Subjectivity Detection in Nuclear Energy Tweets

Ranjan Satapathy[1], Iti Chaturvedi[1], Erik Cambria[1], Shirley S. Ho[2], Jin Cheon Na[2]

[1] Nanyang Technological University,
School of Computer Science and Engineering,
Singapore

[2] Nanyang Technological University,
Wee Kim Wee School of Communication and Information,
Singapore

{ranjan002}@e.ntu.edu.sg, {iti, cambria, tsyho,tjcna}@ntu.edu.sg

Abstract. The subjectivity detection is an important binary classification task that aims at distinguishing natural language texts as opinionated (positive or negative) and non-opinionated (neutral). In this paper, we develop and apply recent subjectivity detection techniques to determine subjective and objective tweets towards the hot topic of nuclear energy. This will further help us to detect the presence or absence of social media bias towards Nuclear Energy. In particular, significant network motifs of words and concepts were learned in dynamic Gaussian Bayesian networks, while using Twitter as a source of information. We use reinforcement learning to update each weight based on a probabilistic reward function over all the weights and, hence, to regularize the sentence model. The proposed framework opens new avenues in helping government agencies manage online public opinion to decide and act according to the need of the hour.

Keywords. Subjectivity detection, nuclear energy tweets.

1 Introduction

With the increasing usage of social media like Twitter, there is an increasing opinion available on the social media. Mining such opinions has become a greater interest for the scientific and industrial community. In present times, sentiment analysis has found huge number of applications. For example, it can be used as a sub-module for systems like customer relationship management (CRM) and recommendation engines.

Sentiment analysis [3] is a branch of affective computing research [18] that aims to classify text (but sometimes also audio and video [21]) into positive, negative, and neutral. Sentiment analysis systems can be largely categorized as knowledge-based [5], statistics-based [16], and hybrid [4]. While most works approach tries solving sentiment analysis as a simple categorization problem, it is actually a "suitcase" of research problem consisting of many NLP sub-problems to be solved, including named entity recognition [13], word polarity disambiguation [25], temporal tagging [27], personality recognition [14], sarcasm detection [20], and subjectivity detection.

Subjectivity detection, in particular, has become important in recent years, as the exponential growth of social media data is generating a greater and greater need for filtering out objective data (facts) from subjective data (opinions). Subjectivity detection can be more challenging than polarity detection (positive-versus-negative classification) but it has been under-investigated due to the assumption that most social data were considered subjective. This research gap comes as a surprise, given that the Web has transformed into a dynamic and evolving network enabling users to be the media themselves. Bearing in mind the current pace of these social data generation, filtering out neutral data is taking priority over processing opinionated data.

Subjectivity detection can be a useful tool for governmental agencies to find out which topics are particularly heated or controversial and, hence, act accordingly to prevent popular discontent. Similarly, businesses can use sentiment analysis in social communication for spam detection, troll filtering, social media monitoring, and business intelligence. On the other hand, the development and automated up-keeping

of opinion gathering websites, can be undertaken by opinion mining techniques, in which opinions are continuously gathered in real time and from the Internet and include broader topics like national issues and not limited just to reviews of products.

Within such subjective topics, however, opinions can be expressed on multiple opinion targets. For this reason, aspect extraction [19] comes as an important subtask of sentiment analysis which consists of detecting aspects in opinionated text, i.e., in identifying the specific aspects of a product or service which the consumer is discussing about. For example: "nuclear energy is good for the environment but safety is a great concern" is a subjective tweet about nuclear energy with "environment" and "safety" as two aspects.

This paper focuses on the application of subjectivity detection on tweets about nuclear energy. In addition to what has already been mentioned, there are three issues which this research aims to tackle, namely:

1. A lot of factual or non-opinionated information needs to be filtered out;

2. Opinions are most times on different aspects of the similar product or service rather than on the whole item;

3. Reviewers tend to praise aspects of the same product or service and criticize others.

Different strata of opinion mining have been proposed, each one having its own merits and demerits. In this paper, we apply an ensemble of deep learning and linguistics [6] to tackle the problem of subjectivity detection on tweets[1]. The subjectivity detection task ensures that factual information possessing neutral polarity is filtered out and only opinionated information is further passed on to the aspect extraction and polarity classifier. It enables the correct distribution of polarity among the different features of the opinion target (instead of having one unique, averaged polarity assigned to it).

