word count classification method.

4.3.2 Naive Bayes

Results from testing Naive Bayes with different dictionaries:

As we can see the different dictionaries makes no difference for Naive Bayes. But if we compare the classifier with the results from the word count classification we can clearly see improvement.

Table 4.7: Naive Bayes classifier results table

Dataset	Type	Failed	Correct	Accuracy
Kiro	Monogram	29	968	0.9709
Kiro	Bigram	29	968	0.9709
Obama	Monogram	59	1306	0.9568
Obama	Bigram	59	1306	0.9568

Table 4.8: Comparison of classifiers table

Classifier	Dataset	Dictionary	Failed	Correct	Accuracy
Word Count	Kiro	Obama Bigram	534	463	0.4644
Word Count	Obama	LoughranMcDonald Monogram	508	857	0.6278
Word Count	Obama	Kiro, Trigram	498	867	0.6352
Naive Bayes	Kiro	Kiro Monogram	29	968	0.9709
SVM	Kiro	Kiro Monogram	7	990	0.9930

4.4 Comparison and Results

The Classifiers
We have looked at three classifiers, Naive Bayes, SVM, and the word count classification. Together they have classified the two dataset we have with varying results.

All methods of classification uses the manually labeled tweet sets as the data source. And we use the self compile dictionaries for feature extraction.

The Results
In the following table, 4.4, we highlight the results from the different classifications. We list the most noteworthy results from our experiments and compare them.

As we can see the classifiers give better results then the word count classification. And SVM is a bit better than Naive Bayes. Although the results from the word count classification indicates that the dictionaries play an important role in the results. We can also see that monograms are better for classifiers, while trigrams are better for word counting.

For the classifiers and the good results with the monogram dictionaries, we think that has to do with the number of features we get from a tweet. The more features we have the easier it is to classify and the more accurate

the result.

4.5 Comments and Discussion

The work that is so far described potential for improvement and some uncertainty elements. We need to address this and criticise the findings we have.

4.5.1 Improvements

There are a lot of improvements that could be done.

The word count classification is merely a convenient way to say something about the dictionaries used. We now know more about dictionaries and dictionary compilation than we did before. But we should expand our knowledge toward the quality of the dictionaries. And ways to eliminate terms that has no polarity value. The dictionaries would also be greatly improved with bigger datasets to compile the dictionaries from.

We should cross classify and test the classification with the SVM classifier a bit more and do quality assurance on the work with it. We should also look at other classifiers and other data to confirm the findings we have so far.

4.5.2 Biased Mind

The datasets of manually labeled tweets are biased based on a persons personal opinion and the state of mind in the moment of classification. Therefore we have to keep in mind that all the results are based on the assumption that everyone agrees on the manual labeling. This is of course a big potential source of errors to be explored more in 9.

And we should explore the psychology of perception and classification. How do people perceive content differently? Is finance easier to label than politics? And what can we do to eliminate the human factor in manual labeling?

4.5.3 Drawbacks

The drawbacks of psychology and the datasets renders some uncertainty in the results. Although some of our observations give insight into the use of natural language in association with dictionaries. Further drawbacks are the threshold of the word count classification. This should be addressed in some smart way.

Code wise we should improve the testing of the classifiers, and remove the static use of dictionaries in the feature extraction.

4.5.4 Conclusions

Dictionaries can be compiled and used in classification of sentiment. While classifiers work better than word counting, we still rely on the manually labeled tweets for accuracy. We also know that the data we use to check the classification are biased and should be improved.

4.5.5 Future Work

Analysis of the dictionaries and improvement of them would be a big task to undertake. While we should also look at the input data and sort the data smartly to remove spam. And find another way to decide the sentiment in the word count classification.

Chapter 5

Trending

We looked at trends in regard to twitter and finance. And tried to see if there was any correlations between them. The feeble attempt is described in this chapter.

We start of by describing and defining a trend in section 5.1. Continuing with trends on an in relation to twitter, 5.2. To be followed by trends in finance, 5.3. An last we Compare twitter trends with finance trends, 5.4.

5.1 The trend is your friend

A trend is a series of changes that moves in the same direction over a period of time. It is often associated with fashion or stock trading. The definition of trend is "the general course or prevailing tendency"[1].

In finance especially the trend can be a deciding factor in buying and selling of goods. You typically do not want to sell during a bullish trend, value going up. And you want to sell before you get to the bearish trend, value going down.

When trading based on the trend, the problem is to find the trend and predict it into the future. And there Twitter comes into the mix. If we could predict the trend based on public opinion mined from twitter we would have an advantage over other traders.

If the trend is your friend, you know how the market will move and make good decisions in accordance with the trend.

[1]Dictionary.com: http://dictionary.referencs.com/browse/trend

5.2 Trends from Twitter

There are two parts to trends with twitter. The trends Twitter themselves create based on words of hashtags that appear in many tweets. And the trend we compile ourselves based on data we choose.

On Twitter
The trends on twitter is largely predictable and gives little new information to the trend compilation. The trend on twitter is also specialised for each user, based on that users subscriptions and location. As an example at the Norwegian national day(Mai 17.) we would have trending words like Norge, *Norway*, and *17Mai*. Today we already know that this will happen again next year.

As the Twitter trend depends on the volume of tweets to be updated we have to look at individual tweets, users or volume of some sort to predict a trend.

Our own
To begin with we narrowed down our area of data gathering to a set of search words. Most of them based on an article on tu.no[2]. Taking tweets from everywhere on twitter might give us large amounts of spam or other content about breakfast or similar.

The article lists the most important oil related account on twitter. In two parts. One for the Norwegian ones, and one for international ones. The international ones are the search terms used in the actual trend compilation and graph plotting.

After sorting the mined tweets by day, we calculated the change in tweets between two days. More specifically we looked at the change in positive and the change in negative tweets between today and yesterday. This giving us an average change graph.

To be even more precise we took the percentage average change of change in positive and change in negative tweets.

```
# difference from yesterday til today.
# change in positive tweets between this and the previous day.
pos_diff = (positive_tweets_today - positive_tweets_yesterday) / (
    total_amount_of_tweets * 1.0)

# change in negative tweets between this and the previous day.
```

[2]Tu.no: http://www.tu.no/petroleum/2014/04/05/dette-er-de-viktigste-twitrerne-for-oljebransjen

```
neg_diff = (negative_tweets_today - negative_tweets_yesterday) / (
    total_amount_of_tweets * 1.0)

# median = the mid point between the positive and negative change points.
# change in sentiment volume between this and the previous day.
median = min([neg_diff, pos_diff]) + abs(pos_diff - neg_diff) / 2
results.append(median)
```

Plotting this we get the average change graph, figure: **??**.

