E1: ENRON Sentiment

Andrea Jemmett
Vrije Universiteit Amsterdam
Amsterdam, the Netherlands
andreajemmett@gmail.com

Enrico Rotundo
Vrije Universiteit Amsterdam
Amsterdam, the Netherlands
enrico.rotundo@gmail.com

ABSTRACT

Email dataset analysis is a challenging task in terms of quantity and poor-structured data. Anyway, the availability of big computational infrastructures such as cluster computers helps to face the former issue. Indeed, such platforms provide high and scalable computing and unload the programmer from the burden of managing most of its parallelisation and distribution. Unfortunately, email datasets usually come as unstructured dataset in the form of text files or, whenever they contain any markup structure, the actual data might not be well formed. In that case, the data could be human-readable but hardly parsable by a machine. Therefore, the analysis should include additional mining steps and many integrity checks, in order to minimise any possible inconsistencies.

In the past years, several email datasets from diverse sources have been publicly released. In this paper, we analyse the famous "ENRON Corpus" which contains 620k messages in about 150 mailboxes belonging to ENRON employees involved in a court case. We extract and analyse sentiments within those messages using functional programming together with a well known engine for large-scale data processing. Thus, the analysis is run in a high performance computing cluster. We present our result as an interactive visualisation of the sentiment spread via emails together with the company's stock price of the same period. Our results show that there is a weak correlation between the company's stock price and the overall sentiments extracted from the email corpus. Correlations become more consistent when we consider individuals mailboxes.

1. INTRODUCTION

Email is, at least on the user side, a simple mean of communication. Its popularity is probably due to the simplicity of usage: users can send textual messages and attachments to other addresses, also from mobile devices [6]. Thus, in the digital era it became very a popular way of communication between privates and companies. Normally, corporate emails are characterised by a specific structure, for example *user@company.com*, where the *user* suffix is a mailbox identifier and *company.com* is a distinguishable company web domain. A corporate mailbox server can handle and store thousands of inbound or outbound messages every day, collecting quite a huge amount of exchanged data.

Email dataset analysis consists in analyse a dumped data in order to extract specific information (e.g., communication patterns, sentiment analysis, etc.). Such analysis is expensive in terms of computation: the data is often composed of a multitude of items that have to be processed individually. Therefore, such tasks are normally run in distributed environments which allow high degrees of parallelisation. Cluster computing provides a platform for executing complex parallel tasks in a programmer-friendly environment [5, 27, 24]. This means the programmer does not explicitly code how to parallelise the computation. Moreover, such systems rely on distributed file systems which provide large storage capabilities and support for redundancy and distributed accesses [25].

Furthermore, email datasets usually come in a semi structured fashion in the sense that the actual data might not be well formed. For instance, recipients attributes and email's body can be difficult to parse. Thus, the analysis should include some validation steps which increase the complexity of the whole process.

In this paper, we present a sentiment analysis on the well-known "ENRON Corpus" which contains 619,446 messages over 158 users [22]. This dataset has been published by the Federal Energy Regulatory Commission[1] during its early 2000s investigation on ENRON Corp. for bankrupt and fraud. Although it contains the mailboxes of ENRON's employees which were involved in the court case, the messages include text from many more email addresses, for example personal or even external to the company. We perform sentiment analysis using state-of-the-art large-scale data processing tools. Due to the size of the dataset, about 50GB, we need to parallelise the computation. Thus, we use a functional programming language which is natively supported by Apache Spark engine. The latter is deployed in a cluster system which runs the whole computation quickly and in a flexible distributed environment.

Our outcome is a visualisation of the sentiment extracted from employees' emails together with the ENRON's stock price of the same period. Our experiment shows there is just a weak correlation between ENRON's stock price drop and sentiments extracted from the company's email dataset.

[1] http://www.ferc.gov/

We also show that this is not always true if we consider individuals mailboxes.

The rest of the paper is organised as follow. Section 2 introduces similar works and the kind of technology used in our work. Section 3 points out some research questions we try to answer by our analysis. Section 4 details our analysis setting with respect to the analysis pipeline and its technical architecture. Finally, Section 5 describes the of experiment run in order to collect our results and Section 6 draws some conclusion on the whole work.

2. RELATED WORK

In this section we present an overview of some works related to this paper: *email dataset analysis*, *sentiment extraction* from text and lastly *large-scale data processing tools*.

