An Investigation of Sentiment Analysis for Political News Feeds

Seán O’Sullivan
B.A. (Mod.) Computer Science
Final Year Project April 2011
Supervisor: Prof. Khurshid Ahmad

School of Computer Science and Statistics
O’Reilly Institute, Trinity College, Dublin 2, Ireland
ABSTRACT

The aim of this thesis was to build a sentiment analysis system to automatically analyse the data from novel streaming data feed sources with a strong emphasis on U.S.A. political news in order to investigate the hypothesis that there is an effect arising out of emotional affect in the feed data. The system could then be evaluated against a real world futures market index in a case study based around the Republican Party 2012 primary nominations contest during the period August 2011 until April 2012. The results that were obtained were generally not conclusive but were interesting and none the less indicate that there may be promise in refinement of such an approach.
# Contents

Title Page. ................................................................. 1
Abstract ................................................................. 2
Table of Contents. ..................................................... 4

1 Introduction and Summary ........................................ 5
   1.1 Introduction .................................................... 5
   1.2 Summary ......................................................... 5

2 Literary Review and Motivation ................................ 6
   2.1 Chapter Overview .............................................. 6
   2.2 Sentiment Analysis ............................................ 6
   2.3 Sentiment and Affect .......................................... 6
   2.4 Language and Sentiment ...................................... 7
   2.5 NLP Approaches to SA ...................................... 7
   2.6 Text Based World ............................................ 8
   2.7 Novel Streaming Web Feeds .................................. 8
   2.8 Political News Feed Data .................................... 8
   2.9 Political Data Set ............................................. 9
   2.10 Hypothesis .................................................... 10
   2.11 Measuring Performance ..................................... 10

3 Method ................................................................. 11
   3.1 System Overview .............................................. 11
   3.2 Tools and Justifications ..................................... 11
   3.3 Building a Corpus of Data ................................... 12
   3.4 Accessing Corpus Data ....................................... 13
   3.5 System Data Processing ..................................... 13
   3.6 Bag of Words Model ......................................... 14
   3.7 Presentation of Results ..................................... 15

4 Case study and Discussion ....................................... 16
   4.1 Overview of Approach ....................................... 16
   4.2 Case Study for 2012 Republican Contest ................... 16
   4.3 University of Iowa Electronic Markets ..................... 16
   4.4 Data Sources for Case Study ................................. 17
   4.5 Political Data Feed Sources ................................ 17
   4.6 4.6 Technical Specifications for the Case Study Corpus ... 18
   4.7 All Candidates Data .......................................... 20 - 26
   4.8 Evaluation ...................................................... 26

5 Conclusion and Afterword ....................................... 27
   5.1 Conclusion ..................................................... 27
   5.2 Afterword ....................................................... 27

Bibliography ............................................................ 28

Appendix ................................................................. 28
Chapter 1

Introduction and summary

1.1 Introduction

The field of Sentiment Analysis has much promise in providing us with ways to extract meaning from data. Moving into the 21st Century we had become heavily reliant on textual data and much of the data we deal with day to day is text based. Web based data feeds provide us with up to the minute information. Much of this information is qualitative in nature, for example political opinions, and readily interpreted by people. The interpretation and collation of this qualitative political data by people seems to have an effect on outcomes of political contests. The goal of this thesis is to investigate the automated measuring of sentiment in such qualitative politics Web feeds.

1.2 Summary

Chapter 2 presents an overview of some of the literature and motivations for sentiment analysis. Chapter 3 gives specific details about the system design choices and outlines the important algorithms. Chapter 4 presents a case study for the system and discussion of the data produced. Chapter 5 concludes the thesis with conclusions drawn from the project and discusses possible directions for future work.
Chapter 2

Literature Review and Motivation

2.1 Chapter Overview

In order to prepare for working in the field of SA (Sentiment Analysis) it has been necessary for me to investigate a wide variety of sources of information. SA is a multidisciplinary field and has required that I became familiar with fields such as NLP (natural language processing), computational linguistics, text analytics with a smattering of psychology, political journalism as well as econometrics. What follows in this section is an outline of some of the important concepts from these areas that are most helpful in understanding this this thesis.

