EasyChair Preprint

№ 2080

Sentiment Analysis using Unlabeled Email data

Rayan Salah and Neamat El Gayar

December 1, 2019

Sentiment Analysis using Unlabeled Email data

Rayan Salah Hag Ali
School of Mathematical and Computer Science
Heriot-Watt University
Dubai, United Arab Emirates
rs144@hw.ac.uk

Neamat El Gayar
School of Mathematical and Computer Science
Heriot-Watt University
Dubai, United Arab Emirates
n.elgayar@hw.ac.uk

Abstract*—Sentiment Analysis (SA) in the context of text mining is an automated process to detect subjectivity information, such as opinions, attitudes, emotions and feeling. Most prior work in SA view it as a text classification problem which needs labeled data to train the model. However, it is tough to get a labeled dataset. Most of the times we will need to do it by hand. Another issue is that the lack of portability across different domains makes it hard to use the same labeled data in different applications. Thus, we need to create labeled data for each domain manually. In this paper, we will use sentiment analysis to analyze the Enron email dataset. This work aims to find the best techniques to label the dataset automatically and avoid manual labeling. The training data is used to build a classifier using a supervised machine learning algorithm. In the labeling phase, we compare the lexicon labeling with k-mean labeling. Lexicon labeling gave better and reliable results. We used this labeled dataset to train the classifier. We used TF-IDF for feature extraction, to train Naïve Bayes and Support vector machine (SVM) classifiers.

Keywords*—Sentiment analysis, k-means, TFIDF, support vector machine

I. Introduction

In today's market, customer satisfaction (CS) measurement has become a significant indicator of business performance[1]. All types of business are putting customer satisfaction as their primary goal[2]. One way to measure CS is to analyze the customer feedback and review of the product or the service. According to Domo, we generate more than 2.5 quintillion bytes of data every day[3], this data is coming from different resources (social media, Emails, Amazon, Netflix, youtube…etc).

Email is one of the most necessary communication tools today. In the business area, there are over 108.7 billion Emails exchange every day[4]. This huge amount of data makes it impossible to be analyzed and sorted manually. Thus, there is an absolute need for an automated process. Sentiment Analysis (SA) is an automated process to analyze and categorize opinions (positive or negative or neutral)[5].

Most prior work in SA view it as a text classification problem which needs labeled data to train the model[6]. However, it is tough to get a labeled dataset most of the times. Hence, we often need to label data by hand, which is very costly and time-consuming. Another issue is the lack of portability across different domains, which makes it hard to use the same labeled data in different domains. Consequently, we will need to create labeled data for each domain manually.

In this research paper, a hybrid sentiment analysis schema of approaches using combined VADER lexicon labeling and Support Vector Machine (SVM) classifier algorithms for Enron Email dataset is presented. The main contributions of this paper reflect in the following three aspects:

1) Experiment with different preprocessing, and feature selection techniques for Email dataset.

2) Searching for the most appropriate labeling method for unlabeled data through the comparison among lexicon (VADER) labeling, and Kmeans labeling.

3) Examining the efficiency of Naïve Bays and SVM classification algorithm for sentiment classification.

The outline of this paper is described as follows. Section 2 reviews previous research on Email classification. Section 3 defines the framework and methodology. In ssection 4 result are discussed and presented. Finally, Section 5 summarizes the conclusions, limitations, and future work of this paper.

II. Related work

Youngjoong Ko and Jungyun Seo[7] have proposed a method to automatically create training sentence sets by using a keywords list of each category. They used feature selection techniques and naïve Bayes as a text classifier. To evaluate their method, they compare it to a supervised learning method using the same feature selection method and the same classifier (naïve Bayes classifier) as they used in their proposed method. They used 2,286 documents collected from the web. The proposed method got F-score of 71.8%, and the supervised learning method got 75.6%. Although there is no significant difference between the proposed method and the supervised method, however, Ko and Seo's method is not entirely unsupervised because they have created the category list manually (by hand).

