Cogn Comput (2015) 7:254–262
DOI 10.1007/s12559-014-9310-z

Twitter Sentiment Analysis for Large-Scale Data: An Unsupervised Approach

Rafeeque Pandarachalil · Selvaraju Sendhilkumar · G. S. Mahalakshmi

Received: 4 March 2014 / Accepted: 25 October 2014 / Published online: 7 November 2014

Abstract Millions of tweets are generated each day on multifarious issues. Topical diversity in content demands domain-independent solutions for analysing twitter sentiments. Scalability is another issue when dealing with huge amount of tweets. This paper presents an unsupervised method for analysing tweet sentiments. Polarity of tweets is evaluated by using three sentiment lexicons—SenticNet, SentiWordNet and SentislangNet. SentislangNet is a sentiment lexicon built from SenticNet and SentiWordNet for slangs and acronyms. Experimental results show fairly good *F*-score. The method is implemented and tested in parallel python framework and is shown to scale well with large volume of data on multiple cores.

Keywords Sentiment analysis · Twitter · SentiWordNet · SenticNet · Parallel python

Introduction

The advent of online social media and mobile communication technologies has triggered a rapid increase in the flow of user generated content of various forms. People often express their reactions, fancies and predilections through social media by means of textual fragment of epigrammatic nature rather than writing long text. Further, unlike broadcasting media, the content generated in the online social media is immediate and unedited. Hence, many established industry players and social scientists have started to analyse this 'wisdom of crowd.'

Twitter is a micro-blogging service in which people share and discuss their thoughts and views in 140 characters without being constrained by space and time. Twitter sentiment analysis (TSA) has been utilised for several applications. A recent work [1] identified four major areas in which TSA was used. They are product reviews, movie reviews, political orientation extraction and stock market prediction. That study has given a detailed review on works carried out in those areas. In addition, generation of tweets on diverse issues invited researchers to think about domain-independent solutions to solve problems such as discovering latent communities based on sentiments [2], identifying same wavelength groups [3] and real-world behaviour analysis [4].

Millions of tweets are generated each day on multifarious issues. Topical diversity in content and linguistic flexibility in expression are two important challenges in analysing tweets. Traditional machine learning approaches fail to give the desired accuracy when applied to out-of-domain data. Further, many twitter sentiment analysers rely on various sentiment lexicons either to supply features to classifier models or to find sentiment scores. But the unigram features and the limited size of these lexicons fail to address the semantic features including negations. New generation opinion mining and NLP (natural language processing) techniques [5–7] entail resources to understand and extract concepts from texts. Moreover, tweets are rich in slangs and acronyms which themselves are capable of representing emotions or sentiments. Hence, TSA demands

R. Pandarachalil (✉)
Department of Computer Science and Engineering, Govt. College of Engineering Kannur, Kannur, India
e-mail: rafeeqpc@gcek.ac.in

S. Sendhilkumar
Department of Information Science & Technology, Anna University, Chennai, India
e-mail: thamaraikumar@cs.annauniv.edu

G. S. Mahalakshmi
Department of Computer Science & Engineering, Anna University, Chennai, India
e-mail: malakshmi@cs.annauniv.edu

a sentiment lexicon for slangs. Only few works discuss the scalability of twitter sentiment analysers in a distributed environment.

We propose an unsupervised and domain-independent approach by using the polarity scores from three lexical resources—SentiWordNet 3.0 [8], SenticNet 2 [9] and SentislangNet. SentiWordNet contains polarity scores of uni-grams. SenticNet 2.0 is a publicly available semantic resource which contains commonly used polarity concepts. Our method exploits SenticNet along with SentiWordNet to analyse twitter sentiments. We have also created a sentiment lexicon for slangs and acronyms called SentislangNet using SenticNet and SentiWordNet. The algorithm has been implemented in parallel python framework and shown to scale well with multiple cores for large volume of data. The paper is structured as follows. "Related Works" section presents a brief summary of recent works in TSA. "Sentiment Lexicons" section discusses three sentiment lexical resources used in this paper. In "Methods" section, we discuss the execution framework and algorithm. "Experiment Setup" and "Experiment Results" section describe the experimental setup and results. "Conclusion and Future Work" section concludes with future directions.

