Magnus L. Kirø May 27, 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 English version

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?

Abstrakt Norsk versjon

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?
github repository for code an thesis.
word count for fun.

iv

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1   Context

This thesis have largely three parts to it. The finance part, where we look at stock market prediction[1], and technical analysis.

It is well known that people are affected by sentiment. Sentiment in finance is not new concept, but it is has not been explored very much yet. So we want to see if there are ways we can predict trends with the help of sentiment. We generate trends based on sentiment and stock exchange data. And we will look for correlations between them.

For the sentiment we need data. Twitter is a great source of data and a better place to express sentiment. So we chose twitter as our primary source of sentiment. To use the data in some smart ways we compile dictionaries from tweets. And use the dictionaries for sentiment classification.

The elements that are the most difficult are the creation a trend based on tweets, and getting representative data.

1.2   Motivation

We have an interest to find out how sentiment can be extracted from twitter and how it effects finance trends. Further we are motivated by the social media aspect of trends and how that effects finance.

---

[1]Wikipedia: http://en.wikipedia.org/wiki/stock_market_prediction

Why is this work done? Why do we benefit from this? Why do I want to do this? We believe that microblogging and social media plays a role in todays trends and we would like to explore that. Twitter as a social media platform is a huge place for companies to post updates and provide customer support, so it is a natural place to gather data for analysis.

If everything works out and we strike gold, we would hopefully find relations between twitter content and financial trends. And use that to prove increase revenue in trade. Our work is relevant for further endeavors about twitter, sentiment and it's relation to finance.

1.3   Research questions

TODO RQ, specify a bit better.

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 technique in 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.

1.4 Methods

For the classification of sentiment we will use bag of words[2], SVM, and Naive Bayes. SVM and Naive Bayes to test if the classifiers perform better than bag of words with dictionaries in classification. Bag of words is mostly used to test dictionaries.

Moving average[3], and average directional index[4] are used to plot trend graphs. The trend graphs can be seen in section 6.6, on page 50. Concepts and explanations of how the trend works can be found in section 5, on page 39

1.5 Findings

We explore the usage of lists of words, dictionaries, in sentiment analysis. And we look at data retrieval from twitter and the trend we can create from it. With varying results we get positive results with the dictionaries, while the trend aggregation lacks the finesse and results it should have had.

1.6 Outline

The nine chapters of this thesis all describe different aspects of the work done. First we have this introduction, 1. Naturally we follow with previous word, 2, the state of the art part, where we discuss the newest research in this field. The data retrieval is next in chapter 3.

---

[2]Wikipedia: https://en.wikipedia.org/wiki/Bag_of_words_model
[3]Wikipedia: https://en.wikipedia.org/wiki/Moving_average
[4]StockCharts.com http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:average_directional_index_adx

Proceeding the data chapter we go for the core of the thesis, the sentiment analysis, 4. The sentiment naturally leads up to the trend in chapter 5. Code, A, comes next to explain how we did things and go deeper in the specifics of implementation. Results and discussion comes in chapter 7. While conclusion, 8, and future work, 9, rounds off the thesis.

Chapter 2

Background and Previous Work

Previous work and the background information needed for the context of this thesis is described in this chapter. We look at how microblogging activity is can be related to finance Bollen et al. [2011]. And how sentiment can be used to predict stock markets.

This chapter contains related research, context knowledge and introduction to the subjects discussed in this thesis. Twitter as a microblogging platform is introduced in 2.1. Section 2.2, describes sentiment and what it can be used for. Finance and trading as a context is described in section 2.3. And last trends are talked about in section 2.4.

2.1 Twitter

Twitter is a social informations network. It's a real-time service for sharing and gathering small messages. These messages can represent everything from 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. 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. In 2012 the 500 million users would generate 3.2 queries each 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 to communicate with customers. Customers can post questions and feedback, while the company posts answers and information. 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 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 microblogging 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. Figure:2.1 and figure:2.2 are examples of tweets as shown on twitter.

The retrieval of tweets seems like a challenge. 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.

---

[1] About Twitter: https://twitter.com/about
[2] Wikipedia: http://en.wikipedia.org/wiki/Twitter
[3] API: Application programming interface

Figure 2.1: Typical tweet from Twitter.

Figure 2.2: Typical tweet from Twitter.

2.2 Sentiment

Opinion mining on the web is not new. In resent years it has become attractive to traders. The use of Twitter and social media is increasing in increasing. Both in business and with private users. This means 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. 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 opinion the person will do optimistic or pessimistic choices. A positive mindset leads to optimistic judgements of future events, and a negative state of mind leads to pessimistic choices. [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.

---

[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/

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 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 potentially predict changes and trends 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

---

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

place in markets, stock markets, derivatives markets or commodity markets.

A new aspect to trading the last decade has been online trading [8]. Speed, ease of use, and low costs made the online brokers popular. Many brokers provide platforms for trade and analysis to select potential investments.

The tools provided are often some form of technical analysis [9]. Technical analysis is the study of past market data. Mostly volume and price. The purpose of technical analysis is to forecast the direction of prices.

Among tools and techniques in technical analysis are charts, market indicators, moving average, and relative strength index. It is also quite common to combine techniques to acquire better predictions. When looking at put/call rations, short interest, implied volatility, and bull/bear indicators of sentiment is also important to take into consideration.

Technical analysis focus at numeric values, such as volume and price, where fundamental analysis [10] analyse company health, financial statements, production rates, earnings, management, and competitive advantages. Day traders and pit traders prefer technical analysis over financial analysis.

Two principles of technical analysis are, 'History repeats itself', and 'Prices move in trends'. 'History repeats itself' refers to the belief that traders will do the same actions again and again. Technicians believe that the repeated behaviour can be recognized as a pattern and be observed on a chart. 'Prices move in trends' is the belief that the price of a commodity will move directionally over a period of time. Relative highs and relative lows are indicators of a trend. Consecutive lower highs indicates a downward trend.

An interesting aspect of trading and finance is the behavioral one[11]. Behavioral finance is the field of research that study the effects of cognitive, social, emotional and psychological factors economic decisions. It also includes the consequences of resource allocation, market price and returns.

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

---

[8]Wikipedia: https://en.wikipedia.org/wiki/Online_trading
[9]Wikipedia: http://en.wikipedia.org/wiki/Technical_analysis
[10]Wikipedia: http://en.wikipedia.org/wiki/Fundamental_analysis
[11]Wikipedia: https://en.wikipedia.org/wiki/Behavioral_finance

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." [12]

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

On twitter we have lots of subcultures that all express themselves on their specific topic. It can be technology, art, finance, or fashion among others. 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 a trend is the combined general opinion of a group. We can analyse the group and see if we can find certain topics or areas of interest that aggregates to a trend.

Stock market prediction [13] is the act of trying to predict or determine the future value of a stock. A way to do this is to look for trends in the data. The trend is a tendency of movement in a particular direction for a financial market.

There are three categories of trends in finance. Primary, secondary and secular trends. Where primary trends have a medium time frame, secondary have a short time frame, and secular has long time frames [14]. Bull market and bear market are terms that, respectively, describe upward and downward market trends. Trends are often found by using technical analysis.

Secular trends are trends that last between 5 and 25 years. A secular bull market consists of many larger bull markets and many smaller bear markets. Primary trends last a year or more. We can also observe market tops and bottoms here. These are trend reversal points. Secondary trends has a duration of a few weeks or months. The secondary market trend is change in price direction within a primary trend. The small changes are often called market corrections. The short term correction is often between 5 and 20 percent.