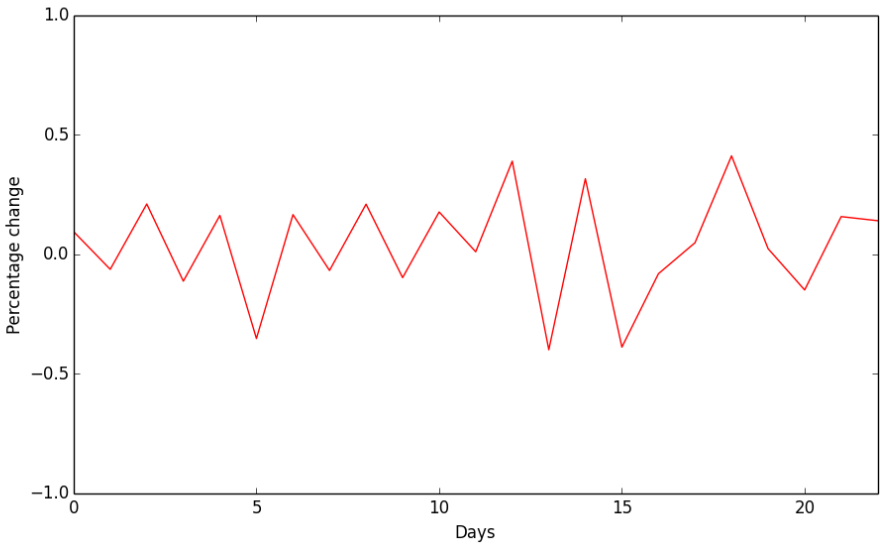

Figure 5.1: Tweet trend plot
The graph show the average change in tweet sentiment between two days, over a 23 day period.

5.3 Trending in Finance

We looked at the change at Oslo stock exchange in the same period as the tweets above(Apr 30 - May 16). For the days the stock exchange was closed we padded out the dataset with the previous open day. So that Saturday and Sunday would have the same values as Friday. This is mainly to make the coding and plotting of the graph easier.

To describe the calculation of the finance trend: we took the percentage of change in closing value from yesterday and today. Then we scaled the

graph by a value of 100 to make the changes visible in the same plot as the tweet trend.

```
# calculate percentage diff between this and the previous day.
diff = (closing_value_today - closing_value_yesterday) / (
    average_value_today_yesterday * 1.0)

# scaled to work with the tweet trend.
results.append(diff * 100)
```

Plotting the graph we got figure 5.3.

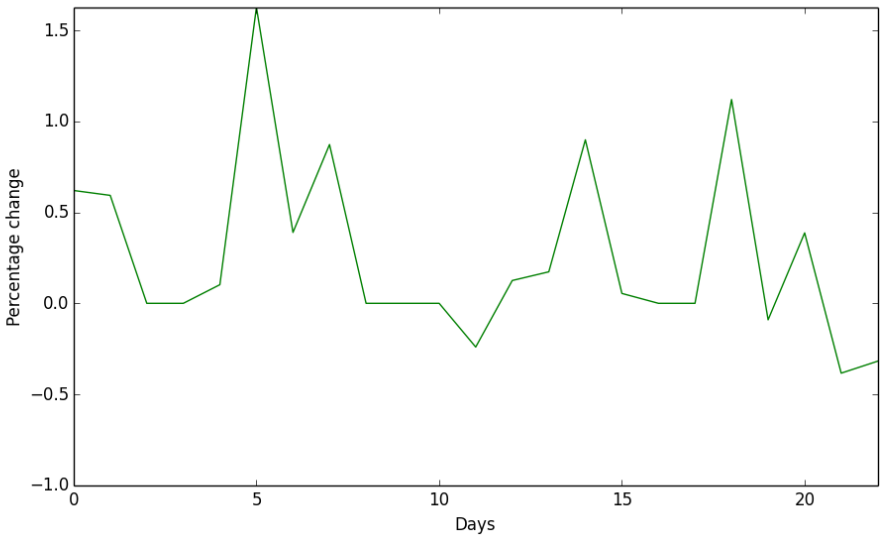

Figure 5.2: Finance trend plot
The graph show the average change in closing value between two days, over a 23 day period. Note that the graph is scaled to be comparable with the tweet trend graph.

5.4 Comparing the trends

First off, the trend plot in figure 5.4.

And continuing with everything that is wrong with it.

As correlations go between the two graphs we have very few. There are two areas where the peaks are quite alike. At least the change is in the same

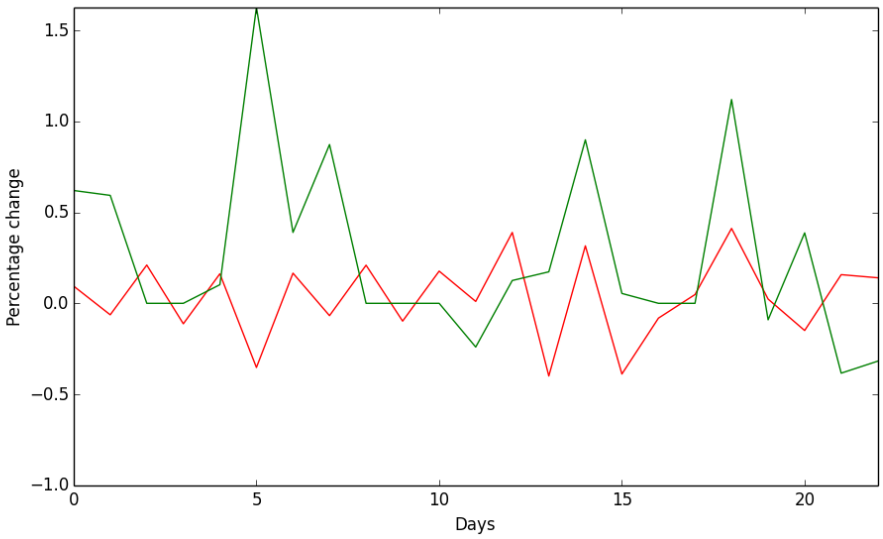

Figure 5.3: Comparing trend plot
The graph shows the two previously compiled trends. The red one is the tweet trend, while the green one is the finance trend.

directions. This is at day 14 and 18. Besides those two points the graphs have no likeness at all.