Related Works

Four different approaches have been used for analysing twitter sentiments: (1) lexicon based, (2) machine learning based, (3) hybrid and (4) by utilising social network parameters. Major works carried out using the lexicon and machine learning methods were discussed in [10, 11]. Accuracy of lexicon-based methods depends on the number of opinionated words in the dictionary. Domain-dependent lexicons become ineffective when applied to other domains. SentiWordNet is considered to be a good domain-independent lexical resource. Still, it contains only sentiment scores of uni-grams. Twitter-specific lexicon was implemented by Ghiassi et al. [12]. They observed that meaningful bi-grams and tri-grams can improve the accuracy of the classifier.

Machine learning approaches require a huge amount of labelled training data to achieve desired accuracy. In twitter domain, this is not always practical. Moreover, machine learning approaches trained with a particular set of data need not perform well for a data set from different domains. Recent work by Kontopoulos et al. [13] created domain ontologies of certain products to determine the polarity of subjects discussed in tweets. This needs not to be an easy solution for all domains. Attempts have also been made to use hybrid techniques by combining lexicon and machine learning methods [12, 14] and improved recall and *F*-measure. Social relations [15, 16] and follower graph [17] in twitter have also utilised to analyse the twitter sentiments. Although much work has been done on TSA, a few have proven the scalability in a distributed environment. Map-reduce framework along with HBase is used by Khuc et al. [18] to build large-scale distributed system for TSA. They also proposed a method for constructing sentiment lexicon automatically.

Two recent works have utilised SentiWordNet lexicon to analyse twitter sentiments. First work by Montejo-Ráez et al. [11] provided a weight for each synset by performing a random walk on WordNet graph. As mentioned earlier, the solution has considered only uni-grams. Second work [10] utilised concepts from DBpedia along with SentiWordNet scores. They made experiment with additional domain-specific and WordNet features and observed that using other sentiment lexicons with SentiWordNet is more useful.

Recent studies on NLP, Big social data analysis and opinion mining [5, 19, 20] explore the importance of concept-level analysis of online social media data. They investigate the importance of machine processable knowledge bases to understand the concepts and common senses from the natural language. We used SenticNet that is a machine-readable semantic and affective resource for analysing sentiments.

Sentiment Lexicons

SentiWordNet

SentiWordNet [8] is a domain-independent public lexical resource used for sentiment analysis. SentiWordNet is derived from WordNet by assigning positive, negative and objective scores to all WordNet synsets. Each sentiment score is a real value in the interval [0.0,1]. Recent version (SentiWordNet 3.0) is enhanced to improve the sentiment score by performing a random walk on WordNet 3.0. The number of terms in the SentiWordNet is very high compared with other lexicons. There are around 117,659 sentisynsets available in SentiWordNet. But SentiWordNet contains only uni-grams and hence provides sentiment score only at syntactic level. For instance, SentiWordNet fails to provide sentiment scores for terms such as 'not well,' 'rainy day' and 'nothing better,' which are very common in tweets.

SenticNet

SenticNet [9, 21] is a public lexical resource made by means of sentic computing [22, 23]. SenticNet 2 consists 14,244 concepts and its associated sentics and semantics.

Table 1 Frequency distribution of *n*grams in SenticNet

*N*grams	Frequency
Four-grams	40
Tri-grams	682
Bi-grams	7,037
Uni-grams	6,482
Others	3

Table 2 Polarity score of sample terms from SenticNet

Term	Polarity score
Not well	−0.539
Nothing better	−0.58
Don't like	−0.546
Not feel well	−0.144
Person nothing better	−0.062
Not smell bad	0.065

Polarity score for each concept is a value in the interval [−1,1]. Unlike SentiWordNet, polarity score of each concept has only a single value either positive or negative. The frequency distribution of *n*grams in the SenticNet is shown in Table 1. SenticNet has been encoded as machine-readable RDF/XML format. We created a SenticNet dictionary by Web scraping the XML file. A number of terms in SenticNet are less when compared to SentiWordNet. But the presence of semantic concepts offer sentiment score of certain terms commonly present in tweets. Table 2 shows sentiment scores of sample terms present in the SenticNet.