2.1 Email Dataset Analysis

About email dataset analysis, literature reports many works focused on exploring, filtering and describing email datasets. Datasets can be noisy and some preparatory work like filtering and reorganising might be helpful to have a better grip on the data. For instance, [16] provides metrics of the "ENRON corpus" as well as a description of its structure. A thorough analysis of such structure highlights the presence of redundant and SPAM messages. Similarly, [28] describes some cleaning strategies for the aforementioned corpus. In particular, the authors analyse the actual difficulties in cleaning a corporate email dataset which in the ENRON case are multiple and mainly related to the text-parsing phase. Indeed, the authors claim there are a certain amount of duplicate emails, addresses and attachments which might come in a slightly different format, making the parsing more challenging. For example, it is possible to identify duplicate messages by checking the MD5 digest of the email's body constrained by same day [8]. Moreover, email bodies often report forwarded text or signatures which are not useful for a sentiment extraction. The authors claim that within the ENRON dataset, only 250k messages are actually unique and they belong to a total of 149 employees. In [15], the authors investigate the feasibility of email folder prediction considering recipients attributes (e.g., *From*, *To*) as well as *Subject* and *body*. Unfortunately, the F1-score achieved using a Support Vector Machine (SVM) seems very poor, ranging from 0.3 to 0.7.

2.2 Sentiment Extraction

Sentiment extraction from text has been well studied in the past 10 years [1, 9, 3, 12, 4]. In [17], the authors present a powerful deep-learning based tool for text annotation which provides sentiment labelling on a sentence-grain. That tool is the state-of-the-art in text annotation and is distributed as a fast and easy-to-use Open Source library. A live demo is also available on the related website[2]. Most emails contain human written text, therefore it is likely to contain some kind of emotions. Its spread is influenced by many factors (e.g., social, behavioural, etc.). For example, [19] shows emotion patterns in email messages occur with different characteristics depending on the genders involved. Specifically, the authors consider the eight basic and prototypical emotions [20] and point out their balance is biased depending on the gender of the sender/receiver genders.

[2] http://nlp.stanford.edu:8080/sentiment/rntnDemo.html

2.3 Large-scale Data Processing Tools

In order to perform email dataset analysis it is often necessary to employ specific tools able to support large-scale data processing jobs. Currently, there are many distributed tools available such as file systems and computing engines which often are shipped altogether as a single product [24, 26, 18, 23, 10, 11] The Hadoop Distributed File System (HDFS) provides reliable storage of very large datasets and it has been developed as Open Source version of the Google File System (GFS) [24, 13]. Altough HDFS is implemented in Java to support portability, it presents some performances drawbacks under certain conditions [21].

MapReduce is a flexible data-processing tool that automatically parallelise map-reduce jobs over key/value pairs, it is usually deployed on a cluster of computers which can deliver sufficient performances speedup [11]. This tools provides a powerful platform for deploying a diverse set of tasks. For instance, most machine learning algorithms can be implemented as map-reduce tasks and therefore run over a MapReduce cluster [7].

Finally Apache Spark represents the state-of-the-art bundle which implement improved versions of the aforementioned functionalities [23]. Indeed, this tool has been designed in order to bundle multiple big-data functionalities and libraries (e.g., SQL, machine learning, graph analysis, etc.) [18], as well as to boost the overall performance [14]. Among many others, Spark provides advanced relational data processing which is close to traditional SQL databases systems [2].

3. RESEARCH QUESTIONS

In this paper we show a sentiment analysis of an emails dataset. Considering the "ENRON corpus" a lot of work has already been done (see Section 2.1). However, in order to give our contribution, we formalise the following research questions:

1. Is it possible to extract useful and consistent sentiment information from a noisy dataset such as "ENRON Corpus"?
2. Does the sentiment extracted from ENRON's emails correlate with the company's stock price of the same period?
3. How individual behaviours affects the overall sentiment observable within a short (i.e., 1 day) timespan?
4. Are there some particular mailboxes that show positive or negative correlation with the stock prices?

In the rest of the paper we provide a description of our setting and we attempt to answer the research questions listed above.

4. PROJECT SETUP

In this section we explain our software setup and the developed architecture. We created and deployed three Spark jobs, each one with a specific task, in order to render them more easily maintainable, manageable and deployable.

