High Resolution Sentiment Analysis by Ensemble Classification

Conference Paper · July 2019

CITATIONS
0

READS
8

4 authors, including:

Jordan J. Bird
Aston University
8 PUBLICATIONS **6** CITATIONS
SEE PROFILE

A. Ekárt
Aston University
80 PUBLICATIONS **927** CITATIONS
SEE PROFILE

Diego R. Faria
Aston University
48 PUBLICATIONS **257** CITATIONS
SEE PROFILE

High Resolution Sentiment Analysis by Ensemble Classification

Jordan J. Bird[1], Anikó Ekárt[2], Christopher D. Buckingham[3], and Diego R. Faria[4]

School of Engineering and Applied Science
Aston University, Birmingham, B4 7ET, UK
{birdj1[1], a.ekart[2],c.d.buckingham[3] d.faria[4]}@aston.ac.uk

Abstract. This study proposes an approach to ensemble sentiment classification of a text to a score in the range of 1-5 of negative-positive scoring. A high-performing model is produced from TripAdvisor restaurant reviews via a generated dataset of 684 word-stems, gathered by information gain attribute selection from the entire corpus. The best performing classification was an ensemble classifier of RandomForest, Naive Bayes Multinomial and Multilayer Perceptron (Neural Network) methods ensembled via a Vote on Average Probabilities approach. The best ensemble produced a classification accuracy of 91.02% which scored higher than the best single classifier, a Random Tree model with an accuracy of 78.6%. Other ensembles through Adaptive Boosting, Random Forests and Voting are explored with ten-fold cross-validation. All ensemble methods far outperformed the best single classifier methods. Even though extremely high results are achieved, analysis documents the few mis-classified instances as almost entirely being close to their real class via the model's given error matrix.

Keywords: Sentiment Analysis, Opinion Mining, Machine Learning, Ensemble Learning, Classification

1 Introduction

The applications of Sentiment Analysis are increasingly growing in importance in both the sciences and industry, for example through human-robot interaction [1] and as a business tool in terms of user feedback to products [2], giving more prominence to the field of Affective Computing. Affective Computing [3] is the study of systems capable of empathetic recognition and simulation of human affects including but not limited to sentimental and emotional information encapsulated within human-sourced data.

In this work, various methods of Sentiment Classification are tested on top of a generated set of word-stem attributes that are selected by their ranking of information gain correlating to their respective class. The best model is then analysed in terms of its error matrix to further document the classification results. The main contributions of this work are as follows:

- Effective processing of text to word-stems and information gain based selection suggests a set of 684 attributes for effective classification of high resolution sentiment.
- Single and ensemble Models are presented for the classification of sentiment score on a scale of 1-5 as opposed to the standard three levels of classified sentiment (Positive-Neutral-Negative). In this study, 1 is the most negative result, and 5 is the most positive.
- Methods of Sentiment Classification are based entirely on text and correlative score rather than taking into account metadata (user past behaviour, location etc.), enabling a more general application to other text-based domains.

This paper will document related works in sentiment analysis modeling, a proposed approach to the experiment, the preprocessing and acquisition of sentiment-based data and the training of various single and ensemble models with analysis. Finally, the impact will be reviewed as well as noting the future works enabled by a general text-to-score sentiment classifier.

2 Related Work

Sentiment analysis, or opinion mining, is the study of deriving opinions, sentiments, and attitudes from lingual data such as speech, actions, behaviours, or written text. Sentiment Classification is an approach that can class this data into nominal labels (eg. *'this remark has a **negative** valence'*) or continuous polarities or score which map to their overall sentiment. With the rise of online social media, extensive amounts of opinionated data are available online, which can be used in classification experiments which require large datasets. Negative polarity was used to analyse TripAdvisor reviews [4] on a scale of negative-neutral-positive, findings show that each review rating of one to five stars each have unique distributions of negative polarity. This unique pattern suggests the possibility of further extending the two polarity three-sentiment system to a further five levels. A similar three-class sentiment analysis was successfully trained on Twitter data [5]. Related work with Twitter Sentiment Analysis found that hashtags and emoticons/emoji were very effective attributes to train classifiers [6]. Exploration of TripAdvisor reviews found that separation via root terms, 'food', 'service', 'ambiance' and 'price' provided a slight improvement for machine learning classification [7].

