A Comparison of Naïve Bayes and Decision Tree Approaches to Twitter Sentiment Analysis in Healthcare as a Binary Classification Problem

Kieran Fox
School of Computer Science
University of Nottingham
psykf2@nottingham.ac.uk

Ashwini Bharat Doke
School of Computer Science
University of Nottingham
psxad11@nottingham.ac.uk

Aditya Kumar Sasmal
School of Computer Science
University of Nottingham
psxas24@nottingham.ac.uk

Shruthi Chinni
School of Computer Science
University of Nottingham
psxsc19@nottingham.ac.uk

Abstract: This study explores the use of sentiment analysis on healthcare-related text data in the era of big data. The goal is to understand public opinion and sentiment toward healthcare topics using Apache Spark libraries[1]. Businesses generate vast amounts of text data from various sources, making sentiment analysis a vital tool in natural language processing. However, modern technologies and architectures are required to process this data effectively. This paper evaluates the effectiveness of Apache Spark libraries for sentiment analysis and demonstrates their ability to extract valuable insights from vast amounts of data, contributing to the advancement of sentiment analysis techniques in healthcare.

The purpose of this study is to create a classification model for measuring concern in relation to healthcare using Twitter data collected in 2015 from major health news organisations. [2] The purpose of the study is to demonstrate the potential of big data techniques to test different classification models, optimizing the result of classification in its accuracy to categorize sentiments into positive and negative polarities from social media posts.

Comparison of these algorithms through calculation of the harmonic mean of precision and recall, denoted by F1 score. This will determine whether the Naïve Bayes algorithm outperforms a Decision Tree approach[3]. Our research will optimize the model parameters to improve their accuracy, to uncover insights into public healthcare opinions. These insights will support the news media's approach to healthcare issues and its influence on public opinion. The results obtained from our optimized models will have significant implications for healthcare policy and practice.

Keywords— Healthcare, Sentiment, Twitter, Naïve-Bayes, Decision Tree

I. INTRODUCTION

Sentiment analysis, also known as opinion mining, has gained significant attention in recent years due to its potential applications in understanding public sentiment and opinion toward diverse topics[4]. In the healthcare domain, sentiment analysis can provide valuable insights into the sentiments expressed in healthcare-related tweets, enabling authority to better understand and improve decision-making public perceptions and experiences. In this study, we focus on two popular algorithms: Naïve Bayes and Decision Trees.

Naïve Bayes is a probabilistic algorithm that utilizes conditional probability to classify words based on their occurrence within different sentiments[5], [6]. Naïve Bayes has been widely used in sentiment analysis tasks due to its simplicity, efficiency, and ability to handle large datasets. The Decision Tree algorithm partitions the feature space into regions based on the categorization of data into labels. This methodology enables the algorithm to classify instances based on the regions they fall into, providing an effective means of sentiment categorization.

A study conducted on sentiment analysis of geotagged tweets from UK cities during the third national Covid-19 lockdown uses three methods: lexicon-based, machine-learning-based, and hybrid[7], [8]. The study concludes that the Support Vector Classification (SVC) model using Bag-of-Words (BoW) or TF-IDF feature models has the best performance, achieving a classification accuracy of 71%[9]. However, because of a lack of training data, the accuracy of classifiers utilizing the Word2Vec embedding approach is low. As a result of this research, we will favor the BoW feature model in converting our text information to a format suitable for machine learning. This aims to improve the accuracy of both classifiers compared to other methods.

The models we will use have been given similar applications, such as the utilization of Naïve Bayes for sentiment analysis on Twitter data. One paper [3] discusses the extraction of sentiment from Twitter using Hadoop and MapReduce architecture. The study focuses on movie reviews, feedback, and comments available on Twitter collected from a microblogging website. and employs Naïve Bayes for sentiment analysis. The paper highlights the importance of preparing the data for analysis, and cleaning of our dataset will follow a similar process, as it is a major challenge to extract useful information from unstructured data. Another study[10] used a series of different methods, including Naïve Bayes, logistic regression and SVM (Support Vector Machine) but concluded that logistic regression performed better than all others for their classification problem. In our study, it is useful to compare the research of others to understand that our results will vary, and though Naïve Bayes has proven successful in similar problems, this does not evidence the idea that it will outperform a Decision Tree approach for our specific data and classification problem.