4.1 Archive Extraction

The first step in our data analysis pipeline comprises the extraction of the dataset from its *zipped* form. The dataset consists of a collection of archives, each one enclosing an ENRON's employee mailbox, containing emails in both native (eml, pdf, docx) and plain text (in which attachments have been converted into plain text too). We decided to extract and work with the plain text messages, filtering out attachments. The `UnzipperDriver` job is responsible for this. The goal of the job is to unzip the archives and store a collection of Scala tuples of the kind `(mailboxName, documents)` where the first element is the mailbox name (extracted from the zip filename) and documents is a `Seq` of extracted documents.

4.2 ETL

Second step is the *ETL* (Extract, Transform, Load). Here the term *Extraction* has a different meaning than that of the previous job. The extraction part in `ETLDriver` is about extracting data from emails in a structured form, that is we need to extract emails body and headers from the raw data. Do do this we created an `EmailParser` object that implements the extraction and parsing logic. In this we apply a series of transformations to the raw emails:

- extract email headers (*Date*, *From*, *To*, *Cc*, *Bcc* and *Subject*);
- separate body from headers;
- clean body from common dataset footer [3];
- clean body from quoted emails, "*Original Message*" and "*Forwarded By*" text and delimiters.

Because we are interested only in emails exchanged by Enron employees, we need a way to retrieve and identify those people from the email headers. Unfortunately this is not straightforward because addresses and people names have different formats like "*first-name last-name*", "*last-name, first-name*", "*first-name last-name, email*" or "*last-name, initial*". To solve this problem we use an external list of *mailbox custodians* that gives us first, last name and its role within Enron. We then search *From*, *To*, *Cc* and *Bcc* headers for known custodians. If for an email it fails to find any known custodian, the message is discarded.

Next processing step consists in transforming the extracted dataset into a form more easily manageable and queryable. We use CoreNLP to tokenize and find sentiment at sentence level in emails. Because the sentiment is at sentence level we need a way to aggregate those into a single sentiment for the whole email. To aggregate sentiments we use the following formula:

$$S = \frac{\sum_{s \in SS} s}{|SS|} \quad (1)$$

where SS is the set of sentence sentiments.

Last step consists in storing the dataset for future analysis. We split the dataset in two: one contains the full email body and no sentiment, the other does not contain the body but has sentiments. We do this in order to have a more lightweight, sentiment-tagged corpus that we can analyse with more agility. We convert both datasets into Spark SQL's `DataFrame` that enables us to perform queries and data aggregation using the SQL language.

4.3 Sentiment Analyser

Last job that we use in our pipeline is the `SentimentResumer`. It is responsible for querying the sentiment-tagged dataset and joins it with Enron stock prices based on date. To do this we group emails by date and then we average the sentiments for that day and then join them with the stock prices data on the date field. Lastly we store the results as JSON so that we can then download them and use them for visualisation purposes. We apply the same computation to the individual mailboxes, to explore the correlation with stock prices of individual ENRON's employees.

4.4 Visualisation

The resulting JSON files are used to visualise the correlation of ENRON stock prices with email sentiment. We created a bar chart that represents individual mailboxes correlations and a simple line chart with two lines and two vertical axes. A line represents the stock prices and another represents email sentiments. Effort has been spent to make the visualisation as easy as possible to navigate and interact with. We built the interface in HTML and Coffeescript, using the d3 visualisation library. Because the codebase is hosted on Github we have chosen to deploy the visualization using Github Pages [4] which offers free hosting of static files.

5. EXPERIMENT

In this section we briefly explain the experiment run in order to obtain our results. As already said in Section 4, we run our experiments on an Hadoop cluster of 90 machines, 720 cores and 1.2PB of storage. Besides this the ETL task (Section 4.2), which is the task that takes the most in time, lasts for 2 hours. Without any parameters and settings control, the sentiment analyser of CoreNLP easily fills the heap space of workers causing them to die prematurely. After running the entire pipeline, we collect the output data in JSON format. Then, we perform a post processing step in which we firstly resample the time series on a weekly (i.e., 7 days) basis using the average as aggregation function. Moreover, we apply an interpolation function (i.e., polynomial, 4^{th} order) to smooth out our time series. The result is shown in Figure 1 where we can notice the dramatic drop of the stock values by the end of 2001. Although the sentiment value seems not to be strongly affected by the stock's drop, we point out it has a decreasing overall trend. Indeed, the Pearson correlation coefficient between the stock price and the sentiment time series is some 0.11.