What we can see from the plots is that the trend compilations have no correlations. From this we can say for certain that the amount of data and way of plotting the trends do not work.

The two things that are likely to play a big part in the results is the data we used and the way we compiled the tweet trend.

To narrow and to little data makes the result rather inconclusive. We should have used something like a million of a billion tweets to get statistically significant results.

And we should look closer on how we aggregate the twitter trend. Further we should look at other fields of a tweet that can be significant for a trend. Like followers and retweets.

From this we learn that data matters a lot and that it is difficult to find a good way to aggregate a trend.

Chapter 6

The Code

The code is complete in the sense of doing what it should. Or making an attempt to. The results of the coding are up for discussion.

For functionality we have code for mining tweets, compiling dictionaries, classifying tweets, and aggregating a trend based on data from twitter.

For this chapter we start of with the structure of the code, 6.1. Then touch the technology and libraries, 6.2, used up to this point. Before we look at the data mining, 6.3, dictionary compilation, 6.4, classification, 6.5, and trend aggregation, 6.6. Last we describe the comparison, 6.7, of the finance trend and ending with the issues, 6.8, of the code.

6.1 Structure

There are four main folders. One for classification, one for mining tweets, one for the dictionaries and on for the files associated with trend.

Code for classification is split into four files. One file for utilities, helper functions that does not touch logic. And the three was of classifying tweets, manual, word counting and with a classifier. List_threshold_results plots the results from the threshold variation in graphs with pyplot.

The dictionary code is split onto two files, helpers and utility in one, and logic and execution in the other.

Trend code is in the trend folder. There we have mining_utils, which is a replica of the same file in the twitter folder. This file only helps with the acquisition of tweets. The trend_mining file executes the mining operations, and acquires tweets from twitter based on a set of search terms. The trend_-tweet_sorting file sorts tweets from all the raw search term data files into files based on date. So all tweets from the same date ends in the same file. And last we have the trend_compilation file, which contains code for compiling

and plotting trend data and financial data.

Last we have the 'twitter' directory which has code for extracting tweets from twitter. This code is used for creating the datasets that the dictionaries are built from. We have the mining_utils which is helper code for connection and writing files and such. While the mining_operations file does the logic of managing the acquired tweets.

The important files are listed under. '- something' are folders, while the others are actual python files. The indentation shows which files are in which folders.

```
- code
    - classification
        classification_utils.py
list_threshold_results.py
        manual_classification.py
        svm_bayes_classification.py
        word_count_classification.py
    - dictionaries
        dictionaries.py
        dictionary_utils.py
    - trend
        mining_utils.py
        trend_classification_utils.py
        trend_compilation.py
        trend_mining.py
        trend_tweet_sorting.py
    - twitter
        mining_operations.py
        mining_utils.py
    graph_plots.py
```

6.2 Technology and Libraries

The technology used, frameworks etc.

The used python libraries are: ConfigParser, ast, codecs, io, os, twython, time, matplotlib, urllib, re, nltk, sklearn. Most of them are quite standard, while some are not. And some are self explanatory.

The libraries that are not shipped with standard python are described in the following paragraphs.

Twython Library for connection and integration with the twitter api. The source and documentation can be found on github: https://github.com/ryanmcgrath/twython

nltk The natural language tool kit (nltk) is a library that provides functionality for working with human language. It has functionality such as classifiers and tokenization tools. See more at http://www.nltk.org/.

sklearn Scikit-learn provides functionality for learning algorithms, machine learning and classification. We use this library to provide the kernels for out classification with classifiers. http://scikit-learn.org/

matplotlib Matplotlib is a library for graph plotting. And that is what we use it for. http://matplotlib.org/.

6.3 Data retrieval

As we have to data sources we split it into finance data and data from twitter.

6.3.1 Twitter

The mining operation can be seen as three steps. This happens in *mining_operations.py* and *mining_utils.py*.

Step 1
Authenticating and connecting to twitter.

```
conf = open('/home/kiro/ntnu/master/code/twitter/auth.cfg').read()
config = ConfigParser.RawConfigParser(allow_no_value=True)
config.readfp(io.BytesIO(conf))

# getting data from conf object.
APP_KEY = config.get('twtrauth', 'app_key')
APP_SECRET = config.get('twtrauth', 'app_secret')
OAUTH_TOKEN = config.get('twtrauth', 'oauth_token')
OAUTH_TOKEN_SECRET = config.get('twtrauth', 'oauth_token_secret')

# creating authentication object for twython twitter.
twitter = Twython(APP_KEY, APP_SECRET, OAUTH_TOKEN, OAUTH_TOKEN_SECRET)
```

Step 2
Query execution.

```
results = twitter.cursor(twitter.search, q=query, count="100", lauage=language)
```

Where we have the query, and language as input variables to the *cursor_-extraction* function.

Step 3
Data extraction and storage.

```
# opens new file with today's date and time now as filename
filename = destination_folder + "/dataset-" + strftime("%d-%b-%Y_%H:%M:%S")
# opens the file for appending.
data_set = open(filename, 'a')

for result in results:
    # skip previously acquired tweets
    if str(result['id']) in str(previous_tweets_list):
        continue
    else:
        previous_tweets_list.append(result['id'])

    # store tweet to file for later use.
    data_set.write(str(result) + "\n")
```

We open the storage file for writing and gives it a name based on the time of creation. Then we traverse all results and store tweets we have not seen before.

6.3.2 Finance

We get finance data from http://www.netfonds.no/. Specifically we get the data about Oslo stock exchangeOSEBX data: http://www.netfonds.no/quotes/paperhistory.php?paper=OSEBX.OSE&csv_format=csv We also sort out the data we do not want by breaking at a certain date.

We do this by using the urllib library in python:

```
# get data file from internet. (csv)
stock_exchange_history = urllib.urlopen(url).readlines()
records = []
for record in stock_exchange_history:
```

```
        # earlier records are not interesting.
        if "2014" in record:
            records.append(record.strip().lower().split(","))
        if "20140414" in record:
            break
    return records
```

6.4 Dictionary compilation

For the dictionary compilation we use the manually classified datasets to create mono, bi and trigram dictionaries.

