Opinion mining of microblog texts on Hadoop ecosystem

Article · January 2016

DOI: 10.1504/IJCC.2016.075096

6 authors, including:

Mehmet Mert Yücesan
TOBB University of Economics and Technology
1 PUBLICATION **0** CITATIONS
SEE PROFILE

Erdogan Dogdu
TOBB University of Economics and Technology
62 PUBLICATIONS **363** CITATIONS
SEE PROFILE

Opinion mining of microblog texts on Hadoop ecosystem

Muhammed Akif Ağca*, Şenol Ataç,
Mehmet Mert Yücesan,
Yusuf Gökhan Küçükayan,
Ahmet Murat Özbayoğlu and
Erdoğan Doğdu

Department of Computer Engineering,
TOBB University of Economics and Technology,
Ankara, Turkey
Email: akif.agca@etu.edu.tr
Email: satac@etu.edu.tr
Email: myucesan@etu.edu.tr
Email: kucukayan@etu.edu.tr
Email: mozbayoglu@etu.edu.tr
Email: edogdu@etu.edu.tr
*Corresponding author

Abstract: Opinion mining started getting more traction due to the increasing popularity of Twitter and similar social network platforms that are producing fast and real-time responses to social events. It is a very challenging area since it is difficult, if not impossible, to identify general public sentiment towards events, entities, etc., using opinion mining techniques over huge numbers of tweets and messages automatically. In this study we present our opinion mining techniques on tweet data with early results. We apply sentiment scoring and clustering algorithms using Hadoop ecosystem for parallel processing. We classify tweets by tagging them as positive, negative, and neutral as a result.

Keywords: opinion mining; sentiment scoring; distributed computing; parallel processing; large scale application development.

Reference to this paper should be made as follows: Ağca, M.A., Ataç, Ş., Yücesan, M.M., Küçükayan, Y.G., Özbayoğlu, A.M. and Doğdu, E. (2016) 'Opinion mining of microblog texts on Hadoop ecosystem', *Int. J. Cloud Computing*, Vol. 5, Nos. 1/2, pp.79–90.

Biographical notes: Muhammed Akif Ağca graduated from Middle East Technical University (METU) in 2011, then started mastering in Computer Engineering at TOBB University of Economics and Technology. He is working on distributed data analytic systems and forecasting methods. Since 1 April 2014, he is working for HAVELSAN Aeronautic and Defence Company as a software engineer.

Şenol Ataç studied Bachelor in Computer Engineering at TOBB University of Economics and Technology. He is mastering in Computer Engineering at the same university. His focus is on cloud computing technologies. He is working for TUBITAK (Scientific and Technological Research Council of Turkey) as a software engineer since 2013.

Mehmet Mert Yücesan is an MS student in Computer Engineering Department of TOBB University of Economics and Technology. He is working on linked data and semantic web. He received his BS in Computer Science at Bilkent University.

Yusuf Gökhan Küçükayan is pursuing a Master of Science degree in Computer Science. He is interested in data mining, big data analysis, cutting-edge technologies and future technologies. He follows new trends in innovative solutions and creative ideas. He works on software development, data science and sentiment analysis.

Ahmet Murat Özbayoğlu graduated from the Department of Electrical Engineering at METU, Ankara, Turkey in 1991, then he received his MSc and PhD degrees from Missouri University of Science and Technology, USA in 1993 and 1996, respectively. After graduation, he joined MEMC Electronics (now became SunEdison), USA as a software project engineer, programmer and analyst. In 2005, he went back to academia by joining the Department of Computer Engineering of TOBB University of Economics and Technology, in Ankara, Turkey. His research interests are machine learning, pattern recognition, financial forecasting, computational intelligence, machine vision.

Erdoğan Doğdu is an Associate Professor in the Computer Engineering Department at TOBB University of Economics and Technology (Ankara, Turkey). He received his MS and PhD in Computer Science from Case Western Reserve University (Cleveland, USA). His research interests include database systems, software engineering, web services, and recently semantic web, web information systems and big data analytics. He served as PI and co-PI in several national and international research projects. He was awarded IBM Shared University Research Award in 2014 with a project in social media analytics. He has published more than 50 international conference and journal papers.

This paper is a revised and expanded version of a paper entitled ‘Opinion mining of microblog texts on Hadoop ecosystem’ presented at the 2nd International IBM Cloud Academy Conference ICA CON 2014, Georgia State University, Atlanta, Georgia USA, 8–9 May 2014.