Another classification model [11]presents a sentiment analysis system for Twitter data using a KNN algorithm with unigram, bigram, and ngram features, with the model categorizing information as positive, neutral, or negative. The system achieved 65.33% accuracy on the #USairline dataset. The authors plan to improve accuracy in the future using deep learning techniques. They used the sklearn library for model selection, label encoding, and evaluation, which included the use of a confusion matrix to derive evaluation parameters such as accuracy, precision, recall, and F1 score.

The objective of our study is to compare the performance of Naïve Bayes and Decision Tree algorithms in sentiment analysis of healthcare tweets. Our study involves the use of a dataset of tweets collected from various news publications, where our Naïve Bayes and decision tree models will predict the sentiments of each processed tweet, using the F1 score as a comparison of precision and recall assessing their performance[12]. The data will undergo rigorous pre-processing and feature extraction to ensure a robust and objective assessment of their performance, making significant contributions to the advancement of sentiment analysis techniques.

II. METHODOLOGY

The purpose of this research is to determine whether the Naïve Bayes algorithm outperforms the Decision Tree model in sentiment analysis tasks. To achieve this goal, our process involved several stages of data pre-processing and feature extraction/selection before training and evaluating our models.

We began by collecting a dataset from a reliable source. The dataset was then carefully studied to ensure its readiness for sentiment analysis. Irrelevant information such as usernames, URLs, and other extraneous data was dropped to focus solely on the text content. Standard pre-processing techniques were also applied, including removing stop words, punctuation marks, and special characters. These measures aimed to eliminate noise and enhance the quality of the dataset, resulting in clean data for further analysis.

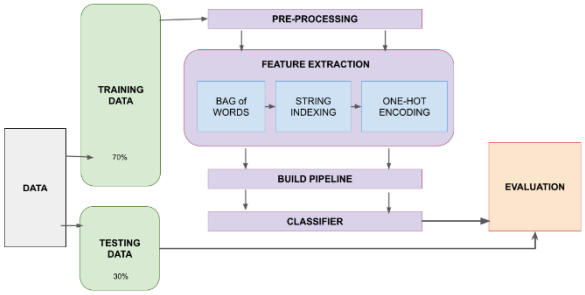

Figure 1: Workflow

After obtaining a balanced dataset, we pre-processed the text for model compatibility. Tokenization, stemming, and lemmatization techniques were applied to convert raw text into a structured format suitable for machine-learning models, allowing for the initial labelling of the data. Tokenization involved splitting the text into individual tokens or words, enabling further analysis on a per-word basis. Stemming was applied to reduce words to their root form, which helps to capture the core meaning of words and reduce feature dimensionality. [5]Lastly, lemmatization mapped words to their base or dictionary form, enhancing semantic analysis by considering different forms of the same word as a single feature[13]. At this stage, the text data was processed enough to be able to calculate sentiment scores for each tweet, with the use of a simple NLP sentiment analyzer. This categorized the data to attach a label of 'Positive' or 'Negative' to each entity.

In avoiding any potential bias towards specific agencies or sentiments, we addressed the issue of data skewness by investigating the distribution of terms across different sentiments. By analyzing the occurrence of terms in each sentiment category, valuable insights were obtained on the correlation between publications and sentiments, developing a better understanding of how this feature of the dataset affects the outcome. Through this, we were able to discover that there were

approximately 2.45 times the positive sentiment labels compared to negative ones.

To train the machine learning models, numeric data is required. As such, the text data contained in the tweets could not be passed into the model. A bag-of-words (BoW) representation was used for this purpose, transforming the text such that each word can be represented as a unique token that can later be processed. For the same reason, the publication names and sentiment labels were converted to numbers, though this was performed using a string indexer approach rather than BoW. Finally, all other columns of our data frame containing text were dropped, leaving features of the BoW representations of the tweets, the indexed publications, and values for the year and month the tweet was posted.

With preprocessing of the data complete, we separated it into training and testing sets using a 70/30 split. This strikes a balance between having enough data to train the model whilst being able to test the performance of the fitted models on a substantial portion of the dataset. Beginning to create pipelines for machine learning, a one-hot encoder was used to convert the categorical publication names into numeric data without introducing or ordering any hierarchy between categories.

We then use a feature transformer that combines the input columns into a single vector column. The models for Naïve Bayes and Decision Tree themselves are defined, with two grids of parameters defined which are appropriate for each model. The same evaluation metric is used for both models, as well as the same features and labels, so we can accurately compare the effectiveness of the models later due to the consistency in the creation of their individual pipelines.