2.2 Sentiment Analysis

SA (sentiment analysis) refers to the application scientific methodology to automatically identify and extract subjective information from data. It aims to determine the attitude of a speaker or a writer with respect to some topic or the overall contextual polarity of a document. The attitude may be a judgement or evaluation, affective state (the emotional state of the author), or the intended emotional communication (the emotional effect the author wishes to have on a reader). It is statistical in nature and uses methodology to estimate the most likely emotional payload in a given communication. SA is important because it can be used to analyse a lot of data quickly with a reasonable degree of statistical accuracy. SA is currently employed in many fields of human endeavour where it can be applied to the analysis of textual or audio data sources. A primary use for sentiment analysis is investigation and explanation of an event after the fact e.g. after a market crash.

2.3 Sentiment and Affect

In SA (Sentiment Analysis) the term sentiment is used in its sense of a sentiment being an attitude, thought or judgement prompted by feeling. Sentiment bearing words or phrases can be viewed as containing payloads of human emotional affect. The word affect is used in the sense of the psychological definition referring to the felt or experienced part of emotion. Affect is a key part of any organism’s interaction with stimuli and operates at a
conscious and unconscious level. Sentiment may be used to communicate affect to people with the goal of influencing their behaviour.

2.4 Language and Sentiment
Words vary a lot in terms of their affect bearing qualities. There are a number of established approaches towards the scientific analysis of sentiment bearing words to determine the affect characteristics of a given word. Sentiment bearing words are used to put a qualitative value on a thing being talked about. A word’s specific context in a phrase changes the meaning and therefore the qualities of it’s affect. One of the most difficult tasks in SA is in disambiguating words. A popular method to disambiguate words is by analysing this context, known as part of speech (POS) tagging. A lot of words have no affect value at all and from a SA perspective serve as grammatical scaffolding for the sentiment in a sentence or utterance. There are open and closed class words with the closed class words providing grammatical cohesion and little, if any, affect value in an utterance. These are words like “this”, “a”, “the”, 12321(numbers) etc. The open class words provide lexical cohesion and consist of nouns, verbs, adverbs and adjectives which are relatively rich in their potential affect. Open classed words make up the ontological specification of the domain. If one performs a frequency count of words in a given text one can discover the ontology of the domain the text comes from. Language is complex but has a structure, otherwise it would not be understandable, and one exploits this structure in SA. Similes and metaphors make excellent hosts for affect. For example the metaphor “bull market” transfers the characteristics of the vehicle (the aggressiveness of a bull) to the tenor (the market in question). There is a knowledge is transfer into the simile (like/as) or metaphor (when he runs he is a speeding bullet).

2.5 NLP Approaches to SA
There are typically two main approaches from NLP to sentiment analysis, lexicon-based and machine learning. The lexicon-based approach uses a sentiment dictionary with a pre-definition of positive and negative words. Documents are rated with range of attribute values that characterise the sentiment. This approach depends on suitability of the dictionary for the application. The second approach uses machine classification techniques. Computers are trained on a part of a human annotated corpus. The performances of the algorithms are refined by testing on the other unseen part of the corpus. The approach depends heavily on having enough human labelled text suitable for the application. Making such human labelled text to a gold standard is very labour
intensive. Most treatments of social media style content choose a lexicon-based approach because of the challenge of obtaining enough human-labelled training data for the large-scale and diverse social opinion needed [cite?].

2.6 Text Based World
We people communicate in many ways. Most of our communication is face to face and interpersonal. Modern people have seen the increasing importance of text based communication. SMS is most widely used data application in the world with over 5 billion subscribers [Whitney (2010)]. It is estimated that up to 80% of all human knowledge now exists in textual form [cite?]. As far as quick access to the information is concerned our capacity to read is quicker than our capacity to listen. Text, being devoid of any facial or bodily expressiveness, depends on the authors writing skill to project an emotional payload into the communication. The skilful creation and distribution of text can be used or misused in influencing people. Ready access to all this textual data combined with the power of personal computing and the improvements in computer communications has led to a lot of attention to SA techniques in many different fields of endeavour such as stock trading or in market research. Many organisations are utilising these types of techniques to gain a competitive advantage.