When looking for usage of twitter trends, we find little to confirm research in this area. No material or indication is found to suggest that trending on twitter is researched in regards to sentiment analysis of tweets and finance.

---

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

[13]Wikipedia:http://en.wikipedia.org/wiki/Stock_market_prediction

[14]Wikipedia: http://en.wikipedia.org/wiki/Market_trend

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 about twitter and tweets, 3.1, is 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, 3.3, 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 in many ways resemble the well known SMS[1].

Tweets are posted to the users profile. When a user post an existing tweet again, it is referred to as a retweet.

---

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

All users can follow other users on Twitter. Tweets from users you follow will appear in your stream of tweets on Twitter. The stream is the collection of tweets from users you are following.

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.

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].

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

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.

Hashtags:

```
u'hashtags': [
     {
       u'indices': [
         82,
         87
       ],
       u'text': u'G01V'
     },
     {
       u'indices': [
         88,
         95
       ],
       u'text': u'G01V11'
     }
```

---

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

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' and 'retweet_count'.

```
u'in_reply_to_user_id': None,
u'retweet_count': 0,
u'id_str': u'390051769780142080',
u'favorited': False,
u'favorite_count': 0,
u'user_mentions': [
],
```

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.

```
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'
},
```

See all the different metadata types in appendix: E.

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 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 must have a twitter account. The application to be used has to be registered with Twitter. Registering the application gives access to API keys. Then you have to used the keys authenticate with twitter before you access the api.

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.

A simple example of how to use the API with Twython is shown below.

---

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

- Authenticate towards twitter.

- Execute search query on twitter.

- Print ID's of all retrieved tweets.

The code of the example is as follows:

```
1  twitter = Twython(APP_KEY, APP_SECRET, OAUTH_TOKEN, OAUTH_TOKEN_SECRET)
2  results = twitter.search(q='Search query', count='15')
3
4  for status in results['statuses']:
5      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.

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:

---

[4]https://twython.readthedocs.org/en/latest/usage/advanced_usage.html
[5]Twitter: https://dev.twitter.com/docs/rate-limiting/1.1

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.

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.

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

Further we can expand the search by using logic.

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

The following query is the one used to create the original tweet sets. The original tweet sets are later used as the basis for the dictionaries. We get all tweets containing one of the words: *Finance*, *Investment*, *Economy*, and *Growth*.

```
1 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.

```
1 query = "Finance OR Investment OR Economy OR Growth"
2 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 duplicate data was a problem. Retweets are mostly duplicate data, so we removed them to increase the quality of the tweet set.

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

```
1 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.

```
1  query = "Finance OR Investment OR Economy OR Growth"
2  results = twitter.cursor(
3                  twitter.search,
4                  q=query,
5                  count="100",
6                  lauage=language,
7                  max_id='the id of the last tweet')
8
9  for result in results:
10     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 the diversity of the dataset could be a problem.

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.

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*.

---

[6]Twitter: https://dev.twitter.com/docs/faq#8650
[7]Neal Caren of University of North Carolina. Tweet file: http://www.unc.edu/~ncaren/haphazard/obama_tweets.txt

In future work it would be natural to connect the search for tweets towards stocks in a better way. Another interesting thing would be to analyse how the obama tweet set can be used to predict the stock market after the election. And see if we could extract trends or other useful finance related information.

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.

The obama tweet set only have the tweet text. We have one tweet per line. And no metadata present.

Caveats
Obama tweets is not ideal for sentiment analysis. It has to much political content to be easily comprehended. 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. The mining in itself is quite easy. Just execute a search and store all the new unique tweets. The problem appear when we need huge amounts of tweets and a setup for mining continuously.

Each request to Twitter can yield 100 tweets. This gives us a theoretical maximum of 1800 tweets per 15 min. Then we have to factor in the fact that

we might already have the tweets we receive from Twitter. And that we might
not get 100 tweets per query. In addition the Twitter API only indexes tweets
a week back in time. Forcing us to mine tweets over a longer period of time.

As we have not set up a server to do the mining for us we, we did it manually
every now and then. Resulting in very few tweets per day ratio. An example of
output from our mining script:

```
@chevron
Info -- Found 18 new tweets
Info -- Metadata file created, containing 1627 tweets
```

To get a broad search we have a list of search words. The words are mostly
usernames, but also some hashtags and other words. A drawback with the
search words are that they might not resemble the area of research, finance. A
subset of the search words are:

```
@annaljunggren        @industrienergi      $sto
@ap_energi_miljo      @industrienergiu     stock
@asmundaukrust        @iraqoil_gas         @svheikki
@bp_america           @janezpotocnikeu     @tageerlend
@chedegaardeu         @knebben             @_tekna_
@chevron              @kobbaen             @tekniskmuseum
decrease              @linetrezz           @terjeaa
@dn_no                market               @theoilcouncil
```

The trend tweets are stored in files named with the search term. Each term
having it's own file with tweets. Before using the trend tweets they are sorted
by date. Example from the folder shows the structure of filenames. All the files
with 'trend-' prefix contain tweets sorted by date.

```
economy.meta          @nikolaiastrup.meta   trend-Apr-20
@eiagov               @_nito_               trend-Apr-21
@eiagov.meta          @_nito_.meta          trend-Apr-22
@eirik_milde          @norskindustri        trend-Apr-23
@eirik_milde.meta     @norskindustri.meta   trend-Apr-24
@eirinolsen           @norskoljeoggass      trend-Apr-25
@eirinolsen.meta      @norskoljeoggass.meta trend-Apr-26
@elonmusk             norway                trend-Apr-27
@elonmusk.meta        norway.meta           trend-Apr-28
@energyindepth        @oeddep               trend-Apr-29
@energyindepth.meta   @oeddep.meta          trend-Apr-30
```

