

Modified framework for sarcasm detection and classification in sentiment analysis

Mohd Suhairi Md Suhaimin¹, Mohd Hanafi Ahmad Hijazi², Rayner Alfred³, Frans Coenen⁴

^{1,2,3}Faculty of Computing and Informatics, Universiti Malaysia Sabah, Malaysia

¹Kuching Community College, Ministry of Education, Malaysia

⁴Department of Computer Science, University of Liverpool, United Kingdom

Article Info

Article history:

Received Sep 15, 2018

Revised Dec 6, 2018

Accepted Dec 17, 2018

Keywords:

Classification

Framework

Malay social media

Sarcasm detection

Sentiment analysis

ABSTRACT

Sentiment analysis is directed at identifying people's opinions, beliefs, views and emotions in the context of the entities and attributes that appear in text. The presence of sarcasm, however, can significantly hamper sentiment analysis. In this paper a sentiment classification framework is presented that incorporates sarcasm detection. The framework was evaluated using a non-linear Support Vector Machine and Malay social media data. The results obtained demonstrated that the proposed sarcasm detection process could successfully detect the presence of sarcasm in that better sentiment classification performance was recorded. A best average F-measure score of 0.905 was recorded using the framework; a significantly better result than when sentiment classification was performed without sarcasm detection.

Copyright © 2019 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Mohd Suhairi Md Suhaimin,

Kuching Community College, Ministry of Education,

Petra Jaya, 93050 Sarawak Malaysia.

Email: mi1511003t@alum.ums.edu.my / suhairisuhaimin@kkkg.edu.my

1. INTRODUCTION

User generated content, acquired from social media, has been extensively analysed so as to identify people's emotions, perspectives, views, beliefs and sentiments towards situations, products, services, other individuals and organisations [1]. Such Sentiment Analysis (SA) focuses on the positive and negative labelling of comments. However, the presence of sarcasm in user comments can adversely affect the quality of the SA. When sarcastic content is included in what would be considered to be a positive statement, the meaning is intended to be negative, and vice versa [2]. The use of sarcasm is particularly prevalent in the context of political exchanges such as in the case of discussion forums. The overall affect of sarcasm is to 'flip' the expressed sentiment [3]. A failure to detect sarcasm will clearly affect the output from SA systems [4]. Extensive work has been reported directed at overcoming the sarcasm problem using a range of techniques [5]. However, to the best knowledge of the authors, no work has been conducted whereby sarcasm detection and classification have been incorporated into a sentiment classification framework as proposed in this paper.

The objective of the work presented in this paper can be summarized as: given an opinionated text comment x , determine whether x expresses a positive or negative sentiment after considering whether sarcasm is present or not. More specifically this paper presents a sentiment classification framework that incorporates sarcasm detection and classification. Initial sentiment classification is performed on the preprocessed texts from which features are selected and extracted. Sarcasm detection and classification are performed later. The aim is firstly to detect the presence of sarcasm, and then, as consequence, to flip the initial classification. Actual sentiment classification is thus performed at the end of the process.

2. RELATED WORK

Sarcasm is a type of verbal irony that implies the opposite meaning of the literal meaning of what was said [6], [7]. Along with hyperbole, jocularity, rhetorical questions and understatements, the idea is to convey a combination of obvious and more subtle interpersonal meanings. However, the study of sarcasm in linguistics [8], [9] and computation [10]–[12] has indicated that the presence of sarcasm in a negative text does not always indicate the opposite of what the speaker meant, hence when undertaking SA we cannot simply reverse the 'polarity'.

There has been some previous work directed at extracting opinions from text (commentary) that may feature sarcastic content. The system proposed in [13] adopts a Natural Language Processing (NLP) approach and features a process comprising eight steps which aim to detecting sarcastic opinions and consequently flipping polarity at fifth step, and determining the final polarity value for each target opinion at eight step. To detect sarcastic statement, the authors suggested adopting the 'contextual valence shifter' proposed in [14]. Valence calculation was performed at the sentence level using a positive and negative valence corpus to flip polarity. The authors also employed the work presented in [15], where pattern and punctuation based features were used, and the work of [16] where lexical and pragmatic features were used, to characterise a sarcastic word. However, no experiments were reported concerning the evaluation of the proposed framework.