2.7 Novel Streaming Web Feeds
A Web feed or (news feed) is a data format used for frequently updated internet Web pages. Content distributors syndicate Web feeds thereby allowing users to subscribe to the feed content. Creating a collection of web feeds available in one collection point is known as aggregation. There are a number of standards in use today the most common of which are versions of the RSS and Atom standards. As the world has shifted its attention to the digital medium so too has grown the popularity of Web feeds for delivering news as well as all other kinds of information both general in nature or tailored to the individual subscribers interest.

2.8 Political News Feed Data
Feeds provide us with novel, low latency, streaming, unstructured and spontaneous sources of data in the form of tweets, blogs and feeds. These sources are well established as part of the modern political news cycle providing us with a diverse range of up to the minute political news and commentary. The power of technology revolutions to facilitate this kind of political commentary is not new. With the spread of the European
style printing press, Pamphleteers spread political sentiments and ideas to a mass audience.

Who are the generators of this content? Everybody with interest in doing so seems. Chiefly this consists of journalists, academics and professional commentators with a fair showing of interested laypeople getting in on the action. This content is subsequently read by the journalists, academics, professional commentators and all the interested laypeople.

The interpretation and collation of this data appears to effect political discourse consequently affecting the political landscape. “Political Journalism” is frequently “Opinion Journalism” and therefore this data is frequently qualitative in nature. A good grasp of using sentiment for communication of affect to the audience is a mark of a great political opinion journalist. These qualitative communications are readily interpreted by human experts (and interested laypeople). Qualitative information like in this political data set is not at all easy for a computer to deal with.

2.9 Political Data Set
Why is the political data set so hard to compute? Well in short, it's not like movie reviews. The qualitative nature of the politics news set makes it a difficult class of data set for automated methods of analysis in comparison to for example the data set of movie reviews. A data set for movie reviews will tend towards a popular consensus due to the shared notions of quality in cinema. A data set for reviews of political candidates will tend not to due to our tendency towards differences in political opinion. There are many diverse and significant effects at work motivating the generator of such data. Personal ideals, political motivation, current public opinion, propagandism and sensationalism are but some factors which may effect the content. This qualitative data can be aggregated, using statistical methods, into a quantitative data set from which inferences can be drawn. There is information to be had from analysing characteristic changes in news flow over time. We can analyse news items bearing key phrases or specific citations to explore their relationship with real world indices.

2.10 Hypothesis
The hypothesis of this thesis is that there is a measurable effect which arises due in part to affect expressed in these novel sources of news data. It’s not a new idea by any means:

“I say that all men when they are spoken of, and more especially princes, from being in a more conspicuous position, are noted for some quality that brings them either praise or censure.” [Machiavelli (1513)]

So the question is can we show that language has an action? Can I implement a software system to measure the affect from qualitative political data on a quantitative political decision with a reasonable degree of accuracy? Can this system be used in an investigation of how affect has changed during the event space of the political cycle in question?

My task therefore was to specify design and prototype a system for political sentiment analysis with special emphasis on sentiment bearing politics feeds containing news summaries from the U.S.A. I would investigate methods to create corpora of structured text data from the feed sources. With these corpora I would investigate machine-based approaches for analysis of this political data set and try to assess their suitability, by comparing the analysis with a real world index.