Surprisingly, the distribution of sentiment values, shown in Figure 2, does not show a significant standard deviation (0.12) over the considered time period. This might be due to either low sentiment-annotation quality by the employed library, or most likely to some inconsistencies in the emails parsing task.

In Table 7 and Table 8 is an ordered (by correlation between sentiment and stock prices) list of mailboxes. The rightmost column denotes the number of data points used

[3]Every email contains a disclaimer from EDRM, the company that cleansed and published this version of the Enron Corpus

[4]http://acidghost.github.io/ENRON-sentiment-analysis/visualization/

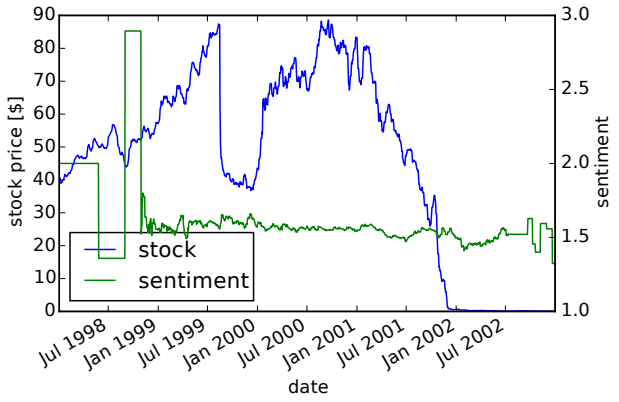

Figure 1: Time series of ENRON's stock price (blue) and sentiment values extracted from emails (green), from 1998-01-01 to 2002-12-31.

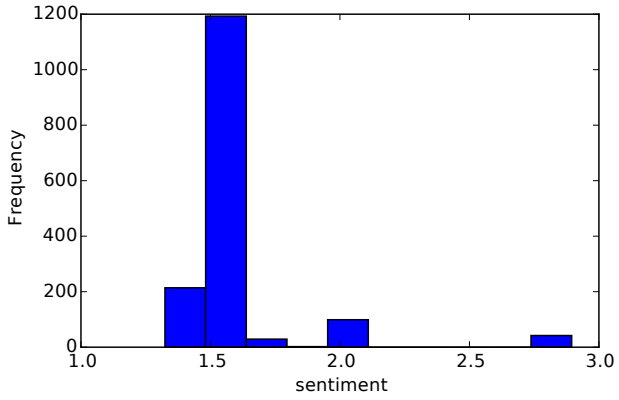

Figure 2: Distribution of sentiment values extracted from emails, from 1998-01-01 to 2002-12-31.

to compute the correlation, that is the number of days in which we have both sentiment and stock prices information. We show in Figures 3 and 4 an example of positive correlation of sentiment and stock prices for the mailbox `buy-r`. In Figures 5 and 6 instead we show an example of negative correlation for the mailbox `giron-d`.

6. CONCLUSIONS

In this section we present some conclusions based on our results, considering the research questions listed in Section 3. Although, our results include a visualisation of sentiment information contained within the "ENRON Corpus". Therefore, we claim the sentiment extraction from an emails dataset is actually doable. However, we highlight that in order to simplify our task we made some assumptions. On one hand, we claim this is absolutely common, and sometimes even necessary, in a large-scale data processing. On the other hand, the number and kind of assumptions might not be negligible. For instance, we have chosen to take into account only those recipients which appears to have a personal mailbox in our dataset. Therefore, people from outside the company and other employees are simply ignored. Given

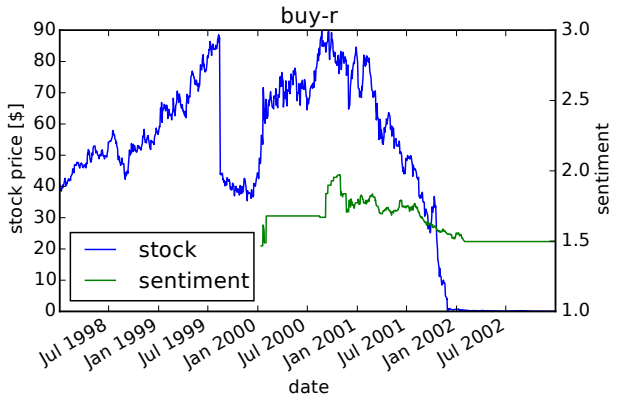

Figure 3: Time series of ENRON's stock price (blue) and sentiment values extracted from emails (green) relative to mailbox `buy-r` with correlation 0.356 and 260 data-points.

