
# Sentiment Barometer in Financial News

## Project Results

Master Course "Machine Learning"
$3^{rd}$ semester, Master of Science in Information and Computer Sciences

Winter Semester 2013/14

C. Schommer, Dept. of Computer Science and Communication, FSTC

## Acknowledgement

These projects have been performed within the course Machine Learning, $3^{rd}$ semester of the Master of Science in Computer Science, University of Luxembourg, in the Winter Semester 2013/14.

The projects are motivated by the research project ESCAPE, which is kindly funded by the Fonds National de la Recherche (FNR) for a duration of 3 years (2012-15).

The author thanks the following students for their participation and support (in alphabetical order): Mr. **Saeed Afshari**, Mr. **Arash Atashpendar**, Mr. **Thierno Diallo**, Ms. **Hera Margossian**, Mr. **Fabien Mathey**, Ms. **Tahereh Pazouki**, Mr. **Valentin Plugaru**, Mr. **Alain Pfeiffer**, Ms. **Kavita Rege**, Mr. **Korok Sengupta**, and Mr. **Fabrice Thill**.

# Contents

# List of Figures

# 1 Introduction

As mentioned in [32], the European Financial Crisis has emerged within the last years, with many ups and downs, with many consequences and decisions for politics and economy. For example, Eurobonds have been suggested, attracting a great deal of attention while financial news appeared in a Tsunami-style of eruptively flowing pace.

Ireland has declared it had been in a recession (September 2008; Source: en.wikipedia.org) and, therefore, officially has asked for monetary support by the Euro Bailout Fund and the European Stability Mechanism (ESM), being the second EU country after Greece.

The European Commission [7] publishes macroeconomic forecasts for the Eureopan Union and the member states three times a year (May, November, and February). These forecasts are produced by the *Directorate-General for Economic and Financial Affairs* (DG ECFIN). In October 2013, it has been announced that Ireland has got out of the Euro Bailout Fund (see also Figure 1.2.3).

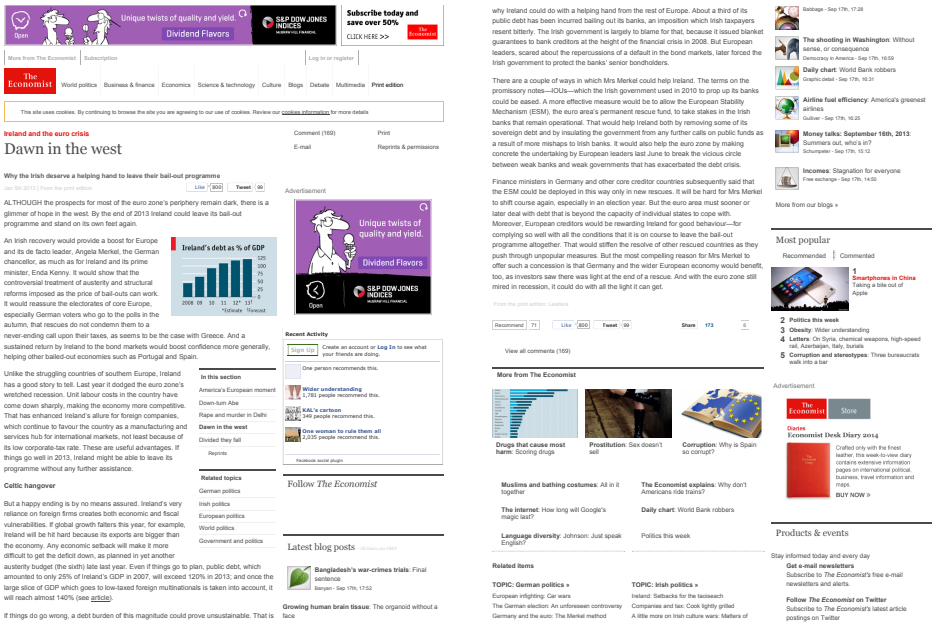



Figure 1: Irish Crisis (The Economist, Jan $5^{th}$ 2013

Besides, financial and political activities have taken place, political communities have emerged, and coalitions established. Also, a certain number of states have been down-rated, Greece (and other European states) are close to insolvency. All these information has been well-noted in Financial News.

## 1.1 Sentiment Detection

We concern with Financial News from Thomson Reuters, which represent a reflection of momentary political, economical, and financial incidents. Financial text news can influence decisions,

„Gipfel ohne Inhalt" worden.
Die Dänin gab sich betont optimistisch; sie habe in New York einen neuen Zungenschlag vernom-

Ministerrat, Kommission und Europaparlament – schon in der ersten Lesung verabschiedet werden soll. Deshalb müsse das „Gesamtgleich-

dass dieser Kompromiss noch weiter verwässert werde, erklärte er gegenüber dem „Luxemburger Wort".

Damit ist vorerst weiter offen, wann und wie der eigentlich für das Jahr 2020 angepeilte Zielwert von 95 Gramm $CO_2$ pro Kilometer er-

Autobranche.
Daimler-Chef Dieter Zetsche sagte: „Wenn wir von den Emissionen ausgehen, sind wir sicherlich, wenn

## EU gibt Irland noch keinen Freifahrtschein

### Irischer Tiger steht langsam wieder auf – mit wackligen Beinen

**Luxemburg.** Erfolgsgeschichten in der Euro-Rettung sind rar und spärlich gesät. Umso größer die Freude am Wochenende, als Irland ankündigte, als erstes Land den EU-Rettungsschirm zu verlassen. Vom 15. Dezember an sei es soweit, kündigte Ministerpräsident Enda Kenny an. Irland nimmt wieder selber Schulden an den Kapitalmärkten auf. Neue Hilfskredite soll es nicht geben. Die „wirtschaftliche Notstandssituation" in seinem Land sei vorbei.

EU-Kommissar Olli Rehn erinnerte die Regierung gestern allerdings daran, dass Irland vor seinem endgültigen Abschied aus dem Hilfsprogramm Auflagen erfüllen muss. Besonders wichtig seien die noch laufenden Stresstests für Irlands Banken, sagte Rehn am Rande des Treffens der Euro-Finanzminister in Luxemburg.

Die Banken hatten Irland ins Desaster gerissen. Der einstige „Keltische Tiger" steht nach wie vor auf wackligen Beinen. Eine Rückkehr zur einstigen Stärke, mit der er An-



**Enda Kenny:** *Die „wirtschaftliche Notstandssituation" ist vorbei.* (FOTO: AP)

fang des Jahrtausends das Leben vieler Iren nachhaltig veränderte, scheint bis auf weiteres nicht in Sicht. Selbst Kenny musste bei allem Stolz zugeben, dass dem Land weiter unsichere Zeiten bevorstehen: „Wir haben noch einen langen Weg vor uns." Er lobte das „große Opfer" seiner Landsleute, stimmte sie damit aber auch gleich darauf ein, noch ein bisschen durchzuhalten. Denn im Haushalt 2014, den Finanzminister Michael Noonan an diesem Dienstag vorstellt, geht es nochmal zur Sache. Kürzungen und Steuererhöhungen im Umfang von 2,5 Milliarden Euro stehen an. Erwartet wird, dass das Geld vor allem aus Änderungen bei der Besteuerung von Wohneigentum, aber auch bei der Sozialversicherung und der Sozialhilfe kommen wird.

Es sei der letzte „wirklich harte" Sparhaushalt, beteuert die Regierung, doch die Wut und Sorgen der Iren beruhigt das nicht. Sie finden, Menschen mit niedrigen und mittleren Einkommen seien in den vergangenen Jahren unverhältnismäßig stark getroffen worden.

So hat die durchschnittliche irische Familie Berechnungen der Wirtschaftsprüfungsfirma Grant Thornton zufolge heute rund 300 Euro weniger im Monat zur Verfügung als vor Beginn der Sparhaushalte 2008. Eine Familie, in der beide Partner 40 000 Euro im Jahr verdienen, muss ein monatliches Minus von 511 Euro hinnehmen. Alleinerziehende Eltern, die im Jahr 40 000 Euro Einkommen haben, haben seit 2008 Steuererhöhungen von 125 Prozent erlebt - und monatlich 300 Euro weniger auf dem Konto. Die Finanzkrise 2008 hatte Irland, das wegen eines kometenhaften Aufschwungs Anfang des Jahrtausends zum „Keltischen Tiger" ernannt worden war, ins Wanken gebracht.

Am Rande des gestrigen EU-Finanzministertreffens in Luxemburg trafen sich der Luxemburger Außenminister Jean Asselborn und der neue irische Minister für Europäische Angelegenheiten, Paschal Donohoe. Bei dem Gespräch ging es im Hinblick auf die Luxemburger EU-Präsidentschaft im Jahr 2015 u.a. um die Erfahrungen der Iren während ihrer diesjährigen Ratspräsidentschaft, um den geplanten Ausstieg Irlands aus dem Euro-Rettungsschirm sowie um den Haushaltsstreit in den USA und mögliche Auswirkungen auf die Weltwirtschaft. Donohoe vertrat den irischen Finanzminister Michael Noonan, der beim gestrigen EU-Finanzministertreffen nicht dabei sein konnte. (dpa/mig)

Figure 2: EU gibt Irland noch keinen Freifahrtsschein (Luxemburger Wort, October 15, 2013).

expose realistically and unaltered current events, and/or contribute to the formation of an opinion. Assume that we have financial texts with an exclusive concentration on facts and objectivity, can we then find indications regarding financial, political, or economic decisions, for example with respect to the European crisis?

In Computer Science research, several directions regarding the analysis of texts have evolved. One of them is *Sentiment Detection* and with it the finding of an inherent polarity of the document. A sentiment detection [32] refers to identifying and extracting subjective information that appears in source materials and to determining the attitude of a person concerning an overall contextual polarity of a document. The sentiment may be a judgement or an evaluation, an affective state, or the intended emotional communication. Following this, a finding of answers to the questions above might be rather simple in a way that a certain number of existing techniques may be applied. More easily, we could argue that we only have to analyse the documents linguistically, statistically, and from a Machine Learning point of view, and that we then may come up with a sentiment decision. [9] describes a diverse number of directions for the detection of sentiment, which are either on document and sentence level or aspect-based or comparative. Also, the learning of a sentiment lexicon acquisition is highlighted. Some fundamental questions regarding *Sentiment Detection* are for example, whether a document is somehow more positive, neutral, or negative or not. Also, can we say that a document is uniformly or diversely polar? With respect to a concrete subject, is the sentiment stable (over time) or are there ups and downs? How polar are texts, i.e., how diverse and differently granular are they, to what extent do they depend on the evaluator's perspective, interest, knowledge status, and expertise?

The development of solid, correct, and intelligent algorithms to monitor document sentiments over time (sentiment barometers) is the major goal of this course. We consider only primary information sources, which reference to the key (Irish) Euro players (politicians, commissioners,

financial persons) to the crisis as reported by financial experts from the Luxembourg School of Finance.

## 1.2 Projects

We believe that the Financial News by Thomson Reuters are objective and that these messages represent a political *mirror of the time*. Different projects have been launched, which are briefly described below. These projects fundament on some assumptions.

### 1.2.1 Bag of Words

- We concern Financial News (published by Thomson Reuters) from the years of February 2009 until January 2013. The data has been prefiltered towards the involvement of Ireland in the European Stability Mechanism (ESM). Nevertheless, some noise (financial news that concern other disciplines like sports) is still inside and, therefore, needs to be removed.
- There exist different granular levels concerning sentiments ([9]). We concern the document sentiment only and do not take into account specific linguistic aspects like - for example - negations or ambiguity (because of lack of time). Instead, we concern Bag of Words, which have been given by financial experts.
- The Bag of Words concern both persons like politicians (Merkel, Sarkozy, and othes) and financial key players (Lagarde, and others) and financial terms like *Bailout*, *European Crisis*, *Euro*, et cetera.
- The Bag of Words might still be filtered and the number of (potentially redundant) entries reduced: a seed of essential terms is to be concernd, which is specifically related to *Ireland*.

The initial *Bag of Words* model can be found in the Appendix.

### 1.2.2 Data Landscape

As mentioned, we use Financial News from *Thomson Reuters* and concern the *Irish Crisis* from February 2009 until January 2013. Prefiltered financial news (Full texts, Headlines, Alerts) have be given to the project teams. The data size has been about 4.2 million lines, whereas the headlines is about 63000 records and the alerts about 29000 lines, respectively. We have organised 6 different research projects, which concern both the full texts (StoryTakes), the Alerts, and the Headlines.

- Story_Takes: these are texts that are either new or that represent an extension to another, previously given, finanical news (*Append*) or are a complete substitute (*Overwrite*). Financial news are identified by a *PNAC* number, which is an unique identifier.

  The following extract shows a Story_Take Financial News with $PNAC = nWLA7298$, published on the $13^{th}$ of February, 2009. There exist some keywords describing the news:

  > 153538, 2009-02-13, 06:07:12, 20090213055748nWLA7298,
  > STORY_TAKE_OVERWRITE, nWLA7298, 2009-02-13 05:57:48, 0000-00-00 00:00:00, BRIEF-Irish Life declines CEO's offer to resign over Anglo,,DUBLIN, Feb 13 (Reuters) - Irish Life & Permanent PLC <IPM.I>: The board of Irish Life & Permanent says it strong disapproved of some of the measures used to support anglo Irish Bank during 2008. The board of Irish Life

& Permanent says it was not informed of the specific manner in which such support had been afforded to anglo Irish Bank. Says its board declined an offer of resignation by group chief executive Denis Casey. Says group Finance director, Peter Fitzpatrick, has resigned. Says head of group treasury, David Gantly, has resigned. Says david mccarthy, chief financial officer, has been appointed as group Finance director to replace peter fitzpatrick. Says it has initiated a full review of its processes and procedures ((Dublin Newsroom; +353 1 500 1529)) ((For more news, please click here [IPM.I])), E UKI RNP PCO,INS FIN BNK BACT MNGISS EUROPE GB WEU IE LEN RTRS,IPM. I IPM.L,,, S, FALSE, RTRS, EN

- Headlines: Headlines typically consist of rare texts; often, only one sentence is used.

  6686398,2010-11-15,11:44:58,20101115114410nLDE6AE0W1,HEADLINE, nLDE6AE0W1,2010-11-15 11:44:10,2010-11-15 11:44:58, EURO GOVT-Bunds tumble, Irish yields fall on bailout hopes, * Bunds fall on Irish bailout expectations

- Alerts: Alerts typically consist of rare text and are often only a sentence.

  157268,2009-02-13,10:08:43,20090213100844nDUB000833,ALERT, nDUB000833,2009-02-13 10:08:44,2009-02-13 10:08:44, IRELAND'S FIN MIN SAYS TO MEET CHAIRMAN OF IRISH LIFE & PERMANENT TODAY TO DISCUSS COMPANY STATEMENT,,,UKI E,MCE ECI WEU EUROPE IE BNK FIN REGS MNGISS INS LEN RTRS,IPM.I,,,S,FALSE,RTRS,EN

### 1.2.3 Term and Document Sentiments

- A term sentiment is a value between -1 (very negative) and +1 (very positive). It expresses the polarity of the term at a specific time. A term sentiment is adaptive, which means that its value changes over time (current term sentiment plus a $\delta$).

- Change of term meanings: as an example, *Christine Lagarde* has been the French Foreign minister, but has become the head of the *International Monetary Fund (IMF)* in July 2011. With respect to the term sentiment, this may result in a complete other sentiment value. The impact of such *external influences* are difficult to handle; we simulate by re-initialize the term sentiment anew.

- Concerning a sentiment initialisation of a term, some strategies are:

  - Individual initialisation: each group performs their own initialisation.
  - Voting: All members vote while assigning a value between -1 and +1 to each term (Wisdom of Crowd).
  - Weighted voting: a domain expert has certainly more knowledge about a term sentiment. The initialisation is to be made as a weighted sum of the experts and the non-experts.

- A document sentiment is a value between -1 (very negative) and +1 (very positive), which is calculated by using the term documents therein. Several strategies have been developped:

  - Average: we use the average value of all term sentiments, which appear in the document. If there is no term in the document, then the document value is set to 0.
  - Weighted Average: apart from the term sentiment, we assign each term a weight. This weight represents the timeliness of the term (term frequencies, inverted document frequencies, etc.) A document sentiment is then the average of existing term sentiments and the corresponding term weights.

- Regarding the temporal monitoring of sentiment values: we have identified 27 time windows, which are defined by the European Council Meetings (European Summits and others) that have taken place between 2009 and 2013.
- With that, we have agreed on having a cumulative sentiment value for the documents (*Story_Takes*, *Headline*, *Alert*) available within these time periods and to have some more fine-grained sentiment values for each document for the weeks (even better: days) therein.

### 1.2.4 European Council Meetings – Time Windows

Regarding the time periods (= separated by the European Council Meetings), 26 dates have been identified within the years of 2009 until 2013 (Source: Wikipedia; confirmed by colleagues from Dept. of Finance). The time periods cab found in the Appendix.

### 1.2.5 Project Ideas

The projects are then

P1 We concern *StoryTakes*, which are the Financial News full messages. We filter these regarding the Irish Crisis and calculate a document sentiment by using the term sentiments. We receive sentiments over time, which are a StoryTake Barometer.

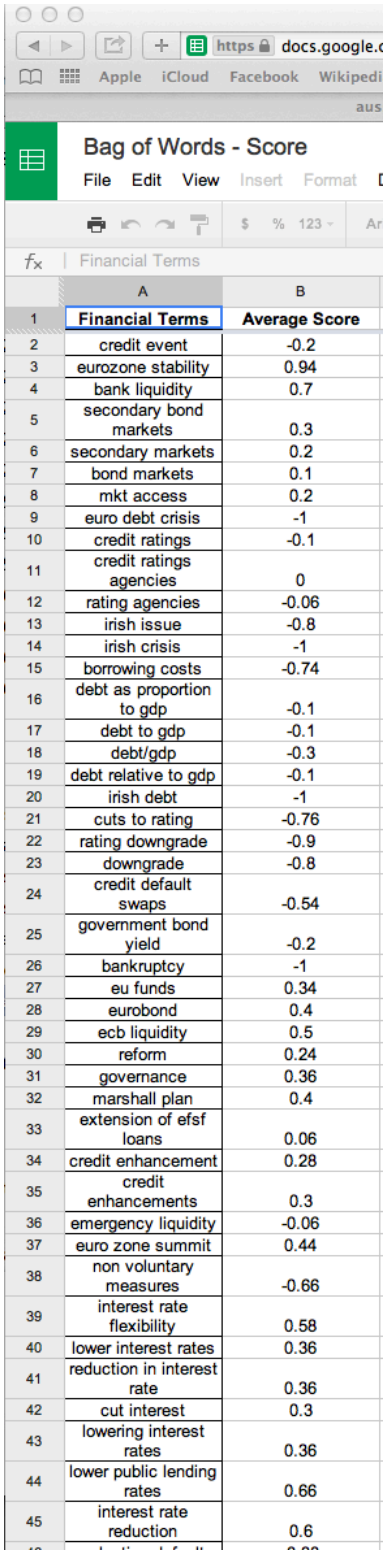
P2 We concern only the Financia News Alerts. An Alert-based sentiment barometer is then calculated and monitored over time. Linguistic problems, for example: negations, are neglected.

P3 Only the headlines are to be considered. As for the alerts, and Headline sentiment barometer (by using the term sentiments) is calculated. Sentiments are monitored over time.

P4 We use the authors (email addresses) and estimate their location (Ireland, UK vs. the rest of the world) and the time. We then calculate the corresponding document sentiments (using the term sentiments) for each author, which are Sentiment Barometer Author Cubes. A monitoring of an author's document sentiment over time is made.

P5 We concern the topic codes and filter the most frequent ones (also: the topics, which concern the European Crisis). We then calculate the corresponding document sentiments and monitor these values over time (Topic Code Sentiment Barometer).

P6 Given a financial news with an unexpected high (very positive) or low (very negative) document sentiment. This could be the key (filter) for a sender (Peter) to hide information therein. The recipient (Susan) reduces the complexity of financial news by selecting exactly those financial news with such an outlier sentiment value. The idea of this project is to simulate this by using (for example) synomym substitutions. Sentiment outliers must be, however, identified first.




| | A | B |
|---|---|---|
| 1 | **Financial Terms** | **Average Score** |
| 2 | credit event | -0.2 |
| 3 | eurozone stability | 0.94 |
| 4 | bank liquidity | 0.7 |
| 5 | secondary bond markets | 0.3 |
| 6 | secondary markets | 0.2 |
| 7 | bond markets | 0.1 |
| 8 | mkt access | 0.2 |
| 9 | euro debt crisis | -1 |
| 10 | credit ratings | -0.1 |
| 11 | credit ratings agencies | 0 |
| 12 | rating agencies | -0.06 |
| 13 | irish issue | -0.8 |
| 14 | irish crisis | -1 |
| 15 | borrowing costs | -0.74 |
| 16 | debt as proportion to gdp | -0.1 |
| 17 | debt to gdp | -0.1 |
| 18 | debt/gdp | -0.3 |
| 19 | debt relative to gdp | -0.1 |
| 20 | irish debt | -1 |
| 21 | cuts to rating | -0.76 |
| 22 | rating downgrade | -0.9 |
| 23 | downgrade | -0.8 |
| 24 | credit default swaps | -0.54 |
| 25 | government bond yield | -0.2 |
| 26 | bankruptcy | -1 |
| 27 | eu funds | 0.34 |
| 28 | eurobond | 0.4 |
| 29 | ecb liquidity | 0.5 |
| 30 | reform | 0.24 |
| 31 | governance | 0.36 |
| 32 | marshall plan | 0.4 |
| 33 | extension of efsf loans | 0.06 |
| 34 | credit enhancement | 0.28 |
| 35 | credit enhancements | 0.3 |
| 36 | emergency liquidity | -0.06 |
| 37 | euro zone summit | 0.44 |
| 38 | non voluntary measures | -0.66 |
| 39 | interest rate flexibility | 0.58 |
| 40 | lower interest rates | 0.36 |
| 41 | reduction in interest rate | 0.36 |
| 42 | cut interest | 0.3 |
| 43 | lowering interest rates | 0.36 |
| 44 | lower public lending rates | 0.66 |
| 45 | interest rate reduction | 0.6 |

Figure 3: Average values of the term sentiment voting (all course participants). Each individual score has had the same weight.

# 2 Sentiment Barometer on Story Takes

Hera Margossian, Fabien Mathey, Valentin Plugaru

The present chapter focuses on the detection and analysis of the sentiment polarity at the document level for the full texts of the Thomson Reuters Irish Crisis (*Story Takes*). We develop several algorithms in order to classify the sentiment polarity over time, with the goal of obtaining a Sentiment Barometer for the mentioned documents which can be used to detect or predict the shifts in sentiment that affect financial markets.

Several preprocessing steps are performed on the raw data in order to extract only the relevant information, on which five machine learning techniques are applied for sentiment classification. First, unsupervised algorithms are applied: a simple classifier which uses the Bag of Words terms and taking negation words into account, and a Tf.IDF-based sentiment analyzer. Next, classification is performed using supervised polarity classifiers based on active learning, Support Vector Machines and Artificial Neural Networks.

The algorithms use as input weighted financial terms to perform the classification of the stories, but are not context dependent and can be applied to different domains by changing the input dataset and knowledge bases.

The results obtained from the application of the aforementioned algorithms on the dataset show a generally negative sentiment over the considered time interval.

**Keywords:** Sentiment Analysis, Machine Learning, Support Vector Machines, Neural Networks.

## 2.1 Problem Description

A recent focus in *NLP* is performing Sentiment analysis on user generated content and extracting valuable information relating to events, people, topics or products.

Many research articles that target opinion mining are analyzing the Twitter social network's messages ([8], [13] and [18]) or interpreting movie reviews ([21]) since these are easily accessible knowledge bases, with very short (140 characters for Twitter messages) or single topic (in the case of the movie reviews) texts.

The focus of this chapter is to analyze the full text of Thomson Reuters financial news articles that relate to the Irish Crisis in the February 2009 - January 2013 period, as described in Chapter 1. We develop and apply several techniques, described in the following subsections, in order to monitor the sentiment across this time period. The results of this evaluation are of prime importance in determining historic economic hot-spots that shaped later events. The methods developed to perform the analyses could also be used in the future in expert systems, trend evaluation and prediction.

Working with the full financial texts is a complex task due to the amount of noise that has to be filtered before the relevant content can be identified. Analyzing the full texts however has the advantage that many features can be extracted and used in machine learning algorithms such as *SVM* and *NN*.

Some of the main challenges of working with the full financial texts have been dataset preprocessing, choosing the appropriate sentiment classification techniques and tuning the algorithms

that implement them, and evaluating the results.

This chapter is organized as follows: section 2.2 describes the algorithms we developed to classify the polarity of the financial texts, section 2.3 discusses the implementation details for the algorithms and their workflow, section 2.4 analyses the obtained results and finally the conclusions given are in section 2.5.

## 2.2 Algorithmic Conception

The following subsections describe the process of sentiment polarity classification. It includes data preprocessing and analysis, a simple classification algorithm with and without Tf.IDF weighting, machine learning algorithms based on active learning, *SVM* and *NN*. Finally, the data postprocessing algorithm is detailed.

The algorithms presented here are independent of the dataset and knowledge base. Changing the input dataset and knowledge bases – the *Bag of Words*, and providing appropriate training sets – is sufficient for sentiment classification on other datasets.

### 2.2.1 Data Preprocessing and Analysis

The dataset and knowledge base provided was preprocessed to help determine the sentiment analysis of the articles. In this subsection the text extraction, sentence Extraction, stop Words Removal and lemmatization, and training set creation are described in detail. Finally, the sentiments associated to the *Bag of Words* are illustrated.

| **Column** | **Information type** |
|---:|---|
| 0 | Unique list ID |
| 1 | Date |
| 2 | Time |
| 3 | Date Time and Story ID |
| 4 | Type of Document |
| 5 | Story ID |
| 6 | Date Time |
| 7 | Date Time |
| 8 | Short Message |
| 9 | Full Text Part 1 |
| 10 | Full Text Part 2 |
| ... | ... |

Table 1: Knowledge Base - Key Information

**Text Extraction** The provided dataset contains different types of articles and information about them. This subsection concentrates on the documents categorized as *Story Take*. They are the ones marked as *Full Text 1* and *Full Text 2* in Table 1.

The extra information provided in the dataset did not need to be referenced for sentiment analysis and was removed. This was done by extracting the full texts from the raw data columns 9 and 10, which are both used to store the articles' text. The corresponding *IDs* and *dates* of each of these articles were also extracted and stored with the *full texts*.

| | | |
|---|---|---|
| 106 billion euro | deficit | lower public lending rates |
| 109 billion euro | downgrade | lowering interest rates |
| 37 billion euro | ecb liquidity | marshall plan |
| assistance programme | efsf | maturity |
| austerity | efsf loans | mkt access |
| bailout | emergency liquidity | non voluntary measures |
| bank liquidity | esm | official financing |
| bank stress test | eu funds | overnight loan facility |
| ... | ... | ... |

Table 2: Bag Of words - excerpt

The data extracted was still needed to be reduced and was thus further processed. This step was important since it helped speed up analysis and reduce the complexity of the algorithms. In the next subsections the pre-processing steps will be discussed in more detail.

**Sentence Extraction** The reduction of the texts was done based on the set of terms from the *Bag of Words*. This was done because most of the evaluation of the sentiment for the articles was done using these terms. The articles were thus reduced to a set of sentences that contained at least one of these terms. Those not containing any term were therefore removed. This reduced the number of documents from 58357 to 43486. The reduced dataset resulting from this extraction contained the ID, date and the set of sentences for each considered article.

**Stop Words Removal and Lemmatization** To reduce noise and speed up analysis the stop words were removed from the sentences. By doing so the size of the data was reduced. Stop words are a good way to reduce data, since their presence does not add any additional information that may help in sentiment analysis. The list of stop words that were removed from the text were taken from the website[1] last accessed January 4, 2014. This list included negation words, which were removed from the list since they may invert the sentiment polarity in a sentence.

The sentences and the terms from the *Bag of Words* were then lemmatized. Lemmatization was chosen as a processing technique in order to attain a more accurate result when performing string comparisons. Lemmatization has been used instead of stemming as this latter process destroys sentiment distinction [24].

**Training Set Creation** Training sets are needed for the supervised learning algorithms, their creation is a simple yet time consuming task. Different training sets have been created for the *SVM* and the *NN*. The labeled training set used in the *SVM*based classfication is split in two parts, the first used for training and the second to verify the accuracy of the model generated. The classification labels used for the training sets created in this chapter are: *positive*, *neutral* and *negative*.

**Bag of Words Sentiments** For the simple classification and the provided Bag of Words (see Table 2), a sentiment per word was assigned. Assigning the sentiments between *-1.0* (negative)

[1] http://norm.al/2009/04/14/list-of-english-stop-words/

and +1.0 (positive) with *0.0* being the absolute neutral, resulted in a knowledge base of the pattern shown in 3.

| | |
|---|---|
| 106 billion euro | 1.0 |
| 109 billion euro | 1.0 |
| 37 billion euro | 1.0 |
| assistance programme | 0.0 |
| bailout | 1.0 |
| emergency liquidity | -1.0 |
| ... | ... |

Table 3: Bag Of words with sentiment

The sentiment polarity of the terms within the *Bag of Words* is needed by the simple classification (subsection 2.2.2) and the Tf.IDF classification (subsection 2.2.3).

### 2.2.2 Simple Classification

The simple classification is a technique based on the values associated to the terms from the bag-of-words. This is a straightforward sentiment analysis method. The algorithm goes through a story text and counts the occurrences of the terms from the *Bag of Words*. Once the term frequency is calculated it is multiplied with the sentiment value corresponding to the term. The resulting value of the terms is then summed and a sentiment value is assigned to the article. The method also takes into account negation words (see Table 4) that may appear in the same sentence as the term considered. The algorithm searches for these words within a 5 word radius to the terms from the *Bag of Words*. If a negation word is found, the value of the term in the same sentence is multiplied by -1, inverting the sentiment polarity.

| | | |
|---|---|---|
| no | isn't | doesn't |
| not | ain't | nothing |
| none | don't | nowhere |
| wont | hadn't | haven't |
| never | aren't | couldn't |
| noone | didn't | shouldn't |
| can't | hasn't | wouldn't |

Table 4: Set of negation words

The reason for using the negation words was due to their impact on the sentiment of a term. To have a clear understanding of their effect on the sentiments, the probability of negation for the terms was calculated. The results showed that there were some terms that were negated with a 20 percent probability. The following example helps clarify the importance of considering them.

**Example Negation**

irish issue : *70%* of the occurrences are *not* negated.

**Example No Negation**

extended maturities : *100%* of the occurrences are *not* negated.

Algorithm 1 shows in detail how the simple classification works. In steps 1 and 2 the term frequencies are calculated and stored in a list. In steps 3 to 5 the presence of a negation word is checked. If one of them is present the sentiment is negated. In step 7 the values are summed up and a document sentiment is calculated. In step 8 the average of the sentiment is assigned to the document.

**Data**: Documents, BagOfWords, NegTerms
**Result**: SentimentPerDocument

```
1 for Doc in Documents do
2 |   fTerms = findTerms(Doc,BagOfWords);
3 |   if any NegTerms in sentence(fTerms) then
4 |   |   term2Negate ∈ fTerms;
5 |   |   negateSentiment(fTerms, term2Negate);
6 |   end
7 |   sentiment = sumUpSentiment(fTerms,NegTerms);
8 |   Doc ← avg(sentiment);
9 end
```

**Algorithm 1:** Simple Classification

### 2.2.3 Classification using Tf.IDF

Tf.IDF is a method commonly used in Information Retrieval. It helps to assign a weight that determines a rank for documents towards a certain query. The term frequency *tf* of a term in a document is the number of times the term appears in the document. The document frequency *df* is the number of documents that contain the term. The inverse document frequency *idf* is the total number of documents divided by *df*. Tf.IDF is the result of multiplying the *tf* of a term to its *idf*. The ranking is based on assigning a weight that determines how important a term is to a document in a collection. This characteristic is the main reason behind choosing this method as another way of evaluating the sentiment of the articles.