2.11 Measuring Performance
In order to measure the affect against a real world index I have based my approach largely upon using ideas from econometrics, in particular the idea of the return [Taylor (2005)]. This application of econometrics ideas to sentiment data is illustrated in [Ahmad (2008)] and this thesis builds on that work. The return is defined as the logarithm of a final value over that of a previous value.

\[
\text{Return} = \log ( \text{ValueToday} / \text{ValueYesterday} )
\]

The return is a dimensionless number useful for comparing two data series for correlation. Positive return values mean the series is increasing in value and negative values mean it is decreasing. Ideally you want to show the affect. Citation counting can be a useful indicator of market sentiment towards a candidate as discussed in [Ahmad (2008)]. Additionally looking at the volume of trade in a commodity gives another measure of market sentiment.
Chapter 3

Method

3.1 System Overview
A system to collect and measure affect from streaming politics news feeds needs to be able to connect to an Internet link in order to access the feed data text. The system must be able to parse this data and store each summary article locally with its date of publishing. The system must implement additional functions useful for accessing and analysing the data in the created corpora. The system must have a mechanism for measuring affect from summary articles. The system must be able to output the results for analysis.

3.2 Tools and Justifications
With the basic system outlined the next step was to identify suitable software tools which would be most apt to the task of developing the system to run a fairly modest desktop PC with a basic DSL internet connection.

The first major decision point was in choosing a language to work with. Currently there are over thirty software toolkits devoted to NLP available to the budding researcher. The vast majority of them are written in Java with most of the others being various flavours of C. There were also a couple of toolkits based on Python. I decided to go with the Python language for a number of reasons. I have had a fair bit of experience with both Java and C like languages which can be very powerful for certain computations but often are quite tricky to get up to speed for a given application domain. They are languages one might use when program speed efficiency is a primary design goal. With the processing power of modern CPUs the general emphasis on program speed efficiency has diminished as the most modern consumer machines have plenty of muscle for most general tasks. I took the view that for the proposed design that any speed advantage of the low level languages, based on C or Java, is outweighed by a gentle learning curve, library module support, ease of use in a modern high level language like Python. Python has a simple syntax and the shallow learning curve well suited to researchers without a strong programming background. To illustrate this fact I should tell you that before I began the project the only Python experience I had was having read the Wikipedia article. Pythons textual data handling makes it apt for text based linguistic data. Python is held in high regard for it's
powerful and extensive function library support enabling users to get easy access to a variety of computational tools. Python’s modular design allows easy addition of these external modules to support specialised tasks such as parsing, plotting graphs etc.

Having settled on a language, I next selected NLTK (Natural Language Toolkit - http://www.nltk.org/) as the framework for development of the NLP components of the system. NLTK defines an infrastructure for NLP work using Python (http://www.python.org/). It provides basic classes to represent NLP data and standard interfaces for common NLP tasks. It can be used to perform many functions such as part-of-speech tagging, syntactic parsing and text classification. There is extensive online free documentation and support for learning the framework. NLTK is an evolving toolkit rather than a system and so is not highly optimised for run-time performance. Mitigating this is Python’s facilities for interfacing with lower level languages for more computationally expensive problems.

The system needed a means to connect to an Internet link in order to access the feed data text. This feed data comes in a fairly wide variety of versions and formats and so the system would need to deal with them. Creating solid code to handle these kinds of formatting issues is far from trivial and I doubted that I would do well in reinventing that wheel. To achieve the required system functionality I made use of the excellent Python module Feedparser (http://pypi.python.org/pypi/feedparser) by Mark Pilgrim.

3.3 Building a Corpus of Data

With the ability to get the feed information from the internet into Feedparser format the next step was to store this textual data on the hard disk in some kind of structured way which is of use for NLP tasks. There are different factors which must be considered when choosing a particular storage structure for a corpus depending on the source of the text. If you wished to store a novel such as “Moby Dick”, you might want to structure it from the title with a hierarchy from the chapter level down through paragraph level down through sentence level ending at individual words. This structure would allow you to pick out chapters, paragraphs, sentences or words as appropriate to your NLP application. For a given streaming feed data source we are interested in the set of summary articles. Summary articles consist of: title text, summary text, hyper-link to article and date of publishing. We are only interested in title, summary and date in the context of this system. As the feed data is based in XML format to begin with I thought that building an XML
corpus structure would be apt for the system. I used the Python ElementTree (http://effbot.org/zone/element-index.htm) module to build up an XML corpus structure from the Feedparser data. The system creates a corpus file which consists of the feed’s identifying title information as the head and each summary article in the feed is represented as an item. Items have child items for title, summary and date.