```
@enr_gop              @ogjonline            trend-May-01
@enr_gop.meta         @ogjonline.meta       trend-May-02
@erlendjordal         @oilandgasiq          trend-May-03
```

All the used search terms a stored in the file: '_search-terms' [8]. Most of the search terms are based on an article[9] from 'Teknisk Ukeblad' where they list the twitter handles that are most significant in the oil industry.

The majority of the Norwegian economy is based on oil and gas exports. This means that other parts of the Norwegian economy is dependent on the oil industry in many ways. As an example, we have all the service industry around oil and gas, such as food supply, air travel, financial services etc. If the oil industry has a hard time, all the dependent companies will have a hard time to.

This leads us to believe that OSEBX, the Oslo stock exchange, is largely dependant on the oil industry also. Which makes our choice of tweets reasonable.

3.1.5   Problems, Shortcomings, and Possible Improvements

Retweets are a source of concern, they provide no information that is not present in other tweets. But they can provide information about sharing and the 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? A wide array of tests and analysis should be done to improve this factor of the problem.

3.2   Dictionaries

The dictionaries are lists of words used in the classification process. There are sets of dictionaries. In each set there are 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 words of all sorts.

---

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

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

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

Monograms are words consisting of one word. 'This', 'That' and 'Finance' are examples of that. Bigrams are words consisting of two words. 'Apple cake' and 'Operatin system' are two examples. For trigrams we have three words to a word, 'big blue boots', 'fast lane car', and 'waterfall boat accident'.

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(4.2.1) classification and the more advanced classification using SVM(4.2.3) and Naive bayes(4.2.3) classifiers.

3.2.1   Downloaded Dictionaries

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

Obama

The obama dictionary was created for the obama tweet set [10] in relation to the us presidential election of 2008.

In the positive list [11] there are 2230 words. Whether or not the frequency of political words are higher than other types of words are uncertain. *wisdom*, *truthful*, *profit*, and *intact* are words found in the positive list. Neither *intact* nor *wisdom* are words that can be described as positive or negative. This gives a clear indication that the dictionary can be improved.

There are 3905 negative words in the list of negative words[12]. 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.

---

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

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

[12]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 [13].

These dictionaries have a lot of potential. 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.

- 1: We import manually labeled the tweets

- 2: We take all the positive tweets and extract words from them.

- 3: We repeat step 2 with the negative tweets.

- 4: We remove words that are present in both the positive and negative dictionaries. Removal of duplicate words. Resulting in two dictionaries of unique words either positive or negative.

---

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

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*)

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 personality of the person labeling the dataset also effects the dictionaries.

3.2.3   List of dictionaries

Table 3.2.3, listing all the different dictionaries used. For each entry in the table there is a dictionary of negative words and a dictionary of positive words.

3.3   Finance Data

The finance data consists of records from Oslo stock exchange, OSEBX, one record for each opening day. The values we use are: *quote_date*, and *close*.

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[14].

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*.

---

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

Chapter 4

Sentiment Classification

Sections in this chapter are as follows. Manual classifications, 4.1, where we see how people classify tweets. Classification with classifiers follows in section 4.2. Containing 'Word count classification', 4.2.1, and classification with classifiers such as SVM and Naive Bayes in section 4.2.3. A comparison of the classifiers and associated results can be found in section 4.3. And a brief discussion come last, in section 4.4.

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 a 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. Further the aspect of sentiment in trading is interesting. Can sentiment improve trading in any way? And how can the sentiment be used to predict trends.

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 effects 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 actual negative tweets as positive then others.

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 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.

Obama tweet set

When labeling the obama tweet set we found that there were many retweets of other tweets in the dataset. Whether or not the data is representative for

twitter in general is difficult to say. Tweets favoring obama are positive for people who support obama and negative for supporters of Romney at the same time. This makes the tweet sset difficult to work with because tweets can be positive and negative at the same time, depending on who are reading them.

Kiro tweet set
The kiro tweet set also has a lot of retweets, but they are not used later. Removing the retweet increased the uniqueness of the data, as the same content do not show up multiple times. The search terms the dataset is based on could be broader to increase the diversity of the tweets.

4.2 Classification

4.2.1 Word Count Classification

We describe the classification process and the different parts of it. Then the results and discuss section looks at the drawbacks and results of this method. The algorithm counts the positive and negative words. More positive words than negative means the tweet is positive and vice versa.

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

Polarity is the difference in positive and negative words. The difference is expressed in the percentage of the total amount of words.

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.

Drawbacks
There are some drawbacks to the word count classification. The dictionaries and the threshold.

The threshold gives inaccuracy because tweets that have a polarity value that is the same as, or close to, the threshold might be classified wrong.

The quality of the dictionaries affects the word count classification. If the dictionaries had better quality we would get better results. An example of improvements would be to remove stop words before creating bi and monograms.

4.2.2 Threshold in Word Count Classification

Threshold is line where tweets are either positive or negative. It is compared to the polarity value. For a tweet to be classified as positive the ration of positive and negative words has to be above the threshold value.

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 greater than the threshold, the tweet is classified as positive.

The threshold is given to the classifier on initialization. We find out which threshold is best in section 6.3, on page 49.

Examples of classification
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

- polarity = pos - neg = 0.4 - 0.0 = 0.4

- threshold = 0.4 positive

- threshold = 0.5 negative

- threshold = -0.1 positive

Classification of t3:

- pos = 1 / 3 = 0.3333

- neg = 1 / 3 = 0.3333

- polarity = pos - neg = 0

- threshold = 0 positive

- threshold = 0.1 negative

- threshold = -0.1 positive

Drawbacks with the Threshold
When classifying with the word count classifier a 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.

The amount of tweets where the threshold an polarity are equal is shown in table on page 34. The number of cases where we get an equal amount of positive and negative words 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.

4.2.3   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 in section 6.5.

Table 4.1: 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

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.

SVM

Support Vector Machine [2], is a classification algorithm that uses a supervised learning model. The supervised learning is used with associated learning algorithms to recognize patterns and analyse data.

SVM builds a model, based on the training data, that split the data into two categories. New data will be assign one of the two categories. This makes the classifies a non-probabilistic binary linear classifier.

In more specific terms SVM creates multiple hyperplanes. The larger the distance from the hyperplane to the training data, the better the classification. The hyperplanes can be used for regression, classification or other tasks.

When using SVM there are different kernel modules that can be used to improve results. The kernel is often chosen based on knowledge about the dataset. In the nltk[3] library for python there are a number of different kernels.

The kernels were tested to find out which kernel gave the best results. Section 6.4 describes the experiment and the results. As seen in the experiment the LinearSVC kernel performed best and was used in further experiments.

Naive Bayes Naive Bayes [4], is based on Bayes theorem. And is a simple probabilistic classifier. It uses strong(naive) independence assumptions with the theorem.

The classifier assumes that the features are independent. That they have no connection to each other and that all features have equal importance. A conditional model is used for with the probability model. Abstractly we get this:

$p(C|F_1, \ldots, F_n)$

In plain English we have the reformulated model that can handle larger datasets:

---

[2]Wikipedia: https://en.wikipedia.org/wiki/Support_vector_machine
[3]Natural Language Toolkit http://www.nltk.org/
[4]Wikipedia: https://en.wikipedia.org/wiki/Naive_Bayes

$$posterior = \frac{prior \times likelihood}{evidence}$$

Even more specifically the implementation in nltk that we use are:

```
1                          P(label) * P(f1|label) *...* P(fn|label)
2  P(label|features)=----------------------------------------
3                          SUM[l]( P(l) * P(f1|l) *...* P(fn|l) )
```

Naive Bayes is efficient to train in a setting with supervised learning. It requires only a small training set. The classifier have worked well with complex real-world problems. Good results despite simplicity have been found theoretically sound. But Naive Bayes is still outperformed by algorithms such as 'random forests'.

4.3   Comparison

The Classifiers
We have looked at three classifiers, Naive Bayes, SVM, and the word classifier. 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.

Experiments
In section 4.2.1, 6.3, 6.4, and 6.5 the methods described in this chapter are tested to validate the research questions, 1.3.

Results
The results of the classifiers are described in 7.3.4. There the good and bad parts of the classifiers are presented and discussed.

4.4   Comments

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 should expand our knowledge toward the quality of the dictionaries. And look for ways to eliminate terms that are neither positive or negative.

We should cross classify and test the classification with the SVM classifier 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.

Biased Mind
The datasets of manually labeled tweets are biased based on a 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 source of errors to be explored more in 9.

We should explore the psychology of perception and classification. How do people perceive content differently? Is finance easier to label than politics?

Drawbacks
The quality of the dictionaries is a problem. 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.

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.

Future Work
Analysis of the dictionaries and improvement of them is a natural place to continue research. 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 look at trends on twitter and in finance and try to see if there are any correlations between them. Before we got into the trends we made some assumptions to set the trend context.

The Norwegian stock exchange, OSEBX, is largely an oil based exchange. And therefore we assume that oil related data will give a good indications of the behavior of OSEBX. Section 3.1.4 says something about why we choose tweets based on oil for the trend aggregation.

Further we wanted to look at trading that is not algorithm based, also known as speed trading. So we started looking at technical analysis. The technical analysis gives a base for decision support for the trader. The techniques of technical analysis gives us good indications about trends, but we want to take it further by involving sentiment.

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

---

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

In finance especially the trend can be a deciding factor in buying or selling of goods. You typically do not want to sell during a bullish trend, while the value is going up. And you want to sell before you get to the bearish trend, when the value goes down. Figure 5.1 shows basic trends on page 40. When markets do not trend they move sideways in trading ranges[2].
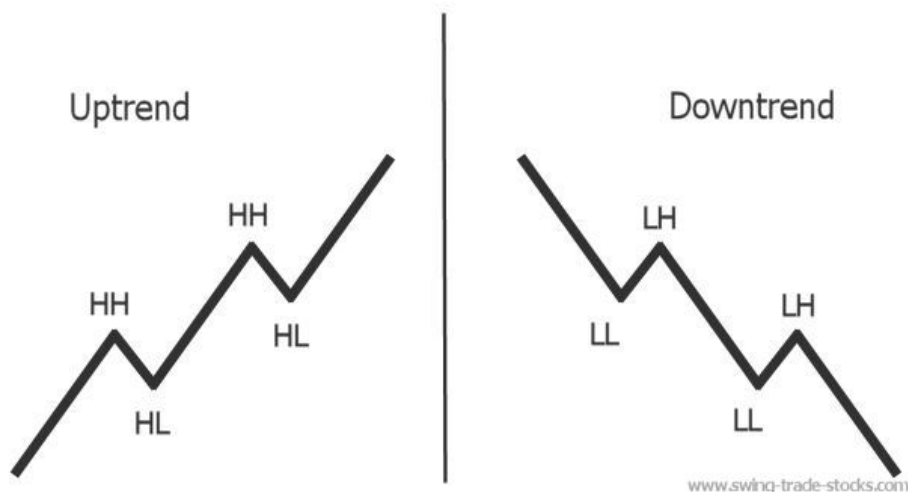
Figure 5.1: Basic trend
Showing Higher Highs(HH), Lower Highs(LH), Higher Lows(HL), and Lower Lows(LL).

Average directional Index Indicator, ADX, gives clear, easy-to-read picture of the market, shown in figure 5.1 on page 41. It shows if the trend strengthens of weakens. Moving averages and parabolic SAR can help to confirm a move or determine a trend. Moving averages indicates change over a period of time. The time frame is adjusted to the time the trader likes to keep stocks. The parabolic SAR looks at the price of a stock and determines the trend as upwards or downwards. When upwards the plotted dots are below the price, and vice versa.

When using ADX or other trend tools it is difficult to spot the trend or at all predict it. The ADX graph has some indicators of trends. The further away

---

[2]Swing-trade-stocks.com http://www.swing-trade-stocks.com/stock-trends.html
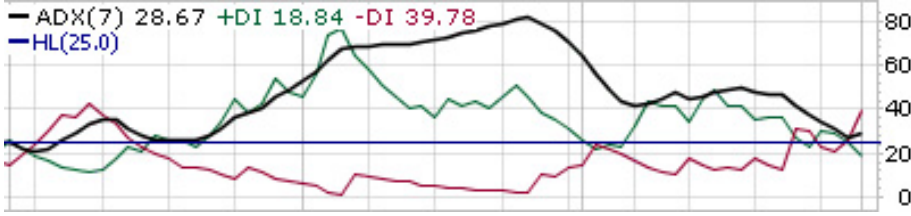
Figure 5.2: Average Directional Index example
An example of ADX. Green line is showing increase in price. Red line is
decrease. The black line is the ADX.

from each other the red and green lines are, the stronger the trend is. When
trading based on the trend, the problem is to find the trend and predict it into
the future. And this is where we combine Twitter with traditional trading. If
we could predict the trend based on public opinion mined from twitter we would
have an advantage.

Data is increasingly important for trend prediction. The better data we
have, the better trend we can predict. By focusing on sentiment we aim for the
cooperation of information available on the Internet and the value of sentiment
to predict trends. This will be more important in the future.

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

5.2 Trending in Finance

The average directional index and the moving average techniques for trend cal-
culation is described.

Moving Average(MA)
MA is the unweighted mean of the previous n days. Given the closing price of
the last n days as:

$$p_M, p_{M-1}, \ldots, p_{M-(n-1)}$$

We have that the simple moving average is :
$$SMA = \frac{p_M + p_{M-1} + \cdots + p_{M-(n-1)}}{n}$$

There are also other variants of the moving average. Such as cumulative MA, weighted MA and exponential MA[3].

ADX

ADX[4] uses the positive directional indicator(+DI) and the negative directional indicator(-DI) in combination to determine a trend. The high, low and closing values are needed for the ADX to be calculated. The directional movement is calculated as follows:

```
UpMove = today's high  yesterday's high
DownMove = yesterday's low  today's low
if UpMove > DownMove and UpMove > 0, then +DM = UpMove, else +DM = 0
if DownMove > UpMove and DownMove > 0, then DM = DownMove, else DM = 0
```

When the directions are calculated we choose the time frame we want to investigate. And get -DI and +DI:

```
+DI = 100 times exponential moving average of +DM
      divided by average true range
-DI = 100 times exponential moving average of DM
      divided by average true range
```

Where the time frame defines the number of periods used in the exponential moving average. And the average true is an exponential average of the true range. Resulting in the ADX:

```
ADX = 100 times the exponential moving average
      of the absolute value of (+DI  DI)
      divided by (+DI + DI)
```

Experiments

In the experiment, section 6.6 on page 50 we plot the graph and look at coding details for the finance trends. In the experiment we used data from the Oslo stock exchange in the same period as we took tweets, Apr 26 - May 26.

---

[3]Wikipedia: https://en.wikipedia.org/wiki/Moving_average
[4]Wikipedia: https://en.wikipedia.org/wiki/Average_Directional_Index

5.3   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 the trend data described in section 3.1.4, page 21.

On Twitter

On Twitter there is a feature called 'Trends'. It shows the words that are used the most lately. A screen capture of it can be seen in figure 5.3, on page 46.

The trends on twitter are in many cases predictable and adds little value to 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.

Our own

Using the trend data described in section 3.1.4 we construct a moving average graph and a ADX graph.

The referenced article lists the most important oil related account on twitter. In two parts. One for the Norwegian ones, and one for international ones. To simplify the trend creation a bit we used mostly English tweets, ignoring most of the Norwegian twitter handles.

After sorting the mined tweets by day we classify all tweets for each day. The classification from each day is referred to as a trend day. The trend day contains the amount of positive tweets, the amount of negative tweets, and the total amount of tweets.

```
trend_days = {
    'trend-Apr-30': {'neg': 112, 'pos': 131, 'tot': 243},
    'trend-May-19': {'neg': 523, 'pos': 1326, 'tot': 1849},
    'trend-May-18': {'neg': 59, 'pos': 110, 'tot': 169},
    'trend-May-15': {'neg': 1151, 'pos': 2255, 'tot': 3406},
}
```

Now we have some sentiment data to create a trend from. To simplify the trend aggregation and the graph plotting, we transform the data to fit into the format used for the finance data.