The Tf.IDF weights tend to filter out common terms. This is because a high weight in Tf.IDF is reached by a high term frequency in the given document, and a low document frequency of the term in the whole collection of documents. This filter is a useful way to reduce the impact of the negative terms which have a very high document frequency. An example of a negative term which has a low Tf.IDF weight is *debt*. Thus by filtering, the negative bias of the text is removed and a clearer view of the positive sentiment of the articles is observed. This is why the result of the Tf.IDF analysis shows a positive value for all the time periods considered. This helps in identifying the time periods where the articles with also positive sentiment were written.

### 2.2.4 Classification using Active Learning

Having a large knowledge base requires a substantial number of manually evaluated articles. Due to the lack of time and resources the "semi-manual" analyzer was developed. This analyzer is based on the concept of active learning. Active learning is a common machine learning technique which is used when unlabeled data may be extracted automatically, but labeling it is too expensive or difficult. The labeling is done by an oracle, for example by experts in the field [28].

The "semi-manual" analyzer finds a list of terms that describe the terms from the *Bag of Words*. This data is extracted automatically from the articles using the bigram list creator of the analyzer. These bigrams are later manually evaluated and stored with their corresponding term. Using this approach helps describe an article by assigning it a polarity label describing its sentiment.

### 2.2.5 *SVM*

Several supervised learning algorithms exist, such as naive Bayes. However, compared to other machine learning algorithms, it performs quite badly [21, 1]. For this reason, the naive Bayes algorithm is not part of this work and instead *SVM* and *NN* which achieve very good results [21, 1] have been chosen.

A Support Vector Machine is a Machine Learning algorithm that works in 3 steps (see Algorithm 2 for reference). The first step is the learning step. For that step, a feature vector is needed and a set of pre-classified data. In the case of this research, there were 3 classes: positive, neutral and negative. The algorithm is given the feature vector and a labeled training set. The *SVM* can then generate a model by itself which allows classification of future unknown texts.

In step 2 the verification and accuracy testing of the generated model is performed. For this step additional oracle labelled articles are needed, distinct from the training set. In some cases accuracies of up to 80% can be achieved [1].

The last step is the sentiment polarity classification step. This is done by applying the algorithm on the unlabelled texts or data entries. Knowing the accuracy of the model used by the algorithm makes it then possible to correct the classification by averaging.

**Data**: Documents, Bag of Words, Training Set, Verification Set
**Result**: Sentiment on Documents, Model

```
1  for Doc,Label in TrainingSet do
2  |   v = featureVector(Doc,BagOfWords);
3  |   trainSVM(Doc,v,Label);
4  end
5  for Doc,Label in VerificationSet do
6  |   v = featureVector(Doc,BagOfWords);
7  |   verifyModel(Doc,v,Label);
8  end
9  computeAccuracy();
10 saveModel();
11 # The actual classification
12 for Doc in Documents do
13 |   v = featureVector(Doc,BagOfWords);
14 |   evaluate(Doc,v)
15 end
16 saveClassification();
```

**Algorithm 2:** *SVM*

The feature vectors used as inputs for the *SVM* algorithm are composed of a corresponding

| | |
|---|---|
| Text | The brown fox is faster than the lazy dog on a sunny day. |
| Bag of Words | summer, car, fox, dog, world, lazy, sunny day, hot |
| Feature Vector | 0,0,1,1,0,1,1,0 |

Table 5: SVM - Feature Vector Example

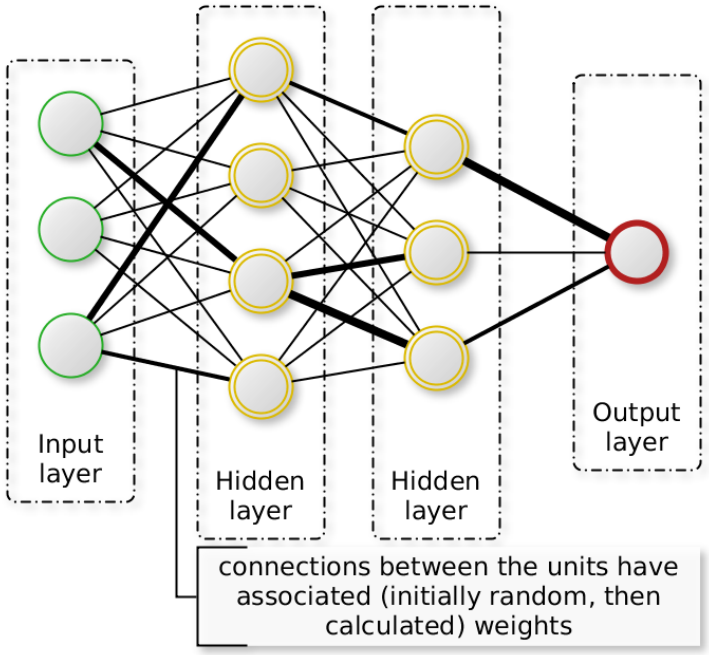



Figure 4: Generic Neural Network model

representation of the provided *Bag of Words* (see table 5 for an example of a feature vector). The feature vector contains binary value elements: *true* (or 1) if the word from the bag of words is part of the current text or *false* (or 0) - (Table 5 shows an example).
The 'positive', 'negative' or 'neutral' sentiment labeled training set used for the *SVM* is generated using the active learning algorithm detailed in Section 2.2.4.

The downside of such an approach is that the algorithm will always provide us with a result even if the result is wrong. This is the same as for humans: classification is a subjective model that is based upon a subjective biased sentiment towards any interpretation of text. And thus if the machine was trained but formed its model in a faulty way, the assignment of the sentiment labels to a document may be faulty as well.

### 2.2.6 Classification using Neural Networks

The last classifier for the sentiment polarity of the financial news that was developed is based on an *ANN*.

Neural Networks are data-driven self-adaptive methods that are able to approximate underlying behavior from training data (learn) and predict behavior beyond it (generalize) [26]. They can be used for supervised data classification by deriving the classification decision rules from user

labelled training data, and applying them to new, unlabelled data.

An *ANN* is composed of neurons (units) which are organized in layers, as shown in Figure 4. Data is passed to the network through an *input* layer, and the result taken from an *output* layer, the number of units in these layers corresponding to the number of inputs, respectively outputs. The input layer is connected to the output layer by weighted connections – used in the calculation of the output – which are initially given random weights, then modified through the learning procedure. Between the input and the output layers an arbitrary number of *hidden* layers can be added in order to increase the computational capacity of the network. The units in the input layer are passive - they do not modify the data, while the units in the hidden and output layers calculate their output value based on an *activation function*.

In the present work, a feedforward multi-layer perceptron *ANN* with backpropagation learning has been used for classification. In this kind of *NN*, the network learns the decision rules by first forward propagating the training data to the output layer and then backward propagating the difference between the expected and generated output in order to update the weights given to each connection.

The workflow for classification using the *NN* is described in the following paragraphs, with supplemental details on implementation being part of Section 2.3.7.

A subset of 180 news items has been extracted from the *Story Takes* dataset – corresponding to approximately 0.3% of the financial stories – and manually classified as having 'negative', 'positive' or 'neutral' document-level sentiment, which are used in the *NN* training phase. The news items are extracted as follows: a set of 60 news items with a time difference between the stories of at least 7 days, in order to span the full 2009 - 2013 time period, and a set of the top 120 news items ordered by story length, with a word count between 500 and 1000 words each.

For each news item in the full dataset, the following features are extracted: the presence (encoded as 0 or 1) of a story's author, $\log_{10}$ based term frequencies (with values in the $[0, \infty)$ interval) are calculated for positive and negative terms (obtained from an external knowledge base [27]) in the same sentence with Ireland-related terms ('Ireland', 'irish', 'AIB', 'KBC') and the presence (encoded as 0 or 1) for each of the 176 unique terms of the *Bag of Words* (people names and financial terms).

When the features are extracted for each news item, the term-frequencies previously computed are normalized across all news items to the $[0, 1]$ interval.

In the next step the *NN* is built with 179 units in the input layer, corresponding to the 179 features serving as inputs, 1 hidden layer with 18 units and output layer with 3 units corresponding to each sentiment polarity category.

Finally, the *NN* is trained for 50000 iterations, with 2 epochs each, on the training set then applied on each news item in the full dataset for sentiment polarity classification.

The *NN*-based sentiment classification can be improved by extracting additional features from the initial dataset, for example dates and locations, however then the training set would have to be much more extensive and cover most of the time interval.

### 2.2.7 Post-Processing

Having such an extensive dataset and respective classification results, combining results within a certain period is necessary in order to obtain a clear visualisation of trends and to allow

prediction. The 2009-2013 time interval was divided by the European Summit dates (as given in the section on the European Council Meetings – Time Windows). The division by these specific dates was performed in order to observe if the Summits had an influence on the sentiment polarity trend. However, combining results by averaging the sentiment values leads to the loss of the relevance of the polarity weights (e.g. strong positive sentiment averaged with a weak negative sentiment will result in a slightly positive sentiment). For this reason the sentiment values of one period between two summits are also shown in detail (weekly level) in Section 2.4.

**Data**: SentimentOnDocuments,TimePeriods
**Result**: CominedResults

```
1 tp = getSummitDates(TimePeriods);
2 for Doc in Document do
3 |   date = getDateFromDoc(Doc);
4 |   tp ← assignDocToTP(Doc,Date);
5 end
6 sumAndAverage(tp);
```

**Algorithm 3:** Post Processing - Combination

Algorithm 3 gives an overview on how the documents' sentiments aggregation works. First, the time periods are read from a file listing the stories' dates. Then each time period is linked to a set of documents. Finally the sentiments of the documents in each time period are averaged and can be used for visualisation or further processing.

While the data can be aggregated in other ways (at the week level, month level, financial quarter level, etc.), in the present work the main focus is exploring the sentiments polarity between the European Summits.

## 2.3 Implementation Details

In this subsection we discuss the implementation of the algorithms described in Section 2.2. The applications we developed have been written in Java and Python programming languages, using external libraries detailed in the next paragraphs.

### 2.3.1 File Formats

In the current work the file format used for intermediate processing results and final sentiment classification results is *CSV* due to its low overhead and simplicity compared to other formats such as the *XML*. For unstructured data, the files are pure text files with one item per line – using the newline character $\backslash n$ as item separator.

### 2.3.2 Data Preprocessing

The sentences from the news texts in the raw dataset have been changed to lower case in order to simplify string comparisons. The 600+ stop words from the norm.al[2] knowledge base were

[2]http://norm.al/2009/04/14/list-of-english-stop-words/ accessed Jan $4^{th}$2014

| Filename | Format | Sample Line |
|---|---|---|
| __finished.txt | *CSV* | ID : LABEL $\in [0.0, 1.0, 2.0]$ |
| featureList.txt | Textfile | TERM $\backslash n$ TERM $\backslash n$ ... |
| sentencesNoStem.txt | *CSV* | ID # DATE # SENTENCE1 # SENTENCE2 ... |
| Stem-bow.csv | *CSV* | TERM ; SENTIMENT $\in [-1.0; 1.0]$ |
| themValues.txt | *CSV* | ID # DATE # SENTIMENT $\in [-1.0; 1.0]$ |
| TrAccuracyData.txt | *CSV* | ID # DATE # LABEL $\in [positive, neutral, negative]$ |
| TrData.txt | *CSV* | ID # DATE # LABEL $\in [positive, neutral, negative]$ |
| txtValues.txt | *CSV* | ID # DATE # LABEL $\in [positive, neutral, negative]$ |

Table 6: Files and their format

also removed. This collection of words excludes the negation words (Table 4) and can easily be updated and adapted since they are stored within a simple text file. This processing and the word lemmatization is done by the *bag_of_words_counter.py* application, which is able to take advantage of multicore systems, and was ran on University of Luxembourg's *Gaia* cluster in order to reduce the long processing time.

Details on the developed applications and their dependencies are given in Table 7. The table displays the applications that need to be executed sequentially in order to perform the sentiment polarity classification. The required input files for each are given and the corresponding result files . The outputs are generally CSV formatted files that allow easy further processing.

**Bag of Words** The Bag of Words contains a list of financial terms. The evaluation of these terms was done by the developers. The evaluation entailed a basic sentiment labelling of each term - see table 3 for an example excerpt.

The bag of words was also lemmatised (see Section 2.2.1 for details). This needed to be done to reduce the amount of terms by ignoring the ones that have the same root: an example of this is given in table 8.

### 2.3.3 *Simple* Classification

After lemmatizing the documents and the *Bag of Words* and removing the duplicate words from the bag of words, the sentiment barometer was implemented. To attach a sentiment to each document, the content of each document was analysed and the terms from the *Bag of Words* were counted. If a term was located within a document, then its sentiment weight was added to the document's overall sentiment value.

**Example**

Assuming the *Bag of Words* would be the one shown in Table 3, and a text containing the following sentences: As an emergency liquidity is required, bailout plans are being formed. The overall savings could reach 37 billion euro.
This text will then be evaluated with the sentiments assigned in table 3. This would result then in: $-1.0 + 1.0 + 1.0 = 1.0$

While the example above will result in a sentiment of 1.0, others would result in sentiments that are outside of the scope $[-1.0 : 1.0]$. For this reason an average value is calculated and then used to define the document sentiment. For the example above the formula would look like this:

| Step | Application | Input files | Output files |
|---|---|---|---|
| 1 | bag_of_words_counter.py | ireland.csv | sentencesNoStopStem.txt |
| 2 | valuator.py | sentencesNoStopStem.txt,<br>stem_bow.csv | themValues.txt |
| 3 | supportVectorMachine.py | featureList.txt,<br>sentencesNoStopStem.txt,<br>TrData.txt,<br>TrAccuracyData.txt,<br>stem-bow.csv | _finished.txt,<br>_Model.txt |
| 4 | combiner.py | themValues.txt,<br>_finished.txt<br>neuralnetClassifDatedID.txt | PeriodicOutput-Summits-Basic.txt,<br>PeriodicOutput-Summits-SVM.txt<br>PeriodicOutput-Summits-SVM.txt |

Table 7: Processing steps for sentiment analysis with the basic evaluator and SVM

$$\frac{-1.0 + 1.0 + 1.0}{3} = 0.\overline{3} \tag{1}$$

The overall sentiment that is assigned to the text is 0.33. Thus this would be classified later as (slightly) positive once a bucket principle is applied.

| term | employees |
|---|---|
| Text 1 | The employees are tired and stop working |
| Text 2 | The employee is tired and stops working |

Table 8: Lemmatisation example

### 2.3.4 *Active Learning* Classification

The 'semi-manual" analyzer for the active learning classification is based on the terms from the *Bag of Words*. It takes as input the sentences which contain one of the terms, then mines the bigrams of these sentences, and counts the number of their occurrences. Based on these numbers, it builds a bigram-list for each keyword. This bigram list is then outputted to a file corresponding to the keyword.

Example: Assume the term is unemployment and one of the sentences it appears in is "unrest popul still high unemploy", then the bigrams from this sentence are; "unrest popul", "popul still", "high unemploy" and so on. The occurrence of these bigrams in the other sentences is counted and the bigrams with the highest count are kept. In case of unemployment "high unemploy" is one of the most common bigram associated to it. Thus it is one of the bigrams that is stored in the list of bigrams for unemployment.

The bigram files are later used as an input to another application, which allows a user to manually determine whether the bigram makes the keyword positive, negative or does not impact the polarity. In the above example "high unemploy" is a negative bigram, since high unemployment gives a negative sentiment.

This approach helps to determine the polarity of an article based on the context of the keywords from the *Bag of Words*. The content is determined by checking which of the bigrams, if any, are found in a sentence containing the keyword. For example, if a positive bigram is found, it means the keyword has a positive meaning in that sentence. If more than one bigram is found in a single sentence then it is set to be positive if there are more positive bigrams, neutral if there are equal number of positive and negative bigrams, and negative otherwise. If no bigrams are found then the polarity of the term is set to neutral. The evaluation of the sentiment of the entire article is done by counting the number of positive and negative terms and assigning the polarity with the greatest number of occurrences.

$$Pol_{kw\_per\_sentence} = pol(max(bigram_{neg}, bigram_{pos}))$$

$$Pol_{pos} = \sum(kw_{pos})$$

$$Pol_{neg} = \sum(kw_{neg})$$

$$Pol_{document} = \begin{cases} max(Pol_{pos}, Pol_{neg}), \& if\ Pol_{pos} <> Pol_{neg}. \\ neutral, \& otherwise. \end{cases}$$

A possible improvement of this method may be asking a financial advisor to set the bigrams' polarity. The application for setting the polarity is user-friendly and doesn't require prior programming knowledge. This makes it easy for any user to give an evaluation, based on their background knowledge of the bigrams.

Another improvement may be to add a value of importance to each term. This value would indicate how much a certain term would affect the main issue (topic) being considered, and should be determined preferably by a financial advisor. Thus during the evaluation the bigrams would only indicate the polarity of the term affecting the article sentiment. This polarity would determine whether the given value of the term should be positive or negative. The remaining evaluation would be adding these values together. This way the article is assigned a value sentiment instead of a term sentiment describing polarity.

### 2.3.5 *Tf.IDF* Classification

The Tf.IDF algorithm was implemented to calculate the Tf.IDF weights for each term from the *Bag of Words*, these values being later used to determine a polarity value for the articles. Each term was also associated with a predetermined value which ranges from -1 to 1 and was based on how positive or negative the term could affect an article. For each term present in an article, the Tf.IDF was multiplied with the associated value, giving it a polarity measure. For example, the term unemployment has a value of -1 and assuming a Tf.IDF value of 0.8, the result would be -0.8 for that keyword in that document. The assessment for each article was then calculated by summing up the polarity measures of all the terms present in its text:

$$Keyword\_Sentiment = keyword\_Tf.IDF \cdot keyword\_value$$
$$Article\_Sentiment = \sum keyword\_sentiment$$

The Tf.IDF algorithm was applied on the full set of news items detailed in Section 2.2.1. Thus the total number of articles was considered during the calculation of the idf. In order to dampen the effect of the Tf.IDF weights, a logarithm base 10 was used.

### 2.3.6 *Support Vector Machine* Classification

To develop an algorithm using a *SVM*, the Python language (version 2.7) has been used, as it contains libraries that facilitate the implementation of such learning algorithms. The required additional libraries used for the *SVM* were the Natural Language ToolKit (*nltk*) which required the *pyyaml* library, and the *scikit-learn* library, requiring *numpy-mkl*.
For the algorithm description of the *SVM*, please refer to Algorithm 2 which shows how the *SVM* works at a high level. In the Python implementation the algorithm has a lot of similarities. The inclusion of scikit-learn allowed the use of the following methods to train, test and classify:

**Training**
svm_train(problem(Labels, featureVector), LINEAR)

**Verification**
svm_predict(Labes,testFeatureVector, Model)

**Classification**
svm_predict([0]*len(vector),FeatureVector, Model)

The application that implements the *SVM* classifier is *supportVectorMachine.py*. For a detailed view on the files needed for the *SVM* execution, refer to table 7. The difficulty involved in using this application is that it is dependent on five files of which three must be created by an user, previous to its execution: *featureList.txt*, *TrData.txt* and *TrAccuracyData.txt*. Each of the files are in *CSV* format. The separation character for all of the files required by the *SVM* except the *stem-bow.csv* is #. For a detailed description on file format for each file, please refer to table 6.

**Classification Accuracy** The accuracy of this *SVM* is shown in table 9. It is clear that the average accuracy of the *SVM* is around 55.30%. Since the goal here is to take as small a training set as possible, and let the *SVM* evaluate the most documents by itself, after testing a training set of 6000 documents was selected as it gave the accuracy closest to the average of 55.30%.

Table 9 describes the sentiment classification accuracy with the training set of a certain size and a testing or verification set of a certain size. It is clear that increasing the testing data is also increasing accuracy. Since three sentiment polarity classes are used, there is a probability of $^2/_3$ that the algorithm could classify the document wrongly. Thus the bigger the training set and the verification set, the more accurate the accuracy test becomes.

| Training Docs | Testing Docs | Accuracy |
|---|---|---|
| 1500 | 500 | 52.60% |
| 1500 | 1000 | 54.50% |
| 3000 | 500 | 53.60% |
| 3000 | 1000 | 54.70% |
| 6000 | 500 | 55.40% |
| 6000 | 1000 | 55.30% |
| 15000 | 500 | 52.40% |
| 15000 | 1000 | 55.40% |
| 30000 | 500 | 60.40% |
| 30000 | 1000 | 57.80% |

Table 9: SVM Classification accuracy for varying number of training documents

The accuracy can be improved by re-evaluating the training data and by re-evaluating the feature vector. Looking at table 5, the feature list is equivalent to the *Bag of Words*. The feature list could be statically improved by simply adding terms which would reflect in the feature vector (see table 5 for an example) or by dynamically updating the feature list by adding new unlisted words with each "learning text"... Means each new text is built out of words and if a word from within the text is not within the feature list, it will be added. Although this might help in most cases, in this case it might still not improve the accuracy as much since many texts are really similar to each other.

### 2.3.7 *Neural Network* based classification

The Neural Network-based sentiment polarity classification workflow uses several Python applications and libraries.

A data exploration and extraction application, *raw_data_extractor.py*, has been developed and is used by other applications and can be ran directly by an user to access the data in the Thomson-Reuters *CSV* file or to visualise it - filtering by the news item type (Story Take, Headlines, etc.) or extracting a specific data row or column.

The *createTrainingSet.py* application is used to create a training set for the *NN* and has two operating modes: the first mode extracts the first news item every *t_interval* days (by default 7), while the second extracts the top *s_num* stories containing between *w_min* and *w_max* stories. Both modes create files containing only the news' text, files which are named in the format *YYYY-MM-DD_StoryIndex* where StoryIndex is the index of the story of *Story_Takes* type in the unprocessed dataset (e.g. 2012-11-01_41681). After executing this application, the user

reads the news text and tags the news item by appending an evaluated sentiment polarity to the file name (e.g. '_neg', '_pos', '_neu': '2012-11-01_41681_neg').

In the following step the *feature_extractor.py* application must be executed to extract the features detailed in Section 2.2.6. It uses a developed *name_finder.py* application to find story author names in the news articles (based on regular expressions) and the Natural Language ToolKit (NLTK) [29], directly - for sentence and word tokenization, and through a developed *lemmatizerHelper.py* wrapper - NLTK's WordNet lemmatizer with the Maxent Treebank Part-of-Speech tagger. The processing steps done by the feature extractor are the following: first, the terms in the *Bag of Words* are put to lower case and lemmatized. Then, the lemmatized news items from the *sentencesNoStopStem.txt* file are read, along with the lists of positive and negative terms from the [27] external knowledge base. Next, depending on the application's execution mode, the feature extraction is performed for the full dataset, for the training set, or both.

When the features are extracted, the negative and positive term frequency fields which contain float values are normalized to the $[0, 1]$ interval as it was shown to be necessary through the vtesting of the Neural Network, which did not perform well with unscaled values in these fields. The execution of the feature extractor results in the creation of the feature matrix files: *trainFeatureList.txt* or *trainFeatureList-FSearch.txt* for the training set and the *FeatureListExtended.txt* or *FeatureListExtended-FSearch.txt* for the full data set. The *-FSearch* files are generated when the application is ran in the mode where it searches for positive and negative terms in the full text of the story as opposed to only the sentence containing Ireland-related terms, as defined in Section 2.2.6.

After the features have been extracted, the *neuralnet.py* application is executed to perform the sentiment polarity classification, using the external Python *PyBrain* [30] library for its flexible Neural Network implementation. The internal steps done by the neuralnet.py application are as follows: first, the training set is loaded, containing the features extracted for the 180 training news items, and the labels that have been assigned manually. Next, the Neural Network is built with the input layer containing the 179 units corresponding to the feature list items, 1 hidden layer with 18 units and the output layer with 3 units corresponding to the negative, neutral and positive sentiment polarity classes, and the Softmax activation function at the output layer, then trained for 50000 iterations, with 2 epochs each. These parameters, as well as the Backpropagation learner's learning rate (0.005) and momentum (set to 0) have been tuned to achieve an error rate below 0.01 on the training set. Testing has been done with the number of units in the hidden layer varying between 179 and 3, and a number of 18 units has been found to create a sufficiently complex neural network to be trained well on the given training set. The Neural Network application creates the following output files:

*neuralnetClassif.txt* containing the evaluated sentiment polarity class (one per line, corresponding to each news item), *neuralnetClassifDated.txt* – a CSV formatted file containing the news's index, date and polarity (given as integer: -1 for negative sentiment, 0 for neutral or 1 for positive sentiment), and the *neuralnetClassifDatedID.txt* file, which contains the news story IDs, the date and sentiment polarity. This latter file is used by the *combiner.py* application to aggregate the sentiment polarity values in the time intervals delimited by the European Council Meetings.

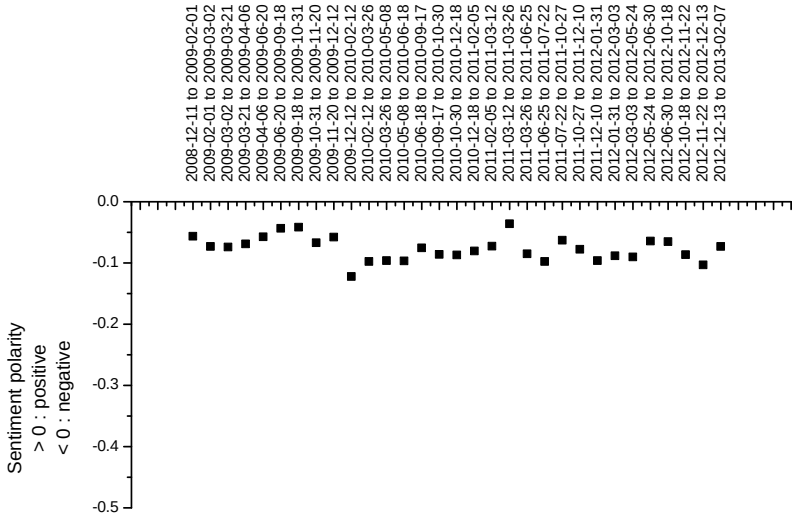



Figure 5: Basic sentiment polarity classification across the 2009-2013 period

## 2.4 Results and Results Discussion

This Section shows and contrasts the results of the five classification techniques applied on the dataset, and discusses observed trends.

Table 10 details the number of positive, neutral and respectively negative sentiment polarity stories that have been classified with each algorithm, showing a high number of negative polarity news stories across the considered interval.

| Algorithm \ Polarity | Positive | Neutral | Negative | Total |
|---|---|---|---|---|
| Simple classification | 6457 | 18687 | 33213 | 58357 |
| TF-IDF classification | 33594 | 8283 | 1609 | 43486 |
| Active Learning | 4418 | 17898 | 13373 | 35689 |
| Training Set SVM | 761 | 2979 | 2260 | 6000 |
| SVM classified | 130 | 44593 | 7634 | 52357 |
| Training Set Neural Net | 58 | 24 | 98 | 180 |
| Neural Net classified | 13204 | 14129 | 31625 | 58958 |

Table 10: Summary of the news stories classified in each polarity category

Figure 5 shows a relatively constant negative (averaged) sentiment when the simple classifier detailed in Section 2.2.2 was used. This is in contrast to the other four algorithms which show some increase or decrease in the averaged sentiment over time.

In Figure 6 the results for the Active Learning classification is shown. From the results it can be seen that for most of the time periods the sentiments are negative except for 2012-11-22 to 2012-12-13 which shows a positive value. It can also be noted that the sentiments seem to be

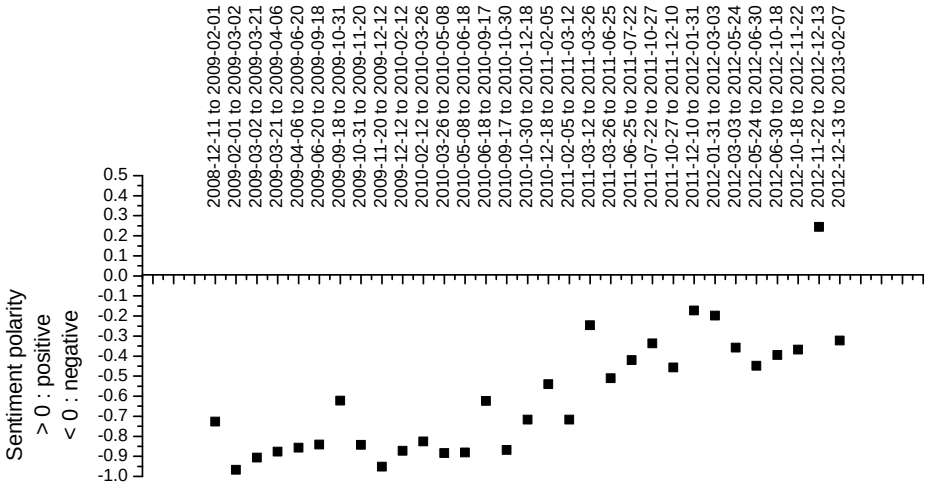



Figure 6: Active learning sentiment polarity classification across the 2009-2013 period

less negative as time progresses. There is an observable difference between the Basic and the Active Learning classifications, with the latter providing a greater variation in the sentiment values. This shows that taking the context of the keywords into consideration provides additional information affecting the evaluation.

Figure 7 shows the results for the Tf.IDF based sentiment evaluation. The results observed show that for all of the time periods the averaged sentiments of the documents are positive. The general negative bias given by very common negative terms from the *Bag of Words* is removed and a higher weight is given to the less frequent terms. This shows that the latter are generally positive. The highest positive value observed is in the period 2011-03-12 to 2011-03-26, immediately after the Irish General Election of 25 February 2011.

Figure 8 shows the results obtained through the *SVM* evaluation, whose training set is based on the classification done with the Active Learning algorithm. The observable trend in this case is a slight tendency towards negative sentiment. The lack of variation in the results may be due to the low number of features used for training and evaluation of the dataset.

The results given by the application of the Neural Network for the polarity classification of the dataset are shown in Figure 9. A generally increasing negative polarity sentiment can be observed, consistent with the results obtained with the SVM algorithm.

The five algorithms that have been applied on the dataset are dependent on different factors and use different characteristics and features for the sentiment polarity classification thus the variation observed in the shown results.

### 2.4.1 Detailed view of sentiment polarity

Due to the averaging of the sentiment values, first at the day level – as multiple stories can appear in the same day – then in the periods delimited by the European Summits – which vary from two weeks to more than a month – a lot of information about the stories' sentiments is lost.