3.4 Accessing Corpus Data
The XML file for a corpus when large can take up a lot of space in computer memory. It is often not efficient to load the whole corpus into working memory to perform NLP. The NLTK provides a class XMLCorpusReader which is used to read the XML file data from a collection of corpus files. This file data is used in conjunction with the class XMLCorpusView which provides a flat list like access to an individual XML corpus files elements such as just the item titles or dates etc.

3.5 System Data Processing
I decided that the lexicon-based approach would be the best method to use given that I did not have access to suitable human annotated test data for the political data set with which to train an algorithm on. The lexicon-based approach requires a lexicon and in this case I choose to use SentiWordNet 3.0 (http://sentiwordnet.isti.cnr.it/). There were many options for the affect dictionary such as General Inquirer or WordNet-Affect and I would have liked to experiment with using multiple dictionaries of affect to compare result. The choice of SentiWordNet 3.0 was based on the fact that it has been updated very recently (June 2010 at time of writing) and straightforward to use with the aid of some useful Python scripts from Christopher Potts (http://compprag.christopherpotts.net/wordnet.html). SentiWordNet 3.0 is a lexical resource devised for supporting sentiment classification applications [Baccianella, Esuli, and Sebastiani]. It is the result of automatically annotating all the WordNet synsets. A synset or synonym set is a group of terms that share similar meaning or similar semantics, e.g. car, ambulance and jeep are all in the same synset for vehicle. WordNet is a freely available gold standard (which is as good as it gets in the field of SA) lexical database for the English language. It is considered a gold standard resource as it has been compiled by humans since 1985. It groups English words into synsets, provides short, general definitions, and records semantic relations between these synonym sets. WordNet is a combination of dictionary and thesaurus that is more intuitively usable for automatic text analysis and artificial intelligence applications. WordNet's database contains 155,287 words organized in 117,659 synsets for a total of 206,941 word-sense
pairs. SentiWordNet 3.0 is itself not a gold standard resource like WordNet as it is devised in an automated fashion rather than being created by human annotation but it should hopefully serve well enough for this system. In essence SentiWordNet will give you a positive and negative score for any term it can match to a synset.

### 3.6 Bag of Words Model

As I wanted to deal with very large amounts of data I choose to use a “bag of words” approach to deal with sentiment analysis for the summary articles. Each summary article in a feed is viewed as an unstructured bag of words. The sentiment extraction algorithm can be summarised as follows:

1. Each summary article is viewed as raw text.
2. Generally one is looking for a specific keyword(s) within the article such as “Romney” or “Republican” in order to go on to more intensive analysis.
3. When an article is found that contains the keyword the next step is to tokenize the text information using NLTKs tokenization functionality to obtain a list of lexical tokens.
4. This list of lexical tokens is then tagged using the default NLTK n-gram tagger. This tagging creates a list of tokens (word sense pair tuples) tagged with the specific part of speech derived from the structure of the input text token. The tagging stage is the most computationally intensive part of the system.
5. Once the tagging is completed, the common English stop words are removed as they contain no useful sentiment data for this approach and their removal offers a speed improvement in the algorithm.
6. The lists of tagged tokens are then sanitized using Christopher Potts script to prepare them for use with SentiWordNet 3.0. This means words end up part of speech tagged as either nouns, adverbs, adjectives, verbs or none (where none of the other types to begin with).
7. Each sanitized tagged token tuple is then checked against the WordNet synsets in order to find the statistically most common word sense for that part of speech.
8. Armed with the most likely synset word sense the positive and negative scores are both recorded and then both divided by the number of tagged tokens in the input set for the article. In this way sentiment is recorded at the article summary level.
9. The values from many articles are accumulated in this way in a dictionary, indexed by the day of the article, yielding both a total positive and total negative sentiment score by day.
Separate to this there is additional functionality in the system for counting the frequency of given keywords over every token in the text. The system stores every lexical token in the text corpus (normalized to lowercase) by its published day. The frequency for the keyword token (lowercased) is then calculated using the NLTK conditional frequency distribution class.