The rest of this paper has been organized as follows: Section 2 reviews the related work; Section 3 explains the proposed architecture; Section 4 provides an evaluation of the architecture; finally, Section 5 illustrates the conclusion and future work.

---

[1]http://dev.twitter.com

2 Related Work

Subjective extraction yields comparable polarity results as full text classification [2] even after reducing the review data by 60%. Previous efforts used general subjectivity clues to produce training data from un-annotated text [23]. Recently, in [15] the authors improved accuracy on subjectivity detection for Twitter data by using hand-crafted features and pre-computed word vectors. However, they considered two classifiers, the first for removing neutral sentences, and another to classify subjective sentences into 'positive' or 'negative'. This approach cannot be applied to new languages and domains. In contrast, we consider deep convolutional neural network (CNNs) that can automatically learn features and hand-crafted features are only used to initialize the weights in the model.

Similarly, in [1] the authors predicted sentiment in finance tweets by cascading two classifiers for subjectivity and sentiment. In finance tweets, the position of the author (e.g., investor or company) is critical, hence they consider Bayes rule to cluster tweets in an unsupervised manner. In order to reduce the time complexity, they divided the training data into multiple sub-sets and then combine the features using a support vector machine (SVM). However, this can lead to loss of information, instead deep learning is a semi-supervised model where each hidden neuron is independent from others and, hence, can be trained in a parallel manner.

In deep neural networks, a word's meaning is simply an indicator that aids to classify document like entities. Similar words appear consecutively when represented in a d dimensional vector-representation.

Vectorising makes it easier to cluster them on the basis of their similarities. The features include suffix, prefix, distance from verbs in the sentence, in order to know the corresponding position of verbs in semantic role labelling. However, there is a corresponding vector illustration in the training as a d dimensional space for each feature.

Recently, CNNs have become increasingly popular for subjectivity detection. In particular, [10] used recurrent CNNs. These show significant accuracy on datasets where the inter-dependency between sentences is high. In [6], authors extended the extreme learning machine (ELM) paradigm to a novel framework that exploits the features of both Bayesian networks and fuzzy recurrent neural networks to perform subjectivity detection.

The order of sentences preceding the one at hand, results in a sequence of sentences. However, due to overfitting of the model, hence in [6] the authors considered fusion of Spanish and English tweets using

deep CNN. In this paper, we extend their work to multilingual and we train the model using reinforcement learning. We evaluate our method on tweets related to nuclear energy. Figure 1 depicts the ratio of subjective vs objective tweets collected from Twitter. It shows that Twitter is a potential source of data for opinion mining. This paper summarizes the significance and contributions of the research work as:

— We introduce a reinforced deep CNN (RDCNN) capable of classifying sentences as subjective (positive or negative) or objective (neutral) in multilingual datasets.

— We propose a new regularization for neural networks based on reinforcement learning.

— In order to validate the model we manually label 7,700 nuclear multi-lingual tweets into positive, neutral and negative classes.
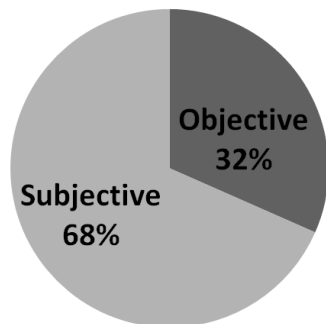
Subjectivity Detection in Nuclear Tweets

Fig. 1. Subjectivity detection on nuclear energy tweets

3 Reinforced Deep Convolutional Neural Networks

In this section, we consider the use of reinforcement learning to regularize learning in a deep CNN. The resulting framework is referred to as a RDCNN. We begin with the description of reinforcement in a single layer neural network and then detail its integration into the complete framework.