In [17] a parsing-based unsupervised approach was proposed directed at Twitter data. The system adopted two approaches to identify sarcastic tweets, the first using parsing-based lexicon generation and the second using interjection word enrichment. Polarity identification was conducted in an automated manner. Supervised learning was then used to classify sentiment as negative, positive or neutral. The proposed system was tested using two sets of tweets, one that featured the sarcasm hashtag and one that did not feature the sarcasm hashtag. The best f-score recorded for sarcastic hashtag tweets was 0.84 using the lexicon generation approach, and 0.90 for the interjection word enrichment approach, outperforming results obtained using the set of tweets without the sarcastic hashtag.

The system presented in [18] was also directed at detecting the presence, or otherwise, of sarcasm in social media comment. The system operated in the following manner: comment acquisition, post-processing of the acquired comments, corpus creation, features extraction and selection and final classification. For the evaluation reported, tweets were used which had been annotated and indexed according to the hashtags produced by users. The lexical features used were n-grams (unigrams and bigrams) contained in LIWC [19] and WordNet-Affect [20]. Emoticons, punctuation and common ground (user reply and name characteristics) were used as pragmatic features. Chi-squared feature selection was applied to identify useful features. Naive Bayes, Logistic Regression and Support Vector Machines (SVM) were employed for the classification. It was found that the binary classification outperformed the three-way classification when an SVM classifier was employed. A best accuracy of 0.783 was reported in the context of polarity based classification (positive versus negative), and an accuracy of 0.730 in the case of sarcastic versus negative classification.

The systems briefly described above [13], [17], [18] use unsupervised or supervised approaches to identifying sarcastic text (commentary). The systems classified text as being either: positive, negative or sarcastic. In this paper some of the ideas presented with respect to these three systems have been adapted so that a positive or negative sentiment classification can be arrived at regardless of whether the text includes sarcasm or not.

3. THE PROPOSED FRAMEWORK

In this paper, we propose a framework to support SA that consists of six main modules: (i) preprocessing of text, (ii) extraction of features, (iii) feature selection, (iv) initial sentiment classification, (iv) sarcasm detection and classification, and (v) final sentiment classification. Figure 1 shows the proposed framework. Initial sentiment classification refers to regular sentiment classification before considering sarcasm detection, while final sentiment refers to final sentiment classification after performing sarcasm detection and classification. In this framework, the most critical module is the detection of sarcasm and classification. It was conjectured that the ability of the proposed work to identify accurately the presence of sarcasm in texts would result in better final sentiment classification with respect to initial sentiment classification. Details concerning each of the above modules, with respect to the proposed framework, are given in Sub-sections 3.1 to 3.6.

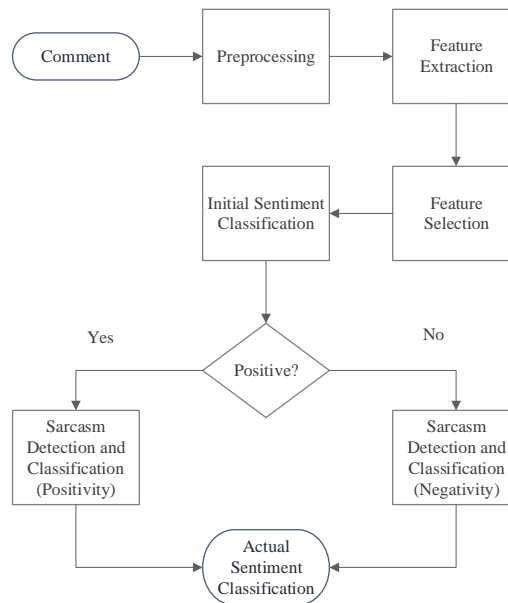


Figure 1. The framework to support SA using sarcasm detection and classification

3.1. Preprocessing

Social media text contains a significant amount of noise including: spelling errors, non-standard words, stylistic words, short form words and repetitions. Classification accuracy is increasingly affected as the presence of noise increases [21]. The presence of noisy text also causes 'dispersion', where same features are treated as different features, which results in poor performance when building a classifier [22]. The preprocessing module employed in this paper involved tokenization, spell checking and stopword removal. Tokenization breaks the corpus into words and symbols such as punctuations and hashtags (#). A correspondence dictionary was used to correct misspelled words. Stopword removal was performed to eliminate meaningless words.