3.7 Presentation of Results
The system outputs graphs for citation count, positive and negative sentiment using MathPlotLib’s PyPlot module (http://matplotlib.sourceforge.net/). I had intended to perform graphing using PyPlot but I found the module documentation far too cryptic for my tastes. More research led me to the conclusion that the best application for graphing data was Microsoft Excel so I opted to also present the data in CSV format with a view to working with it in Excel, which would also provide some very nice statistical analysis functionality.
Chapter 4

Case study and discussion

4.1 Overview of Approach
To analyse the performance of the system with respect to political news feeds the approach used is to find a suitable political event which generates a lot of qualitative data which also has some quantitative measure of performance associated with it.

4.2 Case Study for 2012 Republican Contest
The case study is based around the 2012 Republican presidential primaries. The primaries are the selection processes in which voters of the Republican Party will choose their nominee for President of the United States. This campaign will generate a lot of qualitative politics news data in feeds. The campaign runs from 2011 until the week of August 27, 2012 when at the Republican National Convention delegates of the Republican Party choose the party's nominees for President and Vice President. The person chosen to be the presidential nominee is the winner of this contest. We are interested capturing the changes in sentiment towards candidates in a test set of data gathered from political news feeds over the period of the contest. The time in the case study is bounded from 30 August 2011 to 15 April 2012. With the withdrawal of Rick Santorum, on 10th April, Mitt Romney really the only candidate likely to get the delegate count necessary to win the contest.

4.3 University of Iowa Electronic Markets
The University of Iowa College of Business runs IEM (Iowa Electronic Markets) for research and teaching purposes (http://tippie.uiowa.edu/iem/). The IEM 2012 U.S. Republican Nomination Markets are real-money futures markets where contract payoffs are to be determined by the outcome of the 2012 Republican Nomination process. In particular RCONV12 (http://tippie.uiowa.edu/iem/markets/pr_rconv12.html) is a winner take all market based on the outcome of the Republican National Convention. RCONV12 runs from August 2011 until August 2012. The IEM website provides data for the period of the contest and effectively gives a dollar value for each major candidate. If a candidate is popularly expected to do well their value rises and vice versa. The historical performance of this market can be compared to a systematic analysis of the media sentiment over the time bounds of the contest and provides us with some quantitative measure of individual
candidate’s performance. Betting data is a better indicator of sentiment than opinion polls as people tend to express who they want to win in an opinion poll where as if they have to put their money where their mouth is they will tend to be pragmatic and bet on who they think will actually win.

Candidates are identified with codes as follows:

PERR_NOM – Rick Perry
ROMN_NOM – Mitt Romney
RROF_NOM – If another candidate wins (effectively Rick Santorum)
GING_NOM – Newt Gingritch

4.4 Data Sources for Case Study
In order to get a suitable data set for use in the project it was necessary to accumulate a number of sources of political news data feeds. The sources would obviously have to be streaming RSS/Atom feeds containing articles or summaries of articles with a strong emphasis on American politics. My aim was to try to balance political polarities by having a representative spread of opinion to represent the variety opinions expressed by the generators of this content.