3.1 Deep Reinforcement Learning

Point-wise probability reinforcements (PPR) are commonly used to robustify neural networks. Most robust neural network inference methods consists of assigning a weight to each instance in order to reduce the influence of outliers [7]. The reinforced maximum likelihood model maximizes :

$$p(y = j|\mathbf{x}) = \frac{e^{\mathbf{x}^T w_j}}{\sum_{k=1}^{K} e^{\mathbf{x}^T w_k}},$$
$$-\log p(y|\mathbf{x}, \theta) = -\log p(y|\mathbf{x}, \theta) - \alpha \Omega(\mathbf{r}), \quad (1)$$

where y is the output class label, $\mathbf{r}$ is the vector or PPRs, α is a reinforcement meta-parameter, and Ω is a penalization function. A sparse L1 penalization function is obtained by :

$$\Omega(\mathbf{r}) = \sum_{i=1}^{n} r_i, \quad (2)$$

which shrinks PPRs towards zero with $r_i \geq 0$ L1-regularized PPRs are given by the closed-form expression:

$$r_i = \max\left(\frac{1}{\alpha} - w_k^{old}, 0\right), \quad (3)$$

where w_k^{old} is the current estimate of the model parameters.

3.2 RDCNN Framework

The data pre-processing includes taking away top 50 stop-words and punctuation marks from the sentences. Subsequently, a POS tagger was used to determine the part-of-speech for each word in a sentence. Around 8,000 subjective clues were identified, which were compiled using both manual and automated procedures on both un-annotated and annotated data. Each clue is comprised of word and its corresponding part-of-speech.

The subjective and objective sentences are taken from MPQA corpus. The count of each clue was computed in both the sentences. Here, the top 50 clue with maximum count in the subjective sentences were considered.

As described in Algorithm 1 line number 5-9, we compute the prior probability of each parent word set a_i for child word x_i called "word-motif". Sentences containing all words in a word-motif' with high probability are selected. Lastly, the CNN is jointly pre-trained with both subjective and objective sentences that contain high probability "word-motifs".

Algorithm 1 Reinforced Deep Learning

1

1: Input 1:   Input sequence of T sentences : $\boldsymbol{s} = (s(1), s(2), \ldots, s(T))$

2: Input 2: Maximum sentence length L and pre-trained Google Word-vector features for each word of dimension d

3: Input 3: Word-vectors for each word x_i in sentence $s(t)$ are concatenated to form input to CNN : $s(t) = \{x_{ij}\}_{\mathrm{L} \times d}$

4: Input 4: Frequency of Top 50 Subjectivity Clues in each sentence : $\hat{\mathbf{x}}$

5: Input 5: Output class labels for training sentences : y

6: Outputs: Subjectivity label for each test sentence

7: %*Pre-training CNN using Gaussian Bayesian networks*

8: Compute prior probability of each parent word set a_i for child word x_i called 'word-motif' using:

9: $p(\hat{\mathbf{x}}_i | \boldsymbol{a}_i) = \theta_{i,\boldsymbol{a}_i} \sim \mathcal{N}\left(\mu_i + \sum_{j \in \boldsymbol{a}_i}(\hat{\mathbf{x}}_j - \mu_j)\beta, \Sigma'_i \right)$

10: where Σ_i is the covariance and μ_i is the mean for the word x_i given the parent set

11: Sentence containing all words in a 'word-motif' with high probability are used to pre-train the CNN

12: %*Reinforced Deep Learning of Sentences*

13: Construct the visible layer as a vector of $\mathrm{L} \times d$ neurons for input sentences

14: Construct the convolutional layer with Z (k-gram) neurons

15: Construct the hidden logistic layer with n_h neurons

16: Construct the recurrent neuron layer with n_r neurons

17: Construct the output layer with n_d neurons

18: **repeat**

19:   **for** $t = 1$ to T **do**

20:     Initialize the visible layer with t^{th} training sentence in $\boldsymbol{s}$