SentislangNet

Tweets are rich in slangs, abbreviations and emoticons. Sometimes they themselves decide the sentiments of tweets. Moreover, the slangs used in the social media are evolving. Most of the approaches have not considered the sentiment score of slangs while analysing twitter sentiments. We created a sentiment lexicon for slangs called SentislangNet by Web scraping slangs from Web. The Web site[1] provides slangs, acronyms, emoticons and their definitions (expansions). More than 7,200 slangs were present while scraping the Web. We removed the standard abbreviations such as 'WWW' and 'PhD' since found no significance in the sentiment score calculation. Sentiment score of each slang is evaluated by running the sentiment analysis algorithm on definitions using SenticNet and SentiWordNet and later removed or corrected the noise terms manually. Table 3 shows the sentiment score of few slangs.

Table 3 Polarity score of sample terms from SentislangNet

Slang	Expansion	Polarity score
4EVER	Forever	0.093
GUUD	Good	0.883
XLNT	Excellent	0.496
GR8	Great	0.857
BLEH	Boredom	−0.235
DWBH	Don't worry be happy	0.348

Methods

TSA has two phases: (1) preprocessing phase and (2) sentiment analysis phase. In the first phase, the tweets are normalised and transformed into uni-grams (four-grams, tri-grams, bi-grams and uni-grams). The second phase determines the sentiment polarity of each tweet from the generated uni-grams.

Preprocessing

It is often necessary to normalise the text for any NLP tasks. Since the tweets are often represented in cryptic and informal way, systematic preprocessing of tweets is required to enhance the accuracy of sentiment analyser. The tweets are preprocessed to extract all valid *n*grams that have immense significance to determine the polarity.

1. *Removing links/URL and Hash tags* Tweet may contain URL, hash tags and words start with '@' character. We removed these entities since found no significance in our scoring approach.
2. *Replacing word with contractions* Contractions such as 'didn't', 'ain't' 'couldn't' are common in tweets. Sentiment lexicons are usually free from these type of contractions. Further, SenticNet contains several bi-grams such as 'not well', 'not understand' and 'not lose,' and these types of bi-grams often determine the polarity of tweets. These contractions are replaced with their expanded forms.

```
I can't understand you => I can not understand you
```

3. *Sentiment aware tokenizing* Emoticons, slangs and acronyms present in the tweets are strong indicators of emotion or sentiments. Hence, a modified version of twitter-aware tokeizer by Christopher Potts[2] is used to capture all emoticons and to preserve the case of slangs

[1] www.internetslang.com.

[2] http://sentiment.christopherpotts.net.

and acronyms. SentislangNet is used to ensure the validity of slangs and acronyms.

4. *Elongation replacer* People often use elongation like 'looooooooove' to emphasise words. Elongation can be at the beginning ('oooooh'), end ('toooooo') or in between ('loooove'). Elongation replacer recursively removes repeating characters until no more characters are removed or recognised by WordNet. A WordNet lookup before removing repeating characters is done to ensure unnecessary removal of characters. For instance, removing repeating characters from 'oo-oooooooh' without WordNet lookup gives 'oh.' But the word 'ooh' is a meaningful word.

```
oooooooh what a coooooool breeze => ooh what a cool breeze
```

5. *WordNet Lemmatizing* Wordnet lemmatizer is used to get a valid meaningful root word. Each word (except slang/abbreviation) is lemmatized after tokenizing.
6. *Explicit negation handling* We used an antonym replacer using WordNet to replace word preceded by 'not,' 'never,' etc. Word is replaced if an unambiguous antonym is present in the WordNet. Furthermore, polarity score of negated bi-gram phrases available in SenticNet was used during sentiment analysis phase. This enhanced coverage of more number of negated bi-gram phrases presents in the tweets.

```
I will be never happy => I will be unhappy
```

Finally, *n*grams (four-grams, tri-grams, bi-grams, uni-grams) are generated from the normalised tokens.