Peter D. Turney[8] proposed an unsupervised learning algorithm for classifying reviews as recommended or not recommended. The algorithm first extracts phrases that contain adjective or adverbs from the review, then use pointwise mutual information (PMI-IR) to calculate the semantic orientation of each phrase and finally classify the

review based on the average semantic orientation of the phrases. The author conducted the experiments on 410 reviews from Epinions. The reviews were from different domains (automobile, banks, movies, and travel destination). Automobile and banks reviews got an accuracy average of 82%. Travel destination reviews got an accuracy of 70.53%. However, the movie reviews got a lower accuracy of 65.38% than the other domains. The author justified the low accuracy of the movie review is that a positive review of a movie will often contain unpleasant scenes (e.g., violence, death), thus, will reduce the average of the semantic orientation and will lead to the wrong classification.

Lin and Yulan [9] proposed an unsupervised approach to classify unlabeled dataset (movie review dataset). They presented a joint sentiment/ topic model (JST) to detect document-level sentiment and extract a mixture of topics from the text. The model is a probabilistic model based on Latent Dirichlet Allocation (LDA). To increase the accuracy, they defined a prior information model. They used paradigm word list, which contains a set of positive and negative words. They also used full subjectivity lexicon as prior information. They filtered the full subjectivity lexicon by removing the words that had occurred less than 50 times in the movie dataset. To evaluate their work, they tested it without prior information, and with paradigm word list and finally with the filtered subjectivity lexicon. The best accuracy 82% was achieved with filtered subjectivity lexicon. Although they have got a high accuracy, their model only classified positive and negative document while ignoring the neutral labeled document. Another limitation is that is the model represent the document as a bag of words which ignore the words ordering and thus may lead to the wrong classification.

Li and Liu[10] proposed a clustering-based sentiment analysis approach. The authors used the movie dataset that was created by Pang and Lee[11]. They used TF-IDF for feature extraction then they applied k-mean algorithms to cluster the data into two groups, to identify the positive cluster form the negative cluster they tested the clusters with 25 documents that contained extremely positive or negative content. Thus, the cluster which contains the positive documents is the positive group; the cluster which contains the negative documents is the negative group. Then they used WordNet to obtain term scores and then calculate the average score of each document to determine the positive/negative direction. They compared their approach with a supervised learning approach that used the same dataset and got an accuracy of 77%–82% [11], and their approach got an accuracy of 77.17%–78.33%.

Although sentiment analysis has been studied thoroughly on data from twitter, blogs, movies, product reviews.; very few researches has been conducted on E-mail sentiments analysis. Most work on email processing focused on spam detection and channel allocation. Email-data are mainly found to be topic-oriented and therefore traditional sentiment analysis algorithms cannot be applied without proper adaptation.

Hangal and Lam [12] proposed a system to process email archived. The system goes through the emails and categorizes it based on sentiment features, such as congratulatory emails and family emails, then visualize these emails. They used part of General Inquirer and the LIWC lexicon besides developing their lexicon. Their lexicon covers 45 categories and contains about 500 terms. However, they did not provide any information on the algorithms they used or the evaluation methods they did.

Mohamed and Yang[13] created emotion lexicon by crowdsourcing and use it to analyze and compare emotion word in love, hate email, and suicide notes. Their approach was to check if the words from the text(emails) exist in their emotion lexicon and then calculate the ratio of emotion words to the total number of emotion words in the text. They also experiment with Enron email dataset. They analyzed emails of a former employee at Enron and showed the percentage of positive and negative words in the emails that are sent by this employee.

In this paper we adopt a similar methodology as presented by the work of Sisi Liu and Ickjai Lee[14]. They proposed a hybrid approach to analyze Enron email dataset. For feature extraction, they used term frequency-inverse document frequency term weighting model (TF-IDF). They used K-means algorithms for labeling the dataset and SVM for sentiment classification. The main drawback of their study is that their comparisons was based on pseudo labeled data (no ground truth) and an extremely unbalanced data set. So that most of the classified emails were 'neural' and their model failed to classify negative emails.

In this paper we adopt a different methodology to work with a more balanced data set and report or findings accordingly.