Human-Robot Interaction has increasingly become concerned with Sentiment Analysis as an extra dimension of interaction with a robotic agent. A chatbot architecture was constructed that analysed input and output sentiment of messages and provided it as meta-information within the chatbot's response [8]. The usage of a robot's perception of sentiment is prominent in multiple applications of classification such as mental state [9] [10], facial expressions [11] [12],

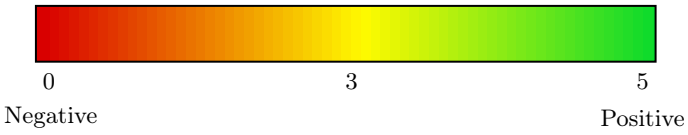

Fig. 1. A diagram to show sentiment gradient, 1 is the most negative score and 5 is the most positive score.

voice/speech [13], and observed physical activities [14] [15].

With increasing availability of computing resources for lower costs, accurate classification is enabled on increasingly larger datasets over time, giving rise to cross-domain application through more fine-tuned rules and complex patterns of attribute values. That is, a model trained on dataset A can be used to classify dataset B. This has been effectively shown through multiple-source data to produce an *attribute thesaurus* of sentiment attributes [16]. Researchers also found that rule-based classification of large datasets are unsuited to cross-domain application, but machine-learning techniques on the same data shows promising results [17].

Observing the results of the related works into social media sentiment analysis (Twitter, TripAdvisor, IMDB etc.) shows the sheer prominence of three level sentiment classification, with only one class for negativity and one for positivity along with a neutral class, with an overall result being calculated with derived polarities. With the large amount of data available correlating to a user's specification of class outside of this range of three, this paper suggests a more extensive sentiment classification paradigm to co-ordinate with user's review scores, to better make use of human-sourced data. To engineer this, the number of data classes in this experiment will be equal to the range of scores available to a user. The end goal would be more points of polarities to give a finer measurement of sentiment. Moreover, many of the state-of-the-art studies experiment with single classifiers, many strong models are produced with Bayesian, Neural Network, and Support Vector Machine approaches but they have not been taken further to an ensemble and explored.

3 Data Acquisition and Processing

3.1 Data Acquisition

A dataset of 20,000 user reviews of London based restaurants was gathered from TripAdvisor[1], in which a review text was coupled with a score of 1 to 5, where 1 is the most negative and 5 is the most positive review. All reviews were in English, and all other meta information such as personal user information was removed,

[1] TripAdvisor - http://tripadvisor.co.uk

this was performed for the more general application of the classifier to all text based data containing opinions. All restaurants from the Greater London Area were chosen randomly as well as the reviews themselves selected at random.

3.2 Preprocessing

Resampling was performed with a 0.2 weighting towards the lower reviews due to the prominence of higher reviews, to produce a more balanced dataset. The resulting dataset of 17,127 reviews with their respective scores can be seen in Table I. It is worth noting that even after weighted re-sampling, there remains

Table 1. Reduced Dataset

Score	***No. of Reviews***
1	2960
2	2983
3	3179
4	3821
5	4283

a higher frequency of positive reviews which will be factored into the analysis of results, specifically in analysis of the classification accuracy of low review scores by way of error matrix observation.

With unprocessed text having few statistical features, feature generation was performed via a filter of word vectors of the string data, based on the statistics of word-stem prominence. Firstly, worthless stopwords were removed from the text using the Rainbow List [18] (ie. words that hold no important significance), and then the remaining words were reduced to their stems using the Lovins Stemmer algorithm [19]. Stopword removal was performed to prevent misclassification of class based on the coincidental prominence of words with no real informative data, and stemming was performed to increase the frequency of terms by removing their formatting eg. time based suffixes and clustering them to one stem.

The process of word vectorisation with the aforementioned filtering produced 1455 numerical attributes mapped to the frequency of the word stem. Further attribute selection was required to remove attributes that had little to no influence of class, which would reduce the computational complexity of classification.

3.3 Attribute Selection

Attribute selection was performed on the 1455 numerical word stem attributes to produce a reduced dataset of 684 attributes. The information gain (IG) of each attribute was calculated and sorted using a simple Ranker algorithm (where higher IG = more correlation to a class).