21:     Update W_l $\forall l$ layers by maximizing the energy function as follows :

22:     $\mathrm{E} = \sum_{z=1}^{Z} \sum_{i,j}^{(\mathrm{L}-k+1),(\mathrm{L}-d+1)} \sum_{r,s}^{k,d} v_{i+r-1,j+s-1} h_{ij}^z w_{rs}$

23:     where the weight for input neuron v_r and hidden neuron h_s is updated using :

24:     $\triangle w_{rs} = \alpha(< v_r h_s >_{data} - < v_r h_s >_{recon})$

25:   **end for**

26:   Fine-tune weights using labels y and *reinforcement learning* given by Eqn 1

27:   Compute change in reconstruction error $\triangle \epsilon$ for each layer

28:   **if** $\triangle \epsilon$ is significant **then**

29:     Add another convolutional layer

30:   **end if**

31: **until** Adding a layer does not change reconstruction error

32: Classify test sentences using trained model.

We first construct a minimal deep CNN with visible layer of $\mathrm{L} \times d$ nodes, where length of the sentence is given by L and d is number of features for each word; first hidden convolution layer of k-gram neurons, second hidden logistic layer of n_h neurons and n_d output neurons.

The n_h features expressed at logistic layer after training form the new low-dimensional input data of T samples.

Next, we construct a recurrent neural network (RNN) with n_h input nodes and n_r hidden neurons with time-delays. The n_r features expressed at the hidden neurons after training form the new output data of T samples.

Each test sample is used to generate n_h features from deep CNN and n_r features from RNN and finally classified using RNN. Lastly, we regularize the negative log-likelihood probability of the output layer in deep CNN as described by Eq.(1) resulting in a RDCNN.

We calculated the change in visible layer reconstruction error $\triangle \epsilon$ on the training samples in order to ascertain the number of hidden layers in the deep CNN and the RNN. At each visible node the $\triangle \epsilon$ is the root mean square error between reconstructed sample and the input training sample.

If a significant change in error was noticed as shown in Algorithm 1 then a new hidden layer was added. The reconstruction error is recomputed and the each layer's weights are learned. The above procedure is iteratively recomputed until no further significant change in classification precision error with the addition of hidden layers occurs. For each hidden layer, optimal number of hidden neurons are determined based on significant principal components in the training data.

Each neuron in the final output layer corresponds to a particular class. In order to sample features with high frequency we use contrastive divergence approach. It samples them in the upper layers, which results in the formation of the phrases at the neurons in the first layer. The bigger sentences are generated at the neurons in the second hidden layer and so on. The algorithm is iterated until no significant change in the weights at the lth layer were found. Lastly, we train the model using labelled multi-lingual tweets.

4 Evaluation

This phase includes data collection from Twitter, data pre-processing, and testing it on the model.

4.1 Data Collection

The dataset consists of tweets crawled from Twitter streaming API [2]. In order to collect tweets relevant to nuclear energy, we carefully chose a list of keywords from Twitter. The crawling started with a single basic keyword *# nuclearenergy* and it resulted in about 2000 tweets over a period of two days. The collected tweets were parsed and the "hashtags" were collected from each tweet.

The hashtags were sorted in terms of frequency. A hashtag was counted only once even if it appeared twice in the tweet. It thereby prevented not giving weightage to a keyword just on the basis of frequency even if it appeared multiple times in one tweet. For example, in the tweet, " The # nuclearenergy is the # greenenergy and will help cut carbon emissions. # nuclearenergy # nuclear", The tag "# nuclearenergy" occurs twice but our model counts it only once, thereby not giving irrelevant weightage to the hashtag.

Bearing this in mind, 219 hashtags were collected with frequencies ranging from 1100 to 10. The cut-off of the hashtag frequency was set to 800, which resulted in 1200 tweets. Then the keywords not related to nuclear energy were manually removed, for example some tweets included *# nuclearenergy* with *# trump*, whereas *# trump* was not an apt keyword for our dataset.