3.2. Feature Extraction

Three categories of NLP based features were considered: syntactic, pragmatic and prosodic. This was the feature combination mechanism proposed in [23] which was adopted because this had demonstrated improvement in sarcasm detection in comparison with comparator mechanisms. The output of this module was a set of feature vectors (one per text) each comprised of Term Frequency - Inverse Document Frequency (TF-IDF) values normalized to document (text) length.

3.2.1. Syntactic Feature Extraction

Syntactic features play an important role in providing information concerning the syntactic structure of documents. In this paper, common syntactic features Part of Speech (POS) tags were used. Four groups of POS tag: NOUN, VERB, ADJECTIVE and ADVERB were selected. The Penn Treebank POS [24] tagset was chosen to tag the tokenized words. Each of the tags was then mapped into each corresponding group. Only the tokenized words associated with the four selected POS groups, as described above, were retained in the text. A word-tag pair representation was used to represent the syntactic features as this has been shown to produce improved sentiment classification performance compared to using words or tags alone [25].

3.2.2. Pragmatic Feature Extraction

Pragmatic features are intended to emphasize the meaning of the content of sentences that may include sarcasm [26]. Emoticons, 'heavy' punctuation, hashtags (#) and repeated words are examples of pragmatic features. Punctuation marks are considered to be pragmatic features, instead of sentence segmentators, because of their potential to indicate sarcasm [27]. Heavy punctuation, for example high occurrences of various punctuation marks, is often an indicator of the presence of sarcasm in text (comment). The punctuation marks considered in this framework were: question marks (?), exclamation marks (!) and quotation marks (' ' and " "). Hashtags (#) were also considered as it is used commonly to indicate the presence of sarcasm [28]. The length of a sequence of punctuation marks was reduced to a maximum of three characters to avoid dispersion.

3.2.3. Prosodic Feature Extraction

Prosodic features involve different pitches, loudness, timing and tempos in writing [29]. Interjections are an example of prosodic features. A list of interjections [30] was employed to compare and extract the interjections from the text.

3.3. Feature Selection

To select only the most significant features, feature selection was applied. To this end Pearson's correlation coefficient was chosen to support feature selection. The features were ranked based on the generated coefficients. Only the top N features were used for classification purposes.

3.4. Initial Sentiment Classification

The initial sentiment classification module classifies text as displaying either positive or negative sentiment. In the work presented in this paper, non-linear SVM was used to generate the classifier because it has been shown to perform well in the context of supervised classification [18]. The variation of non-linear SVM used was LibSVM [31] as provided within the Weka [32] data mining workbench.

3.5. Sarcasm Detection and Classification

This module was derived from an approach to detect and classify sarcasm reported in [12]. It has two sub-processes: sarcasm detection and classification. Figure 2 shows the process of sarcasm detection and classification of a given text after initial sentiment classification has been performed. The aim of this module was to identify and classify texts that contain sarcastic features. Texts that have been identified as positive by the initial sentiment classification will be further classified as either positive sarcastic (sarcastic features occur in the text) or true positive. Similarly, texts that have been identified as negative by the initial sentiment classification module will be further classified as either negative sarcastic or true negative.