1 Introduction

Opinion mining recently gained popularity due to the fast growth of Web social media including Twitter, Facebook, blogs, etc. Automatic understanding of people’s sentiments towards events, entities, etc., using social media data is very valuable and many researchers and industry companies are working towards that goal.

In this paper, we present a method that combines dictionary-based sentiment scoring and clustering-based classification of tweet data. Sentiment scoring is applying dictionary-based scoring to each microblog text, according to the scoring result; the tweets are classified as positive, negative, and neutral. K-means clustering is used for the purpose of looking for similar characteristics of each group. New texts are assigned to the most similar group. There are three groups assigned; these are labelled as positive, negative, and neutral. The samples and distribution for each group is predefined.

Since there are a great number and size of microblog texts in the dataset, we prefer to use a distributed data analytics system. The data is appropriate for batch parallel processing. Current technologies like Hadoop and NoSQL stores are integrated and modified according to the specific needs of the data. Each new coming tweet is stored in a NoSQL store in JSON format and processed with map-reduce methods, results are imported to NoSQL store.

2 Related work

Thelwall et al. (2012) developed an algorithm called SentiStrength that uses a lexical approach differentiating positive and negative words in text and calculating a sentiment score. But they did not include any grammar information. In another study, Özsert and Özgür (2013) used Wordnet as each word is connected to others through a relational graph, and the distance of a particular word to positive set of words and negative set of words indicated the word polarisation. This study was concentrated on words from both English and Turkish. Furthermore, a few studies are proposing large scale distributed computing systems for text analytics. Khuc et al. (2012) proposed a distributed system which has two main components, "a lexicon builder and a sentiment classifier" for twitter sentiment analysis. Lin and Kolcz (2012) applied machine learning algorithms on a large scale Hadoop-based, Pig-centric analytics platform. Within the platform, it was possible to apply common machine learning tasks such as data sampling, feature generation, training, and testing on the collected tweets.

Bermingham and Smeaton (2010) studied classifying microblog texts (tweets) in comparison to long blog texts. Unlike the common sense that microblog texts are not appropriate for sentiment analysis due to the short text size and less information content within these texts, they found that the microblog sentiment analysis was more successful than the blog sentiment analysis. Furthermore, they presented their observations on the challenges of supervised learning for sentiment analysis in microblogs and suggested the community to do more work on microblogs in future sentiment analysis research. O'Connor et al. (2010) analysed the consumer confidence and political opinion surveys, and found a correlation to sentiment word frequencies in tweets. They indicated the possibility of the potential use of text streams in place of traditional polling.

Pak and Paroubek (2010) built a semantic classifier to determine documents as positive, negative, and neutral. They were automatically collecting a corpus, which was used to train the sentiment classifier they developed. General community often uses syntactic structures to describe emotions or state facts. They suggested using some POS-tags as strong indicators of the emotions in the texts. Their classifier was based on the multinomial naïve Bayes classifier that uses N-gram and Pos-tags as features. The study was implemented in English texts, but can be used for other languages (Pak and Paroubek, 2010).

Kouloumpis et al. (2011) investigated the utility of linguistic features for detecting the sentiment of Twitter messages. They have looked at the problems of the existing lexical resources. They conducted a supervised approach to the problem, by leveraging the existing hashtags in the Twitter data for building the training data (Kouloumpis et al., 2011). Martínez-Cámara et al. (2014) surveyed the literature on the sentiment analysis

(SA) of Twitter data comprehensively. They found that some of the main objectives of SA on Twitter were

1 polarity classification

2 event prediction.

The methods used were

1 unsupervised

2 supervised

3 hybrid.

The algorithms used were in a wide range such as

1 SVM

2 naïve Bayes

3 time series

4 graph-based, and so on.

In summary, the community is using lexicon-based analytics to understand semantics in the microblog texts. The methods are generally using iterative methods. The methods necessitate the data to be accessed redundantly many times. Classical databases take long time to process. The distributed data analytics system we are developing is capable of accessing big data in parallel for batch processing. Furthermore, optimising machine learning algorithms for this platform enables faster analysis of the data.

3 Dataset and the method

We collected 573,794 tweets about universities in Ankara in the time period from April 1 to April 12 in 2014. We collected only those tweets that include keywords from a list of reserved subject keywords, in this case the names of a specific number of universities. Raw data is stored in a NoSQL storage system, namely MongoDB.