After removing the irrelevant keywords there were 20 keywords left. Then, the rest of the keywords were given input to the Twitter API. The output of Twitter were stored and analyzed manually to come up with a list of keywords best suited for nuclear energy task. Table 1 is the list of final keywords for the nuclear energy:

Table 1. Final list of Keywords for Nuclear Energy

save the nukes, nuclear power, nuclear energy
nuclear disaster, why nuclear, nuclear for climate
nuclear power plant, nuclear reactor leak, fukusima and nuclear energy
nuclear energy and green energy

It was observed that using "# " restricted the output tweets with #, whereas some people do not enter "# " tags , thereby losing a major part of opinionated posts. The final set of keywords were then given to Twitter's streaming API to get tweets. It collected 30,000 tweets for both the topics over a course of 2 months.

---

[2]https://dev.Twitter.com/streaming/overview

4.2 Data Preprocessing

The data preprocessing phase followed the following rule in order to remove unnecessary tweets:

1. removes usernames (starting with @),

2. urls (eg., https://www.Twitter.com),

3. Removal of the stop words and punctuation marks,

4. Removal of microtext (including emoticons, interjections, and slang).

The above mentioned rules resulted in 12,719 tweets against 30,000 collected tweets, for nuclear energy, which suggests that most of the tweets contained urls, usernames, stopwords, and microtext. This huge dropout in the number of tweets also depicts the importance of a microtext analysis module.

4.3 Microtext

By dropping microtext, a major part of opinionated posts may have been lost since it became one of the most widespread communication forms among users due to its casual writing style and colloquial tone [11]. For instance, it is possible to recognize the sentiment of a tweet through features like retweets (since users frequently retransmit messages they strongly agree with) and hashtags (which can have affect keywords and assembly tweets that often agree with each other) [9]. As follows, the intrinsic noisy nature of this user-generated content (UGC) poses difficult challenges to sentiment analysis applications [8]. Some of the microtext key features of micortexts are as follows:

1. highly relaxed spelling,

2. the reliance on emoticons and out-of-vocabulary (OOV) words involving phonetic spelling (e.g., b4 for before),

3. emotional emphasis (e.g., cooooool for cool),

4. popular acronyms (e.g., otw for on the way) [22, 24, 26].

Yet, the challenge arises when, instead of removing microtext, we try to automatically rectify and reinstate them with the correct in-vocabulary (IV) words [12, 17].

4.4 Parameters

In order to determine the accuracy of the sentences we employed 10-fold cross validation using the trained CNN classifier. The learned features are visualised by only using 6-grams in the test set. The 6-grams demonstrated the highest activation when convolved with learned kernels.

The root mean square error (rmse) method was employed to calculate the difference between the predicted 6-gram kernel vectors and the prior word-vectors for each 6-gram learned. The 6-gram learned was using co-occurrence data. We find that 32% of the tweets are objective and 68% of the tweets are subjective.

4.5 Nuclear Energy

Some neurons in the first layer learn neutral and some learn subjective features. We have illustrated the 6-gram features for nuclear energy as learned by two neurons in Table 2. Next, in order to assess the quality of prediction, we manually labeled $7,700$ tweets out of $12,719$ tweets as Subjective (neutral) and Objective (positive or negative).

Table 2. 6-gram Features learned by the neurons: Nuclear Energy

Subjective					
1	2	3	4	5	6
researchers	retract	study	suggesting	countries	.
lag	retract	study	suggesting	countries	lag
clean	safety	regulatory	committee	japan	weekend
authoritarian	saltier	nuclear	draws	thiel	bloomberg
saltier	nuclear	draws	thiel	bloomberg	.
Neutral					
1	2	3	4	5	6
ethics	cie	2	regulator	safe	keeping
buzz	request	proposals	looms	ampcronies	theyll
nuclear	draws	thiel	bloomberg	-	-
draws	thiel	bloomberg	-	-	-
nuclear	draws	thiel	bloomberg	-	-

We first consider a model trained on 10,000 MPQA English corpus [6] and tested it on all the nuclear tweets. For the second model we report 10-fold cross-validation (CV) on multi-lingual labeled nuclear tweets in English, French, Spanish, German, Malay and Indonesian. We use pre-trained word-vectors for different languages (available at Facebook Research [3]).

