TWO CLASSIFIERS IN ARBITER TREE TO ANALYZE DATA

Tawunrat Chalothorn and Jeremy Ellman
Department of Computer Science and Digital Technologies
University of Northumbria at Newcastle, Pandon Building,
Camden Street Newcastle Upon Tyne, NE2 1XE, United Kingdom

ABSTRACT

This paper reports on the use of ensemble learning to classify the sentiment of tweets as being either positive or negative. Tweets were chosen because Twitter is both a popular tool and a public, human annotated dataset was made available as part of the SEMVAL 2013 competition. We report on an approach to classification that contrasts single machine learning algorithms with a combination of algorithms in an ensemble learning approach. The single machines learning algorithms used were Support Vector Machine (SVM) and Naïve Bayes (NB) while the method of ensemble learning was the arbiter tree. Our system achieved an F score using the arbiter tree at 83.55% which was the same as SVM but quite slightly than Naïve Bayes algorithm.

KEYWORDS: *Tweets, contexts, positive, negative, natural language processing, ensemble learning*

I. INTRODUCTION

The research area of natural language processing (NLP) is composed of various tasks; one of which is sentiment analysis. The main goal of sentiment analysis is to identify the polarity of natural language text. Sentiment analysis can be referred to as opinion mining; studying opinions, appraisals and emotions towards entities, events and their attributes. Sentiment analysis is a popular research area in NLP that aims to identify opinions or attitudes in terms of polarity. Currently, Twitter is a popular microblogging tool where users are increasing by the minute. Twitter allows users to post messages of up to 140 characters each time. These are called 'Tweets', which are often used to convey opinions about different topics. Consequently, various researchers are interested in classifying Tweets by using sentiment analysis.

This paper introduces the novelty of using arbiter tree [1-5], to classify the contexts of Tweet datasets and use SMS datasets to evaluate the system. Arbiter tree [1-5] has been chosen because it has not yet been used in sentiment analysis to classify Tweets or SMS datasets. The basic idea is to divide the training data into subsets, apply the leering algorithm to each one and merge the resulting inducers. The main task is to find the solution to combining the right learning model in order to achieve better results. Our main contribution is to propose and experiment with a combination of two machine learning, based on the use of the arbiter tree algorithm [1-5]. The remainder of this paper is constructed as follows: the detail of the corpus used is discussed in section 2; the methodology with data pre-processing and details of classifier are presented in section 3; section 4 discusses the details of the experiment and results. Finally, a conclusion and recommendations for future work are provided in section 6.

II. RELATED WORKS

Machine leaning is well-known and widely used in various researches. For example, [6] used three machine learning algorithms to classify sentiment of Twitter: Naïve Bayes [7], Maximum Entropy Modelling [8] and Support Vector Machine [9, 10]. Emoticons have been used as labels (positive and negative) in training data to perform supervised learning. There are two features that were used in the experiment: unigram and part-of-speech. The results from unigram showed that, [6] achieved 81.3%, 80.5% and 82.2% from three machine learning algorithms, respectively. On the other hand, the results from the combination of unigram and part-of-speech achieved lower accuracy at 79.9%, 79.9% and 81.9% from three machine learning algorithms, respectively. [6] used single machine leaning algorithm but will the performance achieved better accuracy if used the combination of machine leaning algorithms? This question has not been answered.