Initial datasets are the following:

3.1 Sample tweets

573,794 tweets (in the period of April 1–12, 2014) about universities in Ankara. Tweets are stored in JSON format. Following are two selected sample tweets:


```
{ "_id" : { "$oid" : "531ca82684ae0ac0b4b74a85" },
"tweet" : "Dünyanın en saygın 100 üniversitesinde tek
TÜRK üniversitesi\n ODTÜ diye yazılr Ortadoğu Teknik
Üniversitesi diye okunr \nhttp://t.co/GCisdEmlbN",
"date" : { "$date" : 1394299521000 },
"userid" : 2174545004,
"tweetid" : 442350370007052288,
```

```
"retweetcnt" : 0,
"subjectid" : "1"
}
{ "_id" : { "$oid" : "531ca84184ae0ac0b4b74ce2" },
"tweet" : "Hacettepe Üniversitesi'nin teknoloji ye ve
mühendislere verdiği önemin resmidir. Çöplük ve
Teknokent yanyana! http://t.co/7XolIoZY2W",
"date" : { "$date" : 1394366532000 },
"userid" : 88445878,
"tweetid" : 442631433618993152,
"retweetcnt" : 0,
"subjectid" : "15",
"pictureurl":"http://pbs.twimg.com/media/BiSK8YxIMAAE
-Pa.jpg"
}
```


Sentiment words are stored in the dictionary with keywords and their scores. Dictionary currently includes 2,159 Turkish keywords. Scores for keywords are chosen between –5 and 5, –5 is the most negative, 0 is neutral, and 5 is the most positive.

3.2 Sentiment words dictionary

SentiWorNet (http://sentiwordnet.isti.cnr.it/, http://neuro.imm.dtu.dk/wiki/AFINN) is a dictionary for sentiment words in English with their scores. We automatically translated keywords in SentiWordNet using online language translation dictionaries from English to Turkish and partially edited the words. Currently the dictionary has 2,159 keywords with sentiment scores as in SentiWordNet. Table 1 lists some sample words from this dictionary.

Table 1 Sample sentiment words

Score	*English*	*Turkish*
–2	Abduction	Kaçırma
–3	Abhor	İğrenmek
2	Ability	İktidar
2	Ability	Yetenek
1	Abroad	Gemiiçinde

3.3 Sample words for K-means clustering

We apply learning-based clustering algorithms with samples on Turkish tweets. Sample tweets are characterised in three groups as positive, negative and neutral. Each group includes 100 tweets, a total of 300 tweets. To construct a 100-bit vector for each tweet,

100 most used words are selected. Clustering is implemented according to the vector similarities between the tweet pairs. A selected sample of some of the most used words is listed in Table 2.

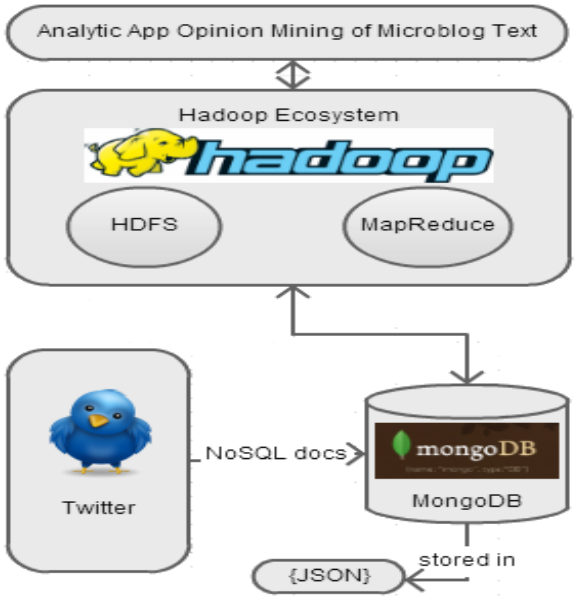
Table 2 Sample sentiment words for vectors

English	*Turkish*
Ugly	Çirkin
Uneasy	Huzursuz
Uneasy	Rahatsız
Uneasy	Mutsuz
Unjust	Haksız

4 Distributed data analytic system

A distributed text mining system is designed for processing and clustering the tweets. Streaming Twitter data is captured and stored in a NoSQL document store MongoDB (https://www.mongodb.org/) in JSON (http://www.json.org/) format. To analyse a specific dataset, it is dumped into a text file in JSON format and distributed to Hadoop Ecosystem by using HDFS. Single node and cluster performances are tested. Multiple copies of the data are stored in HDFS (http://hadoop.apache.org/docs/r1.2.1/hdfs_design.html) and parallel computing is provided by Hadoop. Further modifications are applied to Hadoop job configurations to process the data in parallel.