Table 3 and Table 4 shows the F-measure for our model trained on MPQA English corpus and manually labelled multi-lingual tweets. It can be seen that the F-measure is around 76% for both classes showing that

---

[3] https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md

Table 3. F-measure of our model trained only on English MPQA benchmark

Dataset	F-Measure
Subjective	0.38
Objective	0.62
Total	0.5

Table 4. F-measure of Model trained on multi-lingual labeled tweets (10-fold CV)

Dataset	F-Score
Subjective	0.72
Objective	0.79
Total	0.76

the proposed model works well on Tweets with different languages. Though the model shows low F-score for MPQA dataset which is around 50%.

5 Conclusion and Future Work

The exponential growth of social data has led to a new challenge: subjectivity detection. Subjectivity detection is a complex NLP task that consists of distinguishing subjective data (opinions) from objective data (facts). This paper, proposes a new model with reinforcement learning. The testing phase included Twitter streaming API tweets about nuclear energy.

The data preprocessing phase removed the major part of noise in the collected data, emphasizing the importance of data preprocessing in social media. By filtering out objective data, we found out that 68% of total tweets about nuclear energy were subjective which also shows that Twitter is a consistent source for mining public opinions.

The percentage of neutral tweets is due to the following possible reasons as seen in Figure 1:

1. No nuclear disaster occurred during the data crawling phase. As, public tends to share updates (opinions) more often during a disaster, as could be seen during Fukushima disaster.

2. Many social media users shared media content (e.g., news) related to nuclear energy unlike other topics (e.g., political issues).

The results show a great significance in detection of subjective and objective tweets about "nuclear energy" and, hence, open new avenues in aiding government agencies for decision making in terms of management, planning and logistics related to nuclear power plants. Besides subjectivity detection, sentiment analysis requires handling many other NLP subtasks such as aspect extraction, sarcasm detection, anaphora resolution, and microtext analysis. Future work will involve the development and application of all such subtasks to further improve the accuracy of the proposed analysis. We will also go one step further by applying polarity detection on the collected subjective data, to finally infer whether the public opinion about nuclear energy is positive or negative at any given time.

Acknowledgements

References

1. **Ben-Ami, Z., Feldman, R., & Rosenfeld, B. (2014).** Using multi-view learning to improve detection of investor sentiments on twitter. *Computación y Sistemas*, Vol. 18, No. 3.

2. **Bonzanini, M., Martinez-Alvarez, M., & Roelleke, T. (2012).** Opinion summarisation through sentence extraction. *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval - SIGIR '12*, ACM Press, New York, New York, USA, pp. 1121.

3. **Cambria, E., Das, D., Bandyopadhyay, S., & Feraco, A. (2017).** *A Practical Guide to Sentiment Analysis*. Springer, Cham, Switzerland.

4. **Cambria, E. & Hussain, A. (2015).** *Sentic Computing: A Common-Sense-Based Framework for Concept-Level Sentiment Analysis*. Springer, Cham, Switzerland.

5. **Cambria, E., Poria, S., Bajpai, R., & Schuller, B. (2016).** SenticNet 4: A semantic resource for sentiment analysis based on conceptual primitives. *COLING*, pp. 2666–2677.

6. **Chaturvedi, I., Ragusa, E., Gastaldo, P., Zunino, R., & Cambria, E. (2017).** Bayesian network based extreme learning machine for subjectivity detection. *Journal of The Franklin Institute*.

7. **Frénay, B. & Verleysen, M. (2016).** Reinforced extreme learning machines for fast robust regression in the presence of outliers. *IEEE Trans. Cybernetics*, Vol. 46, No. 12, pp. 3351–3363.

8. **Hutto, C. J. & Gilbert, E. (2014).** VADER: A parsimonious rule-based model for sentiment analysis of social media text. *Eighth International AAAI Conference on Weblogs and Social Media*, pp. 216–225.