Information Gain or Relative Entropy is the expected value of the Kullback-Leibler divergence where a univariate probability distribution of a given attribute

is compared to another [20]. This is a measure of a comparison of states given as follows:

$$IG(T, a) = H(T) - H(T|a) \tag{1}$$

This denotes the measured change in entropy, where IG is the Information Gain of Class T and Attribute a.

A cutoff point of 0.001 Information Gain (the lowest measure) was implemented and this removed 771 attributes (word-stems) that were considered to have no impact on the class. This meant that all remaining attributes had a measurable classification ability when it came to sentiment. Of the highest information gain were the word-vector attributes "disappointing" (0.08279), "worst" (0.06808), "rude" (0.0578) and "excellent" (0.05356) - which, regardless of domain, can be observed to have high sentimental polarity.
The dataset to result from this processing was taken forward for classification experiments.

4 Classification Model Background

A range of models were selected following distinctly different methods of deriving a classification, this was for a range of performances as well as for a future ensemble of non-correlative models.

4.1 Rule Based Models

Rule Based Classification is performed via an 'if-then' approach to labelling [21] (eg. *'if it is -15 degrees' then 'it is Winter'*), where a defined number of rules are generated and refined using their classification entropy (see eq. 4).

Zero Rules (ZeroR) Classification is used as a benchmark, it is the simple process of labelling all data with the most common class, ie. if the most common binary class encompasses 51% of the dataset then ZeroR will have a 51% accuracy. One Attribute Rule (OneR) is a common example of effective classification based on selecting the single best rule, such as classifying the Season by temperature in the aforementioned example. Experimentation of multiple rules are then compared based on their minimum-error, and the best one chosen to classify objects.

4.2 Bayes Theorem

Bayesian Models are classification models based on Bayes Theorem. Bayes Theorem [22] is the comparable probability that data point d will match to Class C. Bayes Theorem is given as follows:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2}$$

Where P is 'probability of', A and B are the evidence ie. the probability of $P(A)$ being true is related to the probability of the H with evidence $P(A|B)$. In terms

of this work, this would take inputs of word-stem parameters and classify the text via its highest probability as calculated by the formula.

Naive Bayes classification is given as follows:

$$\hat{y} =_{k \in (1,\ldots,K)} p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k) \tag{3}$$

Where class label y is given to data object k. The naivety in Bayesian algorithms concerns the assumed independence of attribute values (or existence), whether or not the assumption holds true for a data. Naive Bayes (NB) is the application of Bayes' theorem which selects the class based on the lowest risk, ie. based on the evidence it will select the most likely label for the data. Naive Bayes Multinomial (NBM) is a classification method that differs from NB by defining the distribution of each attribute $p(f_i|c)$ as a multinomial distribution [23] eg. a word count within a text.

4.3 Decision Trees

Decision Tree classification is the derivation of conditional control statements based on attribute values, which are then mapped to a tree. Classification is performed by cascading a data point down the tree through each conditional check it meets until an end node is reached, which contains a Class. The growth of the tree is based on the entropy of its end node, that is, the level of disorder in classes found on that node. Eg. if a certain ruleset on a trained tree ends on a node that has 90% sentiment level 4 and 10% sentiment level 5. The entropy of a node with c Class results is calculated as follows:

$$E(S) = -\sum_{i=1}^{c} P_i \times log_2 P_i \tag{4}$$

A tree will continue to grow either until the level of disorder of the node is 0 (ie. 1 Class), or it has been stopped by other means such as a set stopping length (pre-pruning) or reduction after the entire tree is grown (post-pruning).

A random tree classifier is a decision tree generated in which k-random attributes are selected at each node [24]. The model is simple since no pruning is performed and thus an overfitted tree is produced to classify all input data points. Therefore, cross-validation is used to create an average of the best performing random trees, or with a testing set of unseen data.

J48 is a decision tree operating the C4.5 algorithm [25]. Whilst growing the decision tree, information entropy is used to calculate a best split at each node, ie. that which enriches both splits with differing classes (or each class in a binary classification problem). If features are worthless for the problem with zero information gain, higher nodes on the tree are used to classify the current instance, though, this results in the waste of computational resources.