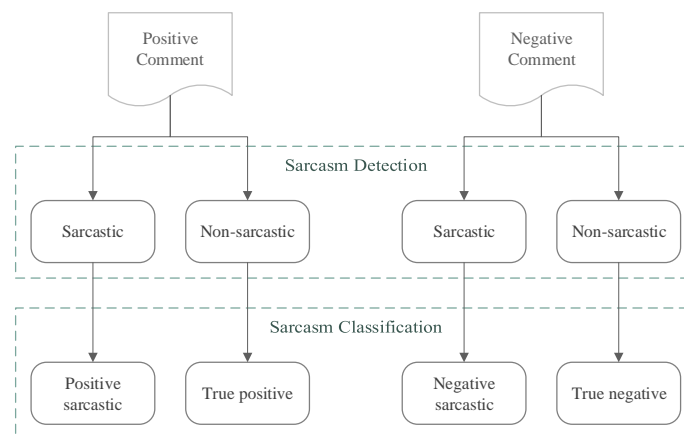


Figure 2. Sarcasm detection and classification of sentiment

3.6. Actual Sentiment Classification

As mentioned in the foregoing sections, sarcastic content tends to reverse the actual sentiment of the texts. Therefore, once sarcasm has been detected, actual sentiment classification will be performed using polarity flipping [33], [34]. The polarity flipping is employed to reverse the initial sentiment classification results based on linguistic hypotheses. Two hypotheses were considered in this paper: to flip all texts identified as containing sarcasm (positive and negative sarcastic) or to flip only positive texts containing sarcasm (positive sarcastic).

3.6.1. Flip Both Positive Sarcastic and Negative Sarcastic

The first hypothesis considers sarcasm as indicating something opposite to what the speaker meant [6], [7]. When sarcastic content is used in a positive statement, the speaker is actually saying something negative, and vice versa [2], [28]. Based on this hypothesis, the polarity of positive sarcastic text will be flipped to negative, and negative sarcastic text will be flipped to positive. Figure 3(a) shows the polarity flip based on this hypothesis.

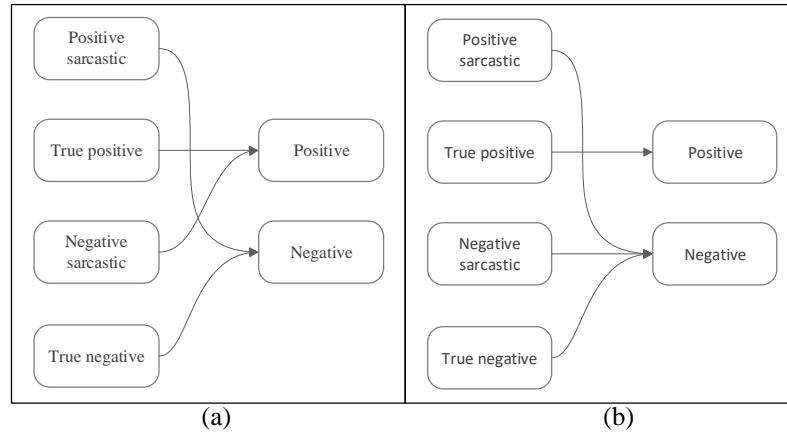


Figure 3. Polarity flipping based on (a) the presence of both positive and negative sarcasm (Hypothesis 1) and (b) the presence of positive sarcasm only (Hypothesis 2)

3.6.2. Flip positive sarcastic only

The second hypothesis states that sarcasm in negative statement does not always deliver the opposite of what the speaker meant. This hypothesis was derived from the sarcasm linguistic studies presented in [8], [9] and the computational experiments reported in [10]-[12]. Based on this hypothesis, only the polarity of positive sarcastic text should be flipped to negative, whilst negative sarcastic text should remain negative. Figure 3(b) shows the polarity flip based on this second hypothesis.

4. EXPERIMENTAL SETUP AND RESULTS

This section describes the dataset used to evaluate the framework and the nature of the experiments conducted; and a discussion of the results obtained.