[11] used Support Vector Machine [9, 10] to classify tweets whether the contexts are related to the company or not. The dataset was obtained from WePS-3. WePS-3 is a workshop that focuses on share tasks on the Searching Information about Entities in the Web. For solving the problem, [11] built corpus by collecting keywords that related to the company by using six profiles. The first profile, keywords that relevant to the company and presented on the company homepage that was provided by WePS-3 was extracted and named as, homepage profile. The second profile, the keywords from meta tags of the webpages were collected and named as, metadata profile. The third profile, [11] used WordNet to find the keywords of the category that the company belong to and named as, category profile. The Forth profile, the keywords that closely related to the company were gotten Google Sets and named as, googleset profile. Google Sets is a source for obtaining common knowledge about the company by identifying and generating the lists of the items that might related to the company such as the companies that similar or competitor or products. In the mid of 2011, Google Sets was discontinued from Google.[1] The fifth and sixth profiles are the collection of the keyword from users' feedback in both positive and negative and named as, positive profile and negative profile, respectively. After getting all profiles, [11] separated the use of these profiles into four tasks: use all profiles, use all profiles except the negative feedback, use all profiles except the category profile and use only home page. The results showed that, the accuracy performance achieved F-score at 59.50%, 62%, 60% and 48%, respectively. In this experiment, Support Vector Machine were used but how much the accuracy could be achieved from using the others machine leaning algorithms? This question has not been answered.

[12] used three machine leaning algorithms: Naïve Bayes [7], Rocchio [13] and Perceptron [14] to classify contents from Facebook by using positive and negative emoticons. Rocchio [13] is not a machine learning but it is text classifier which based on relevance feedback that was introduce by [13]. On the other hand, Perceptron [14] is supervised machine learning with the attempt for finding a hyperplane that separated two sets of point [15]. The datasets were collected by using Facebook API[2]. Facebook API is a platform for building application that available to the Facebook's users. API allow the application to access to the users' information and social connection for connecting to the application for posting the activities or news on users' profile pages of Facebook which subject to the privacy setting of the users [16]. The results showed that, F-score accuracy achieved at 72%, 74% and 60% for using Naïve Bayes [7], Rocchio [13] and Perceptron [14], respectively. If three machine learning algorithms were combined together, will the accuracy performance achieved better than single machine learning algorithms? This question has not been answered.

III. CORPUS

The datasets used in our experiment are from SemEval 2013 [17]. The data were gathered from Twitter; a well-known and increasingly popular microblogging site. Twitter allows its users to post messages, or 'Tweets', of up to 140 characters each time, which are available for immediate download over the Internet. Tweets are extremely interesting in marketing terms, since their rapid public interaction can either indicate customer success or presage public relations disasters far more quickly than web pages or traditional media. Consequently, the content of tweets and identifying their sentiment polarity as positive or negative is a current active research topic.

The datasets are composed of training data, testing data and gold standard. Gold standard refers to the testing data labelled with the correct polarity. However, these datasets were annotated using five Mechanical Turk workers, also known as Turkers [17]. For each sentence, they will mark by using the start and end point of their opinion for the phrase or word, and state whether it is negative, neutral or positive. Then, the words that appear three times from five votes will be assigned the label. In addition to Tweets, SMS messages are used to evaluate the system. SMS messages are also obtained

[1] http://googlesystem.blogspot.co.uk/2011/08/google-sets-will-be-shut-down.html

[2] https://developers.facebook.com/docs/reference/fql

from the organiser of SemEval 2013 [17]. Only the datasets labelled as positive and negative will be used in this research.

IV. METHODOLOGIES

3.1. Data pre-processing

For the process of data pre-processing, emoticons were labelled by matching those that have been collected manually from the dataset against a well-known collection of emoticons. Subsequently, negative contractions were expanded and converted to full form (e.g. don’t -> do not). Moreover, the features of Tweets were removed or replaced by words, such as Twitter usernames, URLs and hashtags.

A Twitter username is a unique name displayed in the user's profile and may be used for both authentication and identification. This is shown by prefacing the username with an @ symbol. When a Tweet is directed at an individual or particular entity, this can be shown in the tweet by including @username. For example, a Tweet directed at ‘som’ would include the text @som. Before URLs are posted to Twitter, they are shortened automatically to use the t.co domain whose modified URLs are a maximum of 22 characters. However, both features have been removed from the datasets. Hashtags are used to represent keywords and topics in Twitter by using # followed by words or phrases, such as #newcastleuk. This feature has been replaced with the following word after the # symbol. For example, #newcastleuk was replaced by newcastleuk.