Figure 1 System architecture (see online version for colours)



5 Sentiment scoring and clustering for opinion mining

Most of the studies either used a lexical approach and/or a machine learning model to implement sentiment analysis. Pang and Lee (2008) provided a list for the implemented studies and presented some sort of benchmarks for opinion mining. They focused on the techniques to find the methodologies on how people think and feel.

In this study we decided to implement sentiment scoring and a supervised clustering model in order to classify each tweet into one of the three categories. The results then can be improved by using some heuristics based on the language and grammar. To process the data in parallel multiple Hadoop jobs are configured to run simultaneously, which is a popular method as Lee et al. (2012) mentioned. That particular study presented the pros and cons about this approach. The obtained results in Figure 2 on page four shows that parallel processing on map reduce is a good choice for the distributed processing of big data. As long as we have enough memory space Hadoop clusters are running faster than single nodes, as a result the more memory and nodes the system have, the better performance it has.

5.1 Sentiment scoring

Our sentiment scoring algorithm works as follows:

1 distribute all tweets to HDFS nodes for parallel processing

2 load sentiment words dictionary to HDFS nodes

3 for each tweet t: tweet score = Sum (w) (score (w)) where word is a word in dictionary and w is mentioned in t

$$\text{Tweet score} = \sum Score(w).$$

5.2 K-means clustering

We are applying an unsupervised learning algorithm, K-means clustering using the predefined vectors. Each tweet is enumerated and converted into a 100-bit vector in accordance to an initial 100 bit vector, composed of the most-used words in the dataset. There are three sample classes for dataset as positive, negative, and neutral. Go et al. (2009) also used a similar method and had the expectation like in our study that machine learning could classify tweets as positive, negative, neutral. Each cluster includes 100 manually selected tweets. Finally, the process comes to an end with analysing vector similarities of all tweets with the sample classes and assigning the tweets to the one cluster which is the most similar.

The clustering algorithm consists of the following steps:

1 statistically count the keywords in all tweets and select the top 100 words that are mentioned in tweets excluding the stop-words

2 manually select and categorise 100 sample tweets for each group as positive, negative, and neutral

3 make a 100-bit keyword vector for each tweet in three groups with the most used keywords. The particular bit value will be 1 if includes the word in the most used vector, 0 if not

4 compare bit vectors of all tweets with the sample bit vectors and import each tweet in one of the most similar class which is positive, negative or neutral.

6 Results and evaluation

We applied the scoring and clustering method on the initial dataset of 573,794 tweets that we collected. We compared the results of both scoring and clustering methods for single node and multiple nodes in terms of performance and correctness. Figure 2 shows the performances of the clusters and single node machines in terms of milliseconds. System properties for each cluster and single node are as follows.

6.1 Evaluation system properties

System properties for single node

- processor: Intel(R) Core (TM) i72670 QMCPU @ 2.20 GHz
- RAM: 7 GB
- operating system: Linux Enterprise Server 11 SP2.

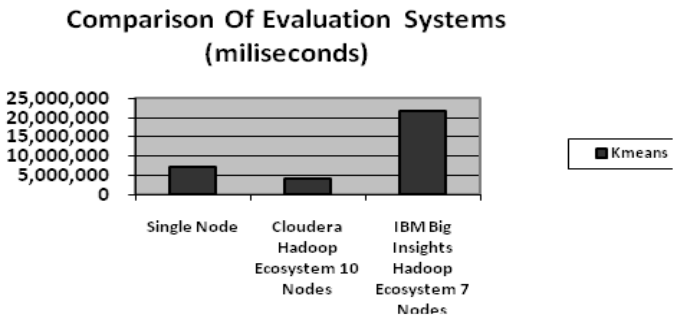
System properties for Cloudera Hadoop ecosystem 10 physical nodes

- processor: Intel®Core™i5-2400 CPU @ 3.10 GHz × 4
- RAM: 7 GB
- operating system: Ubuntu 12.04 LTS, desktop.

System properties for big insights Hadoop ecosystem 7 physical nodes

- processor: Dual-Core AMD Opteron (tm) Processor 2210 * 4
- RAM: 4 GB
- operating system: x86_64GNU/Linux CentOS release 6.5.