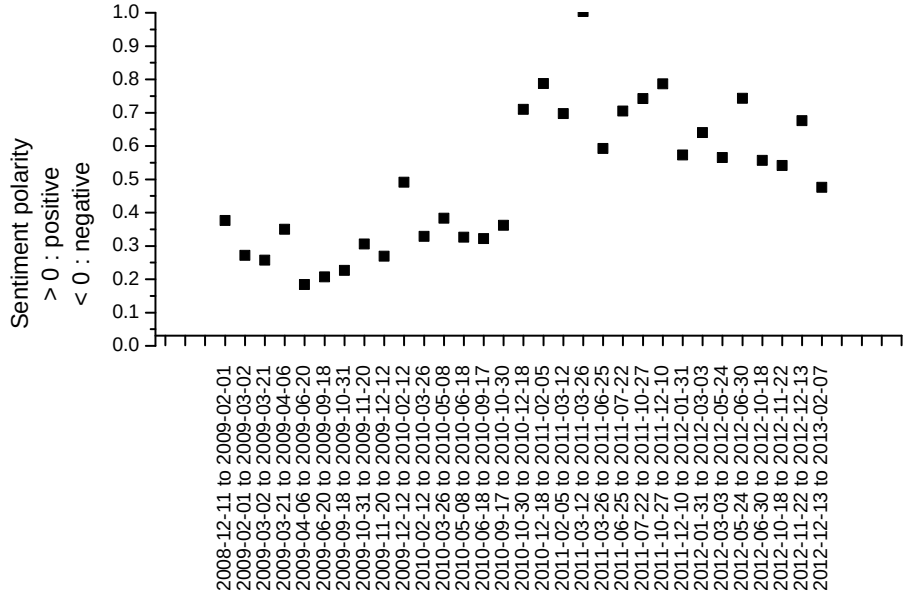



Figure 7: Tf.IDF sentiment polarity classification across the 2009-2013 period

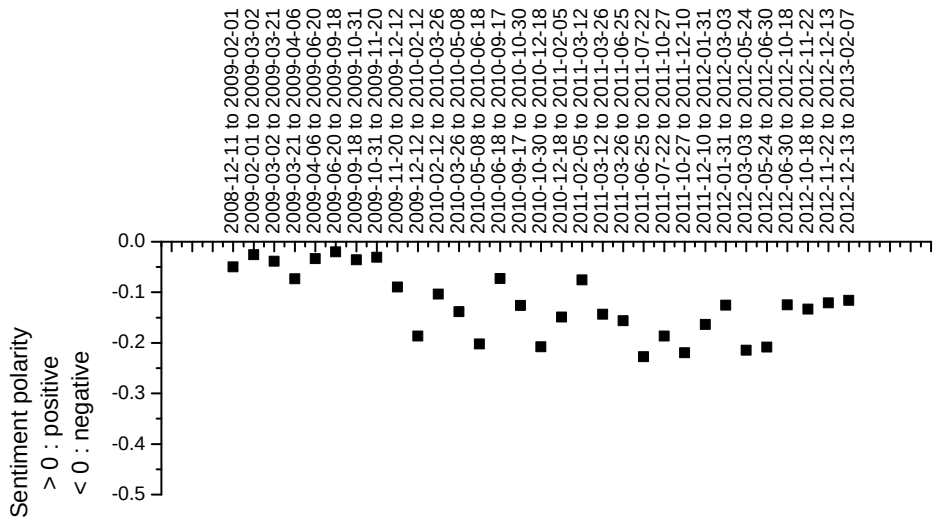



Figure 8: SVM sentiment polarity classification across the 2009-2013 period

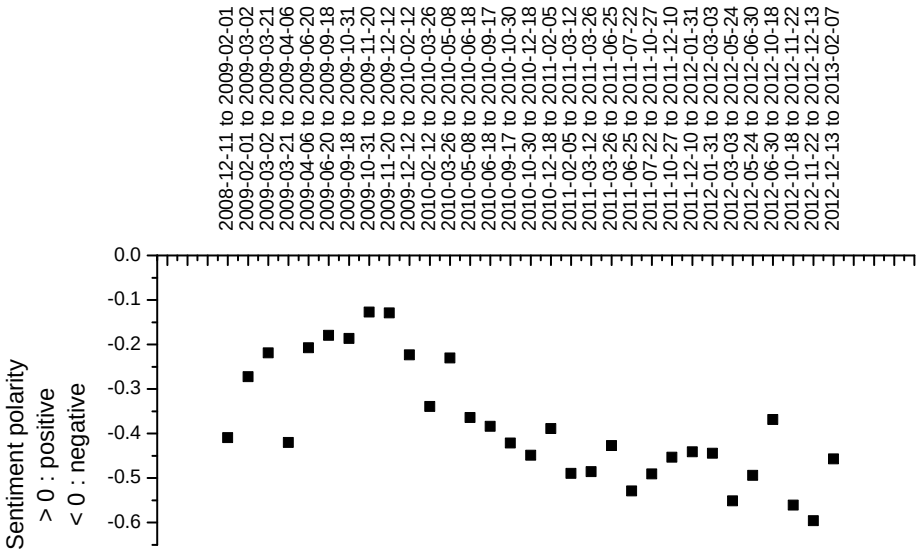



Figure 9: Neural Network polarity classification across the 2009-2013 period

The Figures 10, 11, 12, 13 and 14, show a detailed view of the sentiment polarity between 2011-03-11 and 2011-03-26, as classified by the five algorithms. This interval was chosen as it delimits two European Summits, following immediately after the Irish General Election of 25 February 2011, where the economic crisis led to a turning point in Irish politics. No averaging of the polarity values is performed and box plots are used in order to show for each date in the mentioned interval the sentiment values of the stories appearing in the corresponding date. By using box plots the interquartile range (IQR) with the mean and median sentiment polarity value, minimum and maximum values without outliers and the outliers themselves are shown. While this detailed view cannot be used on long time periods, it can be useful on short spans that are detected in the overall views as given in the beginning of this section.

In Figures 10 and especially 13 for example it can be seen that averaging the sentiment values would have given a neutral polarity classification, but in reality very diverse sentiment stories exist. Because the Active Learning, SVM and Neural Network algorithms classify the sentiment polarity into classes (-1 for negative polarity, 0 for neutral and 1 for positive), in Figures 11 and 14 the boxplots span the entire $[-1, 1]$ range, with the SVM classification (Figure 13) that contains no positive tagged stories, spanning the $[-1, 0]$ range. It is interesting to see in Figure 14, that while most sentiment is negative (due to negative terms such as 'debt' and 'crisis' that appear often) some positive outliers diminish this negativity, and by averaging it would be seen as only a slightly negative sentiment period, while the sentiment is actually strongly negative.

## 2.5 Conclusions and Future Work

This chapter has described the methodology used to classify the sentiment polarity of the full text of the *Story Takes* news items taken from the Thomson-Reuters 2009-2013 dataset that relates to the Irish financial crisis. Several machine learning algorithms have been implemented that perform both unsupervised and supervised sentiment classification and have been applied

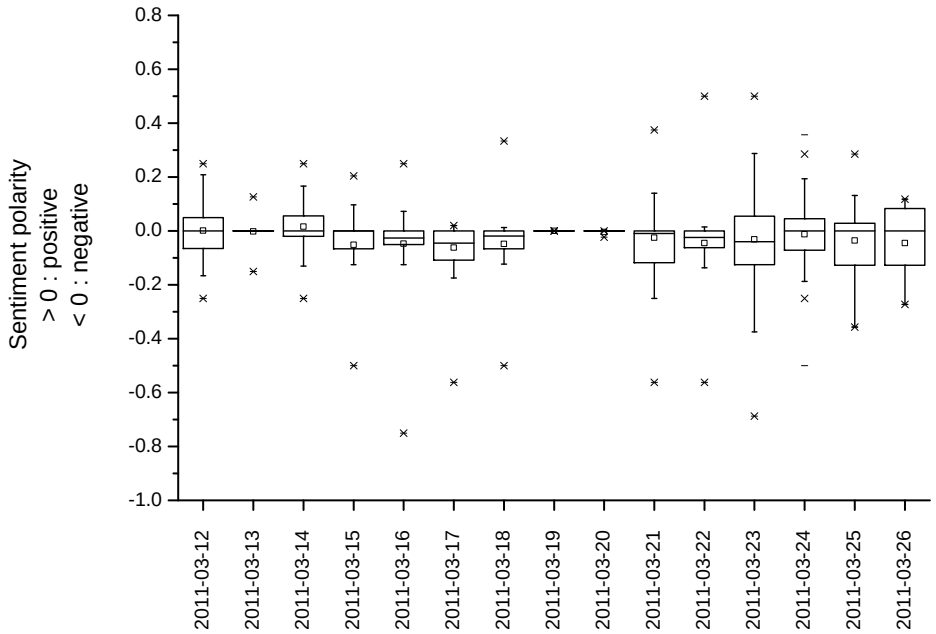



Figure 10: Basic sentiment classification in March 2011

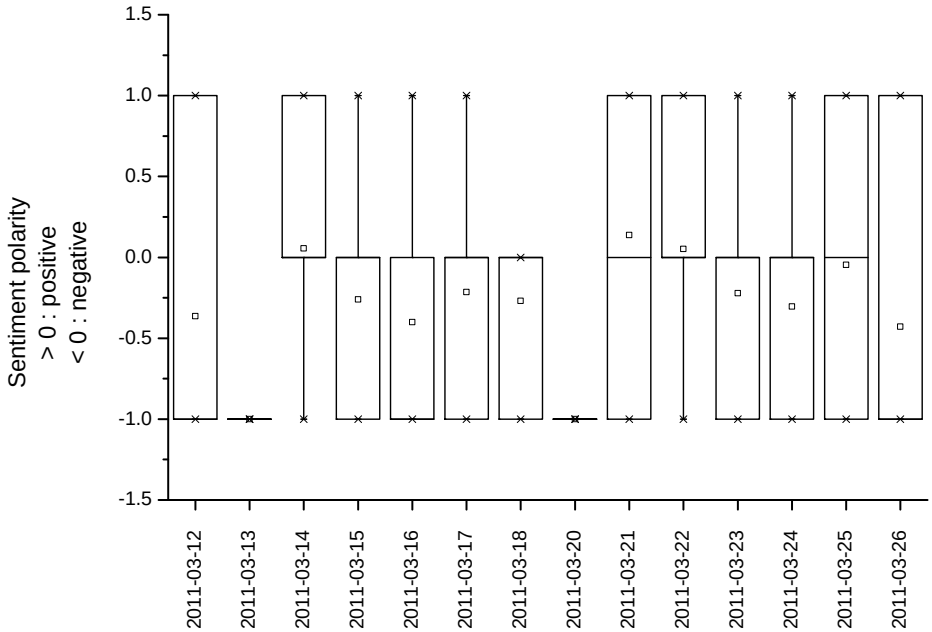



Figure 11: Active learning sentiment classification in March 2011

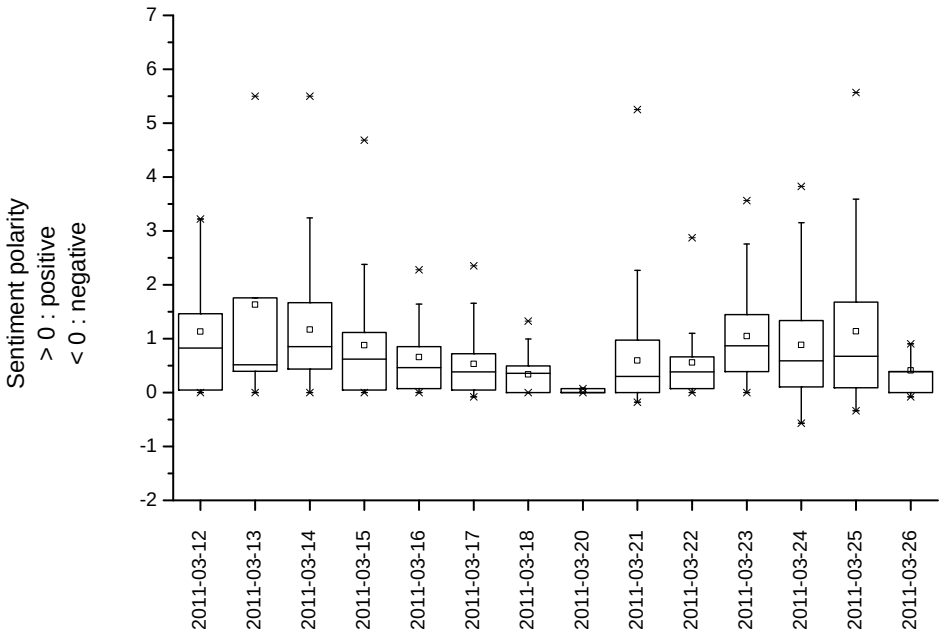



Figure 12: Tf.IDF sentiment classification in March 2011

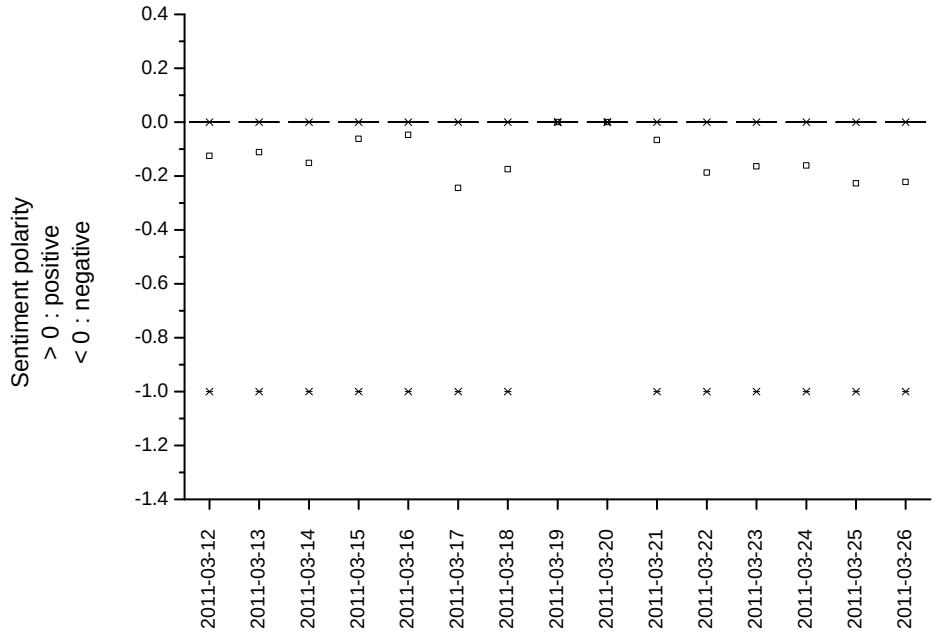



Figure 13: SVM sentiment classification in March 2011

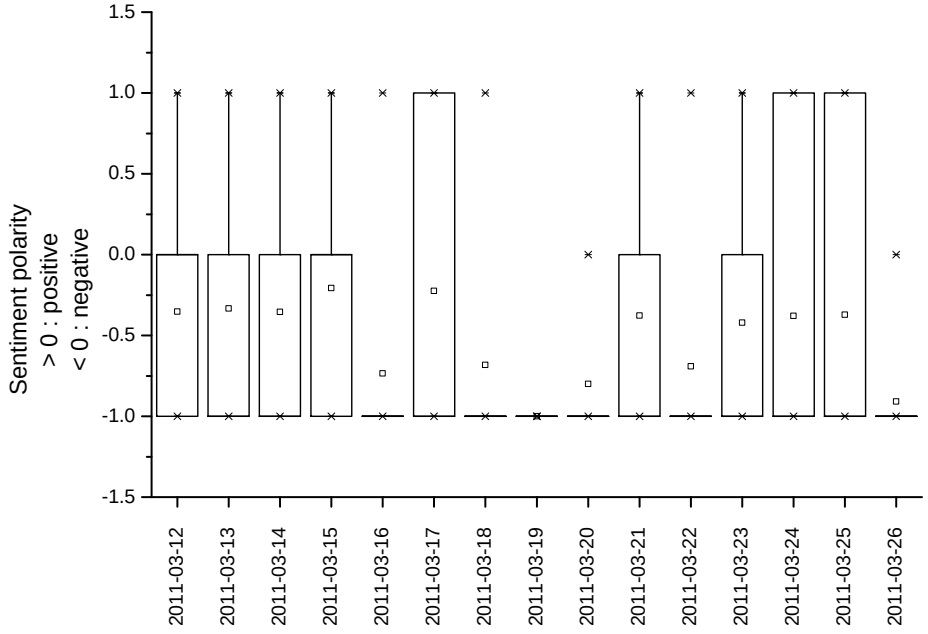



Figure 14: Neural Network sentiment classification in March 2011

to the mentioned dataset.

The results have shown a generally negative sentiment in the 2009-2013 time period, which can be correlated to the deepening Irish financial debt. Two possible visualisations for monitoring the sentiment polarity across this period have been given, contrasting the results of the different classification algorithms.

Some limitations of the present work are that few news items have been manually classified according to their sentiment polarity thus only a small accurate training set has been used for the *NN*. For the *SVM* the training set did not return highly accurate results and thus, in a future improvement, additional manual evaluations would be necessary. Due to the high sensitivity of these algorithms to the used training set, at least 10% of the most representative news stories should be chosen for manual tagging.

In addition to creating more comprehensive training sets for the classification algorithms, in future work the algorithms can be extended by incorporating automatic lexicon expansion [11]. This would result in a more exhaustive feature list that can then be used for in the *SVM* and *NN* classifiers. Also, in future work, an extended *Bag of Words* that encompasses more financial terms can provide a better basis for classification.

All of the classification algorithms that have been proposed in this chapter are dependent on the provided document and training sets, but are independent of the subject matter and can thus be used for the sentiment classification of other topics.

## References


[1] B. Pang, L. Lee, S. Vaithyanathan. *Thumbs Up?: Sentiment Classification Using Machine Learning Techniques. Association for Computational Linguistics, Stroudsburg, PA, USA*. 2002.

[2] E. Boiy, M-F. Moens. *A machine learning approach to sentiment analysis in multilingual Web texts. Springer, Netherlands*. 2009.

[3] C. Whitelaw, N. Garg, S. Argamon. *Using Appraisal Groups for Sentiment Analysis. ACM, New York, NY, USA*. 2005.

[4] I. G. Councill, R. McDonald, L. Velikovich. *What's Great and What's Not: Learning to Classify the Scope of Negation for Improved Sentiment Analysis. Association for Computational Linguistics, Uppsala, Sweden*. 2010.

[5] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A.Y Ng, C. Potts. *Learning Word Vectors for Sentiment Analysis. Association for Computational Linguistics, Portland, Oregon*. 2011.

[6] A. Balahur, R. Steinberger, M. A. Kabadjov, V. Zavarella, E. Van Der Goot, M. Halkia, B. Pouliquen, J. Belyaeva. *Sentiment Analysis in the News.. LREC*. 2010.

[7] M. Koppel, I. Shtrimberg. *Good news or bad news? let the market decide. Springer*. 2006.

[8] M. Thelwall, K. Buckley, G. Paltoglou. *Sentiment in Twitter events. Wiley Online Library*. 2011.

[9] B. Liu. *Sentiment analysis and opinion mining. Morgan & Claypool Publishers*. 2012.

[10] B. Liu. *Opinion mining and sentiment analysis. Springer*. 2011.

[11] H. Kanayama, T. Nasukawa. *Fully Automatic Lexicon Expansion for Domain-oriented Sentiment Analysis. Association for Computational Linguistics, Sydney, Australia*. 2006.

[12] N. Kobayashi, R. Iida, K. Inui, Y. Matsumoto. *Opinion extraction using a learning-based anaphora resolution technique. The Second International Joint Conference on Natural Language Processing (IJCNLP)*. 2005.

[13] L. Barbosa, J. Feng. *Robust Sentiment Detection on Twitter from Biased and Noisy Data. Association for Computational Linguistics, Beijing, China*. 2010.

[14] T. Li, Y. Zhang, V. Sindhwani. *A Non-negative Matrix Tri-factorization Approach to Sentiment Classification with Lexical Prior Knowledge. Association for Computational Linguistics, Suntec, Singapore*. 2009.

[15] F. Li, M. Huang, X. Zhu. *Sentiment Analysis with Global Topics and Local Dependency. Association for the Advancement of Artificial Intelligence (AAAI), Palo Alto, California, USA*. 2010.

[16] D. Ikeda, H. Takamura, L.-A. Ratinov, M. Okumura. *Learning to shift the polarity of words for sentiment classification. International Joint Conference on Natural Language Processing (IJCNLP)*. 2008.

[17] V. S. Subrahmanian, D. Reforgiato. *AVA: Adjective-verb-adverb combinations for sentiment analysis. IEEE.* 2008.

[18] A. Celikyilmaz, D. Hakkani-Tur, J. Feng. *Probabilistic model-based sentiment analysis of twitter messages. IEEE.* 2010.

[19] B. Liu. *Web data mining: exploring hyperlinks, contents, and usage data. Springer.* 2007.

[20] B. Pang, L. Lee. *Opinion mining and sentiment analysis. Now Publishers Inc., Hanover, MA, USA.* 2008.

[21] R. Moraes and J. F. Valiati and W. P. G. Neto. *Document-level sentiment classication: An empirical comparison between SVM and ANN. Science Direct.* 2013.

[22] A. Sharma, S. Dey. *A Document-level Sentiment Analysis Approach Using Artificial Neural Network and Sentiment Lexicons. ACM, New York, NY, USA.* 2012.

[23] C.T. Spathis. *Detecting false financial statements using published data: some evidence from Greece. MCB UP Ltd.* 2002.

[24] C. Pott. *Introduction to sentiment analysis. Stanford University (Slides CS224U).* Feb. 26, 2012.

[25] T. Armstrong. *NLTK Part-of-Speech Tagging. University of Delaware (Slides CISC889-11S).* Oct. 5, 2011.

[26] G. P. Zhang. *Neural networks for classification: a survey. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on.* 2000.

[27] B. Liu, M. Hu, J. Cheng. *Opinion Observer: Analyzing and Comparing Opinions on the Web. Proceedings of the 14th International World Wide Web conference.* May 10-14, 2005

[28] B. Settles. *Active learning literature survey. University of WisconsinMadison.* 2010.

[29] S. Bird. *Nltk: The natural language toolkit. In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics.* 2002.

[30] T. Schaul et al. *PyBrain. J. Mach. Learn. Res. ACM.* March 2010.

# 3 Sentiment Barometer and Alerts

Kavita Rege

The chapter deals with Bag-of-Words and Näive Bayes as two approaches for an Alert-Sentiment Barometer. We find that the Bag-of-Words is not only able to classify the alert sentiments but also has a potential to track the change in sentiment of alerts over time. The Näive Bayes is used only to see the change in distribution of the alerts over time. Both the approaches suffer from the assumption of independence of words used in the alerts and are susceptible to the initial bias introduced in classifying the alerts into one of the five classes based on an individual's perception of the sentiment of the alert.

**Keywords**: Sentiment Analysis, Irish Crisis, News Alerts, Näive Bayes.

## 3.1 Problem Description

In today's world, people's decision-making process largely depends on the opinions and reviews available on social media and online shopping sites such as Twitter, Facebook, Amazon, blogs, news feeds et cetera, therefore gathering of these opinions, reviews and analyzing their sentiments have become a major area of research.

According to [1] *Sentiment Analysis is defined as the task of finding the opinions of authors about specific entities.* These opinion text help business organizations, financial institutions to monitor their reputations, marketing manager of a firm to receive realtime feedback of the product, help people to buy a product or not. Many research papers are written and models are implemented for the analysis of sentiment, particulary, the discovery of reviews and opinions. In this chapter, we will consider the sentiment analysis of financial alerts during Irish crisis from 2009 - 2013.

According to [3], the author discusses about the boom and bust of Ireland during the period from 2008 to 2012. The main factor was the rise in the property market, due to which the Irish banking system financed the property demand through aggressive lending, but the decline in the property market which affected the construction activity severely crippled the Irish banking system which in turn led to economics crisis. Due to the crisis there was rise in unemployment. Banks which suffered considerable losses were Anglo-Irish Bank, which largely contributed to the overall loses in the Irish banking system.

Here, in our analysis, we had considered the news alerts during Irish crisis. We are going to deal with the sentiment change of these alerts during the time period from 2009 to 2013. Taking the set of financial terms given by the expert we will try to correlate these terms to the sentiment of news alerts during the Irish crisis and also try to analyze the financial term impact on news alerts during Irish crisis.

Sentiment analysis is carried out by two different approaches, word categorization method which makes use of Bag of Words and the other is a statistical method. For word categorization, we make use of pre-defined words, that is, these words are separated into different categories according to their sentiments. The statistical method we use is the Näive Bayes approach.

## 3.2 Related Work

Several research efforts has been carried out on sentiment analysis of peoples opinion on social media, blogs new articles etc.[5] deals with classifying documents considering over all sentiment

of the documents, determining whether the review is positive or negative, the movie review data was considered for the experiment and comparative study of machine learning techniques (Naive Bayes, SVM and maximum entropy) and human produced baseline was done. [6] discusses about the using Twitter (microblogging platform) for sentiment analysis explaining about automatic collection of corpus for sentiment analysis and performing linguistic analysis on the data set.

Much research has also been carried out on sentiment analysis of Financial news, articles and reviews. Many of the researcher targeting the stock market analysis and predictions. [4] the author discusses about correlation between the stock price movement and the tone and words used by the author in the news article, for this purpose they use *the Arizona Financial Text (AZFinText) system, a financial news article prediction system* and sentiment analysis tools. Not much work is carried out in the area of sentiment analysis of financial crisis news, especially considering Alerts as corpus.

## 3.3 Algorithmic Conception

### 3.3.1 Bag of Words

In the natural language processing techniques, "Bag of words" technique treats the documents as a set of words. In the bag of words approach, we do not consider the account of word order. This approach can be applied at the chapter level,document level, sentence level and used for the purpose of topic identification and sentiment analysis. For our experiment, we took financial terms $K = \{k_1, k_2, \ldots, k_{131}\}$, which were provided by experts and applied simple and straightforward method of classifying the alerts according to the sentiment scores. We considered the scores provided by four different persons $(K_{p1}, K_{p2}, K_{p3}, K_{p4})$ and took the average of each term score in the set, each term is then classified into five different classes namely very positive (vp), positive (p), neutral (o), negative (n) and very negative (vn). Considering the set of alerts $A$ and checking for the presence of $k_1, k_2, \ldots, k_{131}$ in each alert. if one or more terms are present, we add the scores of the terms present. At the end, we check the scores and analyze whether the sentiment of each term has changes over time.

### 3.3.2 Näive Bayes

We have a set of alerts $A$ and five classes namely very positive (vp), positive (p), neutral (o), negative (n) and very negative (vn) denoted by $C_i \forall i \in vp, p, o, n, vn$. The objective is to find out the class of an alert in the test data. We are assuming that we have at our disposal a set of alerts that can be used to train the data set. The Näive Bayes approach is used to classify the data from the test set into one of the five classes mentioned. We wish to point out that the class of each alert is subjective in nature and is allocated based on individual judgment. In other cases related to insurance, risk, health, the class is either success or failure and is usually clear cut (accident or no accident, default or no default, sick or healthy).

$$P(C_i|A) = \frac{P(A|C_i)P(C_i)}{P(A)} \tag{2a}$$

We are interested in the probability of the class $C_i$ given the alert $A$. Here $P(A|C_i)$ is the probability that the alert belongs to a particular class and is called the *likelihood function*. The term $P(C_i)$ is known as the *prior* and shows the prior probability of the occurrence of the class $C_i$. The term $P(C_i|A)$ is known as the *posterior* and gives the probability of the occurrence of

$C_i$ given the data. The denominator $P(A)$ is a normalising constant and can be derived as

$$P(A) = \sum_{C_i} P(A|C_i)P(C_i) \tag{3}$$

We drop the normalising constant and work towards identifying that model that gives the maximum posterior probability amongst all classes. So we write

$$P(C_i|A) \propto P(A|C_i)P(C_i) \tag{4a}$$
$$P(A|C_i) = P(w_1|C_i) \times P(w_2|C_i) \times P(w_3|C_i) \times ... \times P(w_n|C_i) \tag{4b}$$
$$= \prod_j P(w_j|C_i) \tag{4c}$$

where $w_j$ is the $j^{th}$ word in Alert $A$ and $C_i$ is one of the classes vp,p,o,n,vn.

## 3.4 Implementation Details

### 3.4.1 Data Cleaning

For the experiment, we considered the dataset of financial news alerts between February 2009 and January 2013 of Europe, which consist of of 29222 alerts. After initial cleaning of data, starting with removal of data which did not contain any alerts, for example "SIGN UP FOR BREAKINGVIEWS EMAIL ALERTS: ". Then each alerts consisted of lot of noise data. There were approximately 22 different fields, among them only two fields were important for our analysis namely, the date[the second field] and the alerts[8th field] in the file.
The next task was to extract only those data points which were concerned with Ireland, so Initially, we started checking manually each Alert. we found out that each alert usually start with the name of the company or institute

```
AIB SAYS EFFECT OF CHANGE TO GUARANTEE IS TO INCREASE THE CHARGE
TO 140 MLN EUROS FOR 2009.
ALLIED IRISH BANKS - LOAN PROVISIONS 5.4 BLN EUROS GOLDMAN SACHS
CUTS BMPS <BMPS.MI> PRICE TARGET TO EUR 0.22 FROM EUR 0.33;
RATING NEUTRAL
```

To extract more data points, we took all the alerts where the companies have offices in Ireland and were affected by the Irish crisis. But in the due course we found out that some of these companies does not exist now. So to avoid any ambiguity, we stuck up to a plan and extracted only those alerts which contained word "Irish" or "Ireland". Thus we got 2340 alerts which had either Irish or Ireland in them.

### 3.4.2 Bag of Words Approach

The next task was to give the score to the financial terms which were provided by the experts, these financial terms have impact on the sentiments of the financial news for this we have considered the range of scores

| Very negative | Negative | Neutral | Positive | Very positive |
|---|---|---|---|---|
| -1 | -0.5 | 0 | 0.5 | 1 |

Table 11: Score calculation

| Terms | scores_p1 | scores_p2 | scores_p2 | scores_p3 | Average |
|---|---|---|---|---|---|
| maturity | 0 | 0 | 0 | 0 | 0 |
| unemployment | -1 | -1 | -1 | -1 | -1 |
| european financial | 1 | 0 | 0.5 | 1 | 0.625 |
| cut interest | 0.5 | -1 | 0.8 | -1 | -0.175 |

In Table 1, we have taken scores given by four different persons and took the average score by summing up the four scores and dividing by 4 for each financial term.

Now to bifurcate these scores into very positive, positive, neutral, negative, we took a range of values from [-1,-0.32] as very negative, from [0.321,-0.12] as negative, from [-0.21,0.072] neutral, from [0.0721,0.32] positive and from [0.321,1] as very positive

We just considered each term as a whole and assigned a score, not considering how the meaning of the sentence change if present with the adjective and stop words, for example here we have assigned a score of -1 to unemployment which is very negative, but if we take a sentence *"there is no unemployment problem in the state"*, which becomes a positive sentence. To avoid such problem, the next step was to remove all the stop words. For this task we used the python nltk(natural language tool kit) which has a function to remove stop words.

```
from nltk.corpus import stopwords
stop = stopwords.words('english')
alert = ' '.join([word for word in alert.split()
if word not in stop])
```

After removing stop word, as the trend follows to use stemming(*stemming is the process for reducing inflected (or sometimes derived) words to their stem, base or root form, generally a written word form*[2]) and lemmatization as a part of pre processing, we did not require for our project in hand, since we needed whole of the financial terms to be compared with the alerts.

### 3.4.3 Näive Bayes Method

The data set had 650 alerts that have been split into a training set (400) and a test set (250). Normally in the Näive Bayes approach each data point also belongs to a set (male, female or sick, healthy, default or non-default etc.). Here we have five classifications: very positive (vp), positive (p), neutral (o), negative (n) and very negative (vn). We assign each alert to a particular class (vp,p,o,n,vn) based on the initial choice based on Bag of Words as decided by the experts. We use the following steps to obtain the classification of each alert

1. Compute the prior distribution for classes in the training set. To do this we compute the frequency of each class in all the alerts in the training set. Denote $f_{vp}, f_p, f_o, f_n, f_{vn}$ as the frequencies for vp,p,o,n,vn respectively. Then the prior probability is $p_i = \frac{f_i}{\sum_i f_i}$ $\forall i \in vp, p, o, n, vn$

2. Compute the conditional probability ($p_{alert|i} \forall i \in vp, p, o, n, vn$) of each alert given the specific class vp,p,o,n,vn. To compute the conditional probabilities we use the following steps.