Sentiment Analysis

Sentiment score of each tweet is evaluated by *SENT_SCORE* procedure. A list of *n*grams [four-grams (F), tri-grams(T), bi-grams (B), uni-grams (U)] produced by the preprocessing module is the input to the algorithm. Part 1 (line 1–24) searches SenticNet to find the polarity score of the four-grams, tri-grams and bi-grams present in the tweet. As mentioned earlier, SenticNet has only one polarity score (positive or negative) for each term or concept. But SentiWordNet has a positive as well as negative score for each term. Consequently, for each positive (negative) term in SenticNet, we assume a zero negative (positive) score. *Polscore* function returns the polarity score from the respective sentiment lexicon and increment the *term*. Polarity score of each term is counted only if the term is not contained in the previously considered *n*grams. The function *notContained* implements the same.

Algorithm 1 Sentiment Score Calculation - Part 1

```
1: procedure SENT_SCORE(F, T, B, U)
2:     fgn, tgn, bgn ← NULL
3:     term ← 0, sent_score ← 0
4:     for all f ∈ F do
5:         if f ∈ Senticnet then
6:             pos, neg, term ← Polscore(SenticNet, f, term)
7:             fgn ← Append(f, fgn)
8:             sent_score ← sent_score + pos − neg
9:         end if
10:    end for
11:    for all t ∈ T do
12:        if (t ∈ Senticnet) & (notContained(t, fgn)) then
13:            pos, neg, term ← Polscore(SenticNet, t, term)
14:            tgn ← Append(t, tgn)
15:            sent_score ← sent_score + pos − neg
16:        end if
17:    end for
18:    for all b ∈ B do
19:        if (b ∈ Senticnet) & (notContained(t, (fgn, tgn)) then
20:            pos, neg, term ← Polscore(SenticNet, b, term)
21:            tgn ← Append(b, bgn)
22:            sent_score ← sent_score + pos − neg
23:        end if
24:    end for
```

Part 2 (line 25–43) searches SentiWordNet, SenticNet and SentislangNet for the uni-grams not contained in the previous *n*grams. The *disambWordsenses* function determines the correct synset associated with each term in SentiWordNet. If the term is not present, lookup is made in the SenticNet and SentislangNet. Overall sentiment score is calculated from Eq. (1).

$$\text{sent\_score} = \frac{\sum_{\forall t \in \text{term}} \text{pos}_t - \text{neg}_t}{|\text{term}|} \quad (1)$$

where pos and neg are the positive and negative score of each term, respectively, and term represents the total number of terms or phrases considered for the overall sentiment score. The overall score will be a real value in the range [−1,1]. Value greater than zero denotes a positive sentiment and less than zero a negative sentiment.

Algorithm 2 Sentiment Score Calculation - Part 2

```
25:   for all u ∈ U do
26:     if (notContained(t, (fgn, tgn, bgn)) then
27:       if (u ∈ Sentiwordnet) then
28:         disamb_synset ← disambWordsenses(u, wordnet)
29:         pos, neg, term ← Polscore(SentiWordNet, u, term)
30:       else
31:         if (u ∈ Senticnet) then
32:           pos, neg, term ← Polscore(SenticNet, u, term)
33:         else
34:           if u ∈ Sentislangnet then
35:             pos, neg, term ← Polscore(Sentislangnet, u, term)
36:           end if
37:         end if
38:       end if
39:     end if
40:     sent_score ← sent_score + pos − neg
41:   end for
42:   NetScore ← sent_score/term
43: end procedure
```

Parallel Python

Tweets generated in each moment is huge. Large volume of tweets demands a scalable solution for TSA. Parallel python[3] is an open source module which provides parallel execution of python code on multiple processor (core) as well as cluster of nodes. The number of worker processes decides the number of active parallel execution cores.

In our system, preprocessor and sentiment analysis modules are implemented using parallel python. Figure 1 shows the parallel python framework for sentiment analysis module. Initially, a jobserver is created with worker processes. Number of cores or processors decides the number of worker processes. By default, the number of worker process will be equal to the number of the cores or processors in the system. Each tweet is transformed into a list of *n*grams (four-grams, tri-grams, bi-grams and uni-grams) by the preprocessor module. Here, each job is an instance of SENT_SCORE function to be executed on list of *n*grams. Job-based parallelization is achieved through dynamically allocating processor (core) to each job. Call back function simply collects the polarity of each tweet returned by the worker processes.