Figure 2 Comparison of evaluation systems



Scoring and K-means algorithms are implemented on single node, and distributed Hadoop-based architectures. Single machine environment is provided by IBM Virtual machine image, the main developments are implemented on the machine. Executable jar files are tested on the other platforms. Single node has all Hadoop services for development, tasks are submitted via ssh connections. Figure 2 demonstrates the comparison of evaluation systems in terms of milliseconds. IBM nodes have less memory and slower processor than Cloudera nodes; as a result, it is slower. Faster results are obtained with more memory for Hadoop architectures. Multi-node clusters are implementing jobs faster than single node. However, IBM cluster is running even slower than single node because of having fewer memories on each rack. The results show that Hadoop can run faster on multi-node cluster architecture as long as it has enough memory.

6.2 Comparison of evaluation times

Clustering algorithms can become much faster by using parallelisation methods because of having the appropriate structure for parallel computing. Many recent studies are trying to apply that to their study to design efficient clustering algorithms in parallel as Zhao and friends emphasised (Lee et al., 2012).

Furthermore, large scale dataset clustering can obtain better performance in parallel as Zhao and friends are experimented in another study (Zhao et al., 2009).

In this study we experienced that scoring algorithm is running much faster than K-means clustering algorithm. Table 3 shows the comparison of algorithms runtime in terms of milliseconds.

Table 3 Comparison of evaluation times (milliseconds)

	Single	*Cloudera 10 physical nodes*	*IBM big insights 7 physical nodes*
	Pos/Neg/Neut	*Pos/Neg/Neut*	*Pos/Neg/Neut*
Scoring algorithm	33,754	25,680	124,180
K-means clustering	7,091,750	4,214,549	21,760,335

The scoring algorithm is calculating the score for each tweet; each word within the tweet is searched in the dictionary which has 2,159 keywords.

K-means algorithm is looking for similarities of each tweet based on the bit vector for each tweet. The 100 bit vector which included the existence of the most used words within the tweet is composed for each tweet; bit is 1 if the word is included in tweet, 0 if not. Then, similarities of the tweets are calculated by comparing this particular vector with the other 100 bit most used words' vectors of each tweet.

K-means algorithm has more steps and layers than scoring; it is using Mahout for vector operations, vectorising tweets and vector comparisons, being the main reason for slower computation.

To get faster results, the scoring algorithm is a better choice. However, the lexicon study includes much more manual efforts than clustering. By selecting good and representative samples, K-means algorithm can also provide more accurate results. As long as we have enough processing power and parallel computing clusters like Hadoop

Ecosystem, execution time is not a big issue. As a result, k-means algorithm can be used for clustering to implement machine-based computation instead of human computing.

6.3 Comparison of algorithm performances

Figure 3 shows the results for the scoring algorithm in terms of positive, negative, and neutral tweet classification. Each algorithm can find the correct results up to a certain point. Detection precision is based on the keyword and sample selection respectively for scoring and k-means clustering. Scoring algorithm is using 2,159 words; increasing the number of keywords, adding more synonyms for keys, and applying more precise scores are increasing the quality of lexicon and giving better results.

Figure 3 Total_Number of Tweets_Scoring algorithm: 573,794, positive: 44,612, negative: 35,373, neutral: 493,809 (see online version for colours)

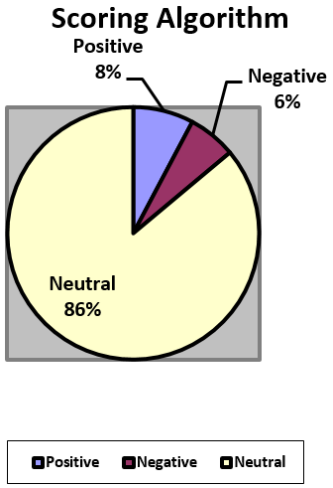


Figure 4 Total_Number of Tweets_k-means algorithm: 573,794, positive: 200,829, negative: 45,902, neutral: 327,63 (see online version for colours)

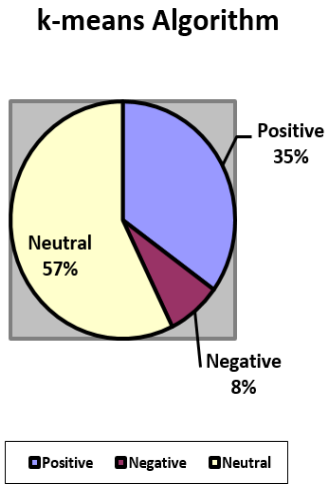


Figure 4 shows the results for k-means algorithm in terms of positive, negative, and neutral tweet clustering. K-means is using the 100 bit most used vector for each tweet. Selection of the most used words, increasing the number of samples in positive, negative, and neutral sample classes can enable us getting better results.