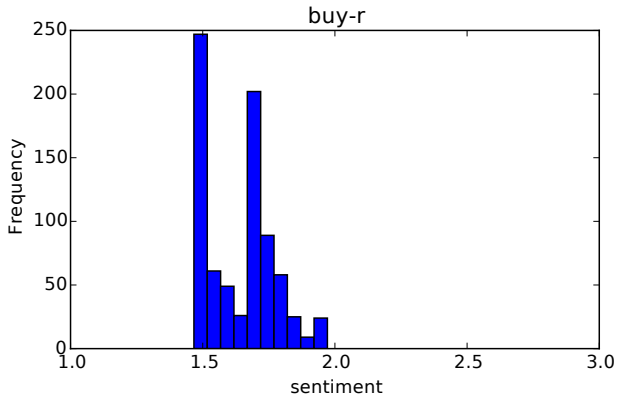

Figure 4: Distribution of sentiment values extracted from emails of `buy-r`.

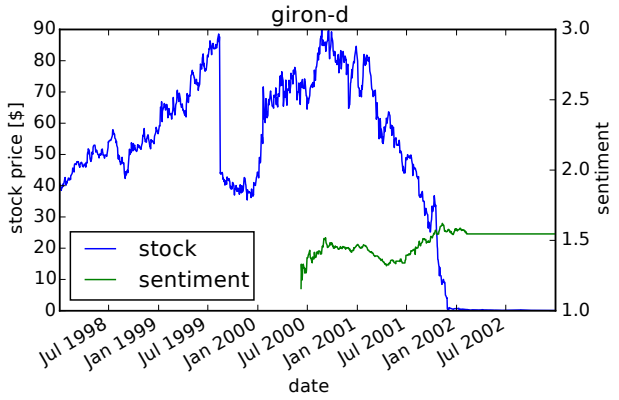

Figure 5: Time series of ENRON's stock price (blue) and sentiment values extracted from emails (green) relative to mailbox `giron-d` with correlation -0.193 and 344 data-points.

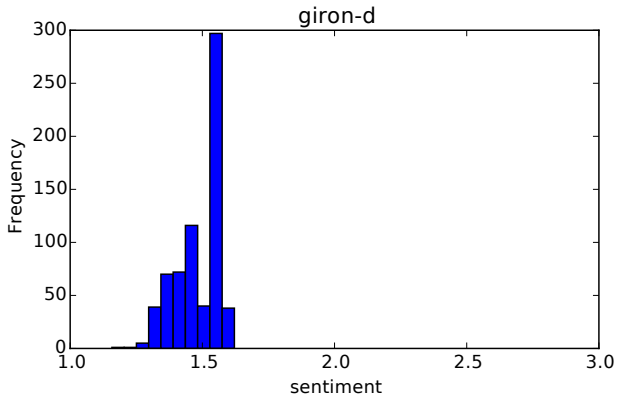

Figure 6: Distribution of sentiment values extracted from emails of `giron-d`.

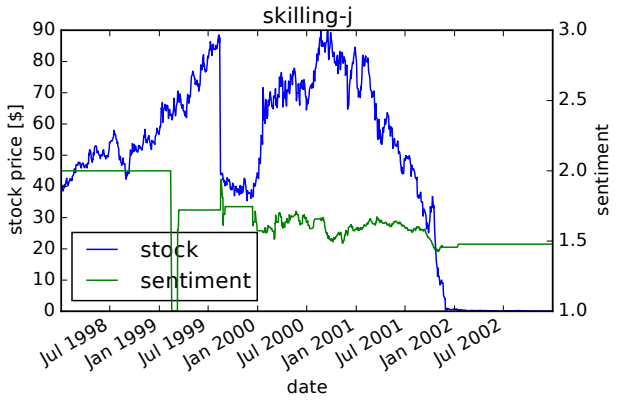

Figure 7: Time series of ENRON's stock price (blue) and sentiment values extracted from emails (green) relative to mailbox `skilling-j` with correlation 0.105 and 357 data-points.