Example

Following example illustrates the algorithm. The preprocessing module produces a list of unigrams (four-grams, tri-grams, bigrams and uni-grams). The slang 'XLNT' that is an abbreviation of the term 'excellent' and the polarity score are taken from SentislangNet.

```
The XLNT looong arc of science:  Englert & Higgs predicted
in 1964 that Higgs boson existed.Today they get Nobel. Wish
live long happy life.
```

First, the algorithm searches the SenticNet for the four-grams present in the list. Polarity score of the only one valid four-gram, 'live long happy life,' in the tweet is evaluated from SenticNet.

```
['live long happy life']    ->    pos[0.811]  ,  neg[0]
```

[3] www.parallelpython.com.

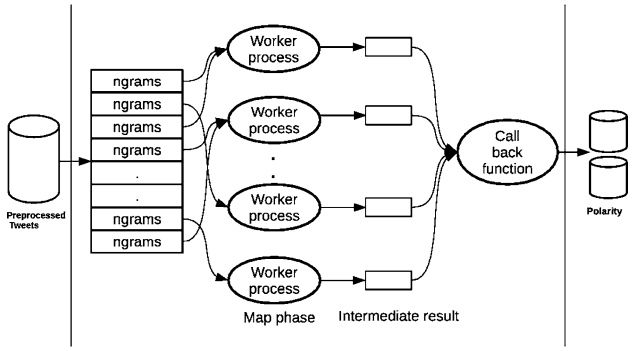

Fig. 1 Parallel python framework for sentiment analysis phase

Similarly, the search has been carried out in SenticNet for tri-grams and bi-grams. The tri-gram 'live long happy' and the bi-gram 'live long' are present in the Senticnet. But their polarity is not considered since they contained in the previous four-gram.

```
['live long happy'] -> #contained in ['live long happy life'].
['live long'] -> #contained in ['live long happy life'].
```

```
the            ->    pos[0.082] neg[0]     #from Senticnet
skill.n.02     ->    pos[0.125] neg[0.0]   #from Sentiwordnet
nowadays.r.01  ->    pos[0.25]  neg[0.0]   #from Sentiwordnet
get.v.01       ->    pos[0.125] neg[0.0]   #from Sentiwordnet
wish.v.01      ->    pos[0.125] neg[0.375] #from Sentiwordnet
XLNT           ->    pos[0.496] neg[0]     #from Sentislangnet
```

As mentioned in the algorithm, polarity score for unigrams and slangs is calculated from SentiWordNet, SenticNet and SentislangNet, respectively. A total number of terms are seven, and the overall sentiment score is 0.2341.

Experiment Setup

Parallel python framework is used to provide a scalable and distributed solution to the whole problem. We created a cluster of eight nodes, and each node has eight cores. Effectively, the work load can be distributed across 64 cores. Two labelled data sets[4] are used for analysing the performance of sentiment analyser. The data sets are created by the author using the method of distant supervision proposed by Go et al. [25]. Statistics about the data sets are shown in Table 4.

Table 4 Details about the twitter corpus

	Number of tweets	Positive	Negative
Data set#1	19,332	9,666	9,666
Data set#2	200,000	100,000	100,000

Baseline Methods

Our approach is compared with two unsupervised methods based on SentiWordNet. Few recent approaches [10, 11, 24] have used SentiWordNet lexical resource to implement sentiment analysis methods. Since their data sets are not public, we implemented and tested two methods to compare the performance.

- First method (SWN) is the basic method using the polarity score from SentiWordNet. Overall polarity score of the tweet is calculated from Eq. (1).
- The second method (RW-SWN) is the implementation of the approach by Montejo-Ráez et al. [11]. They used PageRank value (weight of the synset value after the random walk process over WordNet). Precompiled personalised PageRank vectors for all WordNet lemmas are available online.[5] Overall polarity score is calculated from Eq. (2).

$$P = \frac{\sum_{\forall s \epsilon t} rw_s.(\text{swn}_s^+ - \text{swn}_s^-)}{|t|} \quad (2)$$

where s is the synset in the tweet and rw_s weight of the sysnset s (PageRank value) swn_s^+ and swn_s^-s are positive and negative scores for the synsets retrieved from SentiWordNet.

Experiment Results

In this section, we first examine the performance of our sentiment analyser by comparing with other methods. We also compare the computation time taken by the algorithms in the parallel python environment.