9. **Khoury, R., Khoury, R., & Hamou-Lhadj, A. (2014).** *Microtext Processing*. Springer New York.

10. **Lai, S., Xu, L., Liu, K., & Zhao, J. (2015).** Recurrent convolutional neural networks for text classification. *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, AAAI Press, pp. 2267–2273.

11. **Liu, F., Weng, F., & Jiang, X. (2012).** A Broad-Coverage Normalization System for Social Media Language. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, July, pp. 1035–1044.

12. **Liu, F., Weng, F., Wang, B., & Liu, Y. (2011).** Insertion, deletion, or substitution? Normalizing text messages without pre-categorization nor supervision. *ACL-HLT 2011 - Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Vol. 2, pp. 71–76.

13. **Ma, Y., Cambria, E., & Gao, S. (2016).** Label embedding for zero-shot fine-grained named entity typing. *COLING*, Osaka, pp. 171–180.

14. **Majumder, N., Poria, S., Gelbukh, A., & Cambria, E. (2017).** Deep learning-based document modeling for personality detection from text. *IEEE Intelligent Systems*, Vol. 32, No. 2, pp. 74–79.

15. **Mansour, R., Hady, M. F. A., Hosam, E., Amr, H., & Ashour, A. (2015).** Feature selection for twitter sentiment analysis: An experimental study. *Computational Linguistics and Intelligent Text Processing: 16th International Conference, CICLing 2015, Cairo, Egypt, April 14-20, 2015, Proceedings, Part II*, Springer International Publishing, pp. 92–103.

16. **Oneto, L., Bisio, F., Cambria, E., & Anguita, D. (2016).** Statistical learning theory and ELM for big social data analysis. *IEEE Computational Intelligence Magazine*, Vol. 11, No. 3, pp. 45–55.

17. Petrović, S., Osborne, M., & Lavrenko, V. (2010). The Edinburgh Twitter corpus. *Proceedings of the NAACL HLT Workshop on Computational Linguistics in a World of Social Media*, pp. 25–26.

18. Poria, S., Cambria, E., Bajpai, R., & Hussain, A. (2017). A review of affective computing: From unimodal analysis to multimodal fusion. *Information Fusion*, Vol. 37, pp. 98–125.

19. Poria, S., Cambria, E., & Gelbukh, A. (2016). Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems*, Vol. 108, pp. 42–49.

20. Poria, S., Cambria, E., Hazarika, D., & Vij, P. (2016). A deeper look into sarcastic tweets using deep convolutional neural networks. *COLING*, pp. 1601–1612.

21. Poria, S., Chaturvedi, I., Cambria, E., & Hussain, A. (2016). Convolutional MKL based multimodal emotion recognition and sentiment analysis. *ICDM*, Barcelona, pp. 439–448.

22. Read, J. (2005). Using emoticons to reduce dependency in machine learning techniques for sentiment classification. *Proceedings of the ACL student research workshop*, Association for Computational Linguistics, pp. 43–48.

23. Riloff, E. & Wiebe, J. (2003). Learning extraction patterns for subjective expressions. *Proceedings of the 2003 conference on Empirical methods in natural language processing -*, volume 10, Association for Computational Linguistics, Morristown, NJ, USA, pp. 105–112.

24. Rosa, K. D. & Ellen, J. (2009). Text classification methodologies applied to micro-text in military chat. *Proc. Eight International Conference on Machine Learning and Applications*, Miami, pp. 710–714.

25. Xia, Y., Cambria, E., Hussain, A., & Zhao, H. (2015). Word polarity disambiguation using bayesian model and opinion-level features. *Cognitive Computation*, Vol. 7, No. 3, pp. 369–380.

26. Xue, Z., Yin, D., & Davison, B. D. (2011). Normalizing Microtext. *Analyzing Microtext*, pp. 74–79.

27. Zhong, X., Sun, A., & Cambria, E. (2017). Time expression analysis and recognition using syntactic token types and general heuristic rules. *ACL*.