Frequently, repeated letters are used to provide emphasis in Tweets. These were reduced and replaced using a simple regular expression by two of the same characters. For example, happpppppy will be replaced with happy, and coollllll will be replaced by cool. Next, special characters were removed, such as [,{,?,and !. Slang and contracted words were converted to their full form; for example, ‘fyi’ became ‘for your information’. Finally, Natural Language Toolkit (NLTK) [18] stopwords were removed from the datasets, such as ‘a’, ‘the’, etc..

Furthermore, three sentiment lexicons were used in this experiment. They are Bing Liu Lexicon (HL) (6780 words), collected over many years by [19]. They began to collect lexicons in 2004, during the course of their work on online customer product reviews [19]. MPQA Subjective Lexicon (MPQA) (8221 words) was created by [20] using a set of approximately 400 documents. AFINN Lexicon (AFINN) (2477 words) was created from Twitter between 2009-2011 by [21] for use in the United Nation Climate Conference (COP15).

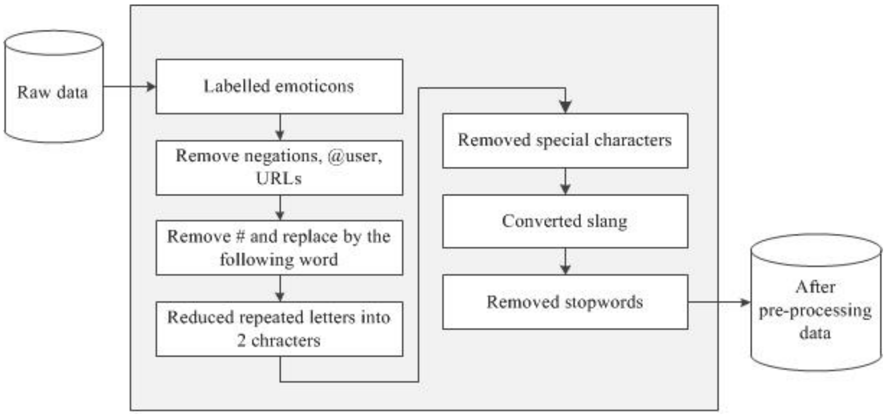
Raw data
Labelled emoticons
Remove negations, @user, URLs
Remove # and replace by the following word
Reduced repeated letters into 2 chracters
Removed special characters
Converted slang
Removed stopwords
After pre-processing data

Figure 1: Flowchart of data pre-processing

3.2. Arbiter Tree

Arbiter tree [1-5] is a method that uses training data output classified using base classifiers with selection rules. Selection rules are used to compare the prediction of based classifiers for choosing the training dataset for the arbiter. Then, the final prediction is decided according to the base classifiers and arbiter by using arbitration rules with the aim of learning from incorrect classifications [1].

In the process of making the training data for arbiter from [1] mentioned using four training data (T1-4) subsets and four classifiers (C1-4). Next, unite the results T1 and T2, and used selection rules to generate a training set for arbiter A12 with the same learning algorithm used in the initial classifiers. This process is similar to arbiter A34, which used the training data that unite from T3 and T4, and then, the first level of arbiter is produced. After obtaining the results from T12 and T34, they will be united to form a training dataset for the root arbiter A14, as illustrated in Figure 2.