III. Framework and Methodology

Fig1 displays the proposed framework. The framework is composed of various techniques including preprocessing, feature extraction, labeling and sentiment classification. The preprocessing step contains the following: duplication removal, handling of missing data, identifying header, signature, quotation, and program code and then removing it, tokenization, stop words removal, stemming. TF-IDF is used for feature selection as it has been widely reported to give the best empirical results[14] .For the labeling process, K-mean labeling and Lexicon labeling have been tested. In lexicon labeling, we will use VADER (Valence Aware Dictionary and sEntiment Reasoner). As for sentiment classification algorithms, SVM, and NB will be tested

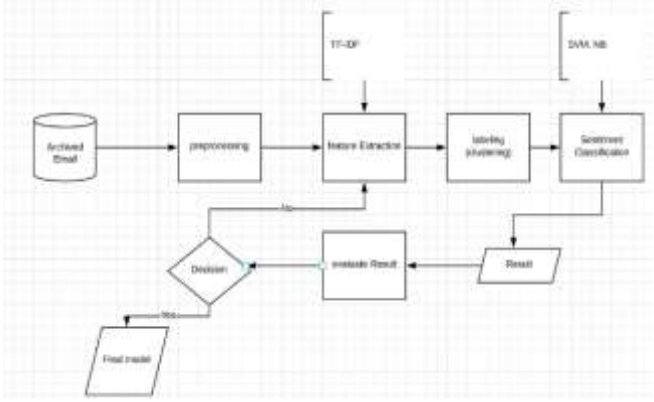

Figure 1 Framework and methodology

A. Discription of the dataset

The aim of this paper is to analyze Emails, and most of the time, it contains sensitive data; for that, there is not many public Email datasets. The most popular one is the Enron dataset. Enron dataset contains 500,000 Emails generated by 158 employees from the Enron Corporation. The dataset was made public by the Federal Energy Regulatory Commission

during its investigation after the company's collapse[15] . The final version was published on May 7, 2015, by https://www.cs.cmu.edu/~./enron/ . The size of the data is 2.23 GB. The dataset contains one folder for each employee. Emails are saves in a Multipurpose Internet Mail Extensions (MIME) format. Each message (E-mail) contains the sender and the receiver email address, date and time, subject, body text, and some other email-specific technical details.

B. Preprocessing

Email data are unstructured and noisy, and it contains two parts the header and the body. The standard preprocessing steps(stemming, lemmatization, and stop word removal) only, are not enough for email data cleaning[16]. We followed the recommendation in [17] for cleaning email data. The authors proposed a cascaded approach to preprocess the Emails. The first step is non-text filtering; in this step, we identified header, signature, quotation, and program code and then removed it. The second step is paragraph normalization; to detect extra lines break, and then remove it. The third step is sentence normalization; for identifying the end of each sentence (sentence boundary) and removing special symbols and non-ASCII-Words. The last step is word normalization; where we return the words to their roots(stem).

C. Feature Extraction

Tim and Irena[18] introduced new feature selection methods; they divided the features into two, unigram features and SentiWordNet Word Groups. Unigram features present terms occurrence and frequency [18]. Part of speech is a technique to find adjective, noun, verb, or adverbs and can also be used as a feature[19]. Opinion words and phrases are words commonly used to express positive or negative opinions. For example, "good, bad, like, hate"[19]. Negations is a false positive that changes the polarity of the sentence. For example, "not good "[19] should change the polarity from positive to negative. Term Frequency-Inverse Document Frequency (TF-IDF): Term frequency is merely counting the number of occurrence of the words in the document, and inverse document frequency is dividing the total number of documents by a number of the document that's a given the word appeared in that documents[18]. TF-IDF is a feature that emphasizes on the rare but important words.