7 Conclusions and future work

We have experienced that Hadoop performance is improving as long as we have enough memory space. Cloudera cluster is faster than single node, but IBM is slower because of having less memory than single node.

Both scoring and k-means algorithms can provide correct results to classify tweets as positive, negative, and neutral. K-means algorithm is enumerating and vectorising each tweet and needs more processor and memory power. In both algorithms, correctness depends on the quality of the dictionary and the sample selections. Better dictionary and better samples give much better results.

To sum up, distributed data analytic systems can provide NoSQL data management efficiently. However, the systems should provide more basic features. For instance, fuzzy text search is not provided in MongoDB. We have to look for all possibilities for each word in the system. We are planning to use and develop better distributed data analytics systems and improve our opinion mining algorithm to get better results. Furthermore, other supervised learning, natural language processing, and content retrieval methods will be applied on the distributed text mining systems to develop large scale applications.

Acknowledgements

Thanks to;

- TOBB University of Economics and Technology (TOBB ETU) Distributed Data Analytics Research Laboratory, http://www.etu.edu.tr/?q=tr/bilgisayar-muhendisligi
- The Scientific and Technological Research Council of Turkey (TUBITAK), http://tubitak.gov.tr/
- HAVELSAN (Aeronautic and Defence Company) http://www.havelsan.com.tr/SirketProfili/default.aspx

for supporting project.

Furthermore, thanks to following companies for software grants.

- IBM, http://www-01.ibm.com/software/data/infosphere/biginsights/
- Cloudera, http://www.cloudera.com/content/cloudera/en/home.html
- Hadoop Ecosystem by Apache Open Source Community, http://hadoop.apache.org/
- MongoDB, https://www.mongodb.org/
- Ubuntu, http://www.ubuntu.com/.

References

Bermingham, A. and Smeaton, A. F. (2010) 'Classifying sentiment in microblogs: is brevity an advantage?', in *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, October, pp.1833–1836, ACM.

Go, A., Bhayani, R. and Huang, L. (2009) *Twitter Sentiment Classification Using Distant Supervision*, CS224N Project Report, pp.1–12, Stanford.

Khuc, V.N. et al. (2012) 'Towards building large-scale distributed systems for twitter sentiment analysis', *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, ACM.

Kouloumpis, E., Wilson, T. and Moore, J. (2011) 'Twitter sentiment analysis: the good the bad and the omg!', *ICWSM*, Vol. 11, pp.538–541.

Lee, K.H., Lee, Y.J., Choi, H., Chung, Y.D. and Moon, B. (2012) 'Parallel data processing with MapReduce: a survey', *ACM Sigmod Record*, Vol. 40, No. 4, pp.11–20.

Lin, J. and Kolcz, A. (2012) 'Large-scale machine learning at twitter', *Proceedings of the ACM SIGMOD International Conference on Management of Data*, ACM.

Martínez-Cámara, E., Martín-Valdivia, M.T., Ureña-López, L.A. and Montejo-Ráez, A.R. (2014) 'Sentiment analysis in twitter', *Natural Language Engineering*, Vol. 20, No. 01, pp.1–28.

O'Connor, B., Balasubramanyan, R., Routledge, B.R. and Smith, N.A. (2010) 'From tweets to polls: linking text sentiment to public opinion time series', *ICWSM*, Vol. 11, pp.122–129.

Özsert, C.M. and Özgür, A. (2013) 'Word polarity detection using a multilingual approach', *Computational Linguistics and Intelligent Text Processing*, Springer, Berlin, Heidelberg, pp.75–82.

Pak, A. and Paroubek, P. (2010) *Twitter as a Corpus for Sentiment Analysis and Opinion Mining,* in LREC, May.

Pang, B. and Lee, L. (2008) 'Opinion mining and sentiment analysis', *Foundations and Trends in Information Retrieval*, Vol. 2, Nos. 1–2, pp.1–135.

Thelwall, M., Buckley, K. and Paltoglou, G. (2012) 'Sentiment strength detection for the social web', *Journal of the American Society for Information Science and Technology*, Vol. 63, No. 1, pp.163–173.

Zhao, W., Ma, H. and He, Q. (2009) 'Parallel k-means clustering based on mapreduce', *Cloud Computing*, pp.674–679, Springer, Berlin Heidelberg.

Zhou, P., Lei, J. and Ye, W. (2011) 'Large-scale data sets clustering based on MapReduce and Hadoop', *Journal of Computational Information Systems*, Vol. 7, No. 16, pp.5956–5963.