4.1. The Dataset

To evaluate the proposed framework, a Malay social media dataset was used [12]. The texts were annotated by three human annotators. Three annotations were produced: (i) the sentiments of the texts (positive, negative or neutral), (ii) the existence of sarcasm or otherwise (sarcastic vs. non-sarcastic), and (iii) where sarcasm was considered to exist the positivity and negativity of the sarcasm (positive sarcasm vs. positive sentiment, and negative sarcasm vs. negative sentiment). Only texts labeled as positive or negative, and agreed by all annotators were considered in the experiments. Hence, a subset of 1970 texts was derived from the original 3000 texts.

4.2. Experimental Setup

The aim of the experiment was to evaluate the effectiveness of the proposed framework in supporting sentiment analysis. To achieve this, five sets of experiments were conducted. The first was to identify the performance of the initial sentiment analysis on the dataset and features used. The second was to measure the performance of sarcasm detection. The third and fourth set of experiments were conducted to evaluate the performances of positive and negative sarcasm classifications. The final set of experiment was conducted to evaluate the performance of actual sentiment classification, where polarity flipping was used to reverse the initial sentiment of texts where the presence sarcasm had been detected. All experiments were conducted using 10-fold cross validation and the Weka Knowledge Flow [35].

4.2.1. Preprocessing

The dataset was first tokenized, followed by spellchecking and stopword removal as described in Sub-section 3.1. The Malay and English dictionaries were used to correct misspelled words. Stopword removal was applied using both Malay [36] and English [37] stopword lists.

4.2.2. Feature Extraction and Selection

During the feature extraction stage, both the original bilingual dataset and its translation to English were considered. The reason was to preserve the original and translated features from the text that might include sarcasm in Malay and/or English. The process of feature extraction consists of two main steps: (i)

extraction of the pragmatic and prosodic (Malay) features from the original bilingual data, and (ii) translation of the bilingual data to English and extraction of English prosodic and syntactic features. In the first step, the pragmatic and Malay prosodic features were extracted from the preprocessed original bilingual dataset. Four pragmatic features were extracted: question marks (?), exclamation marks (!), quotation marks (' ' and " "), and hashtags (#). The Malay prosodic features were identified using the Malay list of interjections. In the end, 40 prosodic features were extracted from the original bilingual dataset. In the second step, the original bilingual dataset was translated to English using Google Translate [38]. Although the resulting translations were by no means perfect, they were judged to produce translations that were sufficiently accurate to support further analysis, better than the translations obtained using Moses or Bing [39]. The English prosodic features were identified using an English lists of interjections. In the end, a total of 26 prosodic features were extracted from the translated dataset. Next, syntactic features were extracted. Four POS tag groups: NOUN, VERB, ADJECTIVE and ADVERB, were extracted using the Python Natural Language Toolkit (NLTK) [40]. Word-tag pairs were extracted to represent each of the texts. The total number of syntactic features obtained was 3695. All the extracted features were then vectorized and normalized to the individual text lengths (TF-IDF). With respect to the feature selection, the top 25%, 50% and 75% of the features were selected based on the Pearson's correlation coefficient ranking. Details of the number of features for each set of experiments, and the size of the dataset in each case, are given in Table 1.

Table 1. The Number of Features Used for Experimentation

Experiment	Dataset size	% Feature Selection size			
		25%	50%	75%	Full
Initial sentiment classification	1970	941	1883	2824	3765
Sarcasm detection	802	514	1028	1542	2056
Sarcasm positivity classification	1168	686	1372	2058	2744
Sarcasm negativity classification	1970	941	1883	2824	3765

4.3. Experimental Results

The experiments were designed to consider only binary classification. Average F-measure (F_{avg}) was used to measure classification performance, formulated as:

$$F_{avg} = \frac{F_i \times c_i + F_j \times c_j}{c_i + c_j} \quad (1)$$

where F_i is the F-measure for class i and c_i is the number of documents in class i , while F_j is the F-measure for class j and c_j is the number of documents in class j . The F-measure (F) is the harmonic mean of precision and recall for each class i and j .

4.3.1. Initial Sentiment Classification

In this set of experiment, the texts were classified as having either positive or negative sentiment. Table 2 shows the results obtained. The best sentiment classification performance was recorded when using the top 25% features, with an F_{avg} score of 0.839. The worst was recorded when all features were used for classification ($F_{avg} = 0.611$).