   (a) All stop words were removed from the alerts

   (b) A set of unique words from all the alerts were identified and their total was denoted by $n_u$

   (c) We next computed the number of unique words in each class and their corresponding frequencies. We denoted this by $n_{(w,f_i)|c}$, where $w$ is the unique word and $f_i$ is its corresponding frequency and $c \in vp, p, o, n, vn$ is the class. Based on a random draw of 400 alerts in the training data set we get the total words in each class as follows

| class | total words |
|---|---|
| vp | 733 |
| p | 1051 |
| o | 904 |
| n | 100 |
| vn | 1617 |

   while the total number of unique words in all the alerts $n_u$ equalled 1292 and remain same across all simulations.

   (d) Next was to compute the conditional probability that each alert $a$ in the test set belonged to a specific class $p_{a|c}$ $c \in vp, p, o, n, vn$. To do this we allocated the probability of each word in the alert based on its class. The probability was $p_{(w,f_i)|c} = \frac{n_{(w,f_i)|c}}{n_u + \sum_{f_i} n_{(w,f_i)|c}}$ and in case the word was not present in the class, then we allocated the probability as $p_{(w,f_i)|c} = \frac{1}{n_u + \sum_{f_i} n_{(w,f_i)|c}}$. The total probability for each alert was the product of the individual probabilities. Hence $p_{a|c} = \prod_w p_{(w,f_i)|c}$. Since the numbers become very small on account of the product of individual probabilities we take the logs and assign the probability as $p_{a|c} = \sum_w \log \left[p_{(w,f_i)|c}\right]$

   (e) We repeated the above steps for each class, that is we computed the conditional probabilities of each alert belonging to each class $c \in vp, p, o, n, vn$.

3. Compute the posterior probability as the product of the prior and the conditional probabilities

4. The alert was classified into a particular class $c \in vp, p, o, n, vn$ depending the maximum of the five individual conditional probabilities for each alert in each class.

## 3.5 Results

### 3.5.1 Bag of Words Approach

As mentioned before we considered only those alerts which had 'Ireland' or 'Irish' in them, so we got 2340 alerts. After checking for 131 financial terms in each alert, we found out that only 652 alerts contained words from the bag of words, among them number of very negative terms:221, negative terms :17, neutral terms:149, positive terms:154 and very positive terms:111. Since each alerts contained around 17-20 words, we found out that, the probability of many terms appearing in the a single alert was very low.

In our case each alert contained only one term, so there was no change of sentiment value of terms over time from 2009 to 2013 and 2340 data set was too small for sentiment analysis.In fig:3, from the graph we can see that for the term unemployment, the score remains same throughout and hence there is no sentiment change. we also found out that out of 131 financial terms only 47 appeared in the alerts under consideration.

### 3.5.2 Näive Bayes Classifier

We obtain the probability of an alert belonging to a particular class depending on its posterior probability. We find that the distribution of alerts into each class for a randomly selected set of training alerts is as follows:

| class | posterior | prior |
|---|---|---|
| vp | 59 | 63 |
| p | 20 | 98 |
| o | 72 | 91 |
| n | 86 | 12 |
| vn | 13 | 136 |

We carry out a bootstrap of the sample by generating 500 randomly chosen alerts. We find that the posterior distribution differs from the prior as shown in Figure 17. In the typical case where one is analysing the sentiment of a particular phenomenon not spread over time such as a movie or political decision, the bootstrap method using the Näive Bayes method makes it more robust. However in the case of the Irish financial crisis, the bootstrap method gives a change in the sentiment values because of the time element. The crisis in Ireland evolved over time and the sentiment will also vary over time. Given the limited sample size of 650 data points, it was difficult to obtain random samples at each time instant to carry out a bootstrap procedure. If the sentiment about the crisis in Ireland were overtly negative at the beginning then the prior distribution would reflect a large negative value which would then diminish and the sentiment would exhibit a less negative or more positive value over time. When samples are generated randomly, we find that the posterior distribution for the negative sentiment has shifted to the right exhibiting a greater negative sentiment, which is also reflected in the other sentiments, showing a reduction in the posterior over the prior.
We find that the posterior distribution is different from the prior distribution as far as classification of alerts into classes is concerned.

## 3.6 Conclusions and Future Work

We have used Bag of Words and Näive Bayes as the two approaches for classifying the sentiment value of the alerts into five classes namely very positive (vp), positive (p), neutral (o), negative (n), very negative (vn). The bag of words gives us a barometer of the future changes in the sentiment values over time as well as the classification of the alerts into classes. The Näive Bayes gives us only the classification of the alert into a particular class. Both methods suffer from the lack of linguistic semantics and assume that word occurrences are mutually independent,

which is a very strong assumption. Also in the Bag of Words, it is not always easy to attribute positive, neutral or negative connotation based on preconceived notions of the words (ex: debt, reduction, assets, debt reduction is good, asset reduction is bad, Ireland suffered from erosion of asset values and non-reduction in debt). This leads to mis-classification of the alerts due to biases introduced. We have also used a bootstrap method to gauge the robustness of the posterior distribution arising from the Näive Bayes method. We find that sentiments are very sensitive to the sampling which imply that the sentiments are definitely changing over time. If the sentiment were overtly negative for a larger part of the time horizon and only a sample is chosen to train the data, then the sample introduces a bias in the training data set. We observe this bias in the comparison of the posterior and prior distributions. Thus the Näive Bayes approach is another method to investigate the potential change in sentiment over time or across a sample.

## References


[1] Ronen Feldman: *Techniques and Applications for Sentiment Analysis* DOI 10.1147/2436256.2436274

[2] http://en.wikipedia.org/wiki/Stemming

[3] Philip. R. Lane: The world Financial Review-Sharing Visions *The Irish Crisis*

[4] @articleSchumaker2012458, title = "Evaluating sentiment in financial news articles ", doi = "http://dx.doi.org/10.1016/j.dss.2012.03.001", url = "http://www.sciencedirect.com/science/article/pii/S0167923612000875", author = "Robert P. Schumaker and Yulei Zhang and Chun-Neng Huang and Hsinchun Chen",

[5] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan: 2002. Thumbs up? sentiment classification using machine learning techniques. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 79-86.

[6] Alexander Pak, Patrick Paroubek: Twitter as a Corpus for Sentiment Analysis and Opinion Mining. Université de Paris-Sud, Laboratoire LIMSI-CNRS, Bâtiment 508.

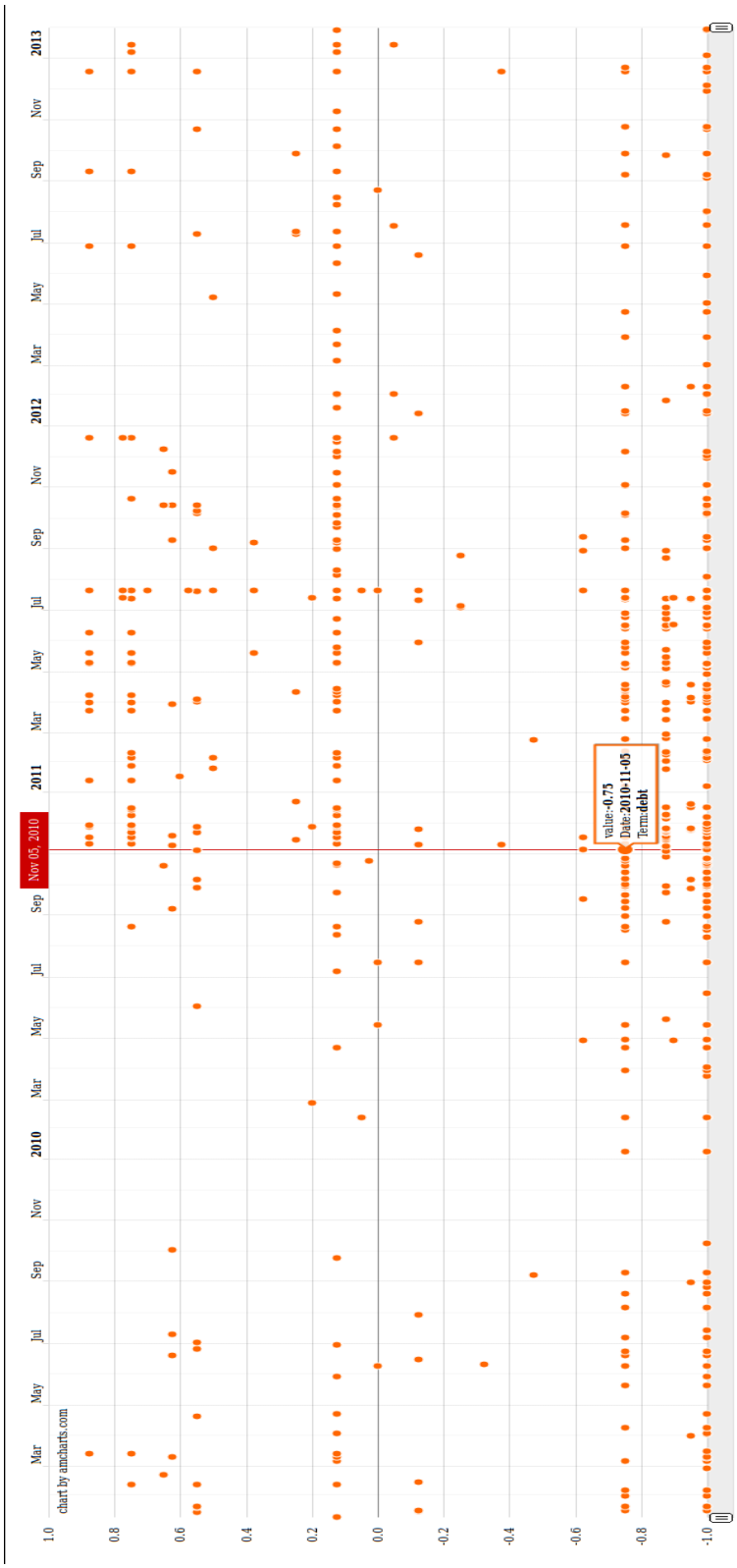



Figure 15: Bag of words Graph

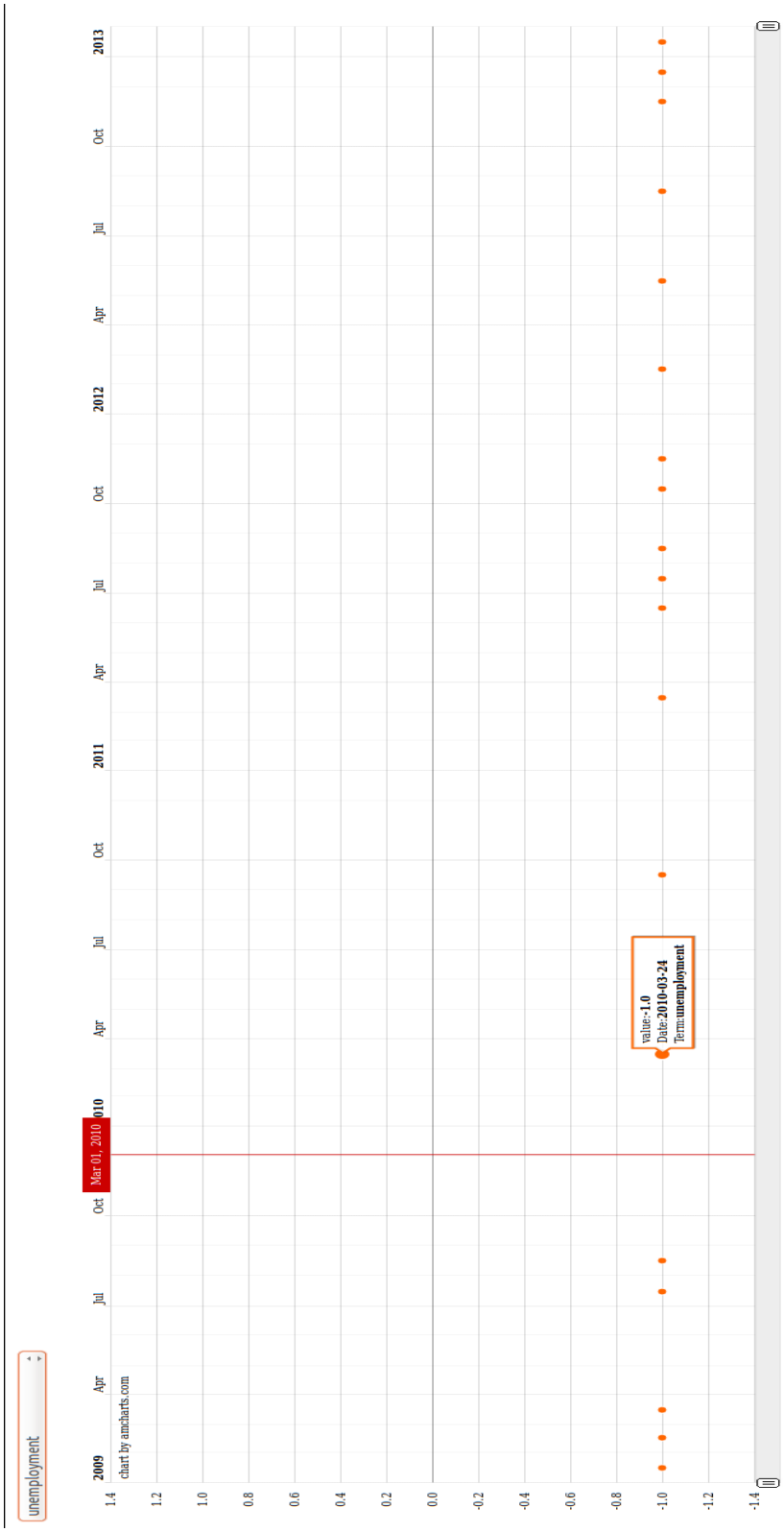
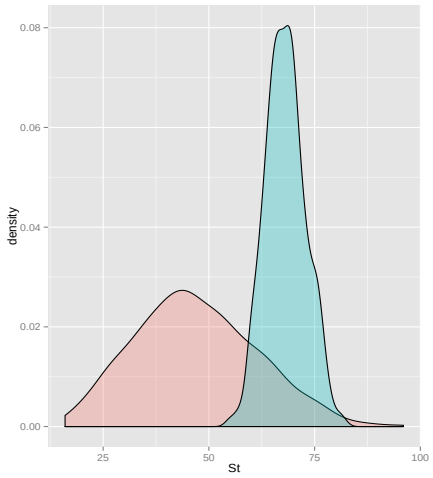
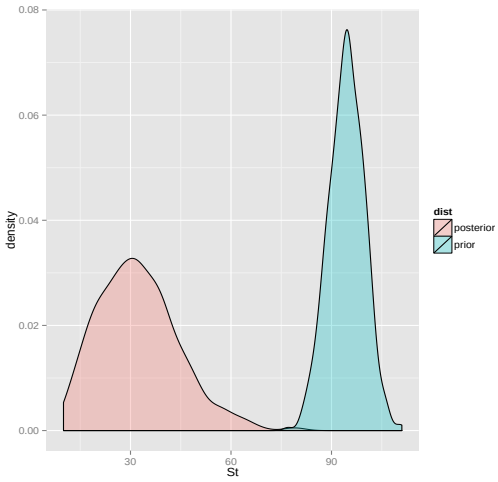


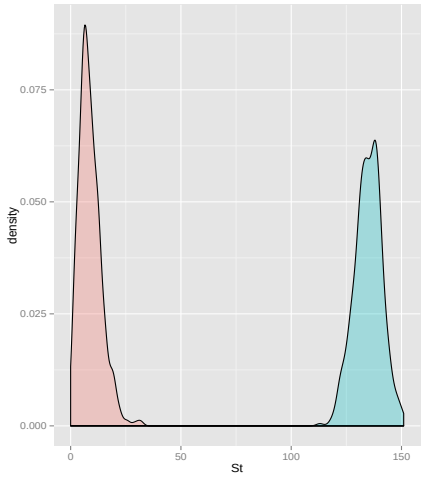

Figure 16: Change in term sentiment over time



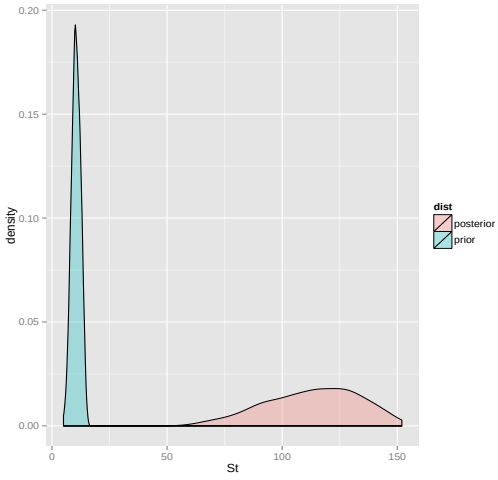

a) very positive sentiment



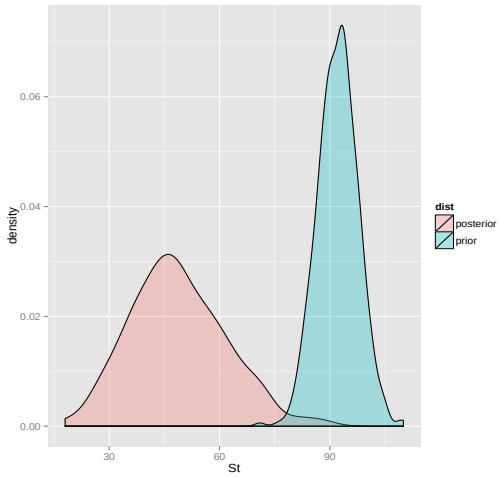

b) positive sentiment




c) very negative sentiment




d) negative sentiment




e) neutral sentiment

Figure 17: Prior and Posterior distribution from bootstrap to Alerts

# 4 Sentiments for the Headlines

Arash Atashpendar and Tahereh Pazouki

The work presented in this chapter describes the conception and development of a set of tools used for performing sentiment analysis on the headlines of news reports provided by Thomson Reuters covering the Irish financial crisis spanning from February 2009 until January 2013.

This is achieved by classifying headlines according to a sentiment polarity ranging from completely negative to completely positive, including the neutral sentiment. The classification is done with values between -1.0 and 1.0. Using the classified documents, a visual temporal sentiment barometer is developed to monitor sentiment swings over time, more specifically, the time intervals between the 27 European summits.

The methodology used for performing the sentiment analysis is a machine learning [6, 5] approach based on the "Bag of Words" method. In essence, it is based on a set of keywords or terms $\mathcal{T}$ that represent the core concepts of the topic in question, i.e. financial information. The aforementioned terms are mapped to sentiment values $BoW : \mathcal{T} \mapsto [-1.0, 1.0]$.

The values are assigned by a group of non-experts - the participants of the Machine Learning course - then the average of all provided values is taken as the seed. Furthermore, the sentiment analysis makes use of $tf.idf$ (Term Frequency - Inverse Document Frequency) values for the terms contained in the bag of words and uses them as term weights, i.e. degree of importance.

Finally, a series of statistical computations are done on the resulting data set to perform data aggregation based on time intervals between the European summits or multiple sentiment values computed for the same day. The developed visual sentiment barometer allows the user to define arbitrary time periods for monitoring the swing of sentiment over time.

**Keywords:** Sentiment Analysis, Bag of Words, Machine Learning, News Reports, Financial Crisis

## 4.1 Problem Description

The sentiment analysis [2] is to be done on a data set containing headlines of news reports provided by Thomson Reuters covering the Irish financial crisis spanning from February 2009 until January 2013. The main objective is to develop a *Sentiment Barometer*, i.e., a view on sentiments over time, taking into account the European Council meetings as time window constraints. News headlines are to be classified according to their sentiment polarity [1], which ranges from completely negative to completely positive, including the neutral sentiment. The classification is done with values between *-1.0* and*1.0*. The sentiment values are supposed to reflect the attitude and sentiment towards the Irish financial crisis during the above mentioned time period. A temporal sentiment barometer is needed to monitor the change of sentiment over time, more specifically, in between the 27 European summits.

The data set containing the news headlines, consists of $\approx 60,000$ records, and it is provided in the CSV format, which is a series of data fields that are comma separated. The data entries contain fields that are not related to the analysis and therefore only relevant fields should be extracted. The data elements of interest include the headline itself and the timestamp indicating the time of their publication. The following shows an extracted example of a date-headline entry for the headlines from the CSV source file: "*2009-03-27, Bank of Ireland want tailored bad debt*

*scheme*"

The methodology used for performing the sentiment analysis is a machine learning technique based on the "Bag of Words" method. In essence, the bag of words is a set of keywords or terms $\mathcal{T}$ that represent the core concepts of the topic in question, i.e. financial information and each term is mapped to a sentiment values $BoW : \mathcal{T} \mapsto [-1.0, 1.0]$. The values are assigned by a group of non-experts - the participants of the Machine Learning course - then the average of all provided values is taken as the seed. Furthermore, the sentiment analysis makes use of $tf.idf$ (Term Frequency - Inverse Document Frequency) values for the terms contained in the bag of words and uses them as term weights, i.e. degree of importance. $tf.idf$ provides a measure of the degree of importance of a term in terms of its rate of occurrence.

The data set should also be filtered to a seed of initial terms using the bag of words and terms related to the context of the Irish financial crisis. In other words, records not containing terms that are either not found in the bag of words or which are not context-related should be discarded as they would not contribute to the sentiment analysis.

The classification of the headlines with values between $-1.0$ and $1.0$ is done so that the change of sentiment over time can be monitored. Basically, once the filtering is done and the data set contains only entries related to the Irish financial crisis, each headline is to be assigned an overall sentiment value which classifies it in terms of reflecting a positive, neutral or negative value. In order to accomplish this task, the sentiment value of every term that is found in the bag of words should be retrieved and weighted with its $tf.idf$ value. Next, the overall sentiment of the headline will be computed by calculating the weighted average of the terms that have a sentiment value.

Finally, using the developed temporal sentiment barometer, one should be able to monitor the change of sentiment reflected by the media's attitude towards the Irish financial crisis and possibly map them to specific events that might be of significance. This allows analysts to shed some light on the underlying causes of certain fluctuations of sentiments over time.

The project report is organized as follows: In subsection 4.2, the algorithmic and conceptual approach put in place for carrying out the sentiment analysis is described, followed by a detailed description of the actual implementation itself in subsection 4.3. Then, the obtained results will be discussed and analyzed in subsection 4.5 and finally, a series of conclusions will be drawn in subsection 4.6 based on the findings presented in this chapter.

## 4.2 Algorithmic Conception

This subsection describes the conceptual phase of the project and the approach that led to the sentiment analysis and the actual development of the resulting temporal sentiment barometer. It begins by explaining how an initial proper understanding of the context plays a crucial role in the choices made in the subsequent steps. Then, the methodology for designing the system and the processing of the data is presented.

### 4.2.1 Understanding

The very first step towards developing a temporal sentiment barometer based on the provided data set is to understand the structure of the data. The data set available in the CSV (comma-separated values) format was studied using the Reuters News Archive User Guide document.

This document provides details and explanations related to the semantics of every field in each record.

### 4.2.2 Bag of Words: Fixing the initial seed

The next step is related to the second data set, i.e. the Bag of Words (BoW) containing the keywords related to the Irish crisis. In this phase, the sentiment values needed to be assigned to each term by all the course participants separately in order to be able to take the average of everyone's input and use these final values as the initial seed for all groups.

An idea aimed at dealing with this phase was suggested by our team and it entailed sharing the Bag of Words in a spreadsheet using Google Docs with the rest of the participants and to have each person provide their own evaluation and sentiment values for each term in a separate sheet. This made it possible to compute the average of all provided inputs and use the resulting BoW and the corresponding values as a common starting point for everyone.

Figure 18 shows the spreadsheet used for assigning the sentiment values to the terms in the Bag of Words. Once the final values for the initial seed are ready, the terms and their mapped sentiment values can be used to filter the Irish financial crisis data set.

### 4.2.3 Data Pre-Processing

Having understood the data set and a ready to use bag of words, the next logical step is to process the actual raw data. The CSV data is analyzed in order to firstly discard corrupt entries and secondly to retrieve only relevant records.

**Filtering relevant records** It was decided to filter the data set using the bag of words. This is done in order to discard records that do not contain any terms that are also found in the bag of words. Otherwise, the headline would not get a sentiment value at all.

Next, the remaining records were filtered once again using a set of predefined terms reflecting the semantics of the Irish crisis and Ireland. This part was required because a large of the entries are completely unrelated to the Irish crisis.

**Cleaning and normalizing records** Certain entries contained extra symbols such as colons, semicolons, etc. which were removed for further simplifying the subsequent processing steps. This was mainly an issue because certain terms were for instance either followed or preceded directly with a comma without a white space in between and this would have made the look up phase erroneous.

### 4.2.4 Store in Memory

The following step in the conception phase of the project was to figure out a way for storing the filtered and cleaned data in such a way that further processing would be more efficient in terms of performance. Hash tables or dictionaries were the ideal choice for storing the Bag of Words. The terms get hashed and used as keys and their corresponding mapped sentiment values as dictionary values. Finally, the corpus itself containing the headlines would be stored in a list-of-lists structure that would contain all the headlines and their terms in exploded form.

| | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Financial Terms | Score | | Human Players | Score | | Countries | Score | | |
| 2 | credit event | 0 | | paul thomson | -1 | <<Poul Thomsen | Austria | 0 | | |
| 3 | eurozone stability | 0.7 | | klaus masuch | -0.5 | | Belgium | 0 | | |
| 4 | bank liquidity | 0.5 | | matthias mors | -1 | | France | 0 | | |
| 5 | secondary bond markets | 0.5 | | preben aamann | 0 | | Germany | 0 | | |
| 6 | secondary markets | 0.5 | | diederik debaecker | 0 | <<Diederik DE BACKER | Ireland | 0 | | |
| 7 | bond markets | 0.5 | | herman rompuy | 1 | | Italy | 0 | | |
| 8 | mkt access | 0.5 | | joaquim almunia | -0.5 | <<Joaquín Almunia | Luxembourg | 0 | | |
| 9 | euro debt crisis | -1 | | olli rehn | 1 | | Netherlands | 0 | | |
| 10 | credit ratings | 0.5 | | jose barroso | 1 | | Portugal | 0 | | |
| 11 | credit ratings agencies | 0.7 | | jean claude trichet | -0.5 | | Spain | 0 | | |
| 12 | rating agencies | 0.7 | | mario draghi | 0.5 | | Switzerland | 0 | | |
| 13 | irish issue | -1 | | dominique strauss kahn | 0.5 | | United Kingdom | 0 | | |
| 14 | irish crisis | -1 | | christine lagarde | 0.5 | | United States | 0 | | |
| 15 | borrowing costs | -0.5 | | juergen ligi | 0.5 | | | | | |
| 16 | debt as proportion to gdp | -0.5 | | jean claude juncker | 0.5 | | | | | |
| 17 | debt to gdp | -0.5 | | axel weber | 0.7 | | | | | |
| 18 | debt/gdp | -0.5 | | jens weidmann | 0.7 | | | | | |
| 19 | debt relative to gdp | -0.5 | | karl theodor zu guttenberg | 0.5 | | | | | |
| 20 | irish debt | -1 | | rainer bruederle | 0.5 | | | | | |
| 21 | cuts to ratings | -0.8 | | philipp rösler | -0.5 | <<Philipp Rösler | | | | |

Figure 18: Bag of Words shared on Google Docs

### 4.2.5 Processing and Sentiment Assignment

After having all the required data prepared and stored in memory, lemmatization was performed both on the corpus as well as the BoW to improve the headline to BoW matching hitrate. This would prove to be of little use in the end.

Next, for each headline, the sentiment values of its terms that are also found in the BoW would be assigned in addition to computing the $tf.idf$ value of each such term to be used as weight. Having the sentiment value and the weight of each term, the sentiment value of each headline would be computed as the weighted average of the sentiment values of its individual terms.

The algorithm for finding collisions between the BoW and the corpus uses the BoW as its starting point for iterating through the corpus. In other words, for each term in the BoW, which could be a single word or a compound and combination of multiple words, the algorithm would go through the headlines in the corpus and look for matches. This decision leads to a linear complexity in the function of headlines. Doing the same thing the other way around, i.e. iterating through headlines and looking for collisions in the BoW, would have led to an exponential explosion since that would have meant taking all possible combinations of the terms in each headline and checking against the BoW.

Certain efficiency related schemes were designed from the very beginning in the conception phase, such as precomputing and storing the IDF values of terms to avoid unnecessary real-time computation and thus save time and computation resources. These will be elaborated on in subsection 4.3.

### 4.2.6 New Data Set and Post Processing

The new data set, i.e. headlines with their sentiment values, will be stored in a new data set and exported once again to the CSV format as it would be needed for the post processing phase, which would be done in a spreadsheet.

### 4.2.7 Data Aggregation in Spreadsheet

The exported data would be processed in a spreadsheet. This decision was grounded in the fact that spreadsheets are well-suited for handling large amounts of structured data and for performing aggregation operations.

The data is aggregated based on the date field for both the entire period of the provided data, as well as in between the European summits. The aggregation over the entire period is done by averaging the weighted sentiment values of headlines published on the same day to reflect the overall feeling towards the Irish crisis on specific days. This would prove to be a necessary operation as the final monitoring phase would be cumbersome with multiple values on the same day.

The aggregated data would be stored separately and exported to CSV again in order to be used by the visualization application that implements the temporal sentiment barometer.

### 4.2.8 Visual Temporal Sentiment Barometer

The exported data from the spreadsheet will be converted to the JSON format as this is the format that will be accepted by the visualization application which will be developed in the form

of a web application. The visualization application implements the temporal sentiment barometer with scrolling, navigation and zooming capabilities. These operations would be needed for precise monitoring of the sentiment value swings during specific time periods, with having the possibility of arbitrarily small or big time windows.

### 4.2.9 Conceptual Design Diagram

The flowchart-like diagram given in Figure 19 illustrates the steps and their relationship with respect to each other as explained above.

## 4.3 Implementation Details

This subsection will describe the implementation phase and it will mainly focus on the used programming languages, data formats for input and output, external libraries and a description of the source code. In this subsection, first the used programming languages will be presented and then the used data format for input and output, followed by covering the used external libraries in addition to describing the structure of the source code and its elements.

Finally, a detailed description of the implementation phase including the intermediary steps will be given.

### 4.3.1 Programming Languages

The main programming language used for the development of the project was Python. Python was mainly used for processing (pre & post) the raw data and computing the sentiment values. On the visualization side, Javascript and HTML were used for the implementation of the visualization application that provides the temporal sentiment barometer.

### 4.3.2 Data Structures for Input and Output

The input data, namely the corpus acquired from Thomson Reuters was available in the CSV format. After the preprocessing step, the corpus was stored in memory using a 2-dimensional structure, i.e. list of lists, which contained all headlines, each of which containing a list of their words. This decision made the processing step easier.

Moreover, the BoW was stored in a hash table to allow for fast lookup while mining for collisions between the BoW and the corpus and also for looking up the sentiment values of the terms.

Once the data is processed, it is exported to the CSV format again, containing the original fields, i.e. timestamp and headline, along with an additional field for the computed sentiment values. The decision for this format was motivated by the fact that the resulting data is post-processed in a spreadsheet and CSV is an the format for doing so. Furthermore, the conversion of the CSV data to JSON for the visualization application can also be done very easily with widely available tools, and a manual implementation of such a tool would not be costly at all.