Starting off the process we run compilation for all datasets:

```
data_files = [
    # [name/description, name of file containing tweets]
    ["kiro", "tweets_classified_manually"],
    ["obama", "obama_tweets_classified_manually"]
]

for item in data_files:
    # get labeled tweets
    # tweets[0] are the positive ones, tweets[1] are the negative ones.
    tweets = get_positive_negative_tweets_from_manually_labeled_tweets(classific

    compile_monogram_dictionaries(tweets, item[0])
    compile_bigram_dictionaries(tweets, item[0])
    compile_trigram_dictionaries(tweets, item[0])
```

Mono, bi and trigrams creation parts are quite similar:

```
for text in tweets[0]:
    positive_dict.extend(text.split(" "))
# negative
for text in tweets[1]:
    negative_dict.extend(text.split(" "))

filename += "-monogram"
save_dictionaries(positive_dict, negative_dict, filename)
```

Where the 'positive_dict =' would be the only part changing.

Bigram:

```
bigrams_list = get_bigrams_from_text(text)
positive_dict = positive_dict + bigrams_list
```

Trigram:

```
trigrams_list = get_trigrams_from_text(text)
positive_dict = positive_dict + trigrams_list
```

We get the bigrams and trigrams by:

```
# if we have no words to work with, return empty list.
text = clean_text(text)
if not text:
    return []

bigrams_list = []
# for all bigram tuples
for tup in bigrams(text.split(' ')):
    # find the tuple texts
    # (u'text1', u'text2')
    m_obj = re.search(r'\(u\'(.+)\', u\'(.+)\'\)', str(tup).decode())
    if m_obj:
        # combine tuple parts to bigram
        word = m_obj.group(1) + " " + m_obj.group(2)
        # if bigram don't exist
        if word not in bigrams_list:
            # add it to list.
            bigrams_list.append(word)
return bigrams_list
```

Where we essentially generate sets of words in tuples and then extract the combined term by string conversion and regex.

Then we conveniently remove duplicates between the positive and negative dictionary before we write the dictionaries to file.

```
primary_dictionary = get_lines_from_file(primary_dictionary_name)
secondary_dictionary = get_lines_from_file(secondary_dictionary_name)

duplicate_words = []
# for all words in the primary dictionary.
for pw in primary_dictionary:
    #print w
    # for all words in the secondary dictionary.
```

```
    for sw in secondary_dictionary:
        # if primary word equals secondary word
        if pw == sw:
            # remove word from both dictionaries.
            primary_dictionary.remove(pw)
            secondary_dictionary.remove(sw)
            duplicate_words.append(pw)
            # as we don't have duplicates within a list we skip to next pw
            break

# rewrite the updated lists to file
write_array_entries_to_file(primary_dictionary, primary_dictionary_name)
write_array_entries_to_file(secondary_dictionary, secondary_dictionary_name)
write_array_entries_to_file(duplicate_words, "duplicate-words")
```

6.5 Sentiment Classification

For both classifiers we have utility code for loading data and dictionaries, but the important parts are the classification. Which is shown below.

6.5.1 Word Count

For each tweet we do the same thing. Count words and find the difference between positive and negative words. Here is the code for doing so.

```
# sanitize text.
tweet = sanitize_tweet(tweet)

# get word count for tweet
word_count = len(tweet.split(' ')) * 1.0

# get word count of pos and neg words.
pos = get_word_count(positive_dict, tweet) / word_count
neg = get_word_count(negative_dict, tweet) / word_count

# storing the polarity value
polarity.append(pos - neg)

if polarity[-1] == threshold:
    edge_cases += 1
```

```
# adding sentiment value (True/False), for the last classified tweet.
if polarity[-1] > threshold:
    # positive tweet
    results.append(True)
else:
    # negative tweet
    results.append(False)
```

6.5.2 With classifier

The execution of the classification can be described in the following steps.

Classifier initialization First we have to initialize the classifier, then train it. We send the class of the kernel as an input argument, such that *classifier_class* is 'SklearnClassifier(LinearSVC())'.

```
# get the training set.
training_set = nltk.classify.apply_features(extract_features_from_text,
tweets)

# create the classifier.
classifier = classifier_class.train(training_set)
```

Where the extract_features_from_text function is rather important. Here the 'dictionary' variable is the combined list of the kiro-monogram-negative.txt and kiro-monogram-positive.txt dictionaries.

```
dictionary = get_list_of_possible_words_in_tweets()
document_words = set(text)
features = {}
# more precision to iterate the word of a tweet then the whole dictionary.
# extracts all words that are both in the dictionary and in the tweet
for word in document_words:
    features['contains(%s)' % word] = (word in dictionary)
return features
```

Classification The code for classifying a tweet is quite simple.

```
# runs classify() on the given classifier class in the nltk library.
results.append(classifier.classify(extract_features_from_text(tweet)))
```

6.6 Trend aggregation

The trend aggregation is a split process. Where the mining of tweets, sorting, classification, and trend compilation happens in separate steps. Most important is the trend compilation. The sorting process only reads tweets and stores them the file that corresponds with the posting date of the tweet. Mining is done on the same principles as described earlier, only with other search terms.

6.6.1 Compilation

First we get positive and negative tweets from the dataset. And store that in a dict, with the filename, or day, as key.

```
pos, neg = 0, 0

# get tweets for this day.
trend_day_tweets = [ast.literal_eval(tweet) for tweet in
                    codecs.open(trend_base + trend_day_filename, 'r',
"utf-8").readlines()]

# sentiment for each tweet.
sentiment = []

# for tweets in this day
for tweet in trend_day_tweets:
    # get words of two or more characters.
    tweet_text = [e.lower() for e in sanitize_tweet(tweet['text']).split()
if len(e) >= 2]

# get sentiment
    sentiment.append(classifier.classify(extract_features_from_text(tweet_text))

    # if the last tweet was positive.
    if sentiment[-1]:
        pos += 1
    else:
        neg += 1

return [pos, neg, pos + neg]
```

Then we calculate the average change of positive and negative tweets between toady and yesterday.

```
keys = sorted(trend_days.keys())
#print keys
results = []
for i in range(1, len(keys)):
    # difference from yesterday til today.
    # dif / tot
    # change in positive tweets between this and the previous day.
    pos_diff = (trend_days[keys[i]]['pos'] - trend_days[keys[i - 1]]['pos'])
/ (
        trend_days[keys[i]]['tot'] + trend_days[keys[i - 1]]['tot'] * 1.0)

    # change in negative tweets between this and the previous day..
    neg_diff = (trend_days[keys[i]]['neg'] - trend_days[keys[i - 1]]['neg'])
/ (
        trend_days[keys[i]]['tot'] + trend_days[keys[i - 1]]['tot'] * 1.0)

    # median = the mid point between the positive and negative change
points.
    # change in sentiment volume between this and the previous day.
    median = min([neg_diff, pos_diff]) + abs(pos_diff - neg_diff) / 2
    results.append(median)

    #print keys[i], "%.4f" % median, trend_days[keys[i]]['tot']
    #print "pos", keys[i], trend_days[keys[i]]['neg'] -
trend_days[keys[i-1]]['neg']

#print len(results)
return results
```