```
Date,close,high,low,open,volume
```

```
20140423,559.41,562.45,557.74,561.13,0
20140424,562.89,565.65,559.84,559.41,0
20140425,566.24,566.24,562.41,562.89,0
20140428,566.82,567.14,564.55,566.24,0
```

To transform the data we do the following:

```
1  trend_days = get_tweet_trend_data()
2  keys = sorted(trend_days.keys())
3  date = ""
4  for i in range(1, len(keys)):
5      if "Apr" in keys[i]:
6          date = "201404" + str(keys[i].split('-')[2])
7      elif "May" in keys[i]:
8          date = "201405" + str(keys[i].split('-')[2])
9      volume = trend_days[keys[i]]['tot'] * 1.0
10     # (positive_tweets / total_amount_of_tweets)*scale
11     high = (trend_days[keys[i]]['pos'] / volume) * 1000
12     # (negative_teets / total_amount_of_tweets)*scale
13     low = (trend_days[keys[i]]['neg'] / volume) * 1000
14     openv = 0
15     close = high - low
16     print str(date) + "," + str(close) + "," + str(high) \
17             + str(",") +str(low) + "," + str(volume / 10) + \
18             "," + str(0)
```

We scale the values to plot a more representative graph. All the proportions are kept intact, so the moving average and ADX is still valid. This method is tested in section 6.6 on page 50.

5.4   Comparing the trends

Comparing the finance trend and the Twitter based trend is easy. We look at the two graphs and draw conclusions. Drawing the conclusions are the difficult part.

We look at the two plotted trends, the moving average and the average direction index. More specifically we compare the white MA of both graphs with each other to. And the blue MA lines of each graph with each other.

With the ADX we will look for similar movements and turning points for the graphs. Also areas where the red and green graphs stay separate.

Further comparison should give some conclusions of the trust worthiness of the sentiment trend. For traders to trust in the sentiment as a new way of predicting trends the sentiment would have to provide profit and scientifically proven results. At this stage there would be few who would rely on sentiment as a predictor of trends. The trust lies in the technical analysis.

With the comparison we have the drawback of examining a stock exchange instead of one stock. Comparing one stock at a time would give better indications of the accuracy of the sentiment trend. This should be explored further in the future.

5.5 Future work

For future improvements of the trend aggregation we should into three aspects. The Norwegian market we started with, and explore the oil tweets. And find out whether or not they are a good indication of the OSEBX.

And we should look into single stocks in the comparison of trends. This way we can explore the stock vs stock Twitter trends, and find new information there.
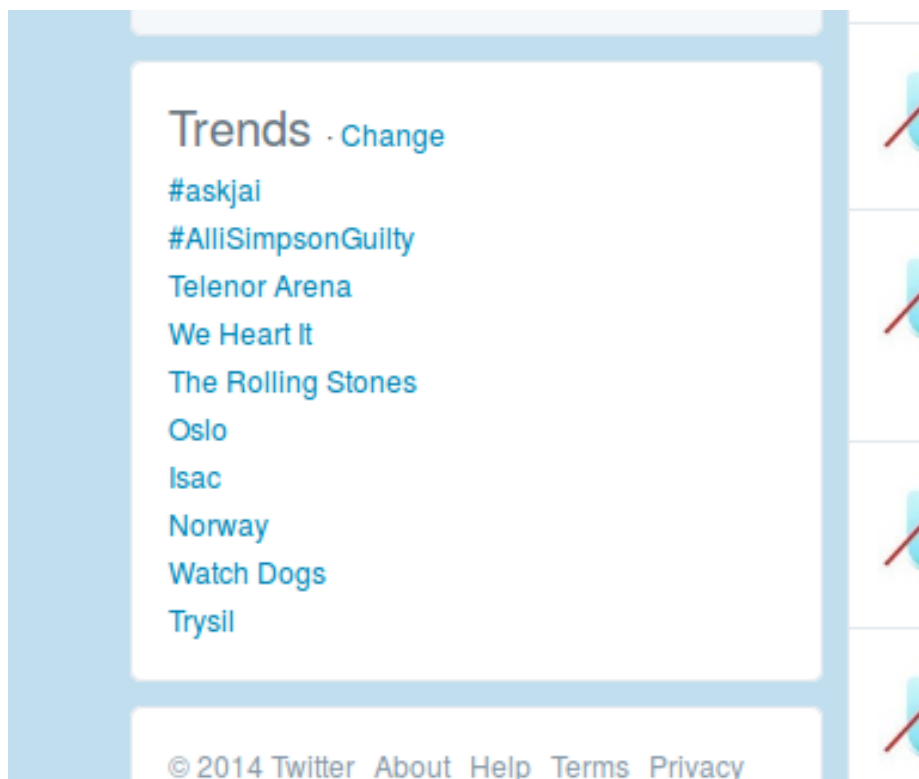
Figure 5.3: Trends on Twitter
Screen capture from Twitter.

Chapter 6

Experiments

This chapter aims to test methods described in earlier chapters. We describe setup and context for the experiments as well as what data was used.

The experiments described in this chapter are the following. Dictionary compilation, 6.1. The word count classification, bag of words method, 6.2. Threshold variation in section 6.3. Kernel choice for the SVM classifier is described in section 6.4. With the testing of classifiers coming next, 6.5. And the trend aggregation comes last in section 6.6.

6.1 Dictionary compilation

Section 3.2 describes dictionaries and what they are used for. This section describes how the dictionaries are tested. The dictionaries are compiled from manually labeled tweets.

As a step in the sentiment classification dictionaries was created for the word count classification 4.2.1. To find out which dictionaries performed well or not we tested the dictionaries with two data sets, the kiro dataset and the obama dataset.

This experiment enlightens part of research question 1, 1.3.1, how we can find the sentiment of a tweet.

Dictionary quality
To test the dictionary quality the word count classification 6.2 does it for us.

All the dictionaries are tested with the two datasets and we can compare the results. How the dictionaries are compiled are described in 3.2.2 and A.4.

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.

The code is described in appendix A.4 on page 79.

Results
Results from the testing of dictionaries are closely linked to the word count classification in 6.2 and described in table 7.3.1 on page 58.

6.2   Word Count Classification

Inspiration from section 2.2, with knowledge of tweets and dictionaries in section 3.1, builds this experiment. The background for the experiment and how it's purpose is described in section 4.2.

This experiment used the technique described in section 4.2.1, the word count method of classifying a tweets sentiment.

This experiment tries to expand on the matter of how to determine the sentiment of a tweet, 1.3.1. And how Twitter can be used, indirectly, to aggregate a trend, 1.3.2. The method looks at at the amount of positive and negative words and use that to decide the sentiment.

For all the dictionaries and datasets, the classification process is run. This gives the accuracy for all dictionaries and the two datasets. This in turn describes the efficiency of the word count classification.

The code for calculating the sentiment is as follows. It is also described in appendix A.5.1, on page 80.