Once these pipelines are created using stages of an indexer, encoder, vector assembler, and cross validator as outlined above, they can each be fit to the training data. These trained models can each be used to produce a set of predictions on the testing data. Finally, we use the evaluator defined earlier to produce an F1 score as an accuracy metric, comparing the precision and recall of each set of predictions.

Overall, this methodology aims to comprehensively evaluate the Naïve Bayes and Decision Tree algorithms for sentiment analysis[14]. By conducting rigorous data preprocessing, feature extraction, and model evaluation, we ensured a robust and objective assessment of their performance, contributing to the advancement of sentiment analysis techniques.

III. Experimental Techniques

In this research, we utilized data collected from various news publications on Twitter, including BBC, CNN and many other companies. Our experiment aims to use this data to analyze and draw insights into the user sentiments expressed on social media.

Data preprocessing is a critical step for our research to ensure accuracy in our sentiment analysis. Our process involved cleaning, filtering, and transforming the raw Twitter data into a format suitable for analysis. To clean the data, we removed URLs, special characters, and stop words. Data transformation was performed using natural language processing techniques such as tokenization, stemming, and lemmatization. The pre-processed data was used for exploratory analysis, visualisation, and modelling, and obtaining initial sentiment labels.

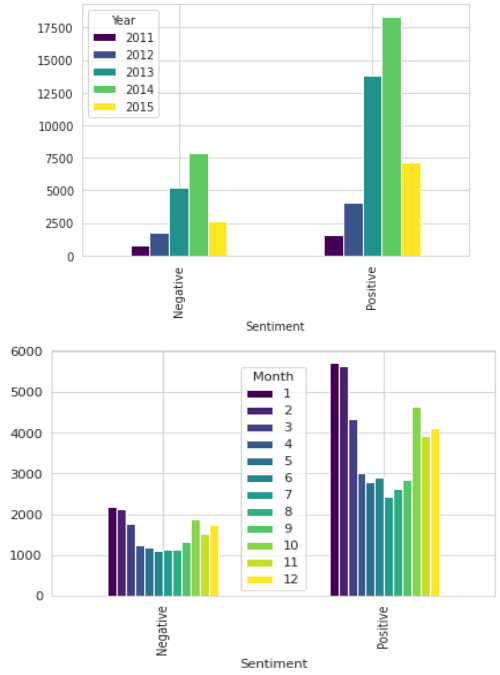

Figure 2: Distribution of Sentiments across Year and Month

[15], [16]Visualisation is an essential part of this process, providing valuable insight into the significance of features on sentiment. While the content of each tweet in BoW form is the main feature to be studied, we can view the distribution of positive and negative sentiments across different timeframes and between publications before creating machine learning pipelines. As shown below in Fig. 1, the distribution of sentiment against the year and month a tweet was posted is relatively consistent with the ratio between the total positive and negative instances in the dataset.

A different method of visualisation is employed when comparing publications and the distribution of sentiment in their tweets. For this, the seaborn library is used instead of Matplotlib, improving the output by

integrating interactivity into the graph. As there is a clear variation in the proportion of positive and negative sentiments between companies, it is important to be able to closely examine the data. Therefore, our plot allows for zooming on areas of interest and hovering over a section of a bar in the plot provides information on the sentiment, publication, and exact number of instances, providing information into how the machine learning models will use data other than the tweet content itself.

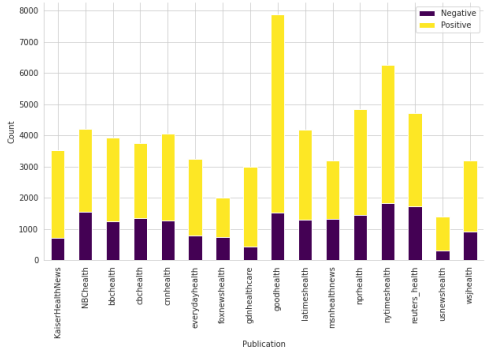

Figure 3: Distribution of Sentiments across publications

Effective comparison of performance requires some factors in both models to be consistent. This ensures a balanced experimentation process, where the outcome accurately indicates which algorithm performs better. As outlined earlier, both models use a machine-learning pipeline consisting of a string indexer, one-hot encoder, vector assembler and cross-validation measure. Of these, only the cross validator differs between the Naïve Bayes and Decision Tree models.