6.7 Comparison

We do the same for the finance data as with the tweet data above. And multiplies the result with 100 to scale the graph plot.

```
for i in range(1, len(stock_records)):
    # calculate percentage diff sinse yesterday.
    diff = (stock_records[i][6] - stock_records[i - 1][6]) / (
        stock_records[i][6] + stock_records[i - 1][6] * 1.0)
```

```
    results.append(diff * 100)
```

And last we plot the data:(xlen is the number of observations)

```
x = [e for e in range(0, xlen)]
plt.axis([0, len(x) - 1, -1.0, 1.0])
plt.xlabel('id')
plt.ylabel('Accuracy')

# plot changes in tweets
#y = get_median_change_tweets(trend_days)
if trend_days:
    y = get_median_change_tweets(trend_days)
else:
    y = get_median_change_tweets(get_tweet_trend_data())
plt.plot(x, y, '-r')  # twitter data

y = get_stock_graph_data()
plt.plot(x, y, '-g')  # finance data

plt.show()
```

And compares the two graphs plotted in the same view.

6.8 Issues

The quality of the code in general is quite good. But whether or not the logic works out all the time is uncertain. Some of the functions and methods can also be improved.

Operating System Impediments There might be some OS related specifics that we do not know about. Linux of the Debian family has been used. So we expect that you know what you are doing when testing the code on Windows or Mac.

If a all a problem, if will be with pre installed packages that will not show up in the requirements or library parts discussed earlier. Thats because we do not know which packages was already installed.

Also some Linux tools has been used in scripts. Those will not work on Windows, but might on Mac.

Chapter 7

Results and Discussion

All our results are in ones and zeroes. And further we discuss why there are only zeroes. And how that affects the outcome and future endeavors for the pirates we are.

7.1 Data Source

TODO twitter as a data source TODO the dictionaries

7.2 Classification

TODO classifiers, svm over others. show it. TODO classification results and why. svm best, bla bla biased.

7.3 Trend

TODO did we get a trend? TODO and it's accuracy to finance plotting.

Chapter 8

Conclusion

We worked hard, and achieved very little.

TODO write stuff about data, datasets and dictionaries. TODO write classification and such stuff. TODO trend conclusions?

TODO overall work progress and process.

Chapter 9

Future Work

All the things I didn't have time to do my self.

9.1 Twitter

TODO further use of twitter as data source and such.

Twitter API TODO the unused parts of the API Look at the different endpoints and see how see how we can use them smartly. Twitter: https://dev.twitter.com/docs/api/1.1

search terms for the tweet sets TODO search parameters and the query. the tweet sets should be expanded to use a wide range of finance words. then create dictionaries from the sets.

implications of special twitter content TODO specifics of tweets to be used? @users, hashtags, links? Does any of them matter much?

9.2 Dictionaries

TODO the future of dictionaries.

9.3 sentiment

TODO future of classifiers and research there. classifiers: should have crosstested. used kiro set for training and obama set for classification.

9.4 trend

TODO future of trend aggregation

Trend Data The analysis of the trend data. There we ahve over 30k of tweets waiting to be analysed.

Bibliography

Luciano Barbosa and Junlan Feng. Robust sentiment detection on twitter from biased and noisy data. 2010. Coling 2010: Poster Volume, pages 36–44, Beijing, August 2010.

Lee Becker, George Erhart, David Skiba, and Valentine Matula. Avaya: Sentiment analysis on twitter with self-training and polarity lexicon expansion. 2013. Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013), pages 333–340, Atlanta, Georgia, June 14-15, 2013. c 2013 Association for Computational Linguistics.

Gregory W. Brown and Michael T. Cliff. Investor sentiment and the near-term stock market. *Journal of Empirical Finance*, 11(1):1 – 27, 2004. ISSN 0927-5398. doi: http://dx.doi.org/10.1016/j.jempfin.2002.12.001. URL http://www.sciencedirect.com/science/article/pii/S0927539803000422.

B. Connor, R. Balasubramanyan, B. Routledge, and N. Smith. From tweets to polls: linking text sentiment to public opinion time series. 2010. URL http://www.cs.cmu.edu/~nasmith/papers/oconnor+balasubramanyan+routledge+smith.icwsm10.pdf.

Leon Derczynski, Alan Ritter, Sam Clark, and Kalina Bontcheva. Twitter part-of-speech tagging for all: Overcoming sparse and noisy data. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*. Association for Computational Linguistics, 2013.

Nicholas A. Diakopoulos and David A. Shamma. Characterizing debate performance via aggregated twitter sentiment. 2010. URL http://dmrussell.net/CHI2010/docs/p1195.pdf.

Qiming Diao and Jing Jiang. A unified model for topics, events and users on Twitter. In *Proceedings of the 2013 Conference on Empirical Methods in*

Natural Language Processing, pages 1869–1879, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D13-1192.

John A. Doukas, Constantinos Antoniou, and Avanidhar Subrahmanyam. Sentiment and momentum. January 10, 2010. Updated May 20, 2011. Available at SSRN: http://ssrn.com/abstract=1479197 or http://dx.doi.org/10.2139/ssrn.1479197.

A. Java, X. Song, T. Finin, and B. Tseng. Why we twitter: Understanding microblogging usage and communities. 2007. 9th WebKDD and 1st SNA-KDD workshop on web mining and social network analysis, 2007.

Annika Jubbega. Twitter as driver of stock price. Master's thesis, BI Norwegian School of Management, 2011.

Yung-Ming Li and Tsung-Ying Li. Deriving market intelligence from microblogs. *Decision Support Systems*, 55(1):206 – 217, 2013. ISSN 0167-9236. doi: http://dx.doi.org/10.1016/j.dss.2013.01.023. URL http://www.sciencedirect.com/science/article/pii/S0167923613000511.

Jianfeng Si, Arjun Mukherjee, Bing Liu, Qing Li, Huayi Li, and Xiaotie Deng. Exploiting topic based twitter sentiment for stock prediction. 2013. Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 24–29, Soa, Bulgaria, August 4-9 2013. c 2013 Association for Computational Linguistics.