Figure 8: Distribution of sentiment values extracted from emails of `skilling-j`.

that those people have most likely contributed to the sentiment coded in the emails, we argue this assumption is rather

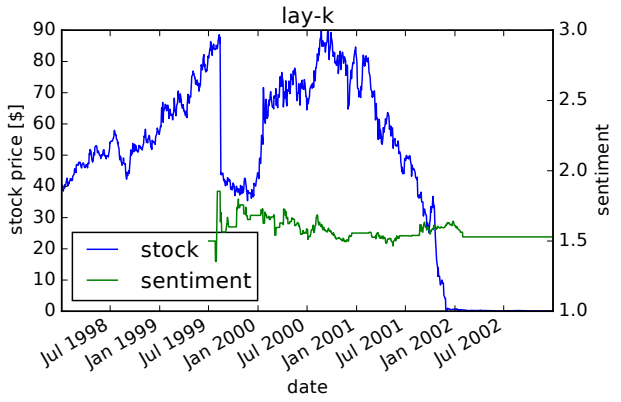

Figure 9: Time series of ENRON's stock price (blue) and sentiment values extracted from emails (green) relative to mailbox `lay-k` with correlation -0.023 and 318 data-points.

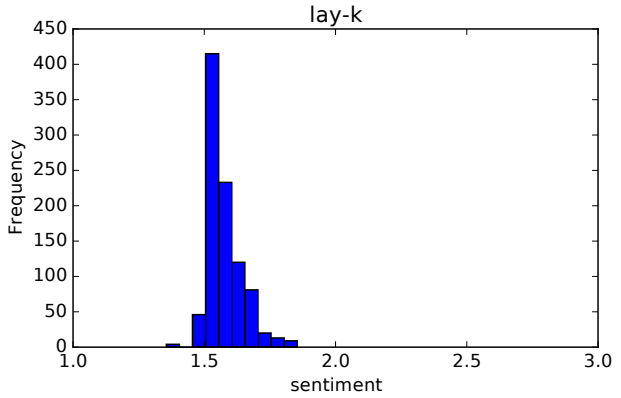

Figure 10: Distribution of sentiment values extracted from emails of `lay-k`.

strict and further investigations should assess weather it is the case. Furthermore, we claim that our visualisation provides useful hints for highlighting a not so strong correlation between emails' sentiment and the ENRON's stock price. We argue this fact might be due to noise in our dataset. Moreover, further experiments should be run with different settings, for example we could try different sentiment extraction techniques and other overall sentiment scoring functions. Lastly, we observe the research question about the influence of individual behaviours cannot find answer in this work. This is mainly due to the lack of time to run further experiments.

As further conclusion, we would highlight some remarks on the usability of employed technologies. In this paper we use Apache Spark which provides a distributed computing engine for running parallel applications. Although this system is deployed in high performance cluster which provides large memory availability, the user is still in charge of estimating its memory needs and consequently to fine tune the memory allocation, also considering machines' physical limits. Another highlight on Spark is the complexity of de-

bugging some applications and the difficulty to read error logs. Since user's applications are run over different machines, logs output are spread all over the cluster and users might suffer of not having direct and quick access to it.

7. ACKNOWLEDGMENTS

We would like to remark our thankfulness to our teachers Peter Boncz and Hannes Mühleisen for their inspiration and support during this work and the LSDE 15-16 course held at Vrije Universiteit Amsterdam. Last but not least we want to thank SURFsara[5] for providing the cluster computing platform necessary to run our experiments.

8. REFERENCES

[1] C. C. Aggarwal and C. Zhai. *Mining text data*. Springer Science & Business Media, 2012.

[2] M. Armbrust, R. S. Xin, C. Lian, Y. Huai, D. Liu, J. K. Bradley, X. Meng, T. Kaftan, M. J. Franklin, A. Ghodsi, et al. Spark sql: Relational data processing in spark. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1383–1394. ACM, 2015.

[3] X. Bai, R. Padman, and E. Airoldi. *Sentiment extraction from unstructured text using tabu search-enhanced markov blanket*. Carnegie Mellon University, School of Computer Science [Institute for Software Research International], 2004.

[4] S. Bird, E. Klein, and E. Loper. *Natural language processing with Python*. " O'Reilly Media, Inc.", 2009.

[5] R. Buyya et al. High performance cluster computing: Architectures and systems (volume 1). *Prentice Hall, Upper SaddleRiver, NJ, USA*, 1:999, 1999.