The Naïve Bayes model specifies a multinomial probability distribution, used to model the frequency distribution of features in the training data, treating each word as a separate feature. Instead of a probability distribution, the Decision Tree model requires specification of the maximum depth of the decision tree and the maximum number of bins to be used to discretize continuous features in the dataset. Through our experimental process, these values are optimized to increase the F1 score of the decision tree model. Each model uses a parameter grid, containing appropriate values for each model, which will search for the optimal combination of hyperparameters for each machine learning model. These are used in cross-validation measures, which aim to reduce overfitting and include an evaluator to calculate the F1 score for each fold.

With these measures being used to fit each pipeline to the training data, we can obtain predictions on the testing set. We then display the first 25 predicted values for each model for a straightforward comparison of the outputs. For the result, we use the evaluator to get an F1 score for each model based on the predictions.

This is useful as it is a combination of the precision and recall scores of a model, whereas a standard calculation of accuracy would be less suited to this dataset since each class of the dataset has a different number of samples. If every prediction of sentiment was positive, the result using an accuracy metric would be 71%, whereas, with an F1 score, this value would, correctly, be significantly lower, representing the inability of the model to distinguish between the two classes in that case.

IV. Results and Discussion

After conducting our experiments using both methods of classification, we obtained two final, optimized values of the F1 score. For Naïve Bayes, this used a parameter grid containing a smoothing parameter including values in increments of 0.2 between 0 and 1, representing the amount of smoothing to apply. For the decision tree classifier, the parameter grid contained values for the maximum depth and maximum number of bins. These values were [2, 5] and [32, 64] respectively.

With these parameters, both models were fitted to the training data. Naïve Bayes took approx. 80 seconds to train, while the decision tree took over 13 minutes. When applied to the testing data, this produced F1 scores of 0.854 for Naïve Bayes and 0.743 for the decision tree.

The numbers of true and false positives/negatives for each model were then represented in a confusion matrix, illustrating further the successes and weaknesses between classification methods.

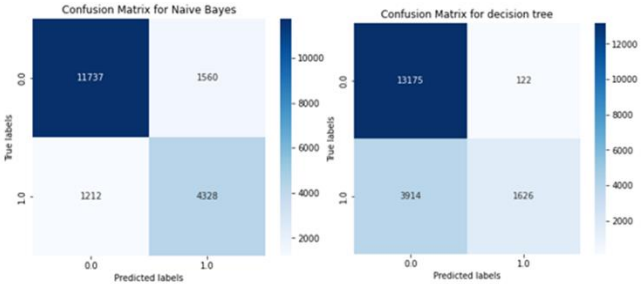

Figure 1: Confusion Matrix for both the models

As is evident from examining these matrices, the decision tree has results skewed towards positive predictions, implying that it is biased towards this output. While Naïve Bayes produces a similar number of false positives and false negatives, the number of false negatives from the decision tree model is very low compared to the false positives. This highlights its bias caused by overfitting, especially because the number of false positives is greater than the number of true negatives, hence the boundaries between branches of the tree must be skewed towards positive.

The commonly used Bayes model is not skewed towards either sentiment, resulting in an overall higher F1 score. Additionally, this model took far less time to train and so it is more efficient than the decision tree approach. For these reasons, it is clear that the Naïve Bayes model was better suited to sentiment analysis of unstructured Twitter data than the decision tree. This is highly significant as there was no valuable metric for which the decision tree outperformed, reflecting its inefficiencies and inaccuracy under the same preprocessing method when compared to Naïve Bayes.

Cleaning and preprocessing using our methodology proved highly effective. With the numeric data passed into the Bayes classifier, it took just over a minute to train the model. This shows that the BoW representation of Twitter data is well-suited to a big data approach to sentiment analysis, and sufficient cleaning of the text before applying this enables vast amounts of data to be processed very quickly. There were over 60,000 entries in the dataset used for this study, so using a 70/30 split of the data, over 42,000 rows were used to train the model. Because such a large amount of data was used, the decision tree model was slow in training, even with parameters optimized for performance, though Naïve Bayes was greatly efficient through strong assumptions of independence made between the features.