Michael Speriosu, , Nikita Sudan, Sid Upadhyay, and Jason Baldridge. Twitter polarity classification with label propagation over lexical links and the follower graph. 2011. Proceedings of EMNLP 2011, Conference on Empirical Methods in Natural Language Processing, pages 53–63, Edinburgh, Scotland, UK, July 27–31, 2011. c 2011 Association for Computational Linguistics.

Timm O. Sprenger and Isabell M. Welpe. Tweets and trades: The information content of stock microblogs. December 2010.

Alexandra Stevenson. The social media stock pickers, Oct 23 2012. URL http://search.proquest.com/docview/1114502067?accountid=12870. Copyright - Copyright Financial Times Ltd. 2012. All rights reserved.; Last updated - 2012-10-23.

Xue Zhang, Hauke Fuehres, and Peter A. Gloor. Predicting stock market indicators through twitter "i hope it is not as bad as i

fear". *Procedia - Social and Behavioral Sciences*, 26(0):55 – 62, 2011. ISSN 1877-0428. doi: http://dx.doi.org/10.1016/j.sbspro.2011.10.562. URL http://www.sciencedirect.com/science/article/pii/S1877042811023895. ¡ce:title¿The 2nd Collaborative Innovation Networks Conference - COINs2010¡/ce:title¿.

Appendix A

Processed Articles

TODO merge into previous work chapter 2. and appendix a tweet usage thingy.

A.1 Article template

file:*filename.pdf* citation:[]
* What did they use tweets for?
* What do they do?
* Event detection. Is the tweet about merging?
* How is learning present? * Is the approach statistical of NLP? * Where can this article be useful later?
* What does this article give answers to?

A.2 A Unified Model for Topics, Events and Users on Twitter

file: EMNLP192.pdf citation:[Diao and Jiang, 2013]
* What did they use tweets for?
Modelling topics, events and users in a unified way.
* What do they do?
LDA-like topic model, Recurrent Chinese Restaurant Process(discover events), Event-topic affinity vectors to model association (events–¿topics), Detecting meaningful events, Grouping events by topic. Tweet separation, topic(personal life)/event(global events)-tweet.

* Event detection. Is the tweet about merging?
Online and offline detection. Online= early detection of major events, efficiency is the main focus. Offline, focusses on getting all the relevant tweets. Don't assume every tweet is linked to an event. LDA?

* How is learning present?

* Is the approach statistical of NLP?

* Where can this article be useful later?
With event detection. Tweet separation. Financial tweets.

* What does this article give answers to?

A.3 Twitter Part-of-Speech Tagging for All: Overcoming Sparse and Noisy Data

file:*twitter-pos.pdf* citation:[Derczynski et al., 2013]

A.4 Tweets and Trades: The Information Content of Stock Microblogs

file:*SSRN-id1702854.pdf* citation:[Sprenger and Welpe, December 2010]

* What did they use tweets for?
"We find the sentiment (i.e., bullishness) of tweets to be associated with abnormal stock returns and message volume to predict next-day trading volume." [Sprenger and Welpe, December 2010]

* How are tweets used?

* Event detection. Is the tweet about merging?

* Where can this article be useful later?
What twitter is used for, Twitter chapter.

Twitter incentives. [Sprenger and Welpe, December 2010, p4]

Description of bullishness, message volume and what it does etc.

[Sprenger and Welpe, December 2010, p52] suggest that stock microblogs can claim to capture key aspects of the market conversation.

Picking the right tweets remains just as difficult as making the right trades.

* What does this article give answers to?
Whether bullishness can predict returns. Whether message volume is related

to returns, trading volume, or volatility. Whether the level of disagreement among messages correlates with trading volume or volatility. Whether and to what extent the information content of stock microblogs reflects financial market developments Whether microblogging forums provide an efficient mechanism to weigh and aggregate information

A.5 Exploiting Topic based Twitter Sentiment for Stock Prediction

file:*filename.pdf* citation:[Si et al., 2013]

* What did they use tweets for?

Predicting the stock market. Stock index time series analysis. daily one-day-ahead predictions.

* How are tweets used?

Dirichlet Process mixture model to learn the daily topic set. Vector regression. Topic-based prediction.

* Event detection. Is the tweet about merging?

* Where can this article be useful later?

Twitter's topic based sentiment can improve the prediction accuracy. [Si et al., 2013, p28]

* What does this article give answers to?

A.6 Twitter as driver of stock price

file:*Twitter as driver of stock price-Jubbega.pdf* citation:[Jubbega, 2011]

* What did they use tweets for?

* How are tweets used?

* Event detection. Is the tweet about merging?

* Where can this article be useful later?

General about twitter.

* What does this article give answers to?

A.7 Twitter Polarity Classification with Label Propagation over Lexical Links and the Follower Graph

file:*twitter polarity classification.pdf* citation:[Speriosu et al., 2011]

* What did they use tweets for?

Polarity classification. Positive/negative.

* How are tweets used?

With label propagation. Distant supervision. Graph based data structure. user–¿tweet–¿bigram/unigram/hashtag/etc.

* Event detection. Is the tweet about merging?

* Where can this article be useful later?

Data section / sentiment /

Twitter section: What people uses twitter for.

Label propagation approach rivals a model supervised with in-domain annotated tweets and outperforms the noisily supervised classifier and a lexicon-based polarity ratio classifier. [Speriosu et al., 2011]

Twitter represents one of the largest and most dynamic datasets of user generated content.

* What does this article give answers to?

A.8 AVAYA: Sentiment Analysis on Twitter with Self-Training and Polarity Lexicon Expansion

file:*Sentiment Analysis on Twitter with Self-Training and Polarity Lexicon Expansion.pdf* citation:[Becker et al., 2013]

* What did they use tweets for?

Contextual Polarity Disambiguation and Message Polarity Classication * How are tweets used?

Constrained learning with supervised learning. Unconstrained model that used semi-supervised learning in the form of self-training and polarity lexicon expansion

* Event detection. Is the tweet about merging?

* Where can this article be useful later?

Technical approach of models and sentiment analysis. State of the art on sentiment analysis with twitter.

* What does this article give answers to?
dependency parses, polarity lexicons, and unlabeled tweets for sentiment classification on short messages

We hypothesize this performance is largely due to the expanded vocabulary obtained via unlabeled data and the richer syntactic context captured with dependency path representations. [Becker et al., 2013]

A.9 Robust Sentiment Detection on Twitter from Biased and Noisy Data

file:*Robust Sentiment Detection on Twitter from Biased and Noisy Data.pdf* citation:[Barbosa and Feng, 2010]

* What did they use tweets for?
Sentiment analysis with focus on noise reduction.