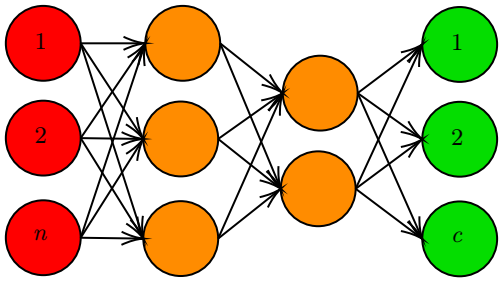

Fig. 2. A Simplified Deep Neural Network Diagram With n Input Attributes, Two Hidden Layers of 3, 2 Neurons and c Output Classes.

4.4 Neural Networks

An Artificial Neural network (ANN) is a system loosely inspired by the natural brain for weighted classification or regression [26]. In observation of Fig. 2, a simplified diagram of a Neural Network - input neurons (attributes) are generated for layer 1, where n is the attributes available to the classifier. A number of hidden layers for classification are generated, in the figures' case, two layers of three and two nodes respectively. Finally, at the end of the diagram, c output nodes that map to c number of classes. Each link between neurons carries a weight, which is trained. The network diagram is considered a Deep Neural Network due to the existence of more than one hidden layer of neurons [27]. For regression problems (eg. prediction of house price), a single output node of real numbers is used.

A Multilayer Perceptron (MLP) is a neural network that through optimisation (or learning) updates the weights between neurons via the process of backpropagation, ie. to derive a gradient which is used to calculate connection weighting values [28]. Backpropagation is the process of calculating the error at the output (the green layer in Fig. 2) and fed backwards throughout the network to distribute new weightings, to refine the system and reduce the errors or loss.

4.5 Support Vector Machines

A Support Vector Machine (SVM) is the classification of entities by calculation of an optimised separator between groups, formed via an n-dimensional hyperplane within the given problem space [29]. An optimal solution considers the margin between the separating hyperplane and the classifer points, with a larger average margin defining a best possible classification vector.

Sequential Minimal Optimization (SMO) is an algorithm to implement an SVM with several classes by reducing the quadratic function to a linear constraint [30]. This is performed by breaking the optimisation problem into the smaller subproblems, which are then analytically solved [31].

5 Ensemble Method Background

An ensemble is a fusion of classifiers at each prediction, where a result is calculated based on the results of the models being fused. This section will detail the theory behind ensemble methods as well as those chosen for this experiment.

5.1 Voting

Voting is a simple democratic process that takes into account predictions of the models encompassed. Each model will subsequently vote on each class or a regression result in turn, and the final decision is derived through a selected operation:

- **Average of Probabilities** - Models vote on all classes with a vote equal to each of its classification accuracies of said class. eg. if a model can classify a binary problem with 90% and 70% accuracies, then it would assign those classes 0.9 and 0.7 votes respectively if voting for them. The final output is the class with the most votes.
- **Majority Vote** - All models will vote on the class it predicts the data to be, and the one selected is the class with the most votes.
- **Min/Max Probability** - The minimum or maximum probabilities of all model predictions are combined and class is selected based on this value.
- **Median** - For regression, all models will vote on a value, and the one selected will be the median of all of their values. Eg. if two models in a voting process vote for values of 1.5 and 2, then the output of the classifier will be 1.75.

5.2 Adaptive Boosting

Adaptive Boosting (AdaBoost) is a Gödel Prize winning meta algorithm for the improvement of classifiers [32]. AdaBoost follows an iterative approach by, at each iteration, selecting a random training subset to improve on the previous iteration's results, and further uses those results to produce a weighting of classification in a combination. This is given by:

$$F_T(x) = \sum_{t=1}^{T} f_t(x) \tag{5}$$

Where F is the set of t models classifying the data object x [33].

5.3 Random Forest

A Random Forest is an ensemble of Random Trees through Bootstrap Aggregating (bagging) and Voting [34]. Training is performed through a bagging process where multiple random decision trees are generated, a random selection of data is gathered and trees are grown to fit the set. Once training is completed, the generated trees will all vote, and the majority vote is selected as the predicted class (See section V subsection A, *'Voting'*). Random Forests tend to outperform Random Trees due to their decreasing of variance without increasing of the model bias.