Performance of the Sentiment Analyser

The performance of the sentiment analyser on two data sets is shown in Table 5. Results show that our unsupervised method (SN-SWN) achieved reasonably good recall and F-measure and outperforms other baseline methods (SWN, RW-SWN). We found that using other sentiment lexicons (SenticNet, SentislangNet) along with

[4] https://github.com/ravikiranj/twitter-sentiment-analyzer.

[5] http://ixa2.si.ehu.es/ukb/.

Table 5 Performance comparison of sentiment analysis methods

Methods	Corpus	Precision	Recall	*F*-score
Performance				
SWN	Data set#1	59.8	56.24	57.97
	Data set#2	58.87	54.87	56.8
RW-SWN	Data set#1	63.7	63.11	63.42
	Data set#2	60.5	59.89	60.09
SN-SWN	Data set#1	64.64	74.35	69.16
	Data set#2	62.9	72.73	67.46

SentiWordNet has improved the accuracy of sentiment analysis. Moreover, SenticNet is a growing and machine-readable public semantic resource [26], and the number of concepts has been increased from the old version to more than 14,000. In future, the addition of new concepts to SenticNet may able to enhance the accuracy further. We have noticed some positive (negative) terms in SenticNet with negative (positive) polarity scores. For instance, polarity score of 'like movie' is −0.053 in senticNet. Updation of such noise terms can also improve the performance of the algorithm.

Computation Time

As mentioned earlier, only few works have proved the capability to run their methods on distributed environment. We compared the running time of the methods by executing different methods in the above-mentioned experimental setup for one lakh tweets. Figure 2 shows the snapshot of the execution statistics of three methods (SN-SWN, SWN, RW-SWN). Here, the number of jobs signifies the number of tweets dynamically alloted to worker process (cores). Time per job represents the average execution time needed for sentiment analysis. The percentage of jobs executed in each node signifies that the distribution of jobs in each node is almost equal. Time elapsed since server creation is the total time taken for TSA which includes the time for I/O activity. Time elapsed since the server creation by different methods is shown in Fig. 3. Time taken by the method SN-SWN is less when compared to other methods.

The RW-SWN method [11] uses a PageRank value for each sysnset in the SentiWordNet. The method searches the precompiled personalised PageRank vectors to obtain the same. The precompiled personalised Pagerank vector is a

Fig. 2 Snapshot of the execution statistics of three methods (SN-SWN, SWN, RW-SWN) in a cluster of 8 nodes

```
Job execution statistics:
job count | % of all jobs | job time sum   | time per job | job server
    12813 |         12.81 |     18853.6290 |     1.471445 | 10.1.10.96:60000
    12493 |         12.49 |     18863.0999 |     1.509894 | 10.1.10.83:60000
    12532 |         12.53 |     18845.9589 |     1.503827 | 10.1.10.88:60000
    12356 |         12.36 |     18873.6035 |     1.527485 | 10.1.10.86:60000
    12363 |         12.36 |     18862.0059 |     1.525682 | 10.1.10.97:60000
    12427 |         12.43 |     18858.8636 |     1.517572 | 10.1.10.115:60000
    12496 |         12.50 |     18852.2393 |     1.508662 | 10.1.10.84:60000
    12520 |         12.52 |     18851.1003 |     1.505679 | 10.1.10.98:60000
Time elapsed since server creation 2375.57363391
--------------------------------------------------------------------------------
Job execution statistics:
job count | % of all jobs | job time sum   | time per job | job server
    12738 |         12.74 |     20819.3448 |     1.634428 | 10.1.10.96:60000
    12564 |         12.56 |     20816.9741 |     1.656875 | 10.1.10.83:60000
    12387 |         12.39 |     20826.6888 |     1.681334 | 10.1.10.88:60000
    12427 |         12.43 |     20826.8803 |     1.675938 | 10.1.10.86:60000
    12500 |         12.50 |     20820.8270 |     1.665666 | 10.1.10.97:60000
    12328 |         12.33 |     20827.7115 |     1.689464 | 10.1.10.115:60000
    12537 |         12.54 |     20819.7755 |     1.660666 | 10.1.10.84:60000
    12519 |         12.52 |     20836.7308 |     1.664409 | 10.1.10.98:60000
Time elapsed since server creation 2621.2859509
--------------------------------------------------------------------------------
Job execution statistics:
job count | % of all jobs | job time sum   | time per job | job server
    12598 |         12.60 |     20981.0295 |     1.665425 | 10.1.10.96:60000
    12484 |         12.48 |     20988.2070 |     1.681209 | 10.1.10.83:60000
    12684 |         12.68 |     20988.4626 |     1.654720 | 10.1.10.88:60000
    12158 |         12.16 |     20993.1743 |     1.726696 | 10.1.10.86:60000
    12486 |         12.49 |     20984.9543 |     1.680679 | 10.1.10.97:60000
    12730 |         12.73 |     21492.4836 |     1.688333 | 10.1.10.84:60000
    12388 |         12.39 |     20987.0329 |     1.694142 | 10.1.10.115:60000
    12472 |         12.47 |     20986.1185 |     1.682659 | 10.1.10.98:60000
Time elapsed since server creation 2702.071136
```