### 4.3.3 External Programming Libraries

The developed source code is mostly made up of in-house developed components with very little reliance on external libraries, however a few libraries were nevertheless used for performance and efficiency reasons.

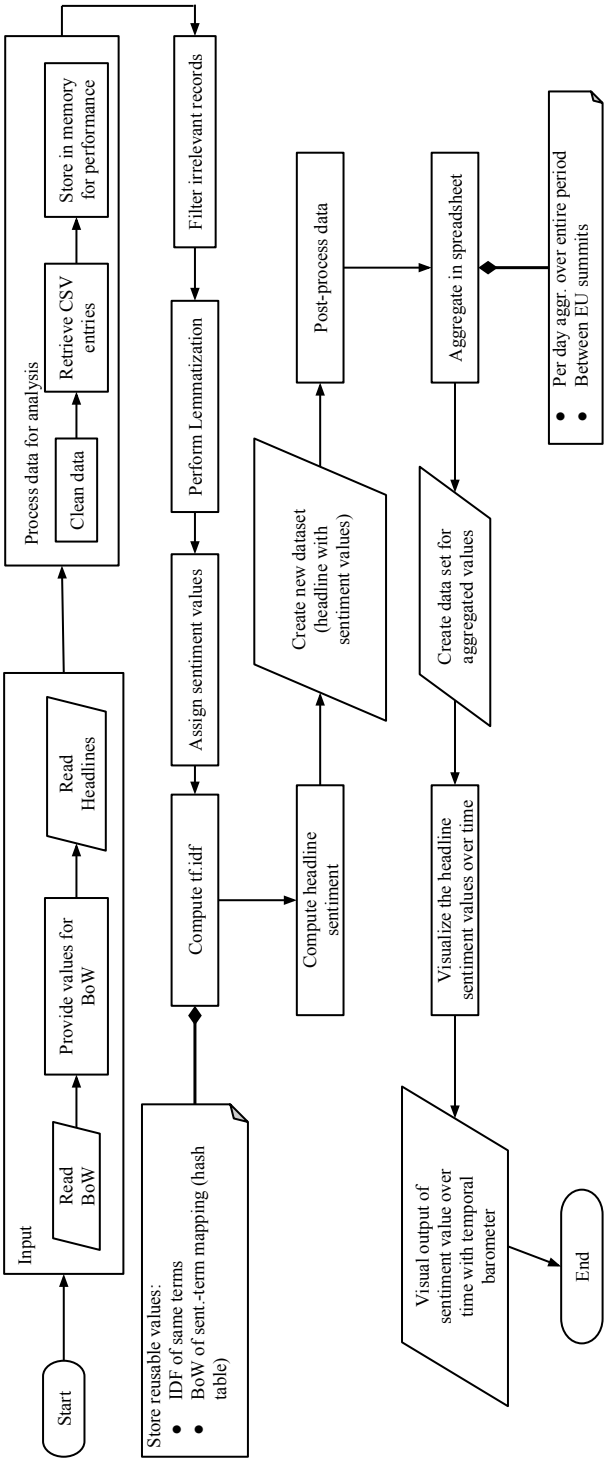



Figure 19: Conceptual Design Diagram for Headlines

**NLTK** The Python NLTK [3] (Natural Language Toolkit) was used for performing certain operations on the corpus such as lemmatization and stemming.

**amCharts** Additionally, the amCharts [4] Javascript library was used in the implementation of the visualization application for the temporal sentiment barometer.

### 4.3.4 Source Code Structure

The source code is made up of Google Docs Javascript functions, a file "sent_analyzer_v2.py" and a series of HTML/Javascript files implementing the visualization application. These will be further detailed in this subsection.

Javascript functions and mathematical formulas written in Google Docs were used for the post-processing phase in spreadsheet, i.e. aggregation, filtering and sorting operations.

The "sent_analyzer_v2.py" file takes care of the main bulk of the work in terms of importing, reading and loading up the raw data as the corpus into memory along with the BoW and performing the preprocessing step in addition to the actual sentiment value computation and assignment and finally exporting the new data set to CSV.

The Javascript/HTML files developed for the visualization application consist of two main files. These files include "sentiment_barometer.html" and
"multi_sent_barometer.html".

The first allows the user to choose from three different data sets - namely headline sentiment values over the entire time period, per day aggregated values and the periods between the European summits - and visualize the selected data set separately so the user can navigate, scroll and zoom in/out on the data. The second application allows the user to do the same thing but it also makes it possible to visualize multiple data set at the same time for comparison purposes.

## 4.4 Overview of Implementation Steps

The implementation was done according to the following steps:

1. Parse & Process the data (Python & NLTK)
   (a) Parse data set & BoW
   (b) Filter entries with no collision with BoW (data reduced to $\approx 10K$)
   (c) Filter non-Irish crisis related records (data reduced to 1015 records)
   (d) Lemmatize Corpus
   (e) Compute term & document sentiments
   (f) Export results to CSV
2. Post-process & aggregate data in spreadsheet
   (a) Entire period of provided data set

[3] `http://nltk.org/`
[4] `http://www.amcharts.com/`

(b) Between European summits

(c) Per day aggregated headline sentiments

3. Visualize the resulting data using the Sentiment Barometer application

## 4.5 Results and Result Discussion

This subsection will discuss the obtained results along with the final developed temporal sentiment barometer and the two visualization applications that allow the user to navigate through the data and monitor headline sentiments over time. Before delving into a discussion surrounding the obtained results, a description of the two developed visual temporal sentiment barometers will be provided. Next, the focus will be on the actual results, what they represent and mean and how the sentiment barometer could be used to look for possibly significant data points.

The spreadsheets displayed in Figure 20 shows the post-processing phase and the new data set containing the sentiment values. The other sheets are used for performing the statistical computations and data aggregations.

### 4.5.1 Visual Temporal Sentiment Barometers

The final results that were aggregated in three different ways in the spreadsheet processing phase were exported to JSON so that they could be used as input to the two developed temporal sentiment barometers, which were developed in the form of sentiment value visualizers over time with navigation, scrolling and zooming capabilities.

**Temporal Sentiment Barometer** This tool allows the user to visualize the sentiment values of the headlines over time. It provides a list of data sets so that the user can select a data set of their choosing that they would like to visualize. Upon selection of a data set, the application automatically loads the corresponding data set and visualizes the sentiments values on the $y$-axis over time, which is defined on the $x$-axis.

The user can then select a time period by moving the two date sliders located above the plot. If the user expands the time interval or restricts it to smaller time windows, the tool redraws the sentiment values of the headlines published within the selected period in real-time and automatically updates the plot. The time line widget that allows the user to change the time windows, displays in its background a miniaturized area chart showing activity, the tendencies, the locality of data and the amount of data. Moreover, the plot is a dynamic plot in that it allows the user to hover the mouse over the data points to see the exact date along with the exact sentiment value. The screenshots given in Figure 21 and in Figure 22 further illustrate the functionality of this tool.

**Multi Data Set Temporal Sentiment Barometer** The second temporal sentiment barometer allows the user to select multiple data sets at the same time so that they can compare different behavior reflected by the data points with respect to each other.

The screenshot given in Figure 23 shows the tool visualizing all three data sets, i.e. entire sentiment-headline data set, per day aggregated data set and the aggregation of the data set between the European summits.

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | 2009-01-14 | EU executive approves Anglo Irish recapitalisation | 0.5000 | | |
| 2 | 2009-01-14 | Ireland won't need IMF financing- IMF spokesman | 0.3000 | | |
| 3 | 2009-01-19 | Irish govt back down on Anglo debt plan -reports | -0.9000 | | |
| 4 | 2009-01-19 | UPDATE 1-Irish government back down on Anglo debt plan | -0.9000 | | |
| 5 | 2009-01-19 | UPDATE 2-Irish government back down on Anglo debt plan | -0.9000 | | |
| 6 | 2009-01-19 | TEXT-Moody's downgrade Anglo Irish Bank to A2/Prime-1/E+ | -0.8000 | | |
| 7 | 2009-01-28 | ANALYSIS-Anglo Irish debt pledge lift market spirit | -0.9500 | | |
| 8 | 2009-01-29 | Euro zone inflation to trough midyear -Irish cbank | -0.5400 | | |
| 9 | 2009-01-30 | Irish debt seen a riskiest in the euro zone | -0.9500 | | |
| 10 | 2009-01-30 | EURO GOVT-Inflation data boost short bonds; Ireland seen risky | -0.5400 | | |
| 11 | 2009-01-30 | EURO GOVT-Inflation data boost short bonds; Ireland seen risky | -0.5400 | | |
| 12 | 2009-02-02 | UPDATE 1-Ireland seen boosting bank bailout this week-media | 0.1400 | | |
| 13 | 2009-02-05 | TEXT-Moody's change outlk on bank debt backed, Irish govt neg | -0.9000 | | |
| 14 | 2009-02-11 | BRIEF-Ireland to announce bank bailout plan at 2000 GMT | 0.1400 | | |
| 15 | 2009-02-12 | TEXT-Moody's downgrade Bank of Ireland to Aa3/C | -0.8000 | | |
| 16 | 2009-02-12 | TEXT-Moody's cut Allied Irish Banks plc snr debt to Aa3 | -0.9000 | | |
| 17 | 2009-02-12 | IMF repeat Ireland doe not need IMF financing | 0.4000 | | |
| 18 | 2009-02-17 | TEXT-Moody's downgrade Irish Life & Permanent to A1/C | -0.8000 | | |
| 19 | 2009-02-25 | Ireland say aim to raise $32 bln debt in 2009 | -0.9000 | | |
| 20 | 2009-02-26 | BRIEF-Citigroup downgrade Allied Irish Banks, Bank of Ireland | -0.8000 | | |
| 21 | 2009-02-27 | Irish Q4 unemployment 7.7 pct, employment slump | -0.8400 | | |
| 22 | 2009-03-04 | Irish unemployment up at 10.4 pct in Feb - PM | -0.8400 | | |
| 23 | 2009-03-06 | INTERVIEW-Fitch say Ireland need to cut deficit by $5 bln | -0.7400 | | |
| 24 | 2009-03-09 | U.S. deficit more of a concern than Irish-Germany | -0.7400 | | |
| 25 | 2009-03-10 | Irish GDP to fall at least 6 pct in 2009 -c.bank | 0.0000 | | |
| 26 | 2009-03-13 | Spain's Solbes say no talk of Irish bailout | 0.2400 | | |
| 27 | 2009-03-16 | Ireland eye 2010 deficit/GDP around 5-6 pct-finmin | -0.3700 | | |

Figure 20: Imported data set with sentiment values ready for post processing

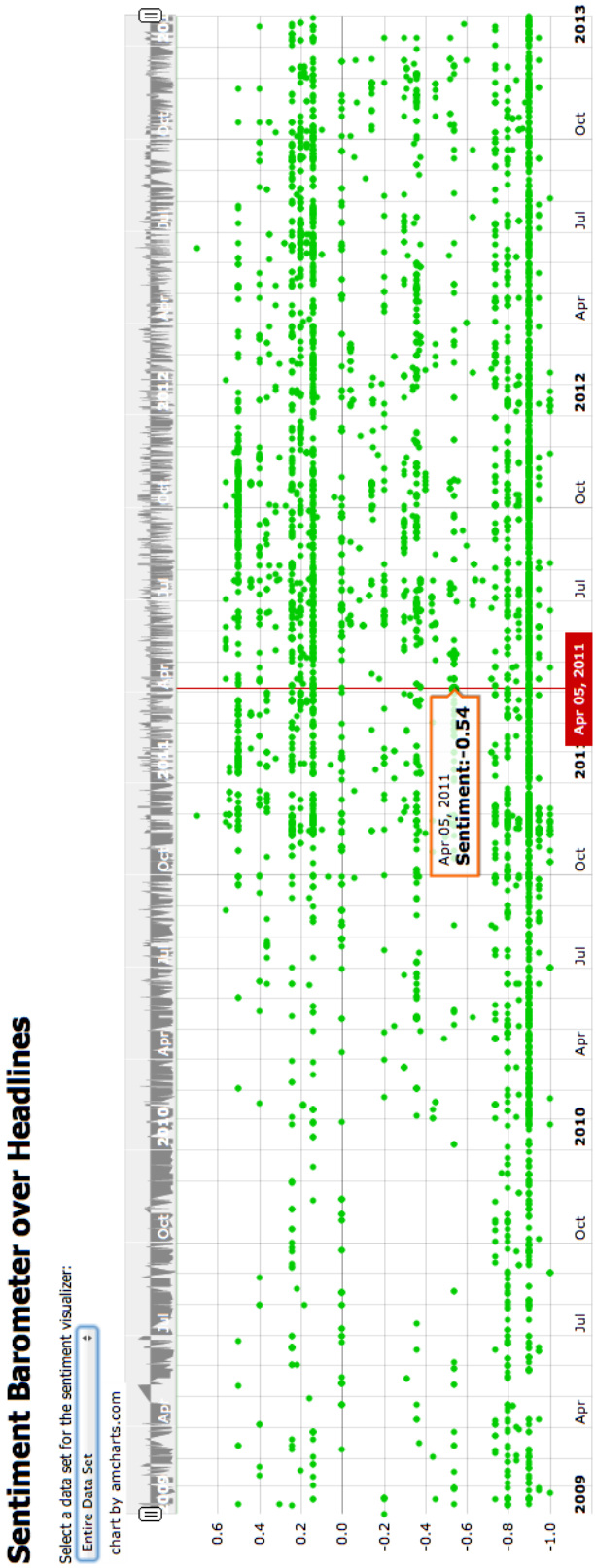



Figure 21: Temporal Sentiment Barometer visualizing the entire headline-sentiment data set

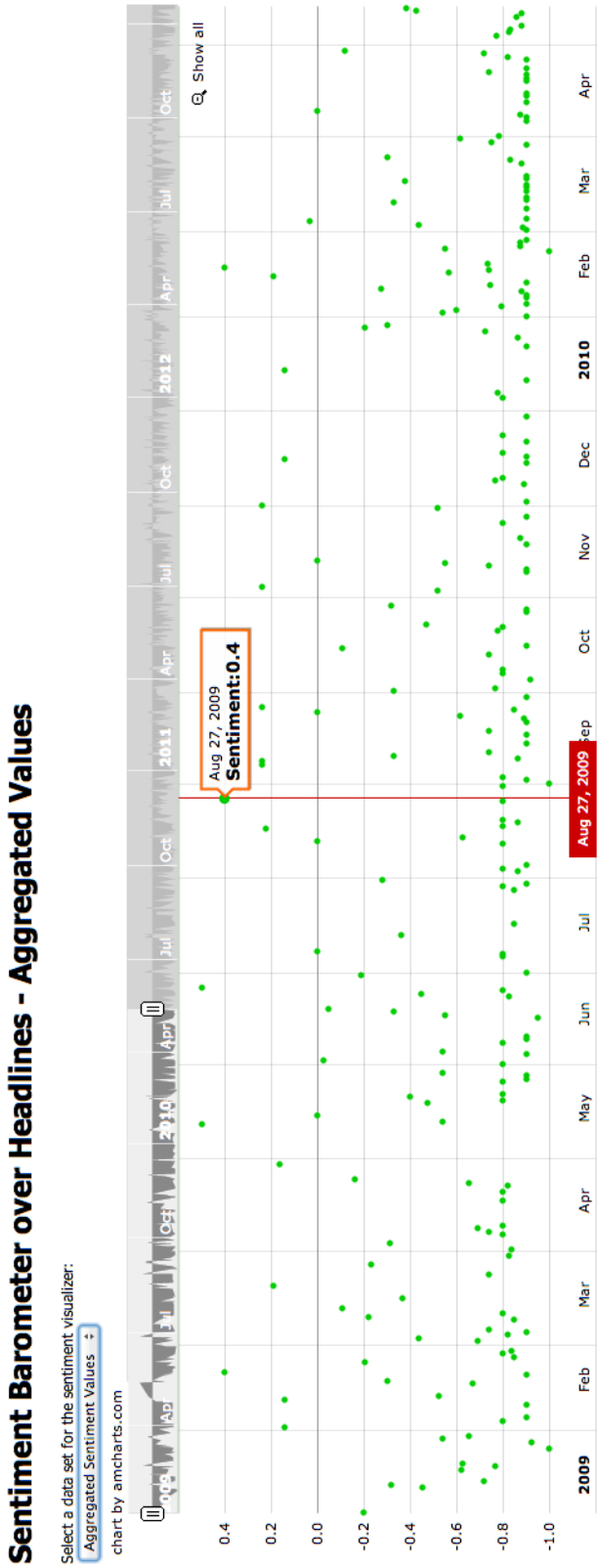



Figure 22: Visualiztion of the per-day aggregated data set with restricted timeframe

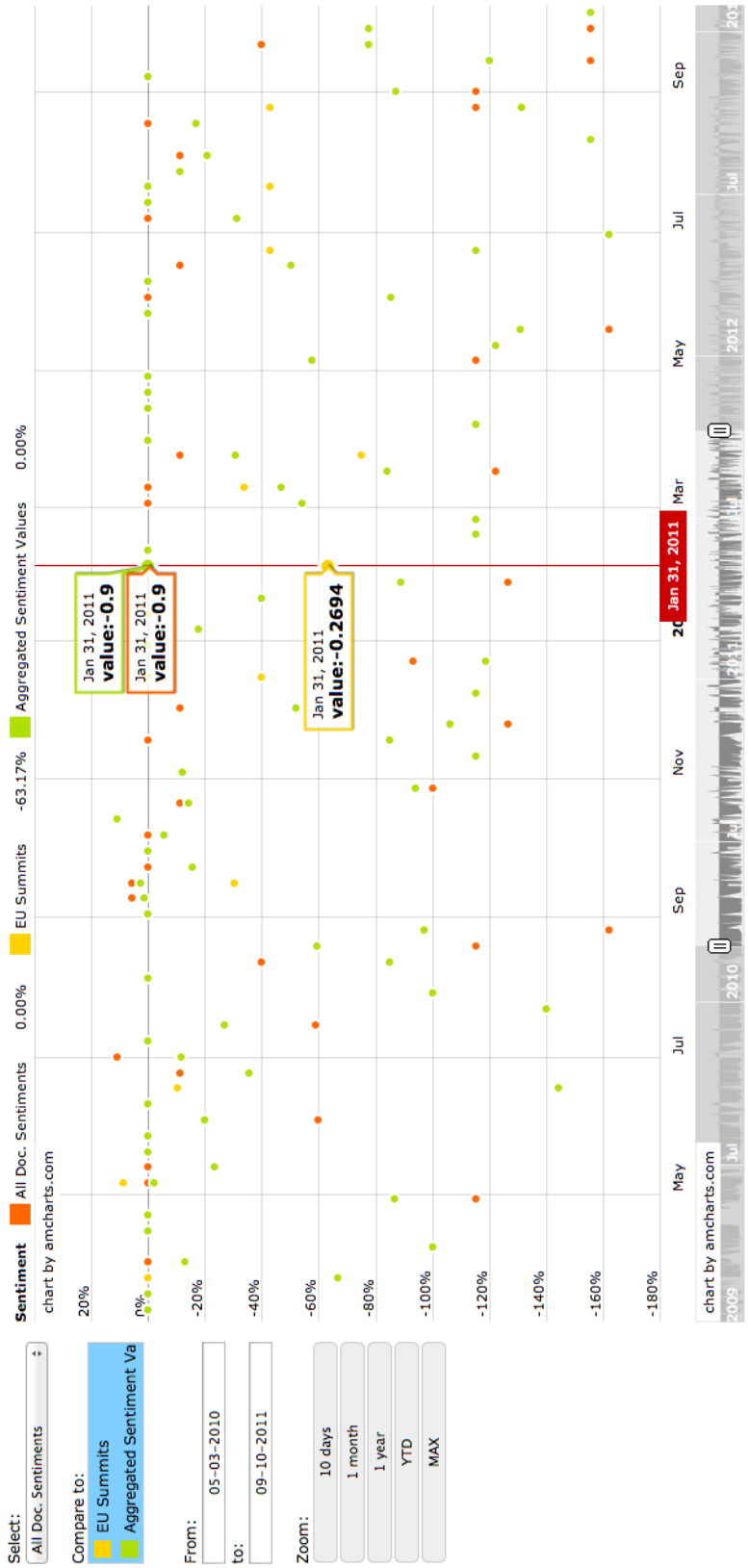



Figure 23: Multiple data sets visualized in the same plot

Furthermore, as shown in Figure 23, on the left hand side of the application, it is possible to select specific time intervals by manually selecting exact dates for the two bounds. It is also possible to change the granularity of the view according to year, month, 10 days, etc.

### 4.5.2 Result Semantics

In the case of the visualization covering the entire data set, the points visualized by the sentiment barometer, represent sentiment values of individual headlines on a given day.

As for the the aggregated results, in the case of the European summits, the visualization represents the aggregated sentiment values between the European summits, i.e. averaged value of headline sentiments in between the intervals. And in the case of the per day aggregated data set, the displayed values represent averaged sentiment values of headlines published on the same day, meaning that for any given day in the data set, there is only a single sentiment value. This is meant to reflect the collective feeling of the media towards the Irish crisis on a given specific day.

### 4.5.3 Result Discussion

The visualized results provide several indicators such as the amount of traffic or activity, i.e. publication rate on the subject of the Irish crisis, the fluctuations and swings of sentiment over time. The obtained results are basically a set of headlines, each of which issued on a specific date and having a computed sentiment value. Further argumentation techniques for evaluating sentiments in financial news articles are given in [3, 4].

As for the actual sentiments and feelings reflected by the news reports, it can be seen that most of the resulting processed data set lies below the value 0, which indicates for the most part a negative sentiment. Furthermore, as shown in Figure 24, the amount of activity, i.e. news reports, increases sharply starting from the first of October 2010.
It can also be observed that the amount of data points located in the positive domain increase past the first of October 2010. However, the overall view indicates a largely negative sentiment throughout the entire time frame. Curiously, as indicated by the blue rectangle in Figure 24, one can observe a higher density of sentiment values around $-0.1$ - n other words completely negative sentiments - when compared to the points past the first of October 2010.

**European Summits used as Indicators** Probably the most useful data set would have to be the one representing the aggregated sentiment values in between the European summits. This visualization is particularly important because it allows one to infer certain information and make certain deduction by correlating the aggregated sentiment values and the swings with the European summits' freely available reports.

Things such as financial packages or war related issues and tensions are strong indicators for the change of sentiment over time. A prime example of this is illustrated in Figure 25.
Since the data point indicated in Figure 25 displays the highest value, i.e. the most positive sentiment value, among all other points, it may of interest to further investigate its significance. The correlation is done as follows: The point represents a value averaging the sentiment values computed since the previous meeting and by looking up the report covering the meeting before this date, one can extract the following from the report: *"**Economic Policy:** The European Council today adopted a comprehensive package of measures to respond to the crisis, preserve financial stability and lay the ground for smart, sustainable, socially inclusive and job-creating*

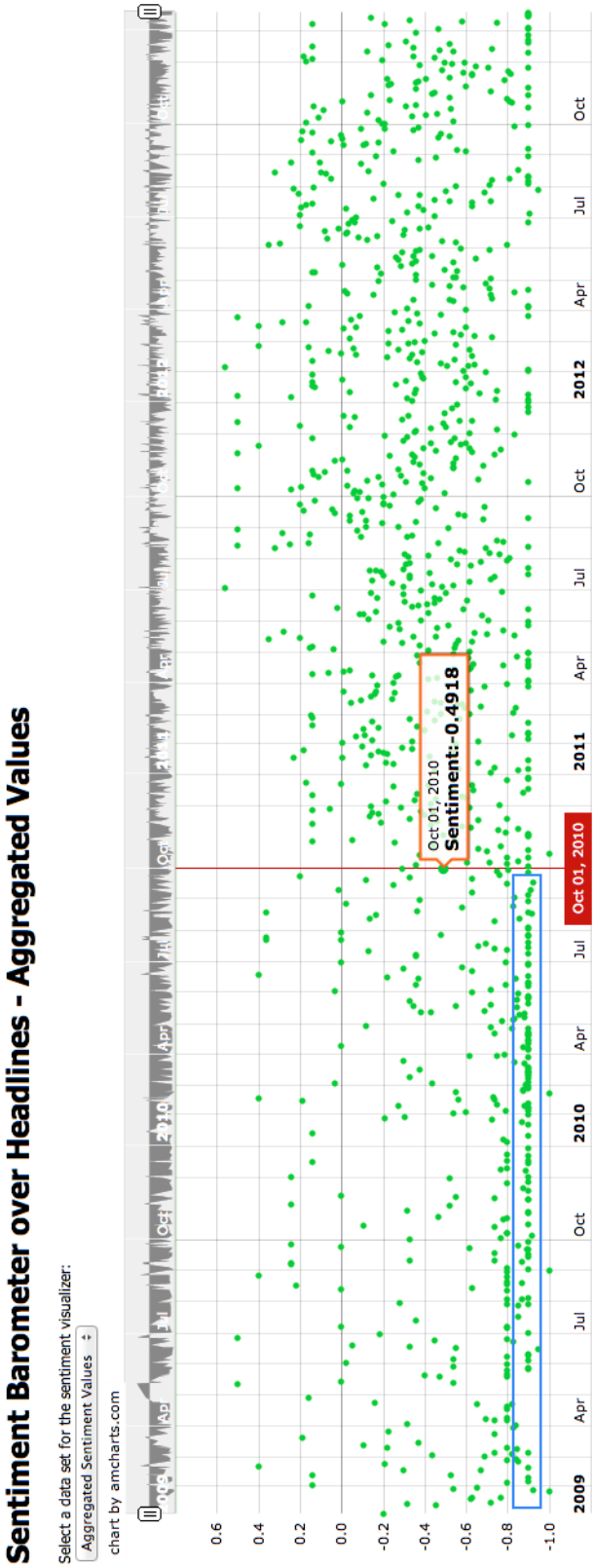



Figure 24: Activity increases starting 1st of Oct. 2010 onwards

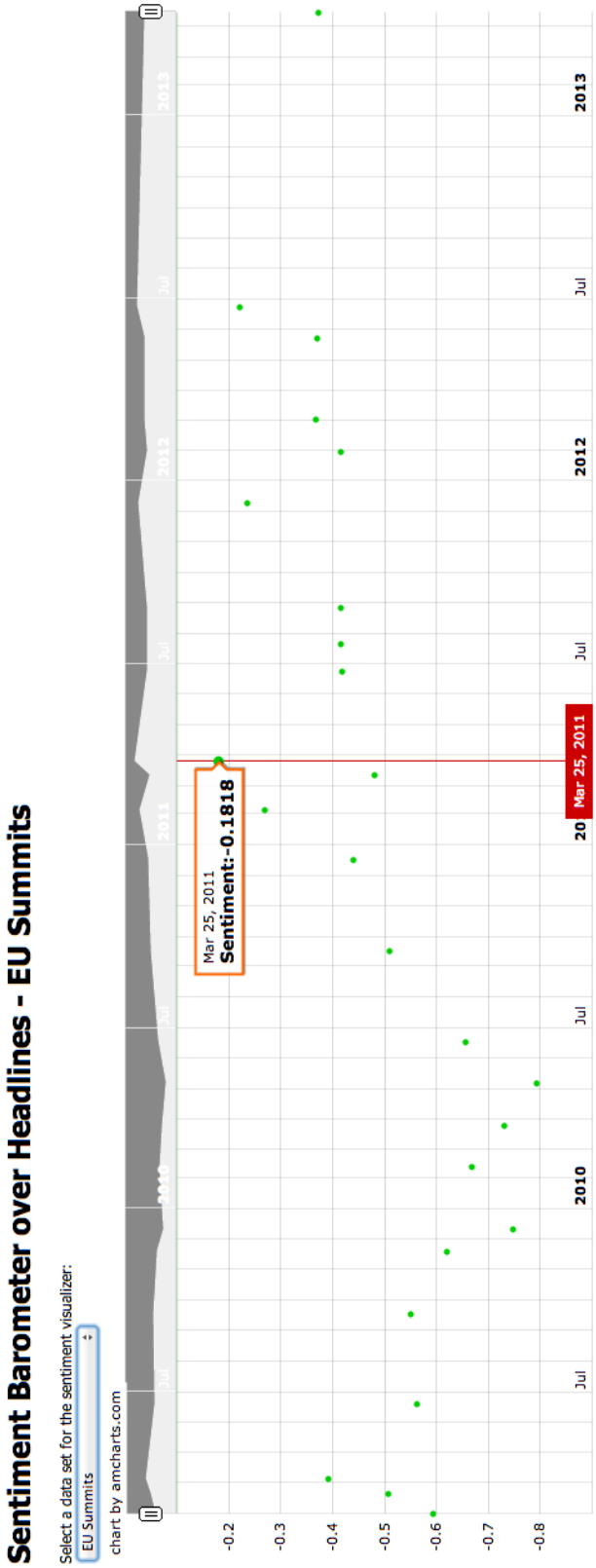



Figure 25: Aggregated sentiment values between the 27 European summits

*growth. This will strengthen the economic governance and competitiveness of the euro area and of the European Union.*"

One can infer that this peak could be the result of the decision towards strengthening the financial situation in Europe.

## 4.6 Conclusions and Future Work

This chapter presented a project aimed at performing sentiment analysis on the news headlines issued by Thomson Reuters covering the Irish financial crisis from February 2009 up to January 2013. The provided descriptions gave insight into many aspects of the project including a complete problem description, a detailed elaboration on the algorithmic and conceptual phase of the project as well as details pertaining to the actual implementation.

Moreover, an extensive discussion of the obtained results and their semantics and significance was provided based on the visualizations made by the developed temporal sentiment barometer. It was shown and discussed how the computed and visualized data points could be correlated to specific events and interpreted in a meaningful way.

### 4.6.1 Limitations

While the presented project fulfills the expected tasks to a very large extent, it still lacks in two specific realms. One aspect is related to the pre-processing phase due to the fact that using more intelligently defined filters and a richer bag of words, one could improve upon the existing rate of false positives and false negatives during the filtering phase. In other words, possibly retrieving more entries that are related to the Irish financial crisis, which could also mean discarding less seemingly unrelated records.

The other issue is related to the visual temporal sentiment barometer and the lack of the possibility to view an excerpt of the visualized headline sentiment values. This would allow the user to see the headline text itself along with the currently displayed value.

### 4.6.2 Strong Points

The developed tool set, provides a straightforward and rather efficient mechanism for parsing the news reports and enriching them with sentiment values. The efficiency of the implementation allows for a convenient testing and debugging process and development cycles.

Finally, the provided visualization tool allows the user to navigate and explore the obtained results in an intuitive manner and zoom in and out on the data with arbitrary precision. This feature makes the application a useful tool for financial analysts wanting to monitor sentiment swings over time and correlate them with financial, political or other significant events.

### 4.6.3 Future Work

This project could be improved in the way it does its preprocessing of the data and also the methodology it uses for initially cleaning and normalizing the input data. This would imply the usage of more sophisticated natural language processing tools.

Furthermore, yet another natural language processing and semantics related improvement, could be the implementation of algorithms that compute semantic similarity between terms so that a smarter processing scheme could be put in place.

## References

[1] Wilson, Theresa, Janyce Wiebe, and Paul Hoffmann. *Recognizing contextual polarity in phrase-level sentiment analysis. Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing. Association for Computational Linguistics*, 2005.

[2] Pang, Bo, and Lillian Lee. *Opinion mining and sentiment analysis.* Foundations and trends in information retrieval 2.1-2 (2008): 1-135.

[3] Schumaker, Robert P., et al. *Evaluating sentiment in financial news articles.* Decision Support Systems 53.3 (2012): 458-464.