5.4 Adopted Approach

Manual tuning noted the performance of Random Forests, Adaptive Boosting, and Vote (average probabilties) and thus they were used in this experiment. Adaptive Boosting was performed on Random Tree as well as Random Forest (*ensemble of ensemble*). Voting on average probabilties were tested with models that had techniques unique to one another; firstly, Naive Bayes Multinomial, Random Tree and Multilayer Perceptron. Secondly, Random Forest, Naive Bayes Multinomial, and Multilayer Perceptron.

6 Results

Training and evaluation of models were performed using 10-fold cross validation ($k = 10$), in which one fold of the data is used for testing a classifier trained on the remaining nine folds of data. This is performed ten times, thus testing and training encompass all ten folds. This is done to prevent a model being overfitted to the dataset in question and therefore having no useful application. Leave-one-out cross-validation ($k = n$ *data points*) was not performed due to computational resource requirement of such validation. Where required, all random seeds for model training were set to 1. Random numbers are generated by the Java Virtual Machine (JVM).

6.1 Single Classifiers

Results of single classifiers can be seen in Table II. The two best models were both Decision Tree algorithms, with the best being Random Tree with an accuracy of 78.6%.

The selected methods had the following parameters set:

- **MLP** - Three layers of 5, 10, and 15 neurons with training at 2000 epochs.
- **RT** - No limits imposed on the number of random attributes selected or the tree depth, a minimum total weight of instances on a node set to 1.
- **J48** - As for RT but with tree depth pruning at a confidence factor of 0.25.
- **SMO** - A complexity parameter of 1.0 with a Logistic calibrator.

6.2 Ensemble Classifiers

The models applied in the ensemble experiments (Table III, column 2) were sourced from the training experiments performed previously, seen in Table II, and are therefore more directly comparable as a sum to their parts.

Results of ensemble methods and their classifiers can be seen in Table III. The best model was a Vote of Average Probabilities by the three previously trained models of Random Forest, Naive Bayes Multinomial, and a Multilayer Perceptron.

The selected methods had the following parameters:

Table 2. Classification Accuracy of Single Classifier Models

Classifier	**Classification Accuracy**
OneR	29.59%
MLP	57.91%
NB	46.28%
NBM	59.02%
RT	**78.6%**
J48	75.76%
SMO SVM	68.94%

Table 3. Classification Accuracy of Ensemble Models

Ensemble	**Classifiers**	**Classification Accuracy**
RF	N/A	84.9%
Vote	NBM, RT, MLP	80.89%
Vote	RF, NBM, MLP	**91.02**%
AdaBoost	RT	79.36%
Adaboost	RF	84.93%

- **Random Forest** - Bag size was 100% of the provided data, tree maximum depth was unlimited, selected features were not limited, and 100 iterations were performed for each forest.
- **Vote** - Class voting was decided upon via the highest average of probabilities.
- **AdaBoost** - 10 iterations of each model were performed for the Adaptive Boost.

6.3 Analysis

ZeroRules benchmarking resulted in an accuracy of 24.88%, all of the models far outperformed the benchmark with the exception of OneR which had an only slightly better accuracy of 29.59% (+4.71), this is due to One Rule Classification having diminishing returns on higher dimensionality datasets, the one in this particular experiment taking place in 684-dimension space.

All ensemble approaches to classification outperformed the single classifiers. Interestingly an 'ensemble of ensemble' approach produced better results when it came to AdaBoost of a Random Forest (+0.03%), and most importantly factoring in the Random Forest within a vote model along with Naive Bayes Multinomial and a Multilayer Perceptron, which produced a classification accuracy of 91.02%.

In terms of the error matrix (Table IV), it is observed that the best model put forward misclassified predictions at a gradient around the real class, due to the crossover of sentiment based terms. Most prominently, classes 4 and 5 were the most difficult to predict, and further data analysis would give concrete examples of lingual similarity between reviews based on these two scores.