* How are tweets used?
Noisy labels. Classifies tweets as subjective or objective. Then distinguishes the subjective into positive and negative tweets. Generalization of tweet classification. Meta-information. How tweets are written. More abstract representation.

* Where can this article be useful later?
Previous work, sentiment analysis, twitter, sentiment features. * What does this article give answers to?
It provides a better way to classify tweets.

A.10 Investor sentiment and the near-term stock market

file:*Investor sentiment and the near-term stock market.pdf* citation:[Brown and Cliff, 2004]

* Where can this article be useful later?
In the finance chapter for historic value and where we have come from.

[**?**, p2] on over-reaction of investors writes: " He(Siegel (1992)) concludes that shifts in investor sentiment are correlated with market returns around the crash. Intuitively, sentiment represents the expectations of market participants relative to a norm: a bullish (bearish) investor expects returns to be above (below) average, whatever "average" may be.". In the light of resent changes in the financial world and the utilisation of sentiment from

social media, the notion that opinions and sentiment of investors and market actors affect the market is not a new observation.

[Brown and Cliff, 2004, p3] indicates that the sentiment does not cause subsequent market returns. For a short-term marketing timing this is bad news. However with the changes in social media over the last decade how is the situation today? With the microblogging sphere of today we can easily see the correlation of sentiment and the market indicators [TODO:Citation]. But does the sentiment cause changes in the market-return? [Brown and Cliff, 2004, p3] also says that optimism is associated with overvaluation and subsequent low returns.

* What does this article give answers to?

[Brown and Cliff, 2004, p] concludes that aggregated sentiment measures has strong co-movement with changes in the market. He also indicates that sentiment doesn't appear to be a good trading strategy. This, in the view of [Zhang et al., 2011] indicates a leap in sentiment research and what is possible with the microblogging of today.

A.11 Predicting Stock Market Indicators Through Twitter "I hope it is not as bad as I fear"

file:*Predicting Stock Market Indicators Through Twitter.pdf* citation:[Zhang et al., 2011]

* What did they use tweets for?

Gather hope and fear for each day using tweets. The sentiment indication of each day is compared to the marked indicators of the same day.

* How are tweets used?

Get the Positive/negative sentiment.

* Event detection. Is the tweet about merging?

* Where can this article be useful later?

Address the question of intention of users on twitter. Good summary of things done in regards to twitter. (Might be a bit outdated, from 2010).

* What does this article give answers to?

That hope, fear and worry makes the stock go down the day after. Calm times, little hope, fear or worry, makes the stock go up.

A.12 Deriving market intelligence from microblogs

file:*Deriving market intelligence from microblogs.pdf* citation:[Li and Li, 2013]

* How are tweets used?

Companies use twitter for feedback and customer relations. Questions can be asked with a hashtag of to a specific user. This makes it easy to sort filter the messages, and therefore easier to get in contact with the customer. Best Buy demonstrated the successfulness of twitter in customer relations by answering questions with a specific hashtag. In 2009 they had answered nearly 20 thousand questions using twitter. [Li and Li, 2013, p1] Market Intelligence is also a major aspect of the microbloggin sphere.

* What did they use tweets for?

Sentiment classification. Topic detection, pos/neg classification.

* Event detection. Is the tweet about merging?

* Where can this article be useful later?

stateOf-twitter / state-sentiment / data /

* What does this article give answers to?

A.13 The social media stock pickers

file:*social_media_stock_pickers.pdf* citation:[Stevenson, 2012]

Opinion mining on the web is not a new phenomenon. But in resent years it has become much more attractive to traders in the financial world. Twitter and the social media's opinion is on the rise. This means a surplus of raw data with easy access. Companies all over the world has started to use twitter and readily available tweets to their benefit. Trading with social media is part of the trend. Although there are some drawbacks and shortcomings. Noise and garbage is one of them. It's difficult to accurately sort through all the data and get only the information relevant for your use. Even if your right 80% of the time, the last 20% can prove devastating. [Stevenson, 2012]

A.14 Sentiment and Momentum

file:*SSRN-id1479197.pdf* cition:[Doukas et al., January 10, 2010]

Not Twitter. Intra-day transaction data. Sentiment affects the profitability of price momentum strategies.

Use of sentiment can predict changes and momentum in the market. Bad news in an optimistic period creates cognitive dissonance in the small investors. This impacts the market by slowing down the selling rate of loosing stocks. [Doukas et al., January 10, 2010, p29]

Sentiment broadly refers to the state of mind a person has. Whereas negative of positive. Based on the current state of mind the person will do optimistic or pessimistic choices. A positive state of mind leads to optimistic judgements of future events. And a negative state of mind leads to pessimistic judgements. [Doukas et al., January 10, 2010, p4]

Further we can see that optimistic sentiment has a 2% monthly average return. While the investor sentiment is pessimistic we see a drastic reduction in returns. Down to 0.34%.[Doukas et al., January 10, 2010, p5] After optimistic periods it is indicated that the monthly return is reduced to -0.49%. On the contrary there is no equivalent change after a pessimistic period. [Doukas et al., January 10, 2010, p6-7] Momentum profits are only significant when the sentiment is optimistic. [Doukas et al., January 10, 2010, p29]

A.15 Is Trading with Twitter only for Twits?

Document Description: Blog post that describes the findings of the atricle [TODO art:ref].

The article has developed a strategy for trading stocks based on the bullishness of the tweet. [TODO glossary bullishness] Bullishness as I understand it is the same as the negativity of the tweet.

The article bases it's findings on three factors. The holding time of a stock (the time from you buy it until it's sold). The history of x days (how many of the past days are used to determine the tweet signal[TODO glossary tweet signal]). And the number of picks (how many stocks you hold a any given time).

It is also indicated that The main article has some good information about how tweets are built up. (Dollar-tagging for representation of a given stock, $AAPL)

Has a good figure of the system.

Indicates that the message volume and trade volume are related.

RefArticle: **??** Twitter mood Predicts the Stock Market.

Tags: buy/sell-signals, tweet signals, dollar-tagged, OpinionFinder, GPOMS,

A.16 From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series

The article uses polling data and two years of tweets as their data.

Basically a comparison of the opinion expressed on twitter and the opinion from phone enquiries.