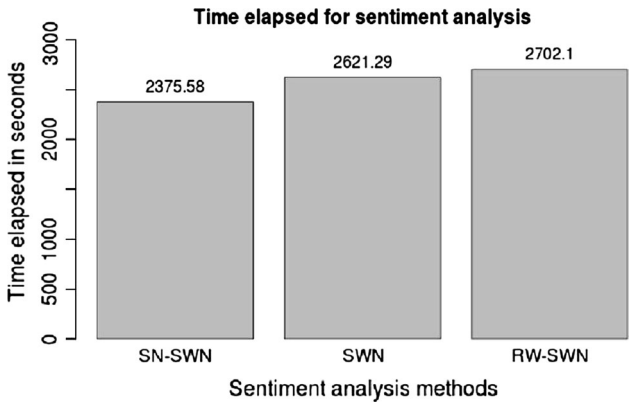

Fig. 3 Comparison of running time for one lakh tweets

very large collection of numerous files distributed across numerous directories and subdirectories. The size of the uncompressed version of the same is around 3.3 GB. We observed that the search space requirement and hence the computation time are high compared with our approach (SN-SWN). The computation time for the other baseline method (SWN) using SentiWordNet is comparatively high since it needs to search for the correct synset before finding the sentiment score. Moreover, the distributed solution provides a scalable approach for large volume of data.

Conclusion and Future Work

Domain-independent solutions for analysing twitter sentiments are required for several social media applications. In this paper, we proposed an unsupervised and distributed solution for analysing twitter sentiments using three domain-independent sentiment lexical resources (SentiWordNet, SenticNet and SentislangNet). We found that SenticNet has contributed polarity scores of several *n*grams commonly present in tweets and hence enhanced the accuracy of sentiment analyser. Including more concepts in SenticNet may increase the accuracy further. Moreover, our method has been shown to scale with large volume of data by executing in a cluster of nodes. We used only the polarity scores from SenticNet to analyse twitter sentiments. Utilising other sentics (affective information) associated with each concept will be a promising future direction for twitter sentiment analysis.