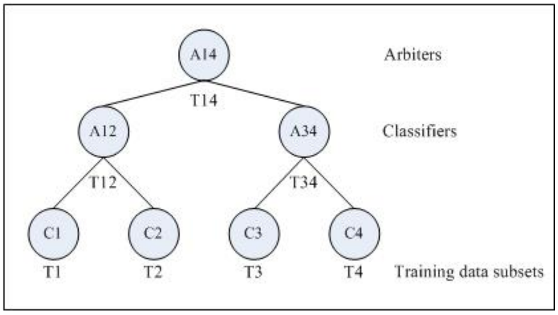

Figure 2: Flowchart to make training dataset for arbiter tree [1]

3.3. Support Vector Machine

For using arbiter tree [1-5] in our experiment, Support Vector Machine (SVM) [9, 10] and Naïve Bayes (NB) [7] will be used as classifiers. SVM [9, 10] is a binary linear classification model with the learning algorithm for classification and regression analysis of data, and recognising the pattern. The purpose of SVM is to separate datasets into classes and discover the decision boundary (hyper-plane). To find the hyper-plane, the maximum distance between classes (margin) will be used with the closest data points on the margin (support vector). In our research, we used the default setting of SVMLight[3] for the SVM classifier model. SVMLight is an implementation of SVM in C.

3.4. Naïve Bayes

Naïve Bayes (NB) algorithm [7] is a classification algorithm based on Bayes' theorem that underlies the naïve assumption that attributes within the same case are independent given the class label [22]. This is also known as the state-of-art Bayes rules [23]. NB [7] constructs the model by adjusting the distribution of the number for each feature. For example, in the text classification, NB regards the documents as a bag-of-words, from which it extracts features. In this research, the NB algorithm was used from the NLTK. NLTK) is a widely-used machine learning, open source, developed using Python and comprising the WordNet interface.

V. EXPERIMENT AND RESULTS

In our experiment, the idea from [1] has been adapted, as we use only two classifiers with one training data. Therefore, from the flowchart for creating the training data in Figure 2 will be changed to that presented in Figure 3 as only two classifers.

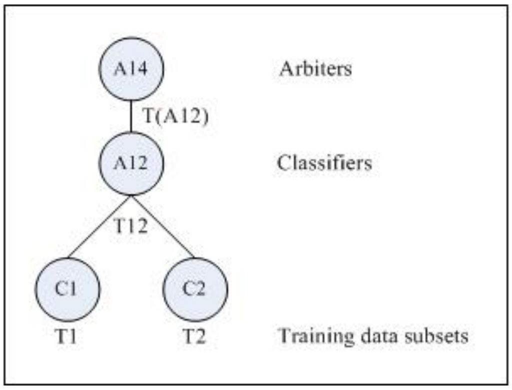

Figure 3: Flowchart to make a training dataset for two classifiers in arbiter tree

[3] http://svmlight.joachims.org/

In order to build the training data, all selection rules from [1] were adapted and used in this experiment. The processes for creating training data are detailed below:

i. Base training data was trained into base classifier, which are SVM [9, 10] and NB [7]. The base training data is yielded from the combination of the sentiment lexicons noted in section 3.1. They were combined by removing the words that duplicate, overlap and contradict in sentiment [24-27].

ii. After obtaining the results from the base classifier, they were united and passed into selection rules. There are three versions of selection rules:

    a. Selection rule 1 is the different results from classifiers 1 and 2

    b. Selection rule 2 is the union of the results from selection rule 1 and the results from classifiers 1 and 2 that they are the same prediction but incorrect

    c. Selection rule 3 is the union of selection rules 1 and 2 and the results from classifiers 1 and 2 that they are the same prediction and correct.

iii. As in the arbiter tree algorithm [1-5], [1] did not mention clearly how to use the selection rules; therefore, they will be adapted from the flowchart presented in Figure 3. The data from selection rules 1 and 2 were trained back in base classifiers; then, their results were combined for processing selection rule 3. This data of selection rule 3 is the final training data for arbiter. The flowchart of these processes is presented in Figure 4.