```
1  positivity = positive_words_count / total_number_words
2  negativity = negative_words_count / total_number_words
3
4  polarity = positivity - negativity
5
6  if polarity > 0:
7      tweet is positive
```

```
8   else:
9       tweet is negative
```

The results of the classification can be found in section 7.3.1 on page 57.

6.3   Threshold Variation

In section 4.2.2 the description an purpose of the threshold variation is described. The threshold aims to find out how positive a tweet has to be before it is actually positive. Or rather how many more positive words than negative words do we need to call the tweet positive.

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 7.3.2, we can see no clear distinction of one value being better than the other ones.

In essence what is done is that the word count classification is run multiple times where the threshold value is changed between -0.9 and 0.9. This give a distribution of results which is plotted in graph 7.3.2 on page 61. The results will give an indication of what threshold is best for word word counting.

The results of the threshold variation is described in section 7.3.2, on page 59. And in table 7.3.2 on page 60.

6.4   Choice of SVM Kernel

In extension of 4.2.3 all the SVM kernels were tested. While testing which kernel was best we used the kiro dataset for training, and the kiro monogram dictionary for feature extraction.

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

6.5   With Classifiers

For the classifiers the two datasets was tested with the two classifiers. This was done to find the best classifier and compare the results with the word count classification. By confirming that classifiers are better than word counting we can eliminate a factor in the classification uncertainties.

The description of the comparison on a conceptual level is described in section 4.3. Also of interest there is the element of the dictionaries in section 3.2. While we have taken inspiration and ideas for the experiment from section 2.2.

Table 6.1: 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 6.2: 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

For the research questions to be enlightened further here is the trend aggregation in section 1.3.1 and the sentiment determination in section 1.3.2.

SVM
Results from testing SVM with different dictionaries are described in table 6.5 on page 50.

Naive Bayes
Results from testing Naive Bayes with different dictionaries are described in table 6.5 on page 51.

6.6   Trend aggregation

Before executing this experiment section 5.4 describes the conceptual parts of the trend comparison. This is the basis for the results of this experiment.

This experiment aims to test the relevance of positive and negative tweets in correlation with change in finance. There are two parts of this experiment.

Table 6.3: 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

The finance part, and the twitter part. Can twitter be used to create a trend, 1.3.2. And how can we compare this to a finance trend, 1.3.3.

In the execution of the experiment the methods described in chapter 5 has been used. More specifically the trend aggregation under trends on twitter, 5.3. And the finance trend aggregation, 5.2.

The tweet data used is described in section 3.1.4, on page 21. And the finance data is described in section 3.3, page 27.

The results of the experiment are discussed in section7.4.

Finance plot   The plotting of the graph is described in better detail in appendix A.6, page 81.

Plotting the finance trend we got figure 6.6, page 52.

Twitter plot   Code for plotting the trend graph is described in appendix A.6, page 81.

Plotting the moving average for the Twitter trend we get figure: **??**, on page **??**.

Figure 6.1: Finance trend plot
This is the plot for OSEBX. The plot show the moving average in the middle
and the Average directional graph at the bottom.

Figure 6.2: Tweet trend plot
This is the Twitter trend plot. The plot show the moving average in the middle
and the Average directional graph at the bottom.

Chapter 7

Results and Discussion

The results of our research and the following discussion of them happen here. We touch the three main parts of the thesis. The data sources, section **??**. Classification of sentiment in section **??**. And the trend in section 7.4. Last we have a general discussion about the results.

7.1   Data Source

Twitter as a data source proved to be OK. It was quite easy to obtain data. But increasingly difficult to get good amounts of data and unbiased data. All the initial data have some restriction to search words or other limits. In total we should have mined a lot more data and manually classified more of it.

As far as the dictionaries go they work well. We have different results for different dictionaries and different datasets, but all in all the way to compile datasets work good. The dictionaries depend on the manually classified tweets, so a lot of the potential improvement lies there. Although we can improve the dictionaries by removing stop words and other words we know have no clear positive of negative sentiment.

In research question 1, 1.3.1, we asked what knowledge we can extract from tweets to find a sentiment. The dictionaries are a good way to compose sentiment from the different words in the manually labeled tweets.

Research question 3, 1.3.3, asks if Twitter has been useful in comparison with technical analysis. To some extent it has. We have mined and classified many tweets in the hope that the aggregated trend would show likeness to plots of

technical analysis. The data from twitter that has been used to acquire positive and negative tweets for the trend aggregation. And has also provided useful insight into the use of the Twitter API.

7.2   Dictionaries

As for the use of the dictionaries in association with the trend and research question 3, 1.3.3, we use the dictionaries for feature extraction in the classifiers. Besides the use in classifiers the dictionaries have no other practical uses in this thesis in relation to sentiment.

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

7.3   Classification

Manual labeling is a bit of a challenge to keep unbiased. The whole classification process is based on a persons opinion, which makes the results somewhat biased. To eliminate this error we should have multiple people do the manual classification and combine the results.