Ensuring fair comparison between the two algorithms meant using the same training data to fit both models, along with avoiding additional measures for one model over the other, such as pruning the decision tree to prevent overfitting. However, some optimization was performed on the decision tree model by altering its parameters. Initially, with a maximum depth of the tree of 1, the model could train only slightly slower than Bayes, though the F1 score this provided was low, at just 0.66. Increasing the maximum depth to 2 yielded a much better score of 0.78, but was too computationally expensive, taking over an hour to fit. With some further changes to the parameter grid, we obtained the final score of 0.74, acknowledging the tradeoff between the computation required and the accuracy produced. With the dataset's size, this can mean that decision trees are not as suitable as other classification methods when the tree is deep and complex.

An issue arose in this study during the visualisation of the distribution of sentiment. With a focus on presenting the data interactively and attractively, our data was converted from Spark data frames to Pandas. This is a major problem in big data applications as the expectation is that numerous nodes will be used to distribute the computation. By converting the data to a format that does not support this method of processing the data, we risk running out of memory and impacting the performance of the program. With a larger dataset, this may be infeasible due to the issue of scalability using a Pandas data frame, thus in this instance, we were fortunate in being capable of fitting all the data on a single machine.

Unintuitively, the amount of time spent training the models was inconsequential when it came to their performance. Even with our most computationally expensive implementation of the decision tree algorithm, the model performed worse than the less optimized Naïve Bayes model. Due to the high F1 score achieved by the Bayes model in its initial implementation, we were able to use it as a ‘baseline’ to compare the decision tree. As the assumptions made by Naïve Bayes simplify the calculations of the probabilities needed for classification, it can be trained quickly and with relatively little computational resources. This makes it remarkably effective in binary classification tasks, which have been well-explored in other studies. Optimization of the decision tree for comparison aimed to match the high F1 score produced, though we were unable to achieve such a high value, further highlighting the benefit of using Naïve Bayes for similar tasks.

Overall, Naïve Bayes outperformed the decision tree in our chosen metric of F1 score, the harmonic mean of precision and recall, as well as showing next to no bias, proving it is generalized well to new data compared to the other model. While decision trees may be easier to interpret, being very intuitive and easy to understand through visualisation of the decision-making process, it was simply unable to match the accuracy of Naïve Bayes and was not as efficient in training.

V. Conclusions

The purpose of this study was to compare two methods of binary classification through the use of two models: Naïve Bayes and decision tree. Our experiment followed a process of data cleaning, preprocessing, feature extraction, visualisation, machine learning and evaluation. Originally the dataset was comprised of the tweet, the time it was posted, and the name of the publication associated with the tweet. Preprocessing and feature extraction enabled us to use the features of the timestamp that were most valuable, represent the tweets themselves with BoW after preparing it with methods of simplifying, and index the publication names as numeric values. A sentiment analyzer could then be used, to obtain binary labels for the data.
The machine learning models were trained on 70% of the data, utilizing cross-validation measures to prevent overfitting and reduce bias. Both models then predicted

the labels on the remaining 30% of the data, with their performance being recorded using the metric of F1 score, a comparison of the precision and recall of the predictions.

From this, it was clear that Naïve Bayes outperformed the decision tree approach, as it was significantly efficient in training compared to the other model, and produced a greater F1 score, indicating it was more accurate. We found that the decision tree produced, even when optimized, was biased towards predicting positive labels, due to the ratio of 2.45 of positive to negative labels throughout the dataset, although this was not a heavy skew, and therefore the performance of the decision tree was poor.

A multinomial Naïve Bayes model was used in this study, which is designed for text classification tasks and can therefore handle features representing word frequencies, i.e., the BoW representation. This is an evolution of a simple Naïve Bayes model and is particularly well-suited to sentiment analysis. In the future, an evolution of decision trees may be capable of matching the performance of a multinomial Naïve Bayes model.

One such example is BFTree, which aims to improve the performance of regular decision tree methods by using a best-first approach. This is designed to be more efficient, which we found to be the main issue of working with a simple decision tree, by prioritizing the most promising branches of the tree rather than exploring all branches. Furthermore, this would improve the accuracy of the model with a priority on the most informative features and instances, which would be especially useful in datasets containing noisy or irrelevant features. Additionally, the design of BFTree intends to be more robust to noisy data, through pruning of unpromising branches early in the training process, reducing the impact of noise or features that are not useful within the dataset. In a big data problem, this would aid in improving scalability, being better suited to handling large datasets, while improving efficiency and providing more accurate predictions than those of the traditional decision tree approach.