After obtaining the final training data for arbiter, they were used in the process of final classification for the final prediction results. During this process, the base classifier will be trained by using base training data, while the arbiter is trained by using arbiter training data to classify the test set. Next, their results will go through the process of arbiter rules for the final prediction results. There are two versions of arbiter rules. The first uses the majority vote of prediction from the base classifier and the arbiter prediction. If the results of predictions 1 and 2 are equal, the results from prediction 2 will be used. Conversely, the arbiter results will be used. In the second version, if the results of predictions 1 and 2 are not equal, the different arbiter results will be used. If the results of prediction 1 are equal to those of the correct arbiter, use the correct arbiter results. In contrast, the results from arbiter tree that are incorrect will be used.

The datasets of Tweets and SMS were tested in arbiter tree [1-5]. Their results are presented in Table 1. Following the comparison between arbiter and base classifier (Table 2), the results of Tweets using arbiter rules version 1 did not make any change and achieved the same F-score as SVM [9, 10] at 83.55%; meanwhile, the results from arbiter rules version 2 achieved a better F-score than NB [7] at 81.94%, but still lower than SVM [9, 10]. Conversely, the results of the SMS dataset showed that, the results from arbiter rule version 1 and 2 achieved better F-score than base classifiers at 85.78% and 85.65%, respectively.

Table 1: The results of Tweets and SMS dataset from arbiter tree

	Tweet dataset Avg. F-score (%)	**SMS dataset** Avg. F-score (%)
Arbiter rules version 1	83.55	85.78
Arbiter rules version 2	81.94	85.65

Table 2: The results of Tweets and SMS dataset from base classifiers

	Tweet dataset Avg. F-score (%)	**SMS dataset** Avg. F-score (%)
SVM	83.55	85.49
NB	81.54	85.05