Word counting works to some extent. But presents varied results based on the dataset and the dictionary used. Again this comes down to the dataset and the dictionaries created. Also the way we separate tweets from each other with a threshold can be improved. All in all word counting works, but it is not very good.

Classifiers improve the classification quite a bit. With the svm classifier we get around 98 percent correct classification on both data sets. Which is good.

The uncertainties lie in the biased, or potentially biased, data. We do not know if the data in itself is fully representable for the content on twitter. Neither do we know if the manual classification is completely correct.

In total we gained some knowledge about dictionaries and their effect on classification. And we can confirm that classifiers work better than word counting.

We tested two ways of finding the sentiment of a tweet. The bag of words method described in section 4.2.1, and the classifier way with the SVM and Naive Bayes classifiers, 4.2.

Research question 2, 1.3.2, the question of trend aggregation, the classification was useful. The sentiment worked to some extent. We do not know how well. And we have to look into the part of credibility in future work.

7.3.1 Word Count Classification

The results from the word count classification can be plotted in the tabled 7.3.1 and in the graph 7.3.1.

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.

Table 7.1: 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
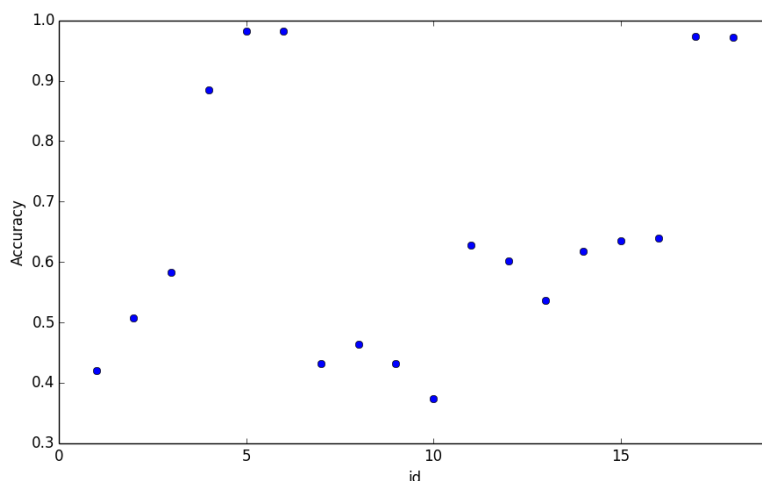
Figure 7.1: Dictionary Accuracy plot
The graph shows the plotted accuracies from the word count classification.
The ids represents the ids from the table above, 7.3.1

7.3.2   Threshold Variation

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

   In figure 7.3.2 we list the results of the experimentation with the threshold.
Table 7.3.2 lists the dictionaries and dataset used for which graphs in figure
7.3.2. 'kiro dataset' and 'obama dataset' columns tells which dataset that was
classified in which graph.

7.3.3   With Classifiers

Not surprisingly the classifiers have better results than the bag of words method
of classification. Further the SVM classifier with the LinearSVC kernel module
performed best. Naive Bayes classifier was nearly as good. This answers part
of research question 1, 1.3.1.

Table 7.2: 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

Table 7.3: 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
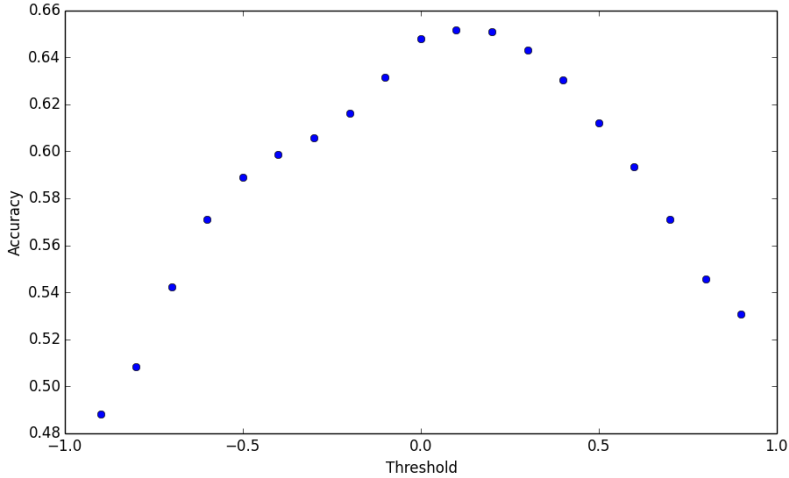
Figure 7.2: Average threshold accuracy
Graph plots of the 'Average threshold accuracy' table.

SVM
From results of the experiment, 6.5, we can draw some conclusions. The mono-gram 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.

Naive Bayes
As we can see from the experiment, 6.5, the different dictionaries makes no difference for the Naive Bayes classifier. But if we compare the classifier with the results from the word count classification we can clearly see improvement.

7.3.4 Comparison

In the following table, 7.3.4, we highlight the results from the different classifi-cations. We list the most noteworthy results from our experiments and compare them.
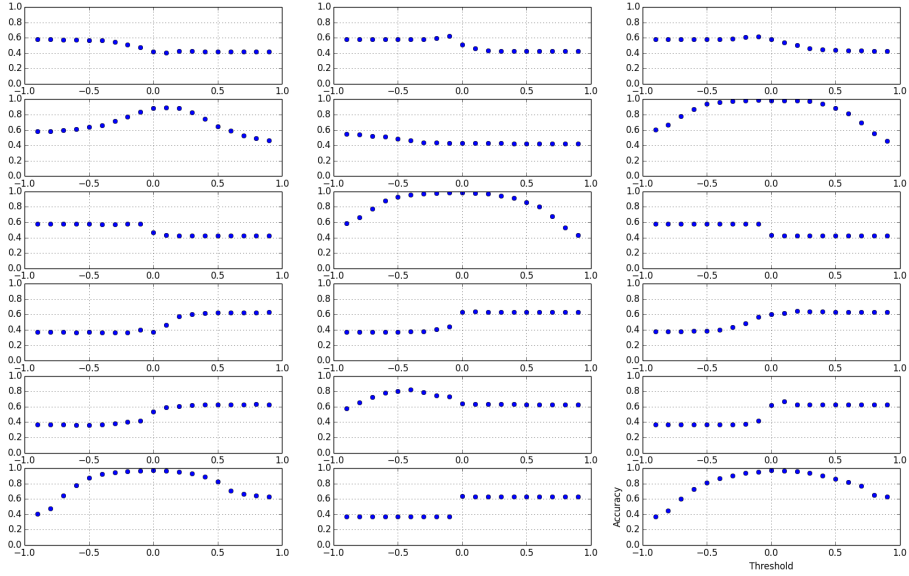
Figure 7.3: Threshold variation accuracy plot
The graphs plot the different variations of threshold. Counting is columns
first; top left is 1, top mid is 7, top right is 13.

As we can see the classifiers give better results then the word count classifi-
cation. 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.

7.4   The Trend

Comparing the two graph plots, figure 6.6 and figure 6.6, we can see minor
similarities. All the moving averages have rightward growth. While for the

Table 7.4: 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

tweet trend the 3 day interval preforms worse than the 8 day interval. There are no certain correlations between the two trends. And it is difficult to conclude on specifics.

For the research questions, 2(1.3.2), and 3(1.3.3) are in some degree true. RQ2, we can aggregate a trend. But the use for it and the accuracy we know nothing of. RQ3 has the answer poor. The sentiment trend compares poorly to the trend from the technical analysis.

7.5   Discussion

From the three parts we have satisfying and unsatisfying results. We have good results and vague results. The vague results should be frowned upon. While the reasonable results should be expanded, retested and improved. Most notably we have clear indications that dictionaries, automatically compiled from a manually classified tweets set, can be used when classifying sentiment.

TODO discuss the achieved research questions.

Chapter 8

Conclusion

We worked hard, and achieved very little.

TODO link up to research questions. TODO write stuff about data, datasets and dictionaries. TODO write classification and such stuff. TODO trend conclusions?

TODO overall work progress and process.

TODO evaluate accordance with the research questions.

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?

Language analysis   TODO How is abbreviations used in tweets? Same as in SMS? example: bc=because

Not computer science

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.

Credibility

Is neutral negative or positive?   TODO if a tweet is classified as neutral, aka the amount of positivity and negativity is the same how do we then classify the tweet?

Word weighting   TODO How can we use weighting of words in sentiment classification ?

How people express sentiment   TODO more of a psychology thing. But still relevant.

Use of sentiment analysis on news   TODO not the whole part of the news, but the title and ingress.

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.

Johan Bollen, Huina Mao, and Xiaojun Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1 – 8, 2011. ISSN 1877-7503. doi: http://dx.doi.org/10.1016/j.jocs.2010.12.007. URL http://www.sciencedirect.com/science/article/pii/S187775031100007X.

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

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, A.1. Then touch the technology and libraries, A.2, used up to this point. Before we look at the data mining, A.3, dictionary compilation, A.4, classification, A.5, and trend aggregation, A.6. Last we describe the comparison, A.7, of the finance trend and ending with the issues, A.8, of the code.

A.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,

73

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
        ma_adx_plotter.py
    - twitter
        mining_operations.py
        mining_utils.py
    graph_plots.py
```