References

1. Mostafa MM. More than words: social networks text mining for consumer brand sentiments. Expert Syst Appl. 2013;40(10):4241–51.
2. Sachan M, Contractor D, Faruquie T, Subramaniam LV. Using content and interactions for discovering communities in social networks. In: Proceedings of the 21st international conference on World Wide Web—WWW '12. New York: ACM Press; 2012. p. 331–40.
3. Rafeeque P, Sendhilkumar S. Identifying same wavelength groups from twitter: a sentiment based approach. In: Proceedings of the 5th Asian conference on Intelligent Information and Database Systems (ACIIDS'13). LNCS Vol. 7803. Berlin: Springer; 2013. p. 70–7.
4. Abbasi M, Chai S, Liu H, Sagoo K. Real-world behavior analysis through a social media lens. In: 5th international conference on social computing, behavioral–cultural modeling and prediction (SBP 2012), USA; 2012. p. 18–26.
5. Cambria E, Rajagopal D, Olsher D, Das D. Big social data analysis. In: Big data computing. London: Chapman and Hall/CRC; 2013. p. 401–14.
6. Wang QF, Cambria E, Liu CL, Hussain A. Common sense knowledge for handwritten Chinese text recognition. Cogn Comput. 2013;5(2):234–42.
7. Cambria E, Mazzocco T, Hussain A. Application of multi-dimensional scaling and artificial neural networks for biologically inspired opinion mining. Biol Inspir Cogn Archit. 2013;4:41–53.
8. Baccianella S, Esuli A, Sebastiani F. SentiWordNet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In: Proceedings of the seventh international conference on language resources and evaluation (LREC'10); 2010, p. 2200–04.
9. Cambria E, Havasi C, Hussain A. SenticNet 2: a semantic and affective resource for opinion mining and sentiment analysis. In: FLAIRS conference; 2012. p. 202–7.
10. Hamdan H, Béchet F, Bellot P. Experiments with DBpedia, WordNet and SentiWordNet as resources for sentiment analysis in micro-blogging. In: Seventh international workshop on semantic evaluation (SemEval 2013), vol. 2; 2013. p. 455–59.
11. Montejo-Ráez A, Martínez-Cámara E, Martín-Valdivia MT, Ureña López LA. Ranked WordNet graph for sentiment polarity classification in Twitter. Comput Speech Lang. 2014;28(1):93–107.
12. Ghiassi M, Skinner J, Zimbra D. Twitter brand sentiment analysis: a hybrid system using n-gram analysis and dynamic artificial neural network. Expert Syst Appl. 2013;40(16):6266–82.
13. Kontopoulos E, Berberidis C, Dergiades T, Bassiliades N. Ontology-based sentiment analysis of twitter posts. Expert Syst Appl. 2013;40(10):4065–74.
14. Zhang L, Ghosh R, Dekhil M, Hsu M, Liu B. Combining Lexicon-based and learning-based methods for twitter sentiment analysis. Technical Report HPL-2011-89HP; 2011.
15. Tan C, Lee L, Tang J, Jiang L, Zhou M, Li P. User-level sentiment analysis incorporating social networks. In: ACM international conference on knowledge and data engineering (KDD'11), California, USA; 2011. p. 1397–1405.
16. Hu X, Tang L, Tang J, Liu H. Exploiting social relations for sentiment analysis in microblogging. In: Proceedings of the sixth ACM international conference on Web search and data mining. WSDM '13. New York, NY: ACM; 2013. p. 537–46.
17. Speriosu M, Sudan N, Upadhyay S, Baldridge J. Twitter polarity classification with label propagation over lexical links and the follower graph. In: Proceedings of the first workshop on unsupervised learning in NLP. EMNLP '11. Stroudsburg, PA: Association for Computational Linguistics; 2011. p. 53–63.
18. Khuc VN, Shivade C, Ramnath R, Ramanathan J. Towards building large-scale distributed systems for twitter sentiment analysis. In: Proceedings of the 27th annual ACM symposium on applied computing, SAC '12. New York, USA; 2012. p. 459–64.

19. Cambria E, White B. Jumping NLP curves: a review of natural language processing research. IEEE Comput Intell Mag. 2014;9(2):1–28.
20. Cambria E, Schuller Y, Xia H. New avenues in opinion mining and sentiment analysis. IEEE Intell Syst. 2013;28(2):15–21.
21. Cambria E, Olsher D, Rajagopal D. SenticNet 3: a common and common-sense knowledge base for cognition-driven sentiment analysis. In: AAAI, Quebec City; 2014. p. 1515–21.
22. Cambria E, Hussain A. Sentic computing: techniques, tools and applications, Springer briefs in cognitive computation. Berlin: Springer; 2012.
23. Cambria E, Hussain A. Sentic PROMs: application of sentic computing to the development of a novel unified framework for measuring health-care quality. Expert Syst Appl. 2012;39(12):10533–43.
24. Chamlerwat W, Bhattarakosol P, Rungkasiri T, Haruechaiyasak C. Discovering consumer insight from twitter via sentiment analysis. J Univ Comput Sci. 2012;28(2):15–21.
25. Go A, Bhayani R, Huang L. Twitter sentiment classification using distant supervision. Technical Report, Stanford; 2009.
26. Poria S, Gelbukh A, Hussain A, Das D, Bandyopadhyay S. Enhanced SenticNet with affective labels for concept-based opinion mining. IEEE Intell Syst J. 2013;28(2):31–8.