[4] Feldman, Ronen. *Techniques and applications for sentiment analysis.* Communications of the ACM 56.4 (2013): 82-89.

[5] Wu, Xindong, et al. *Top 10 algorithms in data mining.* Knowledge and Information Systems 14.1 (2008): 1-37.

[6] Mitchell, Tom. *The discipline of machine learning.* Carnegie Mellon University, School of Computer Science, Machine Learning Department, 2006.

# 5 Sentiment Barometer and Authors

Saeed Afshari and Korok Sengupta

Looking at the Irish Crisis from the perspective of different European council meetings, our objective has been to analyze the sentiment barometer of the authors and contributors of the posts during that phase of time, for each of the council meetings. The reason for this analysis of sentiments of the authors has been to get a look into how the European Council Meetings influenced various writers contributing to the Irish Crisis scenario. The decisions and outcome of the meetings, how they reflected on the common people could be seen from the variation of reactions of the posts that the writers contributed during that particular time span.

**Keywords:** Sentiment Analysis, Bag of Words, Machine Learning, News Reports, Financial Crisis.

## 5.1 Problem Description

The objective of this project was to analyze the sentiments of the authors to the text that we have found. When analyzing the documents for sentiment analysis and looking in depth into each of them, one can not simply ignore the human factor involved in each of the articles that has been written in the above mentioned time frame. The authors, who contributed their articles, do not only have their essence involved in the text but also the sentiments they express are of vital importance. To understand that, we have approached the problem in the subsections mentioned below.

This chapter is organized in a simple progressive manner for the readers to understand what steps have been taken by us in the process of extraction of the data from the complete dataset and how we have processed it for the final analysis. There are five sections: after the *problem description* of our work and the hurdles we have crossed in order to produce such results. In *Section 2*, we will look into the *Algorithmic Descriptions*. In *Section 3*, we will demonstrate how we have *implemented our algorithms*. *Section 4* gives some *results and discussions* about the results achieved. Finally, *Section 5* has some *conclusions and future work* for a further discussion.

### 5.1.1 Sentiment

Every post that comes across in the Irish Crisis when analyzed as per the European Council Meetings, the most important aspect for our search was to analyze the sentiments expressed in those posts for the nature of the sentiments expressed would give us an idea at that time for that meeting, who was thinking what with respect to positive, negative or neutral categories.

*Problems Encountered*: For every one of the European Council meetings, the sentiments of the authors were calculated as per the values assigned to the bag of words. once the calculation was done, on many of the of those occurrences of the European Council Meetings, Sentiment value came as zero (0). The reason being the bag of words were not occurring in those posts.

### 5.1.2 Author Names

Names initially posed a great challenge. There might still be certain types of names that we have not figured out. On further examining the document we saw that almost all the documents

had email addresses of the authors by which we can at least track from whom the article had been generated. Extraction of emails made us stumble upon a lot of other information like the number of people who posted how many times in that time frame. People, who had phone numbers for contact in those articles, gave us another important direction in which the authors were writing from which geographic location.

*Problems Encountered*: Names from emails is a difficult bargain when considering "Authors" for there could be a lot of contributors from official mails, for example say *mail@ryanair.com*. Also one single author could contribute one important article in the span of the European Council dates but his count will be low considering some author contributing 20 articles in that span which may not be that important.

### 5.1.3 Location

On extraction of emails, we came across the telephone numbers due to problem in the white spaces of the parent document. And, then we thought of using these phone numbers for location of the posts. This gave us an idea about which post from which location had the a particular sentiment at a particular point of time. Our objective thus became a little more diverse than just finding authors and their sentiments attached with the articles. It leaped over to various nations that had authors contributing to this Irish Crisis scenario.

*Problems Encountered*: One major issue is that there are a lot of posts that do not have telephone numbers associated with them and thus the information in that zone stays incomplete. Another issue that we came across was a single post having phone numbers with different area codes in them. So it meant that one post was generating possibly from United Kingdom and India which sounds strange. Therefore since we wanted to show the sentiment barometer of each country, we take the same post twice because of two different country locations.

## 5.2 Algorithmic Conception

The algorithmic concept had been conceptualized in a similar way we looked at our problem in totality. It involved definite step wise involvement of our work that is mention in brief below:

1. Extraction of Email Addresses
2. Extraction of dates associated with each post
3. Checking of area codes of telephone numbers.
4. Removal of redundancies with respect to Authors
5. Generation of complete list with with Author, Date, Location
6. Generating Sentiment values from the *Bag of Words*
7. Associating sentiment value for each author for each post
8. Grouping data as per European Council Dates

The reason we followed the above order was to get data that we required in every step. Some steps were merged in the end when we were finally generating the data as per the European Council meeting dates.But if look into the deeper perspective of the enumerated points above,

we can definitely say that the objective for which we were assigned this job, could be explained in those steps.

Our approach was to map every author to their corresponding posts while extracting their location and email address and other information. To do this, first using a *regular expression*, the email addresses in each post were extracted. After that, we used another regular expression to check the area codes of the phone numbers we stumbled upon in every posts. This returned us an unsatisfactory output for we landed with two issues. One was the unavailability of the numbers, there by making the location finder some what weak and the other one was the presence of multiple phone numbers for each post. This gave us the opinion that there could be a writer that was writing the post from one location and the publishing head was in another location and thus two different country codes associated with that post.

Now coming to the most important subsection: Sentiments. The given bag of words (BOW) were taken into account to generate the sentiment values for each post in the main document. Every word has some value that was given by everyone in the class and the average of those words were taken to calculate the sentiment of the posts. There is an issue with this approach which we figured out when the implementation for sentiment calculation for each post was done. Many of posts returned the sentiment value of 0. The reason was that those words in the BOWs were not present in the post content. It is good to mention here that if the algorithm for sentiment analysis that has been implemented by the group who are doing sentiment analysis on full text is used to calculate the sentiment values for every post, that could definitely lead to better results. Since we were concerned with the sentiment of the Author and Author in particular, the bag of words approach was taken into consideration. Once that was done, we generated a complete list of authors, their sentiments, dates of the post and location and *dumped* the data on a csv file in JSON format.

Now the complete file with all the data parameters required was processed in accordance with the European Council Meetings but with the slight modification to check how the sentiments varied two weeks prior to the beginning of the session and three days post the conclusion of it. Once the specified dates were mentioned in the script, it generated 27 different files having the list of the fields that we needed for generation of the graphs:

- Author
- Date
- Location
- Sentiment

The format in which the authors were stored was dependent to this particular text, since authors' emails were included inside the body of the posts. However the algorithm can be easily adapted to different kinds of text by making few changes. The algorithm is robust enough to extract the above mentioned fields from similar documents of the same domain or different domain.

With regards to the processing of the document pre or post: we converted the .csv file into a custom format in which posts were distinguishable by our custom markers. Also the post-processing step is converting all the data in a JSON format that holds all the mappings of authors, locations, sentiment and posts together.

An assumption that we had earlier mentioned was the Author itself and that we narrowed it down by the emails. Now when we narrow it down to emails, there are posts with multiple emails. We do not know if both of them are authors in terms of contribution to the article or who is the author or who is the editor for that article. Another assumption we made was that the location that we got for the posts from the phone numbers were the locations of the authors. It could be that that the author is writing from different location and has used the phone number of his office. This got a little complicated when we figured out different country codes in the phone numbers associated with the posts. We could assume that one number was from the Thompson Reuters head office and the other was the local number, as the United Kingdom country code where the head office of the Thompson Reuters is located appears in the list a lot.

## 5.3 Implementation Details

The Algorithmic Concepts explains in details the process that was undertaken to reach the results. In this subsection we will talk about the implementation of the algorithm and different software that were used to take care of the data, pre-process it, work on it and finally process it for output. Given below are the sub-subsections in which we will look into the aspects of our implementation for the Sentiment Analysis.

### 5.3.1 Programming Languages

We have used the Programming languages *Python* and *R* for this project.

### 5.3.2 Source Code implementation

The built-in CSV library in python does not support utf8. Initially we had used C# to pre-format the document but after that we used the regular python to parse it manually. The whole approach was to *JSONize* the file so that the structure is easy to work on or can be used in other programming languages with ease. The pre-formatting of *ireland.csv* was also done to remove newline(\n) and double slashes (\\)

The initial work before we got into the extraction and other requirements was to have a json file for words with their average sentiment values (file: words.json) and another json file with the countries and country codes associated with it (file: CountryCodes.json). Once that was one, with the help of a simple email *regex*, we extracted the emails from the posts and at the same time converted in lower case if any was having mixed cases. A ranking function then ranks the posts based on the sentiment values that were associated with the BOWs in *words.json*.

For every line in the csv file ireland.csv, the *STORY_TAKE_OVERWRITE* was taken into account and from there the dates matching to the dates of European Council Meetings were placed. Extraction of emails in raw, associating them to post and also keeping a unique list of them was then implemented. The posts were then linked to the email dictionary that was created and consequently the country names were also linked.

Once the final file *makeJsonOut.json* was generated, we observed something that was we had missed before and that was: same author having multiple emails. For example there could be an author having emails as john.smith@xyz.org, john.smith@xyz.co.uk, john.smith@alpha.com. So we now picked up the names from the email list before the '@' sign and matched them against

others to detect the same authors and merged the contents of all the different emails of the same author together.

Once the json file with the desired entries were created, we used that file to generate separate tables in csv format for excel for each of the European council dates by writing another script *authorTimeSentiment.csv* where the name clearly suggests the fields it has. Another file *codeTimeSentiment.csv* was created for the location or area codes that were had in our json database.

Once the data was gathered after running the script, we encountered a problem with the format of the date and the author name being text in the fields that were used for plotting with *R*. So we had to index the authors and the dates. The dates were counted as the number of days in that time span of the European Council meetings.

### 5.3.3 Libraries Used

Given below are the set of libraries that were used in Python and R for this project.
**Python**:

- json: For JSONizing the outputs for flexible useability
- re: For regular expressions
- codecs: For decoding the utf-8 format

**R**:

- Regular R libraries

## 5.4 Results and Result Discussion

There are 27 graphs which show us the variation of the sentiments over the course of the specific European Council Meetings. It is difficult to include 27 of them here based on the Author Sentiment Analysis and another 27 based on Location Sentiment Analysis. How ever, a few distict graphs are shown here in the following to subsections followed by a comparative layout of Author vs. Location.

1. Author Sentiment Analysis
2. Location Sentiment Analysis
3. Comparative layout

The results clearly show us how the sentiments are varied over a period of time, sometimes shifting from neutral to positive and sometimes just staying constant through out the entire period of the European Council Meetings. Sometimes we have observed some countries being involved in writing posts extensively while some show scarce points on the whole plotting graph.

We have used variations of three colours that mark the sentiments expressed in the whole span of the Irish Crisis.

- Red - Negative Sentiments

- Yellow - Neutral Sentiments
- Green - Positive Sentiments

The colours in between are indications of sentiments that tend from one colour to another. For example, if its *orange*, then its an indication of the sentiment value between negative and neutral. If its *light green*, then its an indication of the sentiment value between positive and neutral.

### 5.4.1 Author Sentiment Analysis

If we clearly look into the plots generated during the mentioned time span, we see a decent layout of sentiments. In Figure 27, there is the same plot as in Figure 26: from the graph we can see that on the 10th day of this council meeting, the authors who contributed to the posts had mostly negative sentiments. Hardly anyone contributes on 20th day to plot the sentiments of the documents (based on the sentiment calculated on the bag of words).

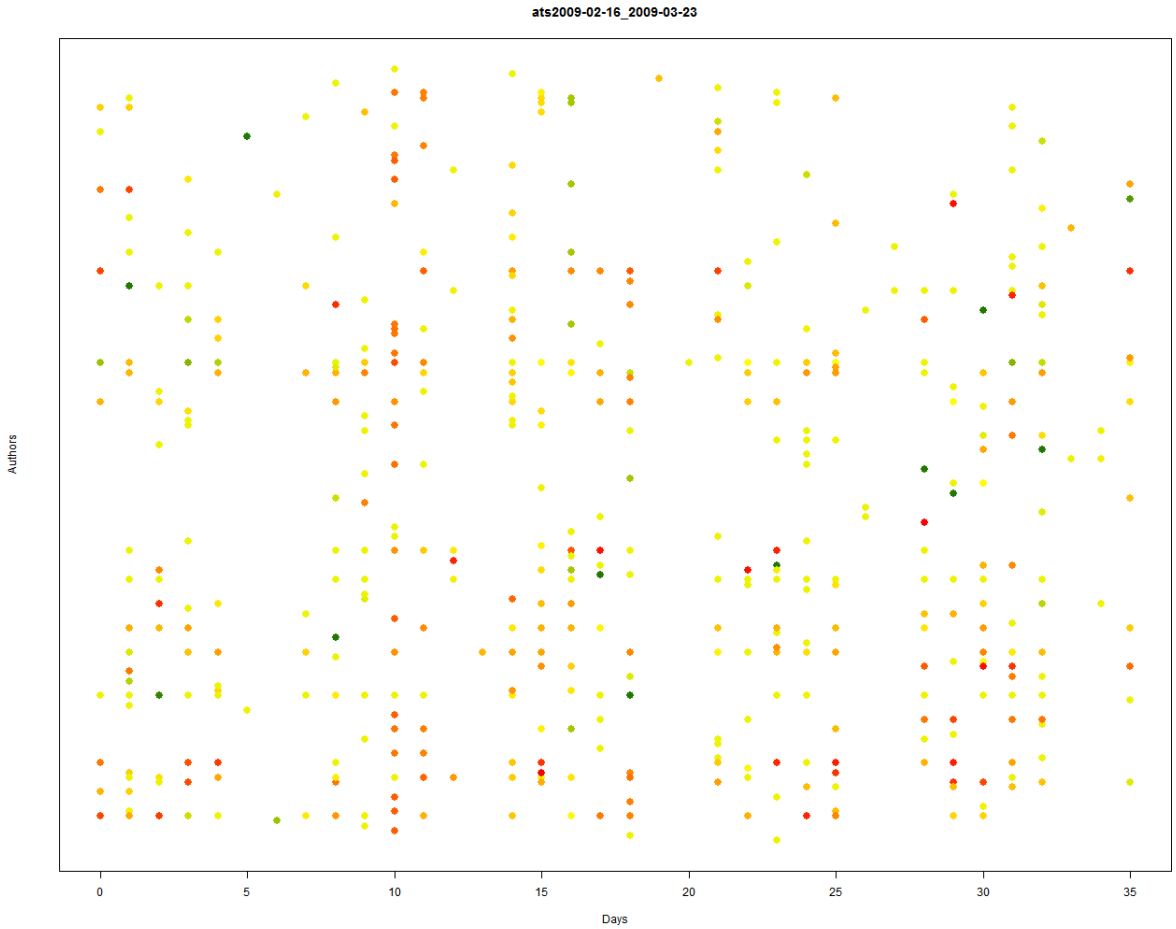



Figure 26: EU Council Date 2009-02-16 to 2009-03-23

Let us look at Figure 28, where in we will see the problem of being densely populated. The sentiment plots get overlapped at multiple places. This time span of the European Council

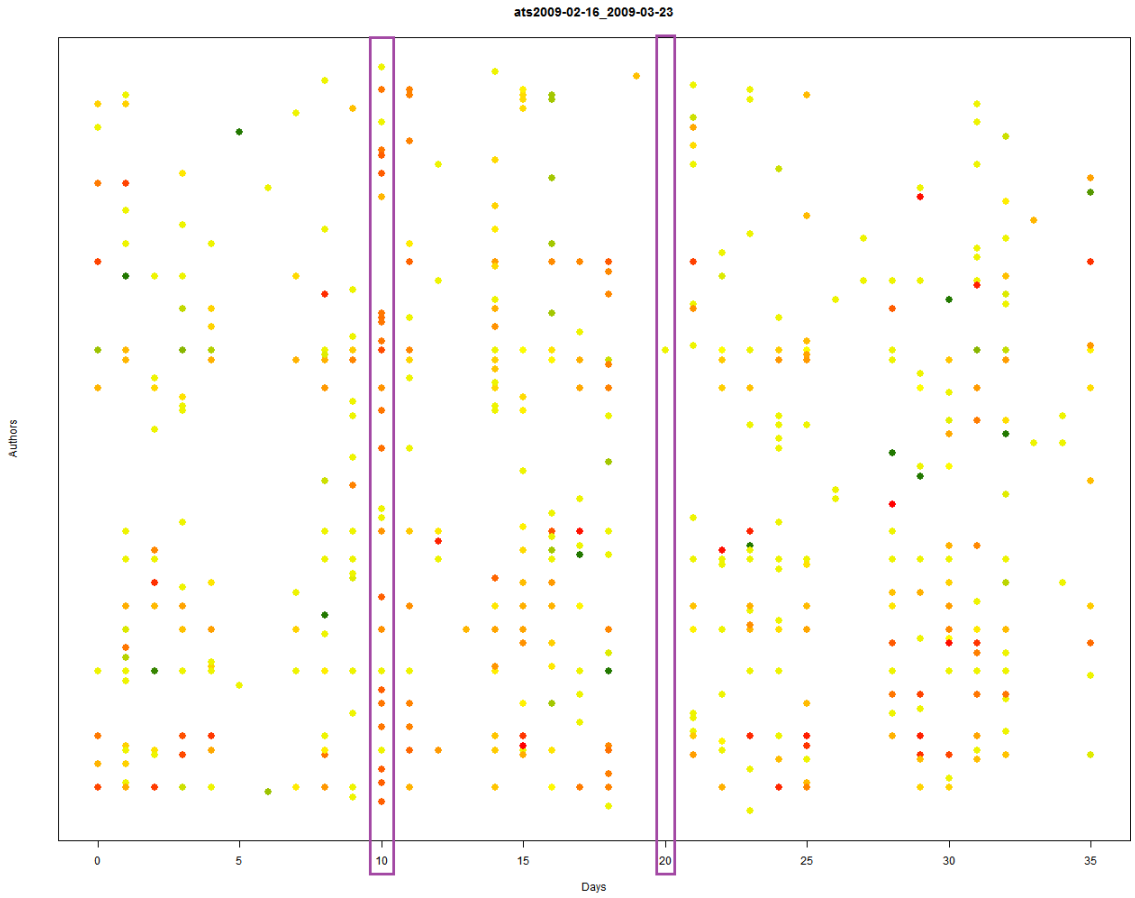



Figure 27: EU Council Date 2009-02-16 to 2009-03-23

Meeting is the longest.

From this Figure 28, we can clearly see the in the last line we can see that the author contributing maintains a somewhat neutral sentiment over the entire course of time of this EU council Meeting. The line marked above shows that the author expresses a kind of positive but varied range of emotions in his articles. From green, it shifts to yellow with an ocassional array of red dots and then yellow again.

### 5.4.2 Location Sentiment Analysis

In this domain of location based sentiment analysis, we will come across a lot of empty spaces in the plot which is the indication that details of the location were not present in the main document.
Its quite evident that the plots in Figure 29 are sparse in comparison to the Author Time Sentiment graphs. However, one can clearly see the variation of the sentiments pertaining to a particular country. From green, to yellow, to orrange & sometimes red.


Figure 28: EU Council Date 2012-06-16 to 2013-03-17

### 5.4.3 Comparative Analysis

A few of the comparative analyis pictures have been attached in this segment, which shows us how the pattern of plotting varies when we take authors into consideration and when we go in for mapping based on locations.
From the two Figures 30 and 31, which we have here of two different time slots of the European Council Meetings, we can clearly see how one can analyze the sentiments of the authors and locations, with the location based plot being less dense than the author one. From Figure 32 we see how the most dense plot is visualized.

The two Figures 33 and 34 give us a complete vew of the sentiments with respect to Authors and Locations in the total time span from 2008 to 2013.

## 5.5 Conclusions and Future Work

Sentiment analysis of posts is basically sentiment analysis of the author of that post or the reactions of the people the author is writing about. The idea for this project was to show the sentiments expressed by the various authors contributing their article during the Irish crisis. From the results that we have obtained by focusing our search on specific time periods associated

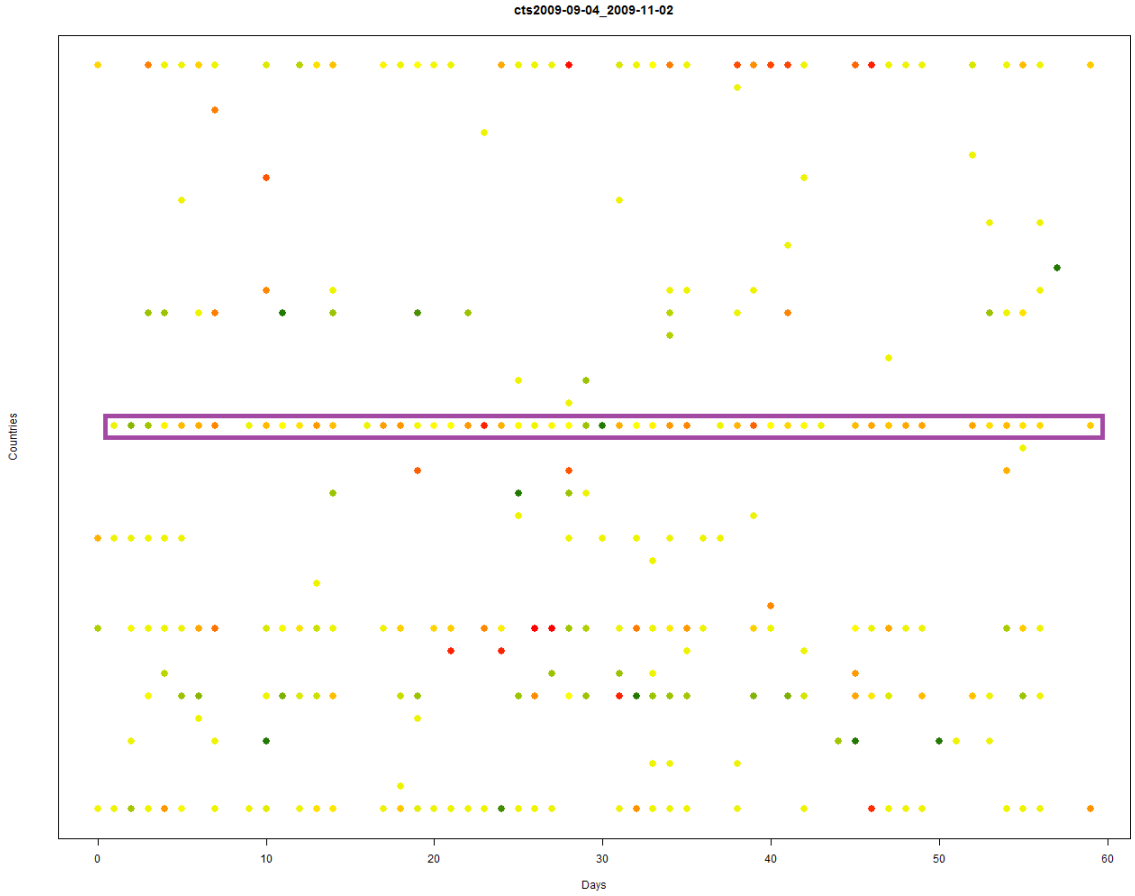



Figure 29: EU Council Date 2009-09-04 to 2009-11-02

with the European council meetings, we can clearly see from the graphs that we have plotted the variation of sentiments of authors as per the posts and time in which they were published. So, if we make a evaluation of the points we have achieved so far, it will be:

- Sentiments of every author
- Location associated with the posts via country codes in the phone numbers
- Time frames associated with European Council Meetings

### 5.5.1 Limitations and Improvements

The basic sentiment evaluation can be made more robust like the other group who has undertaken the sentiments for the full text. Our approach has been very basic considering the Bag of Words only. A definite improvement can be made on the Sentiment evaluation aspect with advanced algorithms. The advantage of our system is that it is very robust in terms of dealing with the changes. The moment the sentiment values are updated, the remaining files can work smoothly and will give us a better output without changing any other parameters whatsoever. Every script associated with this project has been made in such a way that the final product is easily reproducible.

When talking about the Locations that we have found out, it is very important to mention again that there may be some of posts that do not have telephone numbers associated with them, and nothing can be analyzed from those posts about what could have been the locations.

Initially, the graphs were supposed to be 3-D cubes with Author, Location, Time in the three axes and the sentiment value having the color. Due to our lack of expertise in generating such graphs, we had to resort to an alternative by transforming this three dimensional graphs in to two 2-D graphs. One having Author, Sentiment and Time while the other looking into Location, Sentiment and Time. Improvements on the graphical output by some one having better knowledge of how various graphs an be generated would lead to better visual representation for the readers. As far as the raw data is concerned, the reason it was put into csv format was for everyone to have better access and do further work on it as per the requirement.

Improvement can be made in terms of plotting of the data if some other alternative is devised for marking the x-axis and the y-axis with the dates and the author or location names respectively. The problem is that there are too many points to plot and it will not be visually soothing to see so much of text on the graph.