[6] Y.-F. Chen, H. Huang, R. Jana, T. Jim, S. John, S. Jora, R. Muthumanickam, and B. Wei. Enterprise mobile server platform, June 10 2002. US Patent App. 10/165,887.

[7] C. Chu, S. K. Kim, Y.-A. Lin, Y. Yu, G. Bradski, A. Y. Ng, and K. Olukotun. Map-reduce for machine learning on multicore. *Advances in neural information processing systems*, 19:281, 2007.

[8] A. Corrada-Emmanuel. Enron email dataset research. *Retrieved October*, 5, 2004.

[9] S. R. Das and M. Y. Chen. Yahoo! for amazon: Sentiment extraction from small talk on the web. *Management Science*, 53(9):1375–1388, 2007.

[10] J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[11] J. Dean and S. Ghemawat. Mapreduce: a flexible data processing tool. *Communications of the ACM*, 53(1):72–77, 2010.

[12] M. Gamon, A. Aue, S. Corston-Oliver, and E. Ringger. Pulse: Mining customer opinions from free text. In *Advances in Intelligent Data Analysis VI*, pages 121–132. Springer, 2005.

[13] S. Ghemawat, H. Gobioff, and S.-T. Leung. The google file system. In *ACM SIGOPS operating systems review*, volume 37, pages 29–43. ACM, 2003.

[14] S. Gopalani and R. Arora. Comparing apache spark and map reduce with performance analysis using k-means. *International Journal of Computer Applications*, 113(1), 2015.

[15] B. Klimt and Y. Yang. The enron corpus: A new dataset for email classification research. In *Machine learning: ECML 2004*, pages 217–226. Springer, 2004.

[16] B. Klimt and Y. Yang. Introducing the enron corpus. In *CEAS*, 2004.

[17] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60, 2014.

[18] X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. Tsai, M. Amde, S. Owen, et al. Mllib: Machine learning in apache spark. *arXiv preprint arXiv:1505.06807*, 2015.

[19] S. M. Mohammad and T. W. Yang. Tracking sentiment in mail: how genders differ on emotional axes. In *Proceedings of the 2nd workshop on computational approaches to subjectivity and sentiment analysis (acl-hlt 2011*, pages 70–79, 2011.

[20] R. Plutchik. *Emotion: A psychoevolutionary synthesis*. Harpercollins College Division, 1980.

[21] J. Shafer, S. Rixner, and A. L. Cox. The hadoop distributed filesystem: Balancing portability and performance. In *Performance Analysis of Systems & Software (ISPASS), 2010 IEEE International Symposium on*, pages 122–133. IEEE, 2010.

[22] J. Shetty and J. Adibi. The enron email dataset database schema and brief statistical report. *Information sciences institute technical report, University of Southern California*, 4, 2004.

[23] A. G. Shoro and T. R. Soomro. Big data analysis: Apache spark perspective. *Global Journal of Computer Science and Technology*, 15(1), 2015.

[24] K. Shvachko, H. Kuang, S. Radia, and R. Chansler. The hadoop distributed file system. In *Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on*, pages 1–10. IEEE, 2010.

[25] S. A. Weil, S. A. Brandt, E. L. Miller, D. D. Long, and C. Maltzahn. Ceph: A scalable, high-performance distributed file system. In *Proceedings of the 7th symposium on Operating systems design and implementation*, pages 307–320. USENIX Association, 2006.

[26] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, pages 2–2. USENIX Association, 2012.

[27] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica. Spark: Cluster computing with working sets. *HotCloud*, 10:10–10, 2010.

[28] Y. Zhou, M. Goldberg, M. Magdon-Ismail, and A. Wallace. Strategies for cleaning organizational emails with an application to enron email dataset. In *5th Conf. of North American Association for Computational Social and Organizational Science*, 2007.