Uses word counting to distinguish relevant tweets from the rest.

The twitter dataset is huge, typically billions of tweets.

Daily sentiment = positive tweets / negative tweets.

Appendix B

Tweet usage overview

TODO Write what twitter has been used for I previous articles. Minimal Summary of stuff I could find. And reference articles.

- Time series
- Stock index time series analysis
- Message volume
- Message polarity, Bullishness/bearishness
- Predicting the stock market
- Predict next day-trading volume
- Daily one-day-ahead predictions
- Topic based prediction
- Vector regression
- Dirichlet Process mixture
- Label propagation
- Constrained and supervised learning.

Appendix C

Web resources

http://hum.csse.unimelb.edu.au/emnlp2013/papers.html
http://neuro.imm.dtu.dk/wiki/Twitter$_s entiment_a nalysis$
http://provalisresearch.com/products/content-analysis-software/wordstat-dictionary/sentiment-dictionaries/
http://www3.nd.edu/ mcdonald/Word$_L ists.html$

Appendix D

Tweet Data Structure


```
{
  u’contributors’: None,
  u’truncated’: False,
  u’text’: u’WO2013149663A1 Estimating Anisotropic Resistivity Of A
Geological Subsurface $STO #G01V #G01V11 http://t.co/yyPFEJSdIj’,
  u’in_reply_to_status_id’: None,
  u’id’: 390051769780142080,
  u’favorite_count’: 0,
  u’source’: u’<a href="http://w.pat.tc" rel="nofollow">TwittlyDumb</a>’,
  u’retweeted’: False,
  u’coordinates’: {
    u’type’: u’Point’,
    u’coordinates’: [
      5.7326363,
      58.9645836
    ]
  },
  u’entities’: {
    u’symbols’: [
      {
        u’indices’: [
          77,
          81
        ],
        u’text’: u’STO’
      }
    ],
    u’user_mentions’: [
```

```
  ],
  u’hashtags’: [
    {
      u’indices’: [
        82,
        87
      ],
      u’text’: u’G01V’
    },
    {
      u’indices’: [
        88,
        95
      ],
      u’text’: u’G01V11’
    }
  ],
  u’urls’: [
    {
      u’url’: u’http://t.co/yyPFEJSdIj’,
      u’indices’: [
        96,
        118
      ],
      u’expanded_url’: u’http://w.pat.tc/WO2013149663A1’,
      u’display_url’: u’w.pat.tc/WO2013149663A1’
    }
  ]
},
u’in_reply_to_screen_name’: None,
u’in_reply_to_user_id’: None,
u’retweet_count’: 0,
u’id_str’: u’390051769780142080’,
u’favorited’: False,
u’user’: {
  u’follow_request_sent’: False,
  u’profile_use_background_image’: True,
  u’default_profile_image’: False,
  u’id’: 163877216,
  u’verified’: False,
```

```
    u’profile_text_color’: u’333333’,
    u’profile_image_url_https’: u’https://si0.twimg.com/profile_images
/2309783804/355j4shhjrh4rqb5vsys_normal.jpeg’,
    u’profile_sidebar_fill_color’: u’DDEEF6’,
    u’entities’: {
      u’url’: {
        u’urls’: [
          {
            u’url’: u’http://t.co/apqPEHN3aC’,
            u’indices’: [
              0,
              22
            ],
            u’expanded_url’: u’http://w.pat.tc’,
            u’display_url’: u’w.pat.tc’
          }
        ]
      },
      u’description’: {
        u’urls’: [

        ]
      }
    },
    u’followers_count’: 299,
    u’profile_sidebar_border_color’: u’C0DEED’,
    u’id_str’: u’163877216’,
    u’profile_background_color’: u’C0DEED’,
    u’listed_count’: 8,
    u’profile_background_image_url_https’: u’https://abs.twimg.com/
images/themes/theme1/bg.png’,
    u’utc_offset’: 32400,
    u’statuses_count’: 247688,
    u’description’: u’New patent information from WIPO.
IPC-based hashtags for realtime subject searching.’,
    u’friends_count’: 203,
    u’location’: u’Tsukuba, Japan’,
    u’profile_link_color’: u’0084B4’,
    u’profile_image_url’: u’http://a0.twimg.com/profile_images/
2309783804/355j4shhjrh4rqb5vsys_normal.jpeg’,
    u’following’: False,
```

```
    u’geo_enabled’: True,
    u’profile_banner_url’: u’https://pbs.twimg.com/profile_banners/
163877216/1359154591’,
    u’profile_background_image_url’: u’http://abs.twimg.com/images/
themes/theme1/bg.png’,
    u’screen_name’: u’w_pat_tc’,
    u’lang’: u’en’,
    u’profile_background_tile’: False,
    u’favourites_count’: 10,
    u’name’: u’World Patents Mapped’,
    u’notifications’: False,
    u’url’: u’http://t.co/apqPEHN3aC’,
    u’created_at’: u’Wed Jul 07 14:08:23 +0000 2010’,
    u’contributors_enabled’: False,
    u’time_zone’: u’Tokyo’,
    u’protected’: False,
    u’default_profile’: True,
    u’is_translator’: False
  },
  u’geo’: {
    u’type’: u’Point’,
    u’coordinates’: [
      58.9645836,
      5.7326363
    ]
  },
  u’in_reply_to_user_id_str’: None,
  u’possibly_sensitive’: False,
  u’lang’: u’en’,
  u’created_at’: u’Tue Oct 15 09:49:23 +0000 2013’,
  u’in_reply_to_status_id_str’: None,
  u’place’: {
    u’full_name’: u’Stavanger, Rogaland’,
    u’url’: u’https://api.twitter.com/1.1/geo/id/dee2255bd015b52c.json’,
    u’country’: u’Norway’,
    u’place_type’: u’city’,
    u’bounding_box’: {
      u’type’: u’Polygon’,
      u’coordinates’: [
        [
          [
```

```
            5.5655417,
            58.884420999999996
          ],
          [
            5.8687141,
            58.884420999999996
          ],
          [
            5.8687141,
            59.0608787
          ],
          [
            5.5655417,
            59.0608787
          ]
        ]
      ]
    },
    u'contained_within': [

    ],
    u'country_code': u'NO',
    u'attributes': {

    },
    u'id': u'dee2255bd015b52c',
    u'name': u'Stavanger'
  },
  u'metadata': {
    u'iso_language_code': u'en',
    u'result_type': u'recent'
  }
}
```