Table 4. Error Matrix for the Classifications Given by the Vote(RF, NBM, MLP) Model

Predicted Class						
1	***2***	***3***	***4***	***5***		
2919	26	7	3	5	***1***	**Real Class**
42	**2887**	25	19	10	***2***	
55	71	**2873**	126	54	***3***	
14	25	139	**3090**	544	***4***	
17	13	63	288	**3902**	***5***	

7 Conclusion

To conclude, this work presented results from models for classification of multi-level sentiment at five distinct levels after performing effective feature extraction based on lingual methods. The best single classifier model was a Random Tree with a classification accuracy of 78.6%, which was outperformed by all applied ensemble methods and their models. The best overall model was an ensemble of Random Forest, Naive Bayes Multinomial, and a Multilayer Perceptron through a Vote of Average Probabilities, with a classification accuracy of 91.02%.

The findings suggest future work in the development of text-based ensemble classifiers as well as their single classification parameters, due to the trained models in this experiment successfully being improved when part of an ensemble. The effectiveness of Neural Networks for sentiment classification is well documented [35] implying that further work with more computational resources than were available for this experiment is needed due to the low results achieved. Furthermore, leave-one-out cross validation has been observed to sometimes be more effective than k-fold cross validation [36] but proved too computationally expensive for the dataset in this experiment, therefore exploration of this method of training validation is needed. Voting had very promising results, which could be refined further through adding new trained models as well as experimentation with different democratic processes.

Successful experiments were performed purely on a user's message and no other meta information (eg. previous reviews, personal user information) which not only shows effectiveness in the application in the original domain of user reviews, but also a general application to other text-based domains such as chatbots and keyword-based opinion mining. The application of the classifiers put forward in this paper are useful in the aforementioned domains, though future work should encompass a larger range of sources to smooth out some of the remaining domain-specific information.

In terms of contribution, a comparison of the results of this study and the state of the art can be seen in table 5. With a far higher resolution than the 3 (Pos-Neu-Neg) or 2 (Pos-Neg) observed in many works, a high accuracy of 91.02 was still achieved through the method of ensemble. It can be observed that the

Table 5. Indirect comparison of this study and state-of-the-art sentiment classification work (different datasets)

Work	Resolution	Accuracy
This study (Ensemble - Vote)	5	**91.02**%
Read [5]	3	84.6%
Bollegala, et al. [16]	2	83.63%
Denecke [17]	2	82%
This study (Single - RT)	5	78.6%
Kouloumpis, et al. [6]	3	75%

best single classifier fits within related works whereas an ensemble approach with the same dataset far outperforms related works.

8 Acknowledgements

This work was supported by the European Commission through the H2020 project EXCELL (https://www.excell-project.eu/), grant No. 691829.
This work was also partially supported by the EIT Health GRaCEAGE grant number 18429 awarded to C.D. Buckingham.

References

1. C.-W. Lee, Y.-S. Wang, T.-Y. Hsu, K.-Y. Chen, H.-Y. Lee, and L.-s. Lee, "Scalable sentiment for sequence-to-sequence chatbot response with performance analysis," *arXiv preprint arXiv:1804.02504*, 2018.
2. H. Cui, V. Mittal, and M. Datar, "Comparative experiments on sentiment classification for online product reviews," in *AAAI*, vol. 6, pp. 1265–1270, 2006.
3. C. Lisetti, "Affective computing," 1998.
4. A. Valdivia, M. V. Luzón, and F. Herrera, "Sentiment analysis in tripadvisor," *IEEE Intelligent Systems*, vol. 32, no. 4, pp. 72–77, 2017.
5. J. Read, "Using emoticons to reduce dependency in machine learning techniques for sentiment classification," in *Proceedings of the ACL student research workshop*, pp. 43–48, Association for Computational Linguistics, 2005.
6. E. Kouloumpis, T. Wilson, and J. D. Moore, "Twitter sentiment analysis: The good the bad and the omg!," *Icwsm*, vol. 11, no. 538-541, p. 164, 2011.
7. B. Lu, M. Ott, C. Cardie, and B. K. Tsou, "Multi-aspect sentiment analysis with topic models," in *2011 11th IEEE International Conference on Data Mining Workshops*, pp. 81–88, IEEE, 2011.
8. J. J. Bird, A. Ekárt, and D. R. Faria, "Learning from interaction: An intelligent networked-based human-bot and bot-bot chatbot system," in *UK Workshop on Computational Intelligence*, pp. 179–190, Springer, 2018.
9. J. J. Bird, L. J. Manso, E. P. Ribiero, A. Ekart, and D. R. Faria, "A study on mental state classification using eeg-based brain-machine interface," in *9th International Conference on Intelligent Systems*, IEEE, 2018.