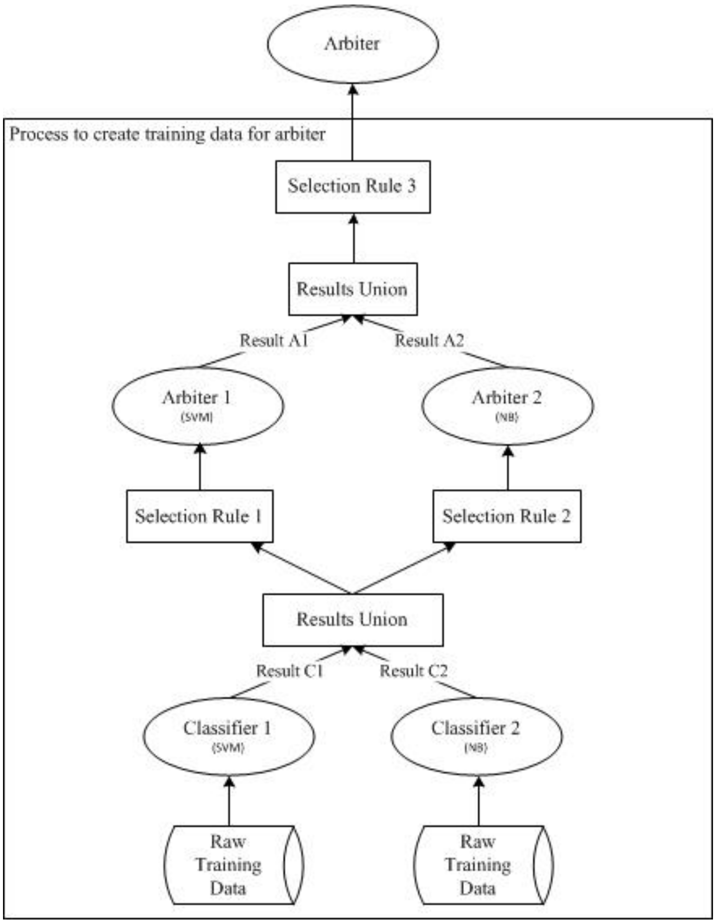

Figure 4: Process for making training data for arbiter

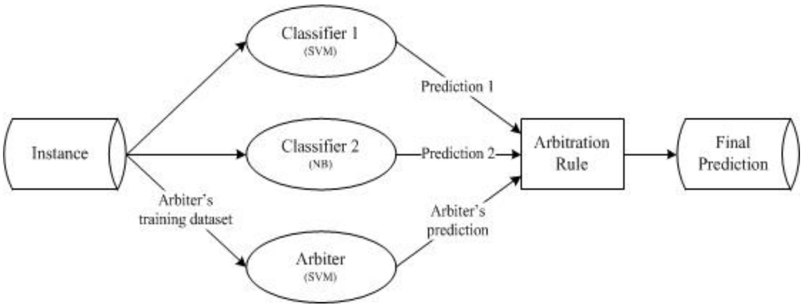

Figure 5: Process for final prediction of the testing data of arbiter tree

VI. CONCLUSION AND FUTURE WORK

In this experiment, the novelty of using the arbiter tree algorithm [1-5] to classify Tweets and SMS datasets has been demonstrated and clearly explained. The use of ensemble learning might not always have achieved the most accuracy; however, the results from the classification of SMS dataset, which we used to evaluate our system, showed that they were able to achieve an F-score of 85.78%, which is better than both base classifiers.

For future work, the sister of arbiter tree [1-5], called the combiner tree [4], will be researched in detail and the combination will be studied with the aim of improving the performance accuracy. Combiner tree [4] is a method that is similar to arbiter tree [1-5] but is trained directly by the training output from base classifiers that have passed the composition rules. The reason that, arbiter tree [1-5] and combiner tree [4] were used, is that the results of them will be used for comparison with the results from stacking [28] for analysing which methods of ensemble learning that achieved better approach in the sentiment analysis task of Tweets.

REFERENCES

[1] P. K. Chan and S. J. Stolfo, "Toward parallel and distributed learning by meta-learning," in *The International Association for the Advancement of Artificial Intelligence (AAAI) workshop in Knowledge Discovery in Databases*, 1993, pp. 227-240.

[2] P. K. Chan and S. J. Stolfo, "Learning Arbiter and Combiner Trees from Partitioned Data for Scaling Machine Learning," in *Conference on Knowledge Discovery and Data Mining (KDD)*, 1995, pp. 39-44.

[3] P. K. Chan, "An extensible meta-learning approach for scalable and accurate inductive learning," 1996.

[4] P. K. Chan and S. J. Stolfo, "On the accuracy of meta-learning for scalable data mining," *Journal of Intelligent Information Systems,* vol. 8, pp. 5-28, 1997.

[5] A. Prodromidis, P. Chan, and S. Stolfo, "Meta-learning in distributed data mining systems: Issues and approaches," *Advances in distributed and parallel knowledge discovery,* vol. 3, 2000.

[6] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *CS224N Natural Language Processing, Project Report, Stanford,* pp. 1-12, 2009.

[7] J. Liangxiao, H. Zhang, and C. Zhihua, "A Novel Bayes Model: Hidden Naive Bayes," *IEEE Transactions on Knowledge and Data Engineering,* vol. 21, pp. 1361-1371, 2009.

[8] R. A. Baldwin, "Use of maximum entropy modeling in wildlife research," *Entropy,* vol. 11, pp. 854-866, 2009.

[9] M. A. Hearst, S. T. Dumais, E. Osman, J. Platt, and B. Scholkopf, "Support vector machines," *Intelligent Systems and their Applications, IEEE,* vol. 13, pp. 18-28, 1998.

[10] N. Cristianini and J. Shawe-Taylor, *An introduction to support vector machines and other kernel-based learning methods*: Cambridge university press, 2000.

[11] S. R. Yerva, Z. Miklos, and K. Aberer, "It was easy, when apples and blackberries were only fruits," presented at the Third Web People Search Evaluation Forum (WePS-3), 2010.

[12] C. Troussas, M. Virvou, K. Junshean Espinosa, K. Llaguno, and J. Caro, "Sentiment analysis of Facebook statuses using Naive Bayes classifier for language learning," in *4th International Conference on Information, Intelligence, Systems and Applications (IISA)*, 2013, pp. 1-6.