Table 2. Results of Initial Sentiment Classification (Positive vs. Negative)

% Feature Selection (FS) size	25%	50%	75%	Full
Experiment	Average F-measure (F_{avg})			
Initial sentiment classification	0.839	0.623	0.754	0.611

4.3.2. Sarcasm Detection

For the second set of experiments, the texts were classified as being either sarcastic or non-sarcastic. The best sarcasm detection was recorded when using the top 50% of the features, with an F_{avg} score of 0.852 as shown in Table 3. As in the case of the results obtained with respect to sentiment classification, sarcasm detection with all features produced the worst performance with an F_{avg} score of 0.664.

Table 3. Results of Sarcasm Detection and Sarcasm Classification

% Feature Selection (FS) size	25%	50%	75%	Full
Experiment	Average F-measure (F_{avg})			
Sarcasm detection	0.755	0.852	0.737	0.664
Sarcasm positivity classification	0.942	0.787	0.776	0.767
Sarcasm negativity classification	0.909	0.797	0.614	0.593

4.3.3. Sarcasm Positivity and Negativity Classification

Table 3 also shows the results for sarcasm classification. For positive sarcasm classification, the texts were classified as being either positive sarcastic or true positive. The best result was again recorded when using the top 25% of the features, with an F_{avg} score of 0.942. Negative sarcasm, where texts were classified as being negative sarcastic or true negative, produced a lower best F_{avg} score of 0.909 compared to positive sarcasm classification. In fact, the negative sarcasm classification produced lower performances than positive sarcasm in most cases regardless of the number of features used. This may be due to difficulties in recognizing the negative sarcastic features from true negative texts compared to positive sarcasm classification.

4.3.4. Actual Sentiment Classification

The results of the actual sentiment classification are shown in Table 4. In this experiment, the texts were classified in terms of positive or negative sentiment. When polarity flipping was applied on both positive and negative sarcastic texts, the best F_{avg} score recorded was 0.899 using the top 25% features. However, the best F_{avg} result with respect to actual sentiment classification was recorded when polarity flipping was applied only the positive sarcastic texts, where the F_{avg} scores of 0.905 recorded using the top 25% features. In most cases, the best performing results were obtained when using only the top 25% of the features. Whatever the case, the actual sentiment classification improved on the initial sentiment classification in all cases.

Table 4. Results of Actual Sentiment Classification

% Feature Selection (FS) size	25%	50%	75%	Full
Experiment	Average F-measure (F_{avg})			
Actual sentiment classification (Flip both positive sarcastic & negative sarcastic)	0.899	0.715	0.671	0.666
Actual sentiment classification (Flip positive sarcastic only)	0.905	0.903	0.903	0.900

4.4. Analysis of Results

Based on the results shown in Tables 2, 3 and 4, it can be observed that the performance of the sentiment classification was improved by 6.6% after considering sarcastic texts (the initial sentiment classification produced a best F_{avg} of 0.839 while the actual sentiment classification produced best F_{avg} of 0.905).

5. CONCLUSION

This paper has presented a framework to support SA by utilizing sarcasm detection and classification. A framework comprised of six modules is proposed: preprocessing, feature extraction, feature selection, initial sentiment classification, sarcasm detection and classification, and actual sentiment classification. A non-linear SVM was used for classification purposes with respect to the reported experiments. Comparison of SA without sarcasm detection (initial sentiment classification) against classification with sarcasm detection (actual sentiment classification) demonstrated that the latter produced a better classification performance.

ACKNOWLEDGEMENTS





This work was supported by Universiti Malaysia Sabah (UMS) through a grant GUG0061-TK-2/2016 and Artificial Intelligence Research Unit, UMS.