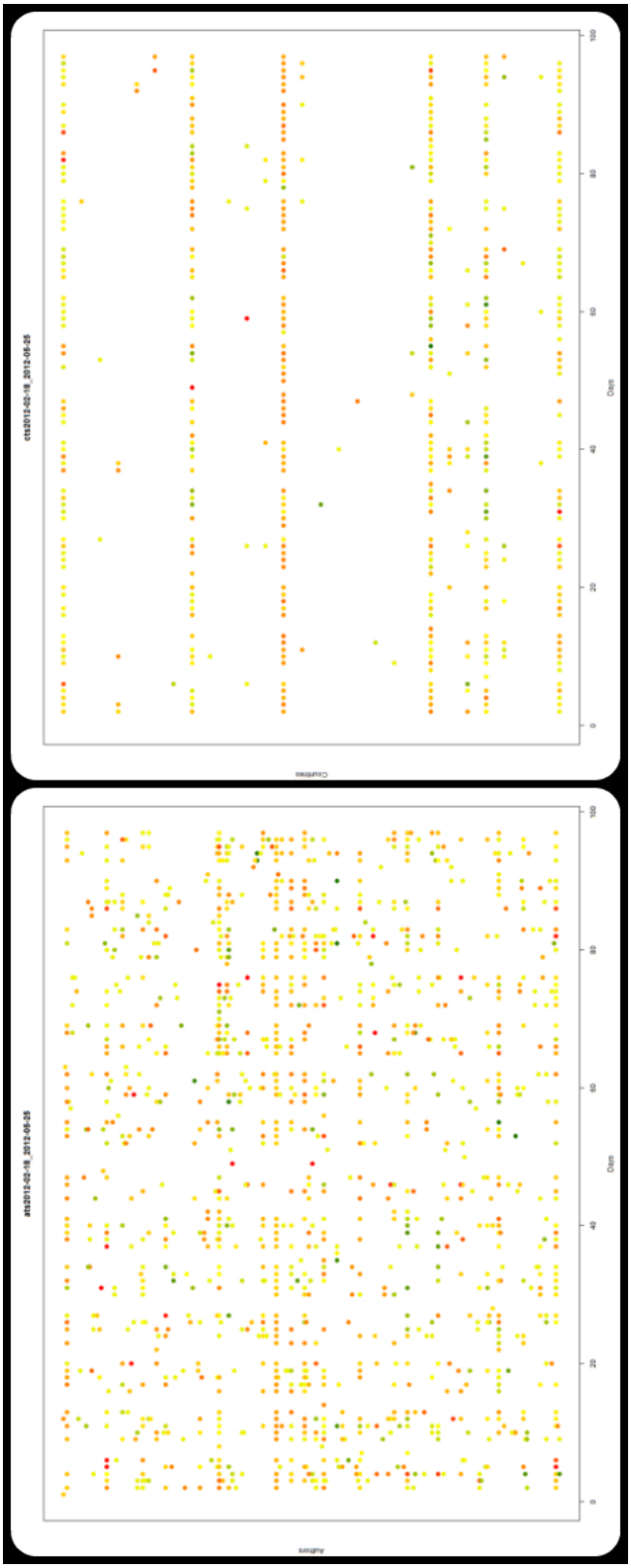


Figure 30: EU Council Date 2012-02-18 to 2012-05-25
30

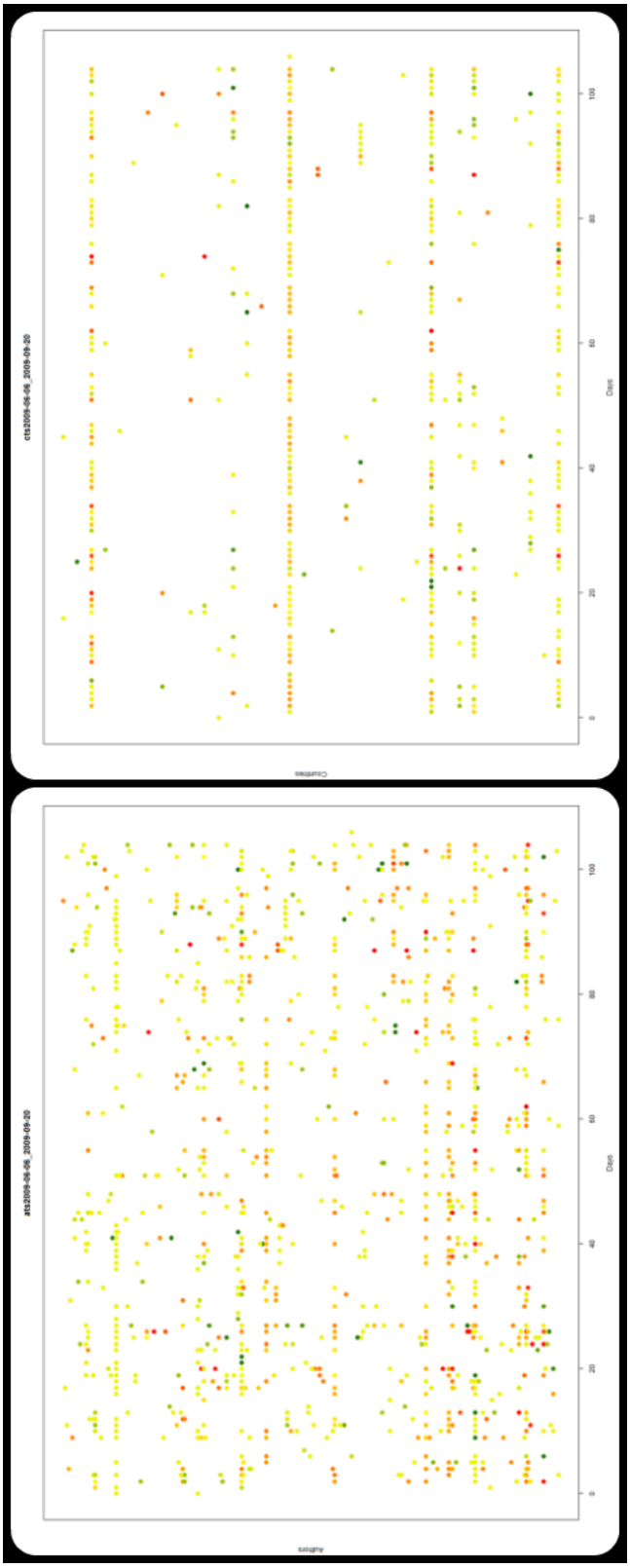


Figure 31: EU Council Date 2009-06-06 to 2009-09-20
31

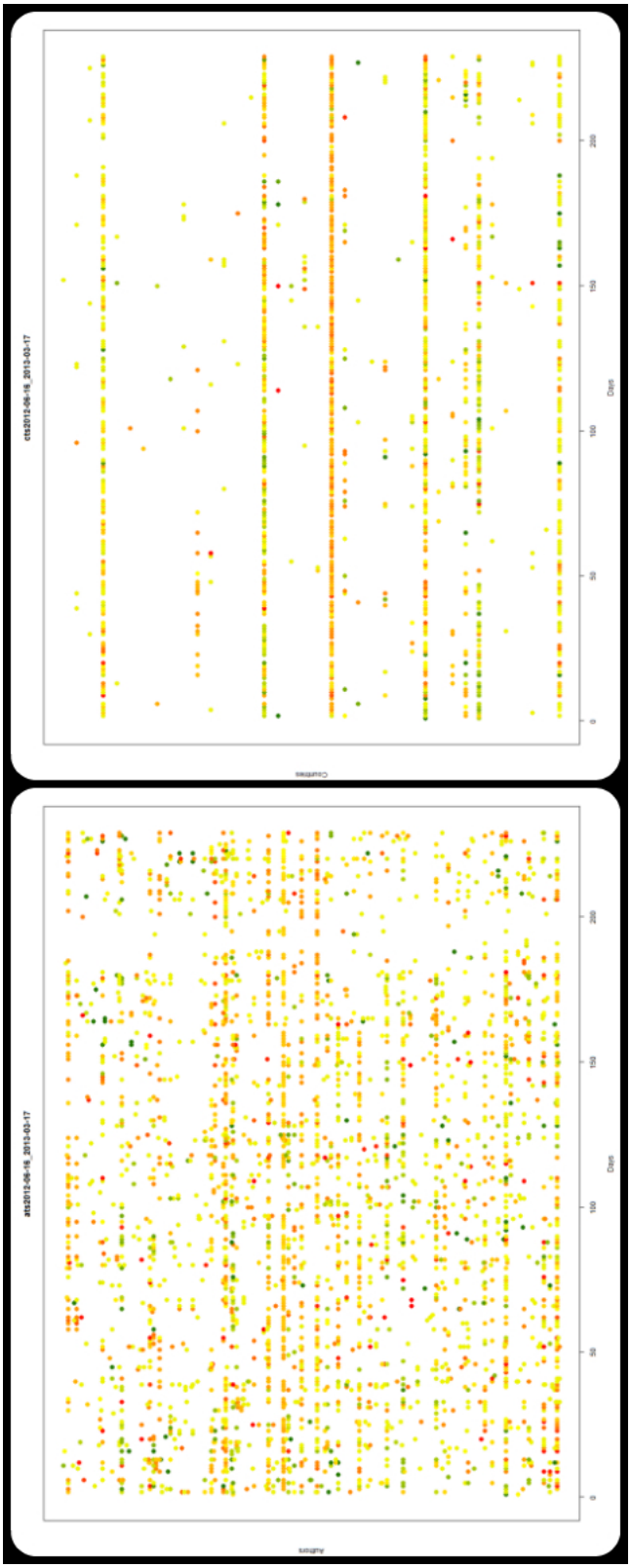


Figure 32: EU Council Date 2012-06-16 to 2013-03-17
32

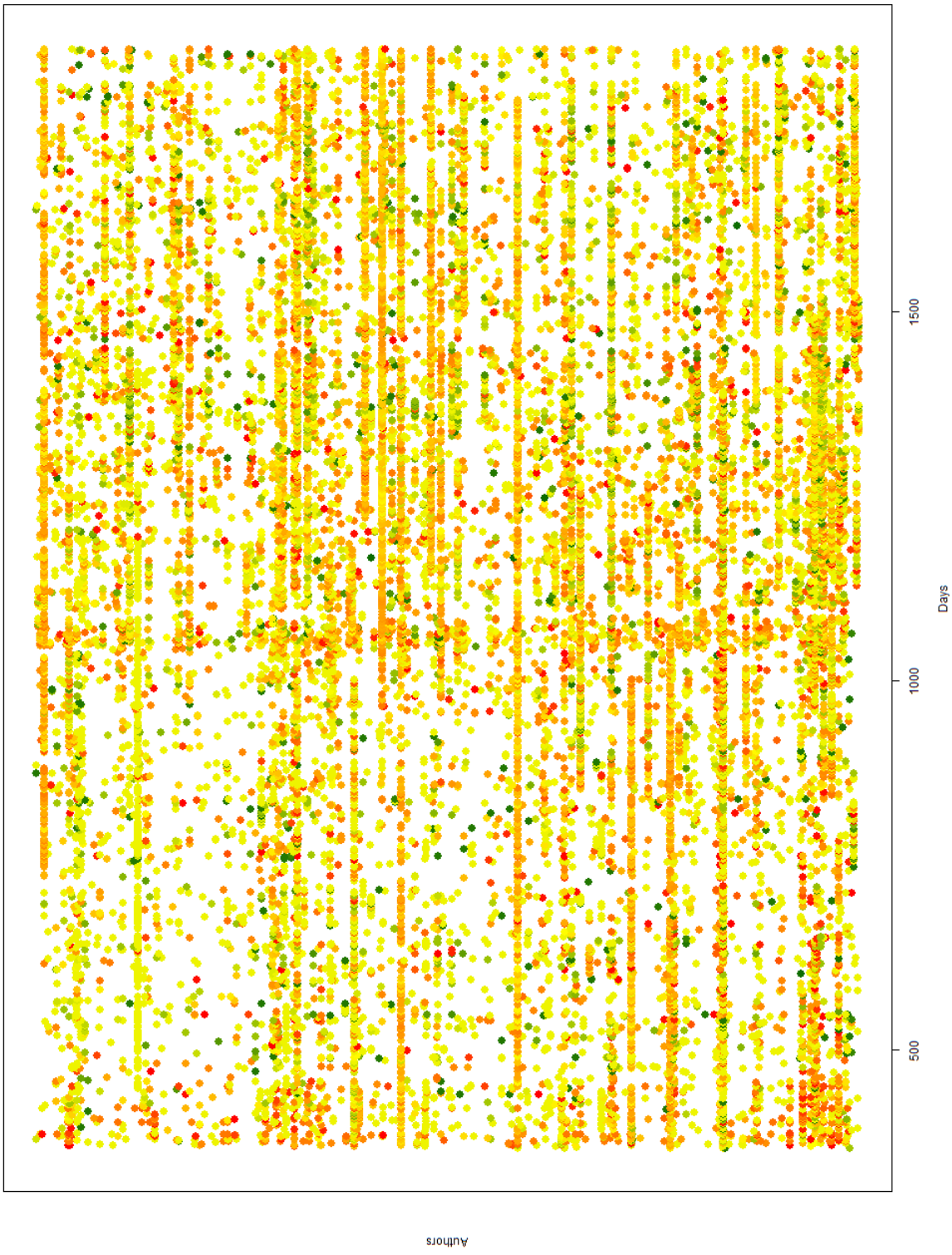



Figure 33: Complete Author Time Sentiment Plot
33


Figure 34: Complete Location Time Sentiment Plot
34

# 6 Sentiment Barometer and Topic codes

Fabrice Thill and Alain Pfeiffer

We present a analysis of topic codes over the time period of the European summits during the Irish financial crisis (2009-2013). We show how a sentiment can be attached to a topic code over a certain time period based on the the sentiment values of the texts they appear in. This gives us information on the rating of topic codes and how they were influenced by the European summits. With this analysis it is possible to detect certain trends which opens possibilities for further investigations of their cause. Further research could link this trends to specific events which would be particularly interesting for market research, for instance to see if the sentiment barometer of topic codes can be linked to stock share values.

**Keywords:** Sentiment analysis, Topic codes, Trend analysis.

## 6.1 Problem Description

The goal of our work was to attach a sentiment value to different Topic codes. Our work is based on the financial news articles from *Reuters* that were provided to us which follows a rigid structure. In particular at the end of most texts a number of topic codes are included that give an overview of the main topics in the text. The idea is that if a topic code appears mostly in texts that have a positive polarity, we could give this topic code a sentiment value that is positive. The sentiment value is then calculated for the topic code for different time periods which makes it possible to analyze how the sentiment values changes and maybe investigate what could have caused these changes.

Since this work is based around the European summits and the Irish crisis, the problem we want to solve is as follows: How have the European summits influenced the narrative of financial news-writer in regards to the Irish crisis. An example which we will show later in our work is the topic code *Europe*. The sentiment value of *Europe* can arguably be seen as a rating of the financial news-writer of events in Europe to have a positive impact or negative impact on the financial market. Since the analysis is done over time periods, it is possible to see which events had a positive effect and which had a negative effect.

We move on now to describe which algorithms were used to solve the problem followed by more specific implementation. A subsection is then dedicated to to our results and finally we discuss our work and possible future improvements.

## 6.2 Algorithmic Conception

The following chapter describes the process of sentiment classification concerning the keywords of texts. It contains the approach of the project, the pre- and post- processing steps and the cleaning process.

### 6.2.1 Understanding the Data

One of the first steps is to properly understand our input data. The financial news articles from *Reuters* all share a structure, more specifically they all have a date, time, text, and most importantly for us a keyword field. Our approach was as general as possible and therefore,

| Keyword | Occurrences |
|---|---|
| GUIDANCE | 4 |
| HAIRCUT | 5 |
| PRIMARY | 2 |
| FRANCE | 58 |
| STABILITY | 7 |
| MEMORANDUM | 1 |
| HEALTHCARE | 1 |
| RIM | 2 |
| PROPERTY | 23 |
| ... | ... |

Table 12: Keyword structure

except for the automated extraction of the different fields which could probably be achieved for any text by minimal changes, our approach is independent from the texts.

### 6.2.2 Extracting the Keywords

We perform a number of pre-processing steps on the provided dataset. For the topic codes, we first performed an extraction which is fairly simple since all the topic codes are preceded by *Keywords:*. After the extraction of the keywords, a cleaning and noise removal is performed to remove all the symbols and other non-keyword occurrences like dates that were placed in the wrong line by mistake. We also performed a manual cleaning step in order to correct spelling mistakes, for instance the at the time luxembourgish prime minister 'Juncker' could be found 164 times in the texts with the correct spelling and twice as 'Junker'. If we had to work with even bigger dataset containing lots of keywords one could try calculating Levenshtein distance of keywords with a high number of occurences to keywords with a low number of occurences in order match mistakes to their correct spelling.

The extracted keyword(s) are then saved (in case of a new unique keyword which has not yet been saved before). After the extraction phase, the numbers of unique topic codes are counted and their numbers of occurrences through all the texts they are included are added to the specific keyword saved. At this point, the list of keywords is completed and has the structure that can been seen in Table 12

From the discovered keyword values, we decided to discard all keywords that did not pass the threshold of at least 20 occurrences. The reason for this decision is that we want to perform an analysis over time periods and less than 20 occurrences would simply not be enough to have a dataset significant enough for analysis. In fact it would yield to an average of less than one text per European summit. Aside from the threshold we set, it is interesting to note that the vast majority of keywords appear only a few times 35, which seems to indicate that the authors follow no fixed rule when assigning the keywords. It seems that the intended purpose of the keywords is more to offer the reader a very quick overview of the main topics and not to provide a classification for the article. Having an expert assign topic codes to each text with a clearer classification might lead to better results in the end because more keywords would resist the mentioned threshold of 20 occurrences. Another possiblity would be to have create a number of categories one would be intrested in and to then use a text classification method to assign one

or more categories to each document [4].The described fact that most keywords only appear a few times in global can also be retraced in the following graph of keywords found.

By means of the Figure 35 you can see that more then 2000 out of 3921 keywords found appear less than twice, about 1000 less than 10 times not including those with a single or two occurrences and about 250 appear between 10 and 20. Thus, only approximately 350 keywords can be used in the end for a proper sentiment analyze in our project.

We also performed a post-processing step on the keywords. For some topic codes that could be placed in a group like the topic codes *France* or *Germany* we, additionally to calculating their sentiment values independently also calculated their values a part of the group *European countries*. The same could be done for other groups like banks or politicians and it makes it possible to see if an increasing sentiment values is for instance specific for a country of if the sentiment value had an positive trend for the entirety of Europe.

### 6.2.3 Classification using tf.idf

Additionally to the financial news article we were provided a bag of words containing relevant financial terms. We assigned a polarity values for each word, either -1 0 or 1 if the words was negative neutral or positive according to our understanding. After having extracted the full text we then search for these words and calculate the inverse document frequency(idf) which is calculated with

$$log\frac{|D|}{t} \tag{5}$$

where $|D|$ is the total number of texts and t is the number of texts the words appears in. Term frequency (tf) is simply the number of occurrences of a word for a given document. For each word and each given document we can thus now calculate the tf*idf value. Usually the tf.idf values is used in order to calculate a document weight which give some information about the relevance of a document relative to a query [3]. In our case however a weight is calculated for every keyword and with the help of our initial values on the bag of words we calculate a sentiment for each document. This is done by multiplying the word sentiment value with the weigth it has in the specific document. The document sentiment is then the average of all calculated scores.

To calculate the sentiment for a topic code, we simply took the average document values for each document the topic code appears in.

## 6.3 Implementation Details

Our algorithms were written originally written in Java and then later in python because of the simplicity of python when it comes to manipulating text and strings. Python also offers a strong system of regular expressions as well as a number of libraries for natural language processing which ultimately makes it the better choice for this problem. All of our data processing was done using python. We also used HTML and javascript to have an interactive visualization of our results.

### 6.3.1 Data Structures

Our data structures consist mostly of formatted text files. These offer both simplicity and speed since regular expression make it easy to extract the different fields and values. They also provide

the advantage of making it easier to manually check results.

Our input data was a single csv file containing all the articles concerning the Irish Crisis from February 2009 to 2013 to January from *Reuters*. One of the first steps was to split the texts into separate .txt files and order them by date by putting each files into a folder corresponding to the correct time span between two european summits. We then extracted all the keywords and put them in a single file, in order to make them easily accessible. Different files were also used in order to see the changes between the cleaning steps and see the number of occurences for each keyword. All input text files were numerated and the according sentiment score was kept saved in a file. All of this made it possible to rapidly calculate any given topic code. The final program took as input the topic code, the ordered list of input files and the scores for each input files and then calculated an average sentiment value for each month, our final output being a text file with the topic code name and the respective sentiment values.

### 6.3.2 External Programming Libraries

The only external library we used is amCharts [1], a javascript/HTML library that offers a number of tools to create interactive charts.

### 6.3.3 Source code implementation

Our source code consists of a number of python files, each one often responsible only for a relatively simple task. The reason for this is that it makes it a lot easier to manually control the results if each python script only performs small tasks. We thus have a script bagofwords.py, which with the help of the bag of words that was provided, and a list of values assigned to each word, calculates the idf values for each word. The script calculatescores.py then calculates a score for each text file using the previously calculated idf values.

We used a number of scripts in order to extract the keywords and clean them. These scripts firstly extract the keywords, then remove any noise like single letters and dates that were placed in the keyword line by mistake. They then further remove unnecessary spaces and symbols and calculate the frequency distribution. Our sort_by_date.py script is then responsible to parse each text file for a date and place the text in the according folder.

Finally calkeywordscores.py and a variation thereof provide us our final results. The first of this script takes a single topic code as input and calculates the sentiment values for each time period by parsing the individual text files that have been sorted by date. If it can find the topic code in the keyword line it looks up the document sentiment value that was calculated previously. The second script outputs a csv file formated as follows : *Date, documentid, documentvalue* This file is then fed into our amchart visualization.

The interactive visualization we used is simply a html file with a javascript script that uses the amChart library [1]. It takes an input file in the format just described above and can be loaded by any Internet browser. The tool enables the user to define the selection of data he wishes to see as well as pan over the entire dataset.

## 6.4 Results and Result Discussion

This sub-chapter describes the results we obtained through the algorithm we described in the previous sub-chapter. Thus the remaining keywords were used to calculate a positive or negative sentiment value for each time period. The following graphs have as their x axis the dates of the

european summits and as their y axis the average sentiment values for a given keyword between each european summit.

All of the following interpretations are made by students with little economic knowledge. They exist to showcase a knowledge extraction from the original input data.

### 6.4.1 Results for country related topic codes

Since our input files were financial news articles concerning the Irish crisis it is only natural that we start with the *Ireland* topic code.

Once can see in Figure 35 that the sentiment values a still quite positive at the start of 2009, to then become more and more negative as it becomes apparent that Ireland will need financial support, which was finally formally requested in november 2010 [2]. The financial help then seems to have had a positive impact, at least for the following months. Towards the end of 2011 a negative trend re-appears, which can arguably be attributed to a number of protest that were led, partially by students to combat the austerity measures.
For the topic code Portugal(Figure 35) we can see that for most of 2009 and 2010 most of the sentiment values were 0 or around 0.

The reason for this being that the crisis only really reached Portugal at the end of the year 2010 and up to that point Portugal was not being actively discussed in the financial news. At the end of 2010 we can see a downwards trend in sentiment values which can be attributed to country reaching a new high of unemployment (11% at the end of the year) and an austerity package announced in september 2010.

We also calculated an average for the overarching topic code Eu-coutries (Figure 35) which is an average sentiment values calculated from the sentiment values of the topic codes France, Ireland, Germany, Europe, Portugal and Spain.

Comparing Ireland and Europe one sees that they share some similarities. In particular both start out with a positive sentiment values to have a more negative trend towards end 2010 and for most of 2012.

### 6.4.2 Results for politicians

Most political figures did not have enough occurrences in the text to pass our threshold of 20 occurrences, the exception being well known-state leaders as for instance the German chancellor Angela Merkel or the American president Barack Obama. Here we will present our results for Jean-Claude Juncker who is particularly relevant in our context since he was president of the eurogroup, a meeting of finance ministers of all countries that have the Euro as a common currency.

Interestingly, the topic code *Juncker* seems to have a rather limited impact on the sentiment values, with all the values being around 0. A possible explanation for this, other then some values being 0 for the lack of documents with topic code *Juncker* is that the decisions taken by the eurogroup do not necessarily have the topic code *Juncker* in the Keywords. If someone wished to measure the impact of political figures in the crisis, one would first have to activly classify the documents where political figures took financial decisions as such.

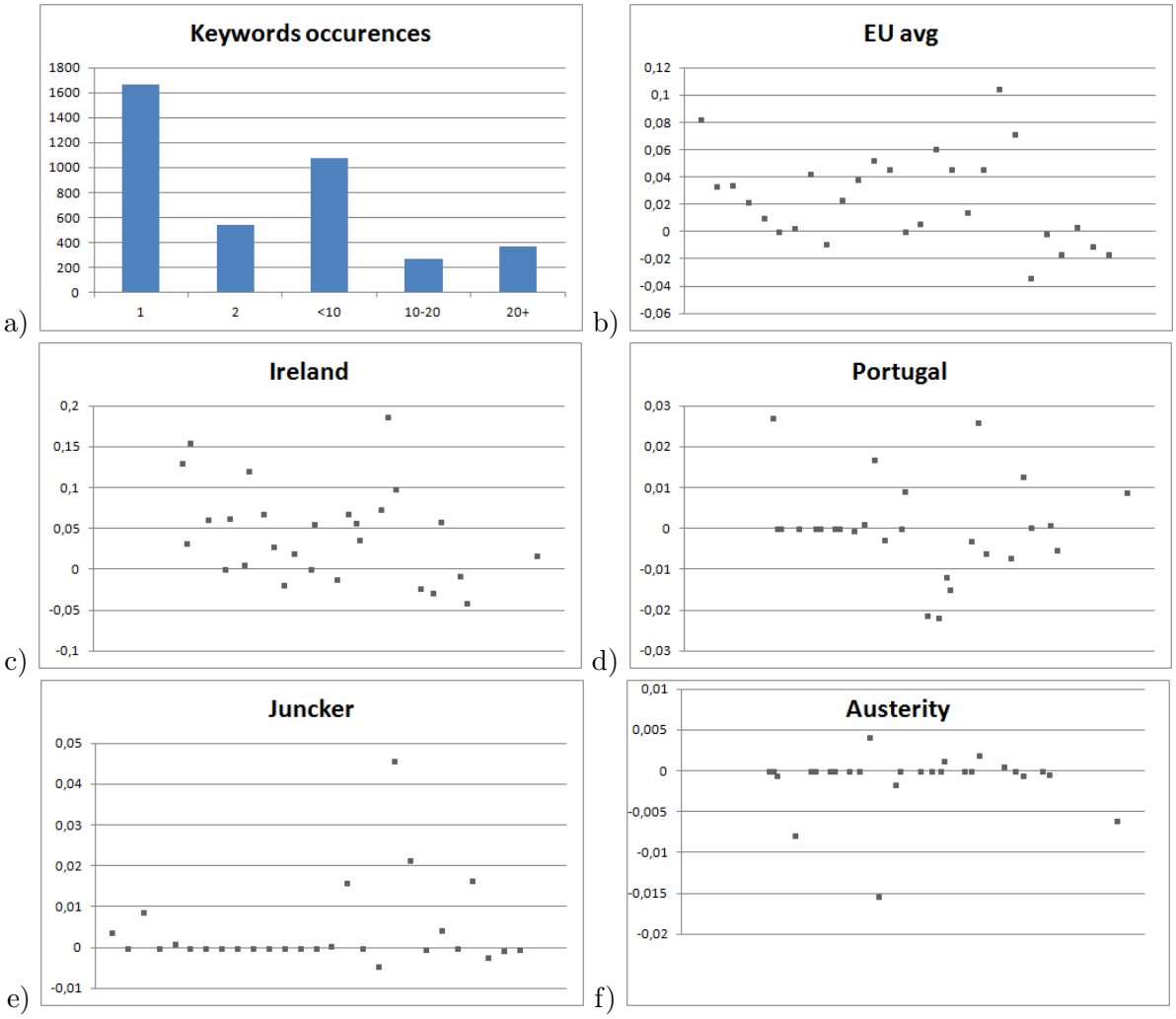



Figure 35: Sentiment analysis: a) topic code occurrences and average values for the topic codes b) *EU conuntries*, c) *Ireland*, and d) *Portugal* across the 2009-2013 period. e) and f) show the average sentiments for *Juncker* and the topic code *Austerity* for the same period.

### 6.4.3 Other Keywords

There are other keywords that might be of interest to an analyst. One example are banks which in the public opinion are often seen as the cause of the crisis. A bank or a company could use these sentiment values provided by our algorithm and try to match different events to a drop or raise in sentiment value. With this knowledge a company could then learn what to do and avoid in order to have a positive sentiment in the eyes of the financial news writer or in the public opinion. Another keyword that was used a lot during the financial news crisis was the Keyword: *Austerity* (see Figure 35).

A lot of european countries, like Greece or Portugal, had to take austerity measures during the crisis. Since austerity usually has a rather negative co-notation it is not surprising that the sentiment value for austerity is mostly negative. However one does also see that most of the sentiment values are around 0. That is due to the fact that there were not a lot of text containing

the keywords austerity, and thus the score for a month without any documents is simply 0. It is therefore arguably that a threshold of 20 was not high enough, and that a higher threshold might have been more appropriate.

### 6.4.4 Interactive visualization using amCharts

All of the previous graphs gave a general overview of trends by giving sentiment values for each time period between the european summits. However we also provided a method that makes a more interactive visualization possible where the user can determine which time period he wishes to look at.

The Figure 36 is a visualization of all the texts that contain the *Ireland* topic code. The x axis is the date an the y axis shows the sentiment value of the specific document. Hovering over a document shows the document id, in this particular case the document with id 42753. This allows the user to immediately consult the text in question should he wish to.

Should for instance on a given day, all articles be perticularly positive or negative, one would simply need to look at a few documents to determine wether these unusual sentiment values come from a certain topic.

## 6.5 Conclusions and Future Work

The goal of our work was to see if sentiment analisis on topic codes would give any relevant information. These keywords (as described in previous sub-chapters) were extracted, filtered and then saved for later analyses. The remaining keywords, which have passed our threshold of 20 occurences, were given a sentiment value for each time period. From these diagrams one is able to make an opinion of the relevance of the articles in global for a certain time interval by only referencing to the keywords in each article. Unfortunately the amount of relevant keywords was unpredictable low and there for the computed results for certain regions were not as good as expected.

A good example is the graph for the topic code *Austerity* in the previous sub-chapter; most of the computed values are not meaningful because the amount of relevant texts was too small to get better results.

One could have set a higher threshold in order to only have a final list of keywords with a very high number of occurences but then we would have optained a very low number of keywords.

However both these issues could be solved. The first solution would be to have experts in the financial field activly set keywords for each document as a mean of classification and not as it is currently as an overview of the document. The overall number of keywords would then be lower and topic codes would generally have a higher number of occurences.

Another solution would be to use machine learning in order to classify the texts into diffrent predetermined groups to then perform sentiment analysis on these groups. This approach could be tried in a future work.

# References

[1] amCharts *JavaScript/HTML charting library. www.amcharts.com*

[2] 2008 to 2013 Irish financial crisis. See *http://en.wikipedia.org/*

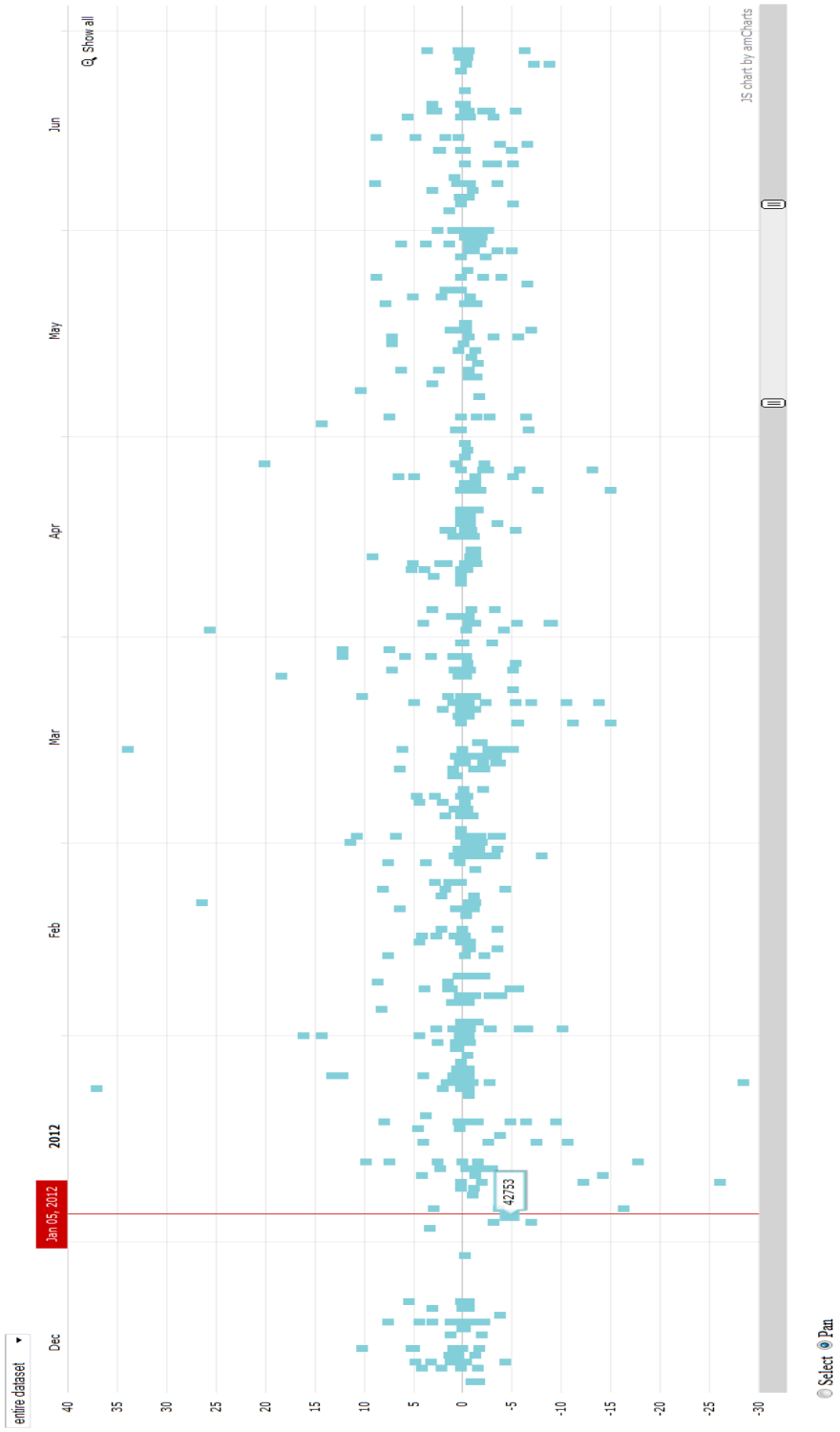



Figure 36: An example of our results using Amcharts

[3] G. Paltoglou, M. Thelwall *A study of Information Retrieval weighting schemes for sentiment analysis. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. 2010.

[4] Sebastiani, Fabrizio. *Machine learning in automated text categorization. ACM computing surveys* 2002 1-47

# 7 Sentiment and Information Hiding

Thierno Diallo

In our days the need of exchanging information safely and secretely is getting bigger. In the same time, the power of the machines explodes. this situation tends to increase the vulnerability of the basic ways of cryptography.

With the growth of internet, electronic documents and the fact that you can find the documents anywhere, so it gets less suspicious to see a document in a given place. The idea of hiding information inside the electronic documents, and to share it with other one, is came to day.

To hide the information, we need predefine a key for the two sides. to define this key, we choose to use the sentiment of the document, and to take only the extrems in positiveness or negativeness. according to the message that needs to be hidden, the sentiment of the document, we can define what is a good or bad document to hide something. To achieve our goals we are going to use the methods from Linguistics steganography.

Linguistic steganography is the way of using natural language to hide information. In order to communicate secretely between people who share the key, most of the time this process is achieved by making some change in the cover text. In this project we have used the synonyms substitution as a linguistic transformation. In order to reduce the information exchange we use the sentiment outilier. In this paper we first introduce the problem description, after we will show our model design in the algorithmic conception chapter. At third we will give some implementation details, the fourth part will be for the results and the discussion. we will end the paper by the conclusion and the future works.

**Keywods:** linguistic steganography, information hiding, sentiment outlier.

## 7.1 Context

We are in the context of Irish crisis. During many years the Irish government politics of low taxes and easy loan from banks, pushed them to invest a lot and to live out of their earning. These behaviors exposed them to indebtedness and decrease the confidence of the local and international investors.

We have to look into all the produced financial data during this period and to find those related to Irish economic crisis. After, we have to monitor the sentiment over the time period and see if there is a correlation between the increase of the confidence throuth the sentiment value and the economic recovery.

## 7.2 Problem Description

The main problem of our work is to establish a covert communication between many people by making a systematic transformation in a natural language text. The problem is that,

These transformations could keep the text innocent from the outside observer. Of course, this communication must satisfy some rules in order to remain innocent after modification. To make simple, the covert communication would fail if the of the existence of the hidden message in the document is getting suspicious.

As all covert communication, before any information exchange. we need to define the rules which on will be based the exchange. So, the to parts need to define some rules and to share its.

- **Key**: In our case the key is the sentiment of the covered document.
- **Dictionary of Substitution**: both of them must have the common defined dictionary for the substitution.
- **Interval for sentiment outlier**: they have to define together an interval in which the document sentiment will be contained.
- **Period of transmission**: they have to share before hand the period of transmission. for example: from january the 1st at noon to january the 1st at 1pm
- **Encryption/Decryption**: obviously, The way of encrypting and decrypting methods need to be shared.
- **BitString length and meaning**: normally, the bit vector is static and the two side must know the length and the meaning. But, In this project we choose to have a dynamic bit vector to avoid the problem of sharing the meaning.
- **Message length**: In the project the length of message (number of word) is static and two side have to share this number.

As for example, assume that *Alice* will come with real sentence which will be converted to a bit vector and encoded in the document. And the *Bob*, will decrypt the bit vector and meaning in the same time. We do not have to share the meaning. So, the length depends to the message. But, the length of the message remains constant and has to be shared by the two parts.