[5] www.surf.nl

APPENDIX

	stock_sentiment_corr	points
quenet-j	0.387	35
brawner-s	0.362	175
buy-r	0.356	260
may-l	0.351	95
whitt-m	0.338	69
ring-a	0.331	99
arora-h	0.328	126
forney-j	0.312	135
platter-p	0.289	125
townsend-j	0.252	122
donoho-l	0.249	121
mims-p	0.244	230
hodge-j2	0.234	188
presto-k	0.227	202
mckay-b	0.226	81
hendrickson-s	0.221	81
guzman-m	0.214	159
swerzbin-m	0.207	120
thomas-p	0.204	153
blair-l	0.199	140
heard-m	0.189	124
grigsby-m	0.182	175
linder-e	0.180	56
crandall-s	0.177	130
lenhart-m	0.173	283
gay-r	0.157	172
williams-b	0.151	198
smith-m	0.151	210
dickson-s	0.148	48
tholt-j	0.147	267
martin-t	0.146	177
lewis-a	0.144	146
sturm-f	0.143	183
perlingiere-d	0.138	402
mcconnell-m	0.138	294
shankman-j	0.135	212
ruscitti-k	0.134	281
wolfe-j	0.133	190
south-s	0.127	21
lokey-t	0.127	161
stclair-c	0.120	152
symes-k	0.118	158
bailey-s	0.115	70
quigley-d	0.112	174
rogers-b	0.111	390
zufferli-j	0.109	132
richey-c	0.107	190
skilling-j	0.105	357
dorland-c	0.097	279
keavey-p	0.096	142
reitmeyer-j	0.094	63
haedicke-m	0.091	425
parks-j	0.090	182
kitchen-l	0.088	242
taylor-m	0.085	726
lucci-p	0.075	70
kuykendall-t	0.066	144
mckay-j	0.065	186
semperger-c	0.064	121
keiser-k	0.056	78
fischer-m	0.053	88
fischer-m2	0.051	46
saibi-e	0.043	120
allen-p	0.034	305
nemec-g	0.031	607
holst-k	0.031	59
shackleton-s	0.030	666
baughman-d	0.028	280
hayslett-r	0.026	283
whalley-g	0.024	307
storey-g	0.022	161
pereira-s	0.021	132
donohoe-t	0.020	124
rodrigue-r	0.017	161
pimenov-v	0.014	65
meyers-a	0.011	73
ward-k	0.011	241
white-s	0.006	211
love-p	0.003	355
benson-r	0.003	133
sanders-r	0.002	294
steffes-j	0.002	129
stepenovitch-j	0.001	191
bass-e	0.000	381

	stock_sentiment_corr	points
mclaughlin-e	-0.001	339
beck-s	-0.003	470
weldon-c	-0.004	205
cash-m	-0.005	334
watson-k	-0.007	220
solberg-g	-0.008	78
schoolcraft-d	-0.009	166
scott-s	-0.012	454
arnold-j	-0.013	317
hain-m	-0.015	179
davis-d	-0.015	242
dasovich-j	-0.019	517
neal-s	-0.021	268
shively-h	-0.021	208
gilbertsmith-d	-0.022	152
lay-k	-0.023	318
slinger-r	-0.032	58
fossum-d	-0.037	270
sanchez-m	-0.037	45
lokay-m	-0.039	436
shapiro-r	-0.040	294
ring-r	-0.040	116
delainey-d	-0.040	203
hodge-j	-0.041	124
mccarty-d	-0.045	96
carson-m	-0.046	193
jones-t	-0.049	574
campbell-l	-0.052	428
harris-s	-0.054	46
germany-c	-0.059	534
horton-s	-0.061	236
king-j	-0.063	59
sager-e	-0.066	485
farmer-d	-0.066	524
williams-j	-0.067	120
causholli-m	-0.069	89
scholtes-d	-0.070	162
tycholiz-b	-0.076	154
hyatt-k	-0.078	261
griffith-j	-0.081	209
lavorato-j	-0.089	359
schwieger-j	-0.093	160
derrick-j	-0.098	180
hyvl-d	-0.111	361
stokley-c	-0.125	137
mann-k	-0.127	346
corman-s	-0.131	184
rapp-b	-0.132	57
gang-l	-0.134	76
panus-s	-0.136	49
staab-t	-0.147	92
geaccone-t	-0.148	142
ybarbo-p	-0.167	145
maggi-m	-0.175	85
cuilla-m	-0.178	123
edrm-enron-v2	-0.179	202
merris-s	-0.180	36
giron-d	-0.193	344
salisbury-h	-0.198	211
badeer-r	-0.199	76
motley-m	-0.199	100
ermis-f	-0.241	118
hernandez-j	-0.262	300
dean-c2	-0.273	85
dean-c	-0.329	187
kaminski-v	NaN	0
kean-s	NaN	0