REFERENCES

[1] H. Elzayady, K. M. Badran, and G. I. Salama, 'Sentiment Analysis on Twitter Data using Apache Spark Framework', in *2018 13th International Conference on Computer Engineering and Systems (ICCES)*, Dec. 2018, pp. 171–176. doi: 10.1109/ICCES.2018.8639195.

[2] X. Ji, S. A. Chun, Z. Wei, and J. Geller, 'Twitter sentiment classification for measuring public health concerns', *Soc. Netw. Anal. Min.*, vol. 5, no. 1, p. 13, Dec. 2015, doi: 10.1007/s13278-015-0253-5.

[3] A. Suresh and C. R. Bharathi, 'Sentiment classification using decision tree based feature selection', *Ijcta*, vol. 9, no. 36, pp. 419–425, 2016.

[4] T. Fukuhara, H. Nakagawa, and T. Nishida, 'Understanding Sentiment of People from News Articles: Temporal Sentiment Analysis of Social Events.', in *ICWSM*, 2007.

[5] J. Ren, S. D. Lee, X. Chen, B. Kao, R. Cheng, and D. Cheung, 'Naive Bayes Classification of Uncertain Data', in *2009 Ninth IEEE International Conference on Data Mining*, Dec. 2009, pp. 944–949. doi: 10.1109/ICDM.2009.90.

[6] H. Parveen and S. Pandey, *Sentiment analysis on Twitter Data-set using Naive Bayes algorithm*. 2016, p. 419. doi: 10.1109/ICATCCT.2016.7912034.

[7] Y. Qi and Z. Shabrina, 'Sentiment analysis using Twitter data: a comparative application of lexicon- and machine-learning-based approach', *Soc. Netw. Anal. Min.*, vol. 13, no. 1, p. 31, Feb. 2023, doi: 10.1007/s13278-023-01030-x.

[8] 'Sentiment analysis using Twitter data: a comparative application of lexicon- and machine-learning-based approach | SpringerLink'. https://link.springer.com/article/10.1007/s13278-023-01030-x (accessed May 11, 2023).

[9] N. Zainuddin and A. Selamat, 'Sentiment analysis using Support Vector Machine', in *2014 International Conference on Computer, Communications, and Control Technology (I4CT)*, Sep. 2014, pp. 333–337. doi: 10.1109/I4CT.2014.6914200.

[10] S. Nigam, A. K. Das, and R. Chandra, 'Machine Learning Based Approach To Sentiment Analysis', in *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, Oct. 2018, pp. 157–161. doi: 10.1109/ICACCCN.2018.8748848.

[11] N. Faizan, A. Löffler, R. Heininger, M. Utesch, and H. Krcmar, *Classification of Evaluation Methods for the Effective Assessment of Simulation Games: Results from a Literature Review*. International Association of Online Engineering, 2019, pp. 19–33. Accessed: May 11, 2023. [Online]. Available: https://www.learntechlib.org/p/207576/

[12] R. Yacouby and D. Axman, 'Probabilistic Extension of Precision, Recall, and F1 Score for More Thorough Evaluation of Classification Models', in *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, Online: Association for Computational Linguistics, Nov. 2020, pp. 79–91. doi: 10.18653/v1/2020.eval4nlp-1.9.

[13] 'Word Polarity Disambiguation Using Bayesian Model and Opinion-Level Features | SpringerLink'. https://link.springer.com/article/10.1007/s12559-014-9298-4 (accessed May 11, 2023).

[14] J. Singh, G. Singh, and R. Singh, 'Optimization of sentiment analysis using machine learning classifiers', *Hum.-Centric Comput. Inf. Sci.*, vol. 7, no. 1, p. 32, Dec. 2017, doi: 10.1186/s13673-017-0116-3.

[15] H. Lee, P. Ferguson, N. O'Hare, C. Gurrin, and A. F. Smeaton, 'Integrating interactivity into visualising sentiment analysis of blogs', in *Proceedings of the first international workshop on Intelligent visual interfaces for text analysis*, Hong Kong China: ACM, Feb. 2010, pp. 17–20. doi: 10.1145/2002353.2002360.

[16] V. D. Nguyen, B. Varghese, and A. Barker, 'The royal birth of 2013: Analysing and visualising public sentiment in the UK using Twitter', in *2013 IEEE International Conference on Big Data*, Oct. 2013, pp. 46–54. doi: 10.1109/BigData.2013.6691669.