REFERENCES

- [1] B. Liu and L. Zhang, "A survey of opinion mining and sentiment analysis," *Mining text data*, pp. 415–463, 2012.
- [2] B. Liu, "Sentiment analysis: Mining opinions, sentiments, and emotions," Cambridge University Press, 2015.
- [3] W. Leila, et al., "How does irony affect sentiment analysis tools?" *Progress in Artificial Intelligence*, vol. 9273, pp. 803–808, 2015.
- [4] A. Farzindar and D. Inkpen, "Natural Language Processing for Social Media," Morgan & Claypool Publishers, vol. 8, 2015.
- [5] A. Joshi, et al., "Automatic sarcasm detection: A survey," *arXiv Prepr. arXiv1602.03426*, 2016.
- [6] R. W. Gibbs, "On the psycholinguistics of sarcasm," *J. Exp. Psychol. Gen.*, vol/issue: 115(1), pp. 3, 1986.
- [7] R. Gibbs, "Irony in talk among friends," *Irony in language and thought: A cognitive science reader*, R. Gibbs and H. Colston, Eds. London: Taylor & Francis Group, pp. 339–360, 2007.
- [8] R. J. Kreuz and S. Glucksberg, "How to be sarcastic: The echoic reminder theory of verbal irony," *J. Exp. Psychol. Gen.*, vol/issue: 118(4), pp. 374, 1989.
- [9] S. Attardo, "Irony as relevant inappropriateness," *J. Pragmat.*, vol/issue: 32(6), pp. 793–826, 2000.
- [10] E. Riloff, et al., "Sarcasm as contrast between a positive sentiment and negative situation," in *EMNLP 2013 - 2013 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pp. 704–714, 2013.
- [11] M. Bouazizi and T. Ohtsuki, "Opinion mining in Twitter: How to make use of sarcasm to enhance sentiment analysis," *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, pp. 1594–1597, 2015.
- [12] M. S. Md Suhaimin, et al., "Mechanism for Sarcasm Detection and Classification in Malay Social Media," *Adv. Sci. Lett.*, vol/issue: 28(2), pp. 1388–1392, 2018.
- [13] E. Cambria, et al., "The CLSA model: A novel framework for concept-level sentiment analysis," *International Conference on Intelligent Text Processing and Computational Linguistics*, pp. 3–22, 2015.
- [14] L. Polanyi and A. Zaenen, "Contextual valence shifters," *Computing attitude and affect in text: Theory and applications*, Springer, pp. 1–10, 2006.
- [15] O. Tsur, et al., "ICWSM - A great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews," *ICWSM 2010 - Proceedings of the 4th International AAAI Conference on Weblogs and Social Media*, pp. 162–169, 2010.
- [16] R. G. Ibáñez, et al., "Identifying sarcasm in Twitter: A closer look," *ACL-HLT 2011 - Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, vol. 2, pp. 581–586, 2011.
- [17] S. K. Bharti, et al., "Parsing-based sarcasm sentiment recognition in Twitter data," *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, pp. 1373–1380, 2015.
- [18] S. Muresan, et al., "Identification of nonliteral language in social media: A case study on sarcasm," *J. Assoc. Inf. Sci. Technol.*, 2015.
- [19] J. W. Pennebaker, et al., "The development and psychometric properties of LIWC2015," *UT Fac. Work.*, 2015.
- [20] C. Strapparava and A. Valitutti, "WordNet Affect: an Affective Extension of WordNet," *LREC*, vol. 4, pp. 1083–1086, 2004.
- [21] S. Agarwal, et al., "How much noise is too much: A study in automatic text classification," *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*. IEEE, pp. 3–12, 2007.
- [22] G. Forman, "Feature selection for text classification," *Comput. methods Featur. Sel.*, vol. 1944355797, 2007.
- [23] M. S. Md Suhaimin, et al., "Natural Language Processing Based Features for Sarcasm Detection: An Investigation Using Bilingual Social Media Texts," *Proceeding of the The 8th International Conference on Information Technology, Amman, Jordan*, 2017.
- [24] https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.htm
- [25] R. Xia and C. Zong, "Exploring the use of word relation features for sentiment classification," *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pp. 1336–1344, 2010.
- [26] D. Jurafsky, "Speech and language processing: An introduction to natural language processing," *Comput. Linguist. speech Recognit.*, 2000.
- [27] P. Carvalho, et al., "Clues for detecting irony in user-generated contents: Oh...!! it's 'so easy' ;-)," *Proceeding of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion - TSA '09*, pp. 53, 2009.
- [28] C. C. Liebrecht, et al., "The perfect solution for detecting sarcasm in tweets# not," *4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 29–37, 2013.
- [29] D. Bikel and I. Zitouni, "Multilingual Natural Language Processing Applications: From Theory to Practice," IBM Press, 2012.
- [30] <http://grammar.yourdictionary.com/parts-of-speech/interjections/list-of-interjections.html>
- [31] C. C. Chang and C. J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol/issue: 2(3), pp. 27, 2011.
- [32] M. Hall, et al., "The WEKA data mining software: an update," *ACM SIGKDD Explor. Newsl.*, vol/issue: 11(1), pp. 10–18, 2009.
- [33] C. F. Burgers, "Verbal irony: Use and effects in written discourse," [SI: sn], 2010.
- [34] F. Kunneman, et al., "Signaling sarcasm: From hyperbole to hashtag," *Inf. Process. Manag.*, vol. 51, pp. 500–509, 2015.