10. J. J. Bird, A. Ekart, C. D. Buckingham, and D. R. Faria, "Mental emotional sentiment classification with an eeg-based brain-machine interface," in *The International Conference on Digital Image and Signal Processing (DISP'19)*, Springer, 2019.
11. D. R. Faria, M. Vieira, F. C. Faria, and C. Premebida, "Affective facial expressions recognition for human-robot interaction," in *Robot and Human Interactive Communication (RO-MAN), 2017 26th IEEE International Symposium on*, pp. 805–810, IEEE, 2017.
12. D. R. Faria, M. Vieira, and F. C. Faria, "Towards the development of affective facial expression recognition for human-robot interaction," in *Proceedings of the 10th International Conference on PErvasive Technologies Related to Assistive Environments*, pp. 300–304, ACM, 2017.
13. A. Nogueiras, A. Moreno, A. Bonafonte, and J. B. Mariño, "Speech emotion recognition using hidden markov models," in *Seventh European Conference on Speech Communication and Technology*, 2001.
14. M. Vieira, D. R. Faria, and U. Nunes, "Real-time application for monitoring human daily activity and risk situations in robot-assisted living," in *Robot 2015: Second Iberian Robotics Conference*, pp. 449–461, Springer, 2016.
15. D. A. Adama, A. Lotfi, and C. Langensiepen, "Key frame extraction and classification of human activities using motion energy," in *UK Workshop on Computational Intelligence*, pp. 303–311, Springer, 2018.
16. D. Bollegala, D. Weir, and J. Carroll, "Using multiple sources to construct a sentiment sensitive thesaurus for cross-domain sentiment classification," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pp. 132–141, Association for Computational Linguistics, 2011.
17. K. Denecke, "Are sentiwordnet scores suited for multi-domain sentiment classification?," in *Digital Information Management, 2009. ICDIM 2009. Fourth International Conference on*, pp. 1–6, IEEE, 2009.
18. A. K. McCallum, "Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering." http://www.cs.cmu.edu/ mccallum/bow, 1996.
19. J. B. Lovins, "Development of a stemming algorithm," *Mechanical Translation and Computational Linguistics*, vol. 11, pp. 22–31, 1968.
20. S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.
21. C. Zhang and S. Zhang, *Association rule mining: models and algorithms*. Springer-Verlag, 2002.
22. T. Bayes, R. Price, and J. Canton, "An essay towards solving a problem in the doctrine of chances," 1763.
23. A. McCallum, K. Nigam, *et al.*, "A comparison of event models for naive bayes text classification," in *AAAI-98 workshop on learning for text categorization*, vol. 752, pp. 41–48, Citeseer, 1998.
24. A. M. Prasad, L. R. Iverson, and A. Liaw, "Newer classification and regression tree techniques: bagging and random forests for ecological prediction," *Ecosystems*, vol. 9, no. 2, pp. 181–199, 2006.
25. J. R. Quinlan, *C4. 5: programs for machine learning*. Elsevier, 2014.
26. R. J. Schalkoff, *Artificial neural networks*, vol. 1. McGraw-Hill New York, 1997.
27. J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

28. F. Rosenblatt, "Principles of neurodynamics. perceptrons and the theory of brain mechanisms," tech. rep., CORNELL AERONAUTICAL LAB INC BUFFALO NY, 1961.
29. C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.
30. J. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," 1998.
31. C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM transactions on intelligent systems and technology (TIST)*, vol. 2, no. 3, p. 27, 2011.
32. Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.
33. R. Rojas, "Adaboost and the super bowl of classifiers a tutorial introduction to adaptive boosting," *Freie University, Berlin, Tech. Rep*, 2009.
34. T. K. Ho, "Random decision forests," in *Document analysis and recognition, 1995., proceedings of the third international conference on*, vol. 1, pp. 278–282, IEEE, 1995.
35. M. Ghiassi, J. Skinner, and D. Zimbra, "Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network," *Expert Systems with applications*, vol. 40, no. 16, pp. 6266–6282, 2013.
36. R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Ijcai*, vol. 14, pp. 1137–1145, Montreal, Canada, 1995.