## 7.3 Main approach

For the linguistic steganography, we choose to use the synonyms substitution. Because, first of all, most of the time it keep the sense of the sentences the same. And after, many languages are profuse in synonyms, As showed in the literature, There is a rich source of information carriers compared to other methods of text transformation.
In the context of the Irish crisis, we have one trader (*Alice*) and a rating agency agent *Bob*. The agent want to inform the trader that the Irish state will be downgraded tomorrow. One of the most used algorithm is the synonyms substitution. So, this example will be based on this algorithm.

## 7.4 Algorithmic Conception

### 7.4.1 Sentiment Outlier

In order to avoid additional information transmission between the two parts, we decide to introduce the notion of sentiment outliers. The main idea is to define an interval [r,r+a] with "a" very small and r as number determined beforehand, and to take only the documents in this interval.

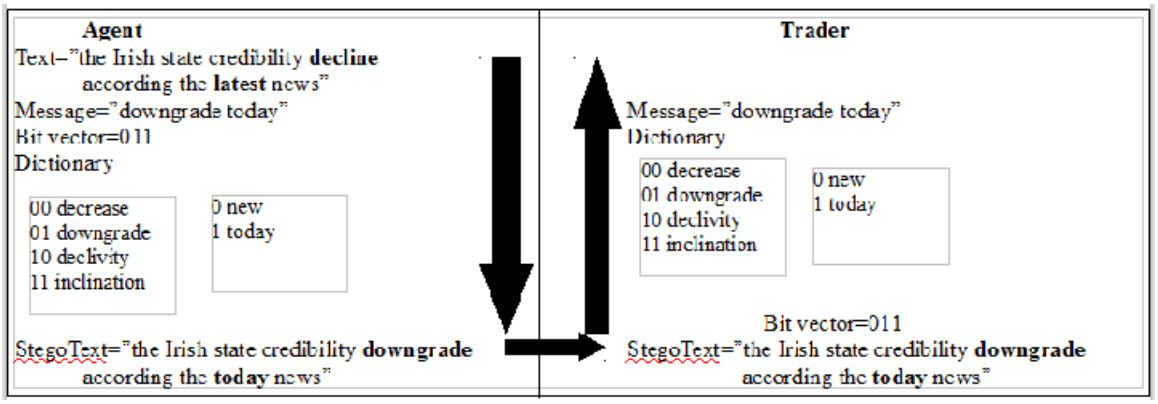



Figure 37: first example of covert communication

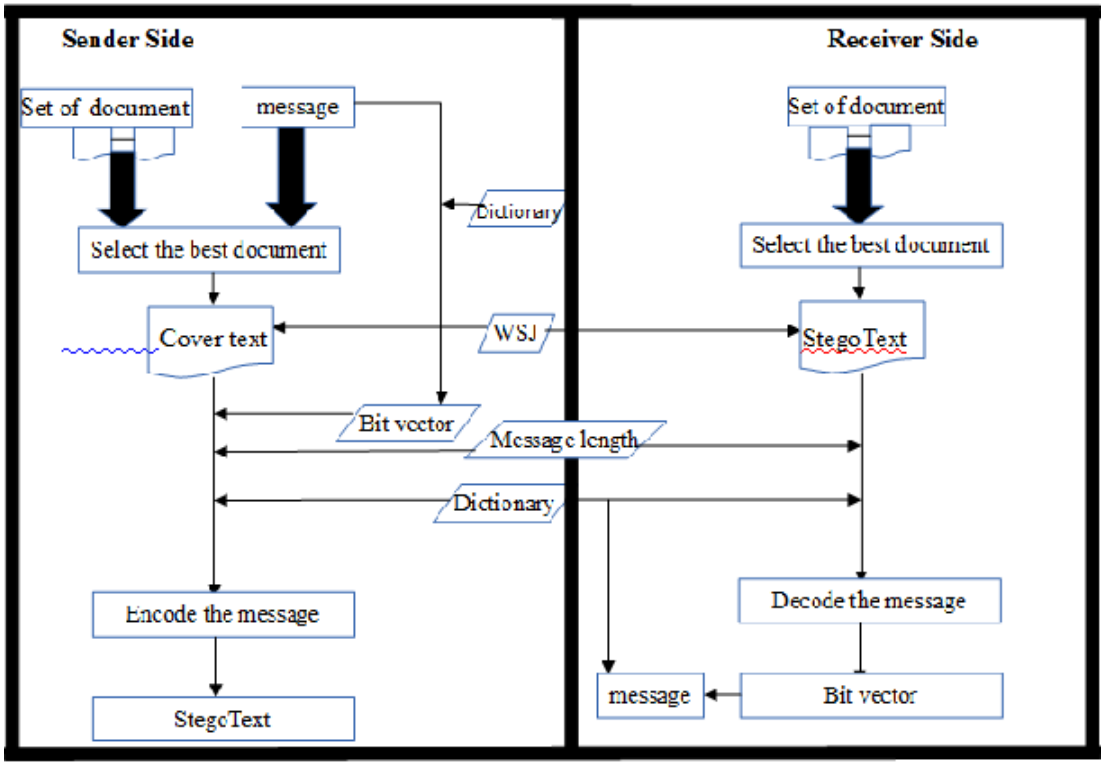



Figure 38: System design

### 7.4.2 The System Design

As you see above, we have two parts. The sender side and the receiver side. So, we will have two main algorithms:

- For the linguistic transformation, we choose the synonyms substitution algorithm. because many languages are profuse in synonyms, As showed int the literature, There is a rich source of information carriers compared to other methods of text transformation. The existing alternatives are text watermaking method, Syntactic Bank-based or the Text-to-Text Generation.
- The presented approach targets the financial domain. but, it is done in a way of easily extended just by adding the data to the knowledge base and without modifying anything in the main system.
- The pre-processing steps are clean the data and to select only the financial data in relation with ireland economics crisis.

### 7.4.3 Sender Side

```
SenderSide(DocumentSet,message):
        BestDocument: Financial Document
        DocumentSet: Set<Financial Document>
        message:String
        ClassifyDocument(DocumentSet) //from -1 to 1
         BestDocument=selectBestDocument(DocumentSet,message)//we take the
document with the sentiment absolute value between [r,r+a], and according to the message.
        bitVector=getBitVector(message)
        HideMessage(BestDocument,bitVector)//By using Synonyms substitution
end;
```

Figure 39: This picture show us summary of the algorithm for the sender side. for example, the sender first selects the best document according all the inputs, and hides the message!

### 7.4.4 Receiver Side

## 7.5 Implementation Details

- for this project we used the python language.
- as input, we took the 7th and 8th columns of the alerts csv file, we made the process of selecting the best sentences. as output we produce the stego modified text, the chart for the document sentiment classification and the embedding capacity.

```
ReceiverSide(inteval,periode):
        interval:int[2]
        periode:date[2]
        Document: Financial news Document
        Document=FindBestDocument(interval,periode)
        bitvector=decrypt(BestDocument)
        message=getmessage(bitvector)
end;
```

Figure 40: This picture show us summary of the algorithm for the receiver side. for example, the receiver first selects the best documents according all the inputs, for all the documents in the same time period and corresponding to the sentiment outlier, he tries to decrypt, if it is a success, he retrieves the bitvector and the message.!

### 7.5.1 How to run the code

- To test the document classification you can run test_classification.py module.
- To test the substituion you can run the stegolib.py module.
- To test the all system encoding and decoding. you have to run first the main_encode.py module and after run the main_decode.py.

### 7.5.2 Tools and External libraries

For the implementation we used many tools. Some tools are build by myself and some others are from the nltk library.

- **Synonyms Dictionary**: This is a class that we build to handle easily the words. For each word provided by the bag of word, we have found all it synonyms, and assign a unique bit vector.
- **Extraction**: Our own extraction module for Sentiment term from the googledoc.
- **NLTK Libraries**: Wall street journal corpus, because we want data which is related to the financial field.
- **SentiWordNet**: we build our own sentiwornet treatment. Optional in case of system extension. Of course reduced to the financial terms.

### 7.5.3 Choices

- Interval to reduce the information exchange: the interval is been fixed by the two parts. There no scientific approach behind.
- The Limit of hidden capacity: The system provide a number which is the limit. All the documents with a sentiment value bellow this limit are considered not relevant enough to transmit all the needed message. At this moment, this limit is fixed according to the

average value of the hidden capacity. But, as a future work it might be interesting to find a more robust scientific approach to calculate the limit.

- The way of building the dictionary: In a reduced stegoSystem. Which mean a system with a reduced number of synonyms. We can avoid the word ambiguity and make the encryption and decryption easier, simply because there are no different synonyms from the same word (with the same bit vector).

### 7.5.4 Sentiment outlier

Basic document sentiment analysis, which take a document tokenize all the terms. And, according to the token and their sentiment value, produce the document sentiment.

Some Details of the sentiment calculation module are:

- We have the dictionnary tagger which take a splitted text. Tag all the tems according their grammatical nature of their sentiment polarity. it takes as parameters the files which contains positive words, negative words, the inversion words and the incrementation words.
- We have also the sentence score function. Which take a tagged sentence, and return the corresponding score according to the terms inside. for each term inside according its tag, it adds or reduces a value to the global score. until the end of the sentence, and it returns the cumulative score.
- To summarize all, we have the sentiment score function. which take a document and according to each sentence, it returns the document sentiment.

### 7.5.5 Steganography encryption/decryption

Basic synonym substitution. There are two module, first the Dictionnary Utility modules which is the module for dictionnary, the second is the Steganography library which content the encryption and decryption methods.

**Details of the Dictionnary Utility modules** It is the module which handle all the dictionnary transaction. we can find there the method like findWordBySynonym, findAllSynonyms, getBitVector and getSynonymsByBitVector et cetera.

- There just one main class which contains all methods and two parametesr which are a python dictionary. the first parameter is for the words and their synonyms, the second is the position of each word in the first one. we need the second parameter for the bit vector construction.
- It implements the methods for adding a word or synonym, there are also methods for finding a synonym by word and its inverse, find a word by bit vector and find a bit vector by the word.

**details of the Steganography library** In this module, we have the necessary methods for the substitution and the recovering of the message.

- We first initialize the dictionary for substitution, and we add the words and their synonyms.

- we implemented the substitution method. it takes two parameters, namely the document which content the original text, and the document which content the message to hide. it splits the document in sentences, and for each sentence it splits the sentence in words and looks for possible substitution. if it finds substitution, the substitution is made and bit vector updated.
- The recovering process takes one parameter, which is the probable stego document.it splits the document in sentences, each sentence in set of words. for each word, i has to look into the dictionary to find its bit vector, and retrieve it in case of it exists.

### 7.5.6 Main system

The Two mains module are being build with all the other module. the Main System for encoding, and Main System for decoding.

**Details of the main System for encoding** It is the main of the system which reuse all the other done modules. In order to find the best document to hide the message. And, to hide the message. As input, it takes a set of documents, which is materialized in our test suite by the many lines of the alerts csv file. As output, it produces the same many documents with one stego modified document.

- It begins by cleaning the data. for example: removes the non alphanumerics characters.
- It reduces the data by selecting only these which are related to the irish crisis.
- It calculates the sentiment for each of them, it applies the defined bound for the sentiment outlier.
- If the document match with our requirements, it hides the message.

**Details of the main System for decoding** As input, it takes the outputs of the main encode. It detects the best documents according to the defined interval. according to the message length, it decodes the hidden message an returns the message with the bit vector.

- For each document from the input. it calculates the sentiment in order to see if it match to the requirements.
- In case of matching, it tries to decrypt the message and returns the bitvector. If the length of the returned message is equal to the our message. The message is returned and the decryption stoped.

## 7.6 Results and Result Discussion

For the results, we first closely look some variables like hidden capacity, imperceptibility, Robustness. And after,we show the results of the encoding and decoding executions.

- the main results are hidden capacity with 0.7302 average value, imperceptibility 0.75 this one is being calculed by small survey with 5 people, Robustness.
- According to the literature a steganography sustem is evaluated on three dimensions. First, its capacity to embedde information efficiently, the innocence of the document.

- even if there are a little gap between our results and the financial news over the period. We saw that the trend remains the same.

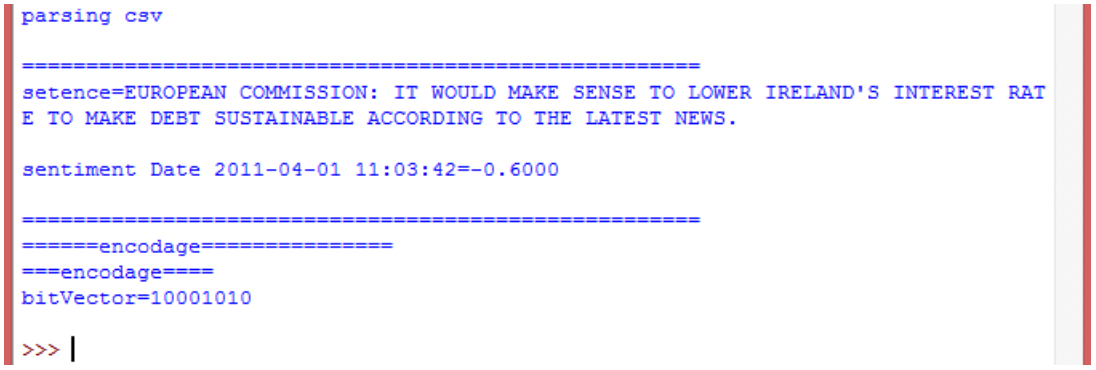



Figure 41: Illustration of the encoding process with a given document and message!!

**Encode Run test image**

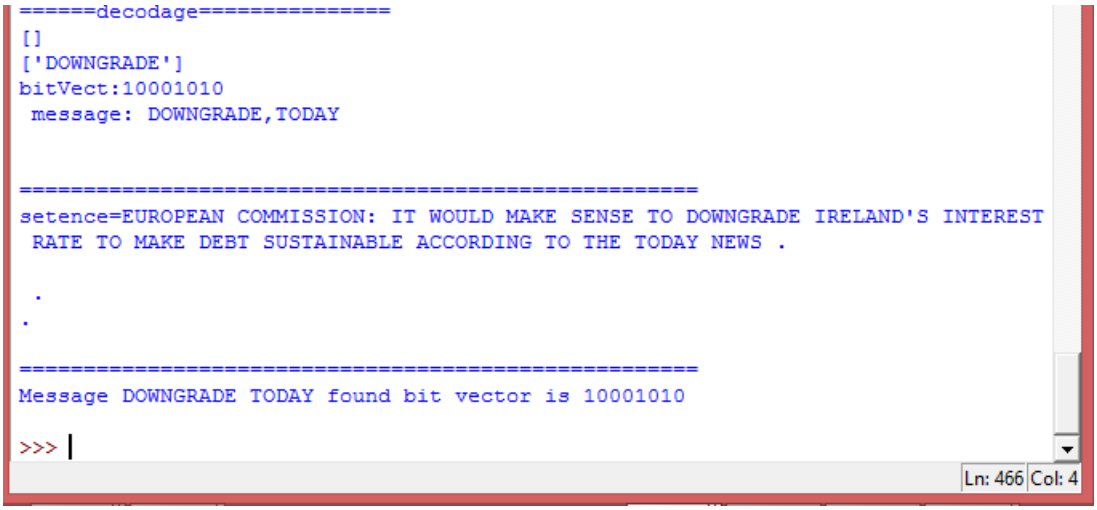



Figure 42: result of the decode run

**Decode Run test image**

### 7.6.1 Hidden capacity

In other term, we can call it embedding capacity. Which is the measurement of the sufficient level to achieve an efficient information transmission.

**Example** message="attack now" document="the strike remains active until today 5pm". If we take synonym(attack)=strike and synonym(now)=today then hiddenCap=100%. If hiddenCap=0% We can not hide anything in our document. So, the hidden capacity helps us to choose

between two documents which is the best to hide a specific message. Of course according to the document sentiment. there will be a penality for the final hidden capacity of each document.

### 7.6.2 Imperceptibility

the ability to remain innocent after modification. it is evaluated by human judgements.

**Example** original document="the strike remains active until today 5pm". modified document="the to attack remains active until to now 5pm". If, we give the modified document to someone. if the sentenc make no sens, or if we have grammatical,syntactic or semantic mistakes. the reader can be suspicious about the document mainly if the original document is from a serious source . And this step is the first which leads the covert communication exposure. This example was just for human. but, even with the machines. all analysis have to conclude that the document has not been modified.

### 7.6.3 Robustness

the ability of the system to resist against the steganalysis attaque. For the robustness, we had two choices. the First was to base it on the algorithm, the second was to was the internal tools like the substitution dictionnary. We choose to base it on the internal tool, because we used the very basic subtitution algorithm. It can be very easy to decrypt.

The particularity of the dictionnary is, each word inside has a unique bitvector which is the the result of the concatenation of two bitvector.

So, For a machine it's nearly impossible to decode a document encoded by our system, if it did not implement our dictionnary first. Because, the dictionnary construction algorithm is complex and not easy to guess.

As well for human, the only way to decrypt a document is to guessthe construction of our dictionnary.

It can be possible, but even for the machines it takes time to have the right construction algorithm, because of the many other existing combination.

### 7.6.4 Hidden Capacity chart

This is shown in Figure 43.

### 7.6.5 Discussion

Our results for the Hidden capacity and the imperceptibility are satisfying. Because, many of the approaches showed in the literature are bellow our results. for example Practical Linguistic Steganography using Contextual Synonym Substitution and Vertex Colour Coding from Ching-Yun Chang, Stephen Clark. but, we ca not ignore that our system is very basic. So, we did not apply many constraints which tend to reduce the hidden capacity. And their algorithm is more consistent, stable and it is designed for a large scale system.

## 7.7 Conclusions and Future Work

The main idea of this work was to have a first version a steganography System.

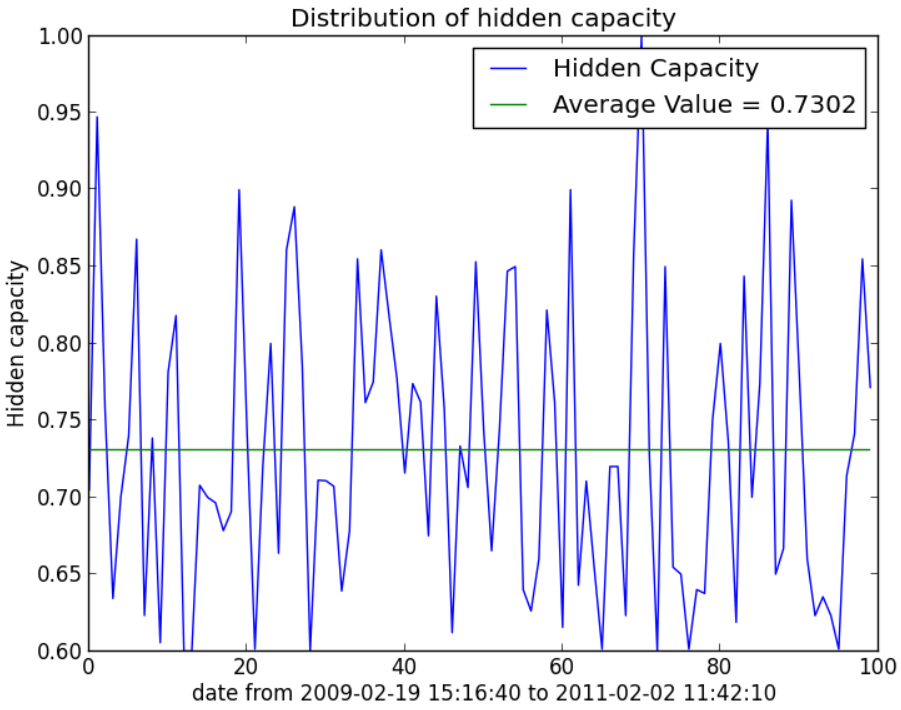



Figure 43: In this chart we can see the trend of hidden capacity over the time.

But, we can ignore some weaknesses like the imperceptibility and robustness evaluation, and in this version of the project we didnt take on account the word disambiguating. as future work the first thing will be to introduce the word disambiguating, and find better way to calculate the robustness and evaluate the imperceptibility. one of The idea for continu this work is to include the antonyms process. or implement the process of sentence naturalness classication.

The strength of the approach is the way of building the synonyms dictionary, because it helps us to have good robustness. the observations are that our synonyms substitution algorithm is not sophisticated anough. One of the weaknesses is the number of people for the imperceptibility evaluation.

## References

[1] Bing Liu *Sentiment Analysis and Subjectivity. Department of Computer Science University of Illinois at Chicagot.*

[2] Ching-Yun Chang, Stephen Clark *Practical Linguistic Steganography using Contextual Synonym Substitution and Vertex Colour Coding University of Cambridge Computer Laboratory*

[3] Ei Nyein Chan Wai, May Aye Khine *Steganography Approach by Using Syntax Bank and Digital Signature.*

# A Bag of Words

- Financial Terms: credit event; eurozone stability; bank liquidity; secondary bond markets; secondary markets; bond markets; mkt access; euro debt crisis; credit ratings; credit ratings agencies; rating agencies; irish issue; irish crisis; borrowing costs; debt as proportion to gdp; debt to gdp; debt/gdp; debt relative to gdp; irish debt; cuts to rating; rating downgrade; downgrade; credit default swaps; government bond yield; bankruptcy; eu funds; eurobond; ecb liquidity; reform; governance; marshall plan; extension of efsf loans; credit enhancement; credit enhancements; emergency liquidity; euro zone summit; non voluntary measures; interest rate flexibility; lower interest rates; reduction in interest rate; cut interest; lowering interest rates; lower public lending rates; interest rate reduction; selective default; irish bond buyback; debt buy back; debt buy back; bond buyback; debt buyback; assistance programme; irish bailout; irish bailout; irish financing programme; solution on ireland; irish package; banking tax; banking sector tax; overnight loan facility; bankruptcy; credit rating; fiscal target; contamination; contagion; debt sustainability; corruption; inflation; current account; debt as a proportion to gdp; debt to gdp; debt; gdp; deficit; bailout; public deficit; unemployment; euro area; sdb; sovereign debt crisis; irish collateral; bailout; stabililty and growth pack; collateral; debt buy back; credit enhancement; debt sustainability; debt; privatisation; european stabililty mechanism; esm; austerity; task force; task force; technical help; structural reforms; lower interest rates; stability fund; stability facility; european financial; efsf loans; european financial stability fund; efsf; 109 billion euro; official financing; financing; recapitalize irish banks; bank stress test; recapitalisation; lengthen the maturities; debt exchange; extending maturity; extension in maturity; exchange of irish debt; bond swap; roll over; rollover; debt rollover; strengthen the maturity; extend the maturities; extended maturities; maturity; private sector participation; 106 billion euro; 37 billion euro; haircut; private sector involvement; contribution to the private sector; private sector action; private sector role; voluntary participation of the private sector; private sector exchange; private debt reduction; private creditors contribution; psi.

- Political & Financial Key players: paul thomson; klaus masuch; matthias mors; preben aamann; diederik debaecker; herman rompuy; joaquim almunia; olli rehn; jose barroso; jean claude trichet; mario draghi; dominique strauss kahn; christine lagarde; juergen ligi; jean claude juncker; axel weber; jens weidmann; karl theodor zu guttenberg; rainer bruederle; philippe roesler; guido westerwelle; peer steinbrueck; wolfgang schaeuble; angela merkel; christine lagarde; francois baron; pierre moscovici; francois fillon; jean marc ayrault; nicolas sarkozy; francois hollande; jyrki katainen; jutta urpilainen; matti vanhanen; mari kiviniemi; wouter bos; jan kees de jaeger; jan peter balkenende; mark rutte; josel proell; maria fekter; werner faymann;

- European Country Name: Austria; Belgium; France; Germany; Ireland; Italy; Luxembourg; Netherlands; Portugal; Spain; Switzerland; United Kingdom; United States;

# B European Council Meetings – Time Windows

Regarding the time periods (= separated by the European Council Meetings), 26 dates have been identified within the years of 2009 until 2013 (Source: Wikipedia; confirmed by colleagues from Dept. of Finance). The time periods are then (last day is the European Council Meeting):

- 2009:
  - 01: February 1 - March 1
  - 02: March 2 - March 20
  - 03: March 21 - April 5
  - 04: April 6 - June 19
  - 05: June 20 - September 17
  - 06: September 18 - October 30
  - 07: October 31 - November 19
  - 08: November 20 - December 11
- 2010:
  - 09: December 12 (2009) - February 11
  - 10: February 12 - March 25
  - 11: March 26 - May 7
  - 12: May 8 - June 17
  - 13: June 18 - September 16
  - 14: September 17 - October 29
  - 15: October 30 - December 17
- 2011:
  - 16: December 18 (2010) - February 4
  - 17: February 5 - March 11
  - 18: March 12 - March 25
  - 19: March 26 - June 24
  - 20: June 25 - July 21
  - 21: July 22 - October 26
  - 22: October 27 - December 9
- 2012:
  - 23: December 10 (2011) - January 30
  - 24: January 31 - March 2
  - 25: March 3 - May 23
  - 26: May 24 - June 29
- 2013:
  - 27: June 30 (2012) - March 14

## References


[1] J. Brooke, M. Tofiloski, and M. Taboada: Crosslinguistic sentiment analysis: From English to Spanish. In *Proceedings of RANLP*. 2009.

[2] X. Ding, B. Liu, and P. S. Yu: A holistic lexicon-based approach to opinion mining. In *Proceedings of the Conference on Web Search and Web Data Mining*. 2008.

[3] X. Ding, B. Liu, and L. Zhang: Entity discovery and assignment for opinion mining applicationS. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2009.

[4] E. C. Dragut, C. Yu, P. Sistla, and W. Meng: Construction of a sentimental word dictionarY. In *Proceedings of ACM International Conference on Information and Knowledge Management*. 2010.

[5] W. Du, S. Tan, X. Cheng, and Y. Yun: Adapting information bottleneck method for automatic construction of domain-oriented sentiment lexicon. In *Proceedings of ACM International Conference on Web Search and Data Mining*. 2010.

[6] A. Esuli, and F. Sebastiani: Determining term subjectivity and term orientation for opinion mining. In *Proceedings of ConF. of the European Chapter of the Association for Computational Linguistics*. 2006.

[7] Eureopan Commission – Economic and Financial Affairs (Ireland): http://ec.europa.eu/economy_finance/eu/countries/ireland_en.htm

[8] R. Feldman, B. Rosenfeld, R. Bar-Haim, and M. Fresko: The Stock SonarSentiment Analysis of Stocks Based on a Hybrid Approach. *IAAI-12, pp. 16421647*. 2011.

[9] R. Feldman: Techniques and Applications for Sentiment AnalysiS. *Communications of the ACM*. April 2013, Vol. 56, NO. 4.

[10] C. D. Fellbaum: Wordnet: An Electronic Lexical Database. *MIT Press, Cambridge, MA*. 1998.

[11] S. Feng, R. Bose, and Y. Choi: Learning general connotation of words using graph-based algorithmS. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics (Edinburgh, Scotland, UK. pp. 10921103* 2011.

[12] Z. Hai, K. Chang, and J. Kim: Implicit feature identification via co-occurrence association rule mining. *Computational Linguistics and Intelligent Text Processing, pp. 393404*. 2011.

[13] V. Hatzivassiloglou, and K. McKeown: Predicting the semantic orientation of adjectiveS. In *Proceedings of the Joint ACL/EACL Conference, pp. 174181*. 1997.

[14] M. Hu and B. Liu: Mining and summarizing customer reviewS. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 168177*. 2004.

[15] M. Hu and B. Liu: Mining opinion features in customer reviewS. In *Proceedings of AAAI, pp. 755760*. 2004.

[16] N. Jakob and I. Gurevych: Extracting opinion targets in a single-and cross-domain setting with conditional random fieldS. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*. 2010.

[17] N. Jindal and B. Liu: Identifying comparative sentences in text documentS. In *Proceedings of ACM SIGIR ConF. on Research and Development in Information Retrieval*. 2006.

[18] J. Kamps, M. Marx, R. J. Mokken, and M. de Rijke: Using WordNet to measure semantic orientation of adjectiveS. *LREC*. 2004.

[19] S. M. Kim, and E. Hovy: Crystal: Analyzing predictive opinions on the WeB. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. 2007.

[20] J. Lafferty, A. McCallum, and F. Pereira: Conditional random fields: Probabilistic models for segmenting and labeling sequence datA. In *Proceedings of the 18th International ConF. on Machine Learning. Morgan Kaufmann, San Francisco, CA, pp. 282289*. 2001.

[21] B. Liu: Sentiment analysis and subjectivitY. *Handbook of Natural Language Processing. N. Indurkhya and F. J. Damerau, eds.*. 2010.

[22] B. Liu: Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language TechnologieS. Morgan & Claypool Publishers*. 2012.

[23] R. Narayanan, B. Liu, and A. Choudhary: Sentiment analysis of conditional sentenceS. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 180189*. 2009.

[24] O. Netzer, R. Feldman, M. Fresko, and Y. Goldenberg: Mine your own business: Market structure surveillance through text mining. *Marketing Science*. 2012.

[25] B. Pang and L. Lee: A Sentimental Education: Sentiment Analysis using Subjectivity Summarization based on minimum cutS. In *Proceedings of the Association for Computational Linguistics, pp. 271278*. 2004.

[26] B. Pang, L. Lee: Opinion mining and sentiment analysiS. *Foundations and Trends in Information Retrieval 2(1-2), pp. 1135.*. 2008.

[27] B. Pang, L. Lee and S. Vaithyanathan: Thumbs up? Sentiment Classification using machine learning techniqueS. In *Proceedings of EMNLP-02, 7th Conference on Empirical Methods in Natural Language Processing (Philadelphia, PA, 2002). Association for Computational Linguistics, Morristown, NJ, 7986*. 2002.

[28] W. Peng and D. H. Park: Generate adjective sentiment dictionary for social media sentiment analysis using constrained nonnegative matrix factorization. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*. 2011.

[29] A. M. Popescu, O. Etzioni: Extracting product features and opinions from reviewS. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*. 2005.

[30] G. Giu, B. Liu, J. Bu, and C. Chen: Opinion word expansion and target extraction through double propagation. *Computational Linguistics* 37, 1, pp. 927. 2011.

[31] E. Riloff, and J. Wiebe: Learning extraction patterns for subjective eExpressions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2003.

[32] R. Bersan, D. Kampas, C. Schommer: A prospect on how to find the polarity of a financial news by keeping an objective standpoint. In Proceeding of *International Conference on Artificial Intelligence and Agents (ICAART)*. 2013.

[33] P. Stone: The general inquirer: A computer approach to content analysis. *Journal of Regional Science 8, 1*. 1968.

[34] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede: Lexicon-based methods for sentiment analysis *Computational Linguistics 37, 2, pp. 267307*. 2011.

[35] Thomson Reuters User Manual.

[36] O. Tsur, D. Davidov, and A. Rappoport: A great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In *Fourth International AAAI Conference on Weblogs and Social Media*. 2010.

[37] P. Turney: Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviewS. In *Proceedings of the Association for Computational Linguistics, pp. 417424*. 2002.

[38] X. Wan: Using bilingual knowledge and ensemble techniques for unsupervised Chinese sentiment analysiS. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (Honolulu, Hawaii, 2008). Association for Computational Linguistics, pp. 553561*. 2008.

[39] T. Wilson, J. Wiebe, and P. Hoffmann: Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing, pp. 347354*. 2005.

[40] Y. Wu, Q. Zhang, X. Huang, and L. Wu: Phrase dependency parsing for opinion mining. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*. 2009.

[41] H. Yu, V. Hatzivassiloglou: Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 2003.

# Index