[35] I. H. Witten, *et al.*, “Data mining: Practical machine learning tools and techniques,” Boston: Morgan Kaufmann, 2011.
 [36] http://nlp.cs.nyu.edu/GMA_files/resources/malay.stoplist
 [37] <http://www.nltk.org/book/ch02.html>
 [38] <https://translate.google.com>
 [39] A. Balahur and M. Turchi, “Comparative experiments using supervised learning and machine translation for multilingual sentiment analysis,” *Comput. Speech Lang.*, vol/issue: 28(1), pp. 56–75, 2014.
 [40] S. Bird, *et al.*, “Natural language processing with Python: analyzing text with the natural language toolkit,” O’Reilly Media, Inc., 2009.

BIOGRAPHIES OF AUTHORS

	<p>Mohd Suhairi Md Suhaimin is a lecturer of Information Technology and General Studies at Kuching Community College, Sarawak. He has completed postgraduate study in Computer Science at Universiti Malaysia Sabah (UMS). His research interests span both data mining and data visualization. He currently leads the project 'Sarcasm Detection and Classification to Support Sentiment Analysis' focusing in bilingual, Malay and English. He has explored the presence and implication of sarcasm in sentiment analysis.</p>
	<p>Mohd Hanafi Ahmad Hijazi is an Associate Professor of Computer Science at the Faculty of Computing and Informatics, Universiti Malaysia Sabah in Malaysia. His research work addresses the challenges in knowledge discovery and data mining to identify patterns for prediction on structured and/ or unstructured data; his particular application domains are medical image analysis and understanding and sentiment analysis on social media data. He has authored/ co-authored more than 30 journals/ book chapters and conference papers, most of which are indexed by Scopus and ISI Web of Science. He also served on the program and organizing committees of numerous national and international conferences.</p>
	<p>Rayner Alfred is an Associate Professor of Computer Science at the Faculty of Computing and Informatics, Universiti Malaysia Sabah in Malaysia. He leads and defines projects around knowledge discovery and information retrieval that focuses on building smarter mechanism that enables knowledge discovery in structured and unstructured data. His work addresses the challenges related to big data problem. He has authored and co-authored more than 85 journals/book chapters, conference papers and editorials.</p>
	<p>Frans Coenen has a general background in AI and Machine Learning and over the last ten years has been working in the field of Big Data Analytics as applied to unusual data sets, such as: (i) graphs and social networks, (ii) time series, (iii) free text of all kinds, (iv) 2D and 3D images, particularly medical images, and (v) video data. He is also interested in data mining over encrypted data. He currently leads a small research group working on many aspect of machine learning. He has some 380 refereed publications on Machine Learning and AI related research. Frans Coenen is currently professor within the Department of Computer Science at the University of Liverpool.</p>