[13] G. Salton, *The SMART retrieval system : experiments in automatic document processing*: Prentice-Hall, Inc., 1971.

[14] F. Rosenblatt, *The perceptron, a perceiving and recognizing automaton Project Para*: Cornell Aeronautical Laboratory, 1957.

[15] R. Rojas, "Perceptron Learning," in *Neural Networks: A Systematic Introduction*, ed: Springer Berlin Heidelberg, 1996, pp. 77-99.

[16] C. E. Ortiz, "Introduction to Facebook APIs," ed: http://www.ibm.com/, 2010, pp. 1-20.

[17] T. Wilson, Z. Kozareva, P. Nakov, A. Ritter, S. Rosenthal, and V. Stoyanov, "SemEval-2013 Task 2: Sentiment Analysis in Twitter," in *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval)*, 2013.

[18] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python*: O'Reilly, 2009.

[19] M. Hu and B. Liu, "Mining and summarizing customer reviews," presented at the Proceedings of the tenth ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD) international conference on Knowledge discovery and data mining, Seattle, WA, USA, 2004.

[20] T. Wilson, P. Hoffmann, S. Somasundaran, J. Kessler, J. Wiebe, Y. Choi, *et al.*, "OpinionFinder: a system for subjectivity analysis," presented at the Proceedings of HLT/EMNLP on Interactive Demonstrations, Vancouver, British Columbia, Canada, 2005.

[21] F. Å. Nielsen, "A new ANEW: Evaluation of a word list for sentiment analysis in microblogs," presented at the 7th International Conference Mechatronic Systems and Materials (MSM 2011), Kaunas, Lithuania, 2011.

[22] M. Elangovan, K. I. Ramachandran, and V. Sugumaran, "Studies on Bayes classifier for condition monitoring of single point carbide tipped tool based on statistical and histogram features," *Expert Systems with Applications,* vol. 37, pp. 2059-2065, 2010.

[23] A. Cufoglu, M. Lohi, and K. Madani, "Classification accuracy performance of Naive Bayesian (NB), Bayesian Networks (BN), Lazy Learning of Bayesian Rules (LBR) and Instance-Based Learner (IB1) - comparative study," in *International Conference on Computer Engineering & Systems (ICCES)*, 2008, pp. 210-215.

[24] P. Melville, W. Gryc, and R. D. Lawrence, "Sentiment analysis of blogs by combining lexical knowledge with text classification," presented at the Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, Paris, France, 2009.

[25] B. Yuan, Y. Liu, H. Li, T. T. T. PHAN, G. Kausar, C. N. Sing-Bik, *et al.*, "Sentiment Classification in Chinese Microblogs: Lexicon-based and Learning-based Approaches," *International Proceedings of Economics Development and Research (IPEDR)* vol. 68, 2013.

[26] E. Refaee and V. Rieser, "An Arabic Twitter Corpus for Subjectivity and Sentiment Analysis," in *In 9 th International Conference on Language Resources and Evaluation (LREC'14)*, 2014.

[27] L. Wang and C. Cardie, "A Piece of My Mind: A Sentiment Analysis Approach for Online Dispute Detection," in *The 52nd Annual Meeting of the Association for Computational Linguistics* Baltimore, USA, 2014.

[28] D. H. Wolpert, "Stacked generalization," *Neural networks,* vol. 5, pp. 241-259, 1992.

AUTHORS

Tawunrat Chalothorn is currently a postgraduate researcher at the University of Northumbria at Newcastle.

Jeremy Ellman has a BSc in Experimental Psychology from the University of Sussex, an MSc in Computer Science from Essex University, and a Ph.D. in Computer Science from the University of Sunderland. Jeremy is currently senior lecturer at the Department of Computer Science and Digital Technologies at the University of Northumbria at Newcastle, UK.