4.5 Political Data Feed Sources
There are over 50 individual US politics feed sources in the data set. The feed source URLs have mostly been gathered using a combination of searches, recommendations and bundles. Additionally feed URLs have been gathered from major news organisations by copying the links from their websites. In all cases I have manually inspected each feed to ensure that the content is related primarily to U.S. politics. The feed source includes popular blogs, streaming news aggregation services and a variety of other generators of similar political news and commentary. One of the best sources for news feed data available to researchers is from Google. Google Reader is a Web-based feed aggregator. It is feature rich, for feed management in general, but in particular features an advanced search function to gather and organise feeds to an online account in a manner similar to Gmail email storage. It is capable of providing recommendations for feeds based on the user’s historical interaction with the service. It provides bundles of feeds based on expert recommendations for specific domains of interest. Additionally one can export a listing of all the feeds you have subscribed to in an XML style text file.
One of the characteristics of streaming news is that publishers of feeds do not typically serve more than a short number of day’s worth of news summaries. One additional important feature of the Google reader service is that it is capable of displaying historical feed data going back in many cases for several years. Typically news aggregator software simply gets the current feed data and then builds a local repository of this data over time and thus cannot be used to look at historical posts. This prompted me to try to find a way to extract this historical information from the service. I have discovered that there is an unofficial API for programmatically interacting with the service and can use this to create scripts that download historical feed data. I have found that this approach is very useful in general for building large corpora based on feeds for natural language processing. Using the Google API the initial test, using the CBS News, feed produced a corpus with over 90,000 summary articles comprising over 4 million words over a six month period.

4.6 Technical Specifications for the Case Study Corpus

- Total tokens in corpus: 49,591,490
- Total number of summary articles in corpus: 186,925
- Time span: 30 August 2011 – 15 April 2012
Frequency distribution of feed sources by number of words in the corpus.
4.7 All Candidates Data

IEM Futures Market RCONV12 Index Values at Daily Close of Market

% of Words Containing the Citation of the Candidates Second Names
Mitt Romney
Newt Gingrich
Investigating the hypothesis involves looking to find the changes in sentiment which precede the changes in the real world values. I think that the graphs show at least some evidence of the affect effect. I think that they indicate the nature of the political data set which is noisy with sentiment. It is no surprise that the positive and negative sentiment returns are quite closely correlated given the variety of news and commentary in the data set. It is interesting to note that the dollar volumes traded were high when the citation counts were high showing some correlation. The best indicator of where the smart money was going was in the overall relative citations counts as a percentage of words for the candidates, which lends weight to the work of [Ahmed (2008)]. The leading candidate was the generally the guy with the most citations, note how Perry starts the series with more citation count and a higher index value than Romney. As Perry’s stock drops the citations in the text corpus follow suit.
Chapter 5

Conclusion and Afterword

5.1 Conclusion

The political data set is indeed difficult and noisy; a lot of things can affect the market price outside of what we pick up in textual sentiment. There is no representation in this model for the sentiment created by the action of shifting money around in response to the larger political context. There may indeed be some correlation in the data but more investigation must be done in order to substantiate any such claim.

The sentiment detection engine in this system is only as good as SentiWordNet 3.0 so a switch of the governing sentiment lexicon would be an interesting follow up in order to measure relative performance on the same data set.

5.2 Afterword

There are many other algorithms from NLP which might be applied for example for part of speech tagging. It would be interesting to look at subsections of the feed data to see if certain feeds correlate to the index more closely than the average. More research is needed to gather more feed sources to try to get a good spread that reflects the infosphere of political news feeds.

I think it would be useful to work on analysing the ontology of the text in the case study to find the most common domain terms. It is important to see what the sentiment analysis step is doing with these important often sentiment bearing words.

The corpus building approach used in the system can produce some very impressive numbers when it comes to the creation of a large corpus of data thanks to some Google API magic. The current system design is nicely generic enough to be quickly applied in any other feed based data for ad hoc sentiment analysis. All one needs is list of feed URLs and some keywords and you are away, of course I have learned that it helps to get familiar with statistics first.
BIBLIOGRAPHY

[Whitney (2010)] Lance Whitney for CNET Reviews:
http://reviews.cnet.com/8301-13970_7-10454065-78.html

[Machiavelli (1513)] Niccolò di Bernardo dei Machiavelli, The Prince, 1513


[Ahmad (2008)] The ‘return’ and ‘volatility’ of sentiments: An attempt to quantify the behaviour of the markets?


APENDIX

Attached is a copy of the source code used in this project in CD format.