A.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/.

A.3 Data retrieval

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

A.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.

```
1  conf = open('auth.cfg').read()
2  config = ConfigParser.RawConfigParser(allow_no_value=True)
3  config.readfp(io.BytesIO(conf))
4
5  # getting data from conf object.
```

```
 6  APP_KEY = config.get('twtrauth', 'app_key')
 7  APP_SECRET = config.get('twtrauth', 'app_secret')
 8  OAUTH_TOKEN = config.get('twtrauth', 'oauth_token')
 9  OAUTH_TOKEN_SECRET = config.get('twtrauth',
10                                  'oauth_token_secret')
11
12  # creating authentication object for twython twitter.
13  twitter = Twython(APP_KEY,
14                    APP_SECRET,
15                    OAUTH_TOKEN,
16                    OAUTH_TOKEN_SECRET)
```

Step 2
Query execution.

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

Where we have the query, and language as input variables to the *cursor_extraction* function.

Step 3
Data extraction and storage.

```
 1  # opens new file with today's date and time now as filename
 2  filename = destination_folder + "/dataset-" + \
 3      strftime("%d-%b-%Y_%H:%M:%S")
 4  # opens the file for appending.
 5  data_set = open(filename, 'a')
 6
 7  for result in results:
 8      # skip previously acquired tweets
 9      if str(result['id']) in str(previous_tweets_list):
10          continue
11      else:
12          previous_tweets_list.append(result['id'])
13
14      # store tweet to file for later use.
15      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.

A.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:

```
1  # get data file from internet. (csv)
2  stock_exchange_history = urllib.urlopen(url).readlines()
3  records = []
4  for record in stock_exchange_history:
5      # earlier records are not interesting.
6      if "2014" in record:
7          records.append(record.strip().lower().split(","))
8      if "20140414" in record:
9          break
10 return records
```

A.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:

```
1  data_files = [
2      # [name/description, name of file containing tweets]
3      ["kiro", "tweets_classified_manually"],
4      ["obama", "obama_tweets_classified_manually"]
5  ]
6
7  for item in data_files:
8      # get labeled tweets
9      # tweets[0] are the positive ones,
10     # tweets[1] are the negative ones.
11     tweets = get_tweets_from_manually_labeled_tweets(
12             classification_base + item[1])
13
```

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

Mono, bi and trigrams creation parts are quite similar:

```
1  for text in tweets[0]:
2      positive_dict.extend(text.split(" "))
3  # negative
4  for text in tweets[1]:
5      negative_dict.extend(text.split(" "))
6
7  filename += "-monogram"
8  save_dictionaries(positive_dict, negative_dict, filename)
```

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

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

Trigram:

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

We get the bigrams and trigrams by:

```
1  # if we have no words to work with, return empty list.
2  text = clean_text(text)
3  if not text:
4      return []
5
6  bigrams_list = []
7  # for all bigram tuples
8  for tup in bigrams(text.split(' ')):
9      # find the tuple texts
10     # (u'text1', u'text2')
11     m_obj = re.search(r'\(u\'(.+)\', u\'(.+)\'\)',
12                       str(tup).decode())
13     if m_obj:
14         # combine tuple parts to bigram
15         word = m_obj.group(1) + " " + m_obj.group(2)
16         # if bigram don't exist
17         if word not in bigrams_list:
18             # add it to list.
```

```
19                 bigrams_list.append(word)
20  return bigrams_list
```

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

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

```
1  primary_dictionary = get_lines_from_file(
2       primary_dictionary_name)
3  secondary_dictionary = get_lines_from_file(
4       secondary_dictionary_name)
5
6  duplicate_words = []
7  # for all words in the primary dictionary.
8  for pw in primary_dictionary:
9       #print w
10      # for all words in the secondary dictionary.
11      for sw in secondary_dictionary:
12          # if primary word equals secondary word
13          if pw == sw:
14              # remove word from both dictionaries.
15              primary_dictionary.remove(pw)
16              secondary_dictionary.remove(sw)
17              duplicate_words.append(pw)
18              # as we don't have duplicates within a list,
19              # we skip to next pw
20              break
21
22  # rewrite the updated lists to file
23  write_array_entries_to_file(primary_dictionary,
24                              primary_dictionary_name)
25  write_array_entries_to_file(secondary_dictionary,
26                              secondary_dictionary_name)
27  write_array_entries_to_file(duplicate_words,
28                              "duplicate-words")
```

A.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.

A.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.

```python
1  # sanitize text.
2  tweet = sanitize_tweet(tweet)
3
4  # get word count for tweet
5  word_count = len(tweet.split(' ')) * 1.0
6
7  # get word count of pos and neg words.
8  pos = get_word_count(positive_dict, tweet) / word_count
9  neg = get_word_count(negative_dict, tweet) / word_count
10
11 # storing the polarity value
12 polarity.append(pos - neg)
13
14 if polarity[-1] == threshold:
15     edge_cases += 1
16
17 # adding sentiment value (True/False),
18 # for the last classified tweet.
19 if polarity[-1] > threshold:
20     # positive tweet
21     results.append(True)
22 else:
23     # negative tweet
24     results.append(False)
```

A.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())'.

```
1  # get the training set.
2  training_set = nltk.classify.apply_features(
3      extract_features_from_text,
4      tweets)
5
6  # create the classifier.
7  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.

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

Classification   The code for classifying a tweet is quite simple.

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

A.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.

A.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.

```
 1 | pos , neg = 0, 0
 2 |
 3 | # get tweets for this day.
 4 | trend_day_tweets = [ast.literal_eval(tweet) for tweet in
 5 |     codecs.open(trend_base + trend_day_filename,
 6 |                  'r',
 7 |                  "utf-8").readlines()]
 8 |
 9 | # sentiment for each tweet.
10 | sentiment = []
11 |
12 | # for tweets in this day
13 | for tweet in trend_day_tweets:
14 |     # get words of two or more characters.
15 |     tweet_text = [e.lower() for e in sanitize_tweet(
16 |         tweet['text']).split() if len(e) >= 2]
17 |
18 |     # get sentiment
19 |     sentiment.append(classifier.classify(
20 |         extract_features_from_text(tweet_text)))
21 |
22 |     # if the last tweet was positive.
23 |     if sentiment[-1]:
24 |         pos += 1
25 |     else:
26 |         neg += 1
27 |
28 | return [pos , neg , pos + neg]
```

Then we transform the tweet data to fit the structure of finance data.

```
 1 | def transform_tweet_data():
 2 |     #Date ,close ,high ,low ,open ,volume
 3 |     #'trend -Apr -29 ': {'neg ': 21, 'pos ': 101, 'tot ': 122} ,
 4 |     #print "data ,h-l,pos ,neg ,open ,tot"
 5 |     trend_days = get_tweet_trend_data()
 6 |     keys = sorted(trend_days.keys())
 7 |     date = ""
 8 |     for i in range(1, len(keys)):
```

```
 9              if "Apr" in keys[i]:
10                  date = "201404" + str(keys[i].split('-')[2])
11              elif "May" in keys[i]:
12                  date = "201405" + str(keys[i].split('-')[2])
13              volume = trend_days[keys[i]]['tot'] * 1.0
14              # (positive_tweets / total_amount_of_tweets)*scale
15              high = (trend_days[keys[i]]['pos'] / volume) * 1000
16              # (negative_teets / total_amount_of_tweets)*scale
17              low = (trend_days[keys[i]]['neg'] / volume) * 1000
18              openv = 0
19              close = high - low
20              print str(date) + "," + str(close) + "," + str(high) \
21                      + str(",") + str(low) + "," + str(volume / 10) \
22                      + "," + str(0)
```

A.7 Comparison

For the comparison of trend graphs we have acquired some code. What we need is the moving average and the ADX.

The moving average is calculate by:

```
1 def movingaverage(values, window):
2     weigths = np.repeat(1.0, window) / window
3     smas = np.convolve(values, weigths, 'valid')
4     return smas  # as a numpy array
```

And for the ADX we have this:

```
 1 def ADX():
 2     window = 14   # modified by kiro
 3     PositiveDI, NegativeDI = calcDIs(window)   # modified by kiro
 4
 5     xxx = 0
 6     DXs = []
 7
 8     while xxx < len(date[1:]):
 9         DX = 100 * ( (abs(PositiveDI[xxx] - NegativeDI[xxx])
10                      / (PositiveDI[xxx] + NegativeDI[xxx])))
11
12         DXs.append(DX)
13         xxx += 1
14
```

```
15      #print len(DXs)
16      ADX = ExpMovingAverage(DXs, window)   # modified by kiro
```

The hundred and some lines used to plot the fancy graphs can be seen in the code file: https://github.com/magnuskiro/master/blob/master/code/trend/ma_adx_plotter.py.

A.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.

Appendix B

Processed Articles

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

B.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?

B.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?

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

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

B.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

B.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?

B.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?

B.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?

B.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]

B.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.

B.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.

B.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.

B.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?

B.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]

B.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]

B.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,

B.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 C

Tweet usage overview

TODO Write what twitter has been used for in 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 D

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 E

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'
  }
}
```