SentiWordNet Word Groups (WordNet) is a lexical database for the English language available for research purpose; it groups the words into sets of synonyms called synset[20]. SentWordNet (SWN) is the extension of the automatic annotation of WordNet that adds a numerical score for positivity, negativity, and objective measures for each synset[29]. Some features of SWN[18] are SWN Word Score Groups (SWN-SG) which is grouping words that have the same positive or negative scores together. Thus we can have one feature that corresponds to more than one value since the values have the same score[18]. Another feature is SWN Word Polarity Groups (SWN-PG) when using SWN, to identify whether the words are more positive than negative and vice-versa. As a result, we can define two features, positive and negative, and the counts of them. So, when we have a word that is more positive than negative, we can add one to the positive feature and the same for the negative feature. In the end, we can have two features, the number of positive words and the number of negative words in a document[18]. The feature SWN Word Polarity Sums (SWN-PS) is similar to SWN-PG except here we sum the positive and the negative scores. So we can have two features, the first one contains the sum of SWN positive scores of all the words that have more positive scores than negative scores. The second one contains the sum of SWN negative scores of all the words that have more negative scores than positive scores[18].

D. Feature selection methods:

Feature selection is the task where we remove the unnecessary features to increase the accuracy and speed of the classifier[18]. Some of the methods in feature selection are [19]: Point-wise Mutual Information (PMI); which is a statistical method commonly used in modeling the association between words[21]. PMI between two words word1 and word2 is defined as follows[8]:

$$\text{PMI (word1, word2)} = Log_2\left[\frac{p(word1\,\&\,word2)}{p(word1)p(word2)}\right]$$

Equation 1 PMI

PMI is used by Turney [8] as feature selection in different reviews datasets (automobile, banks, movies, and travel destination) and got an average of 74% accuracy. Chi-square on the other hand is a statistical feature selection method that measures the lack of dependency between the word in a document and the class of the document. χ_i^2 between word *w* and class *i* is defined as follows[22]:

$$\chi_i^2 = \frac{n \cdot F(w)^2 \cdot (p_i(w) - P_i)^2}{F(w) \cdot (1 - F(w)) \cdot P_i \cdot (1 - P_i)}$$

Equation 2 Chi-square[23]

Chi-square and PMI are different ways of measuring correlation in the text. Chi-square is a normalized value [19]. Another feature is Latent Semantic Indexing (LSI), LSI is a feature transformation method. It analyzes the relationship between the words and documents by creating a set of concepts. It assumes words that have the same meaning will appear in a similar text. It uses the principal component analysis (PCA) technique[19].

E. Labeling the dataset:

As our dataset is unlabeled, the first step is to label the data. We selected two different labeling approaches, lexicon labeling approach and unsupervised labeling approach.
Unsupervised labelling approach; we decided to follow the same procedure as Liu and Lee [14] they had used the same dataset and used K-mean to label the dataset. We followed the same preprocessing and feature extraction (TF-IDF) approach. Liu and Lee[14] labeled the dataset positive, negative and neutral; this means they used K-means with n = 3 (n number of the cluster). We wanted to make sure n=3 is the optimal number of clusters that fit the dataset. For that, we used the elbow method. The elbow method works by

fitting the model to a different range value of K, then calculating the sum of squared errors (SSE). The SSE is plotted for each value of K. The best K will be the elbow[24]. The result we got proved K =3 is the optimal number of clusters (see the below figure).

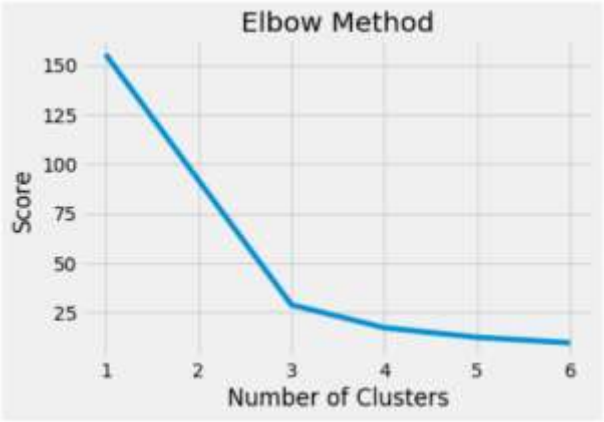

Figure 2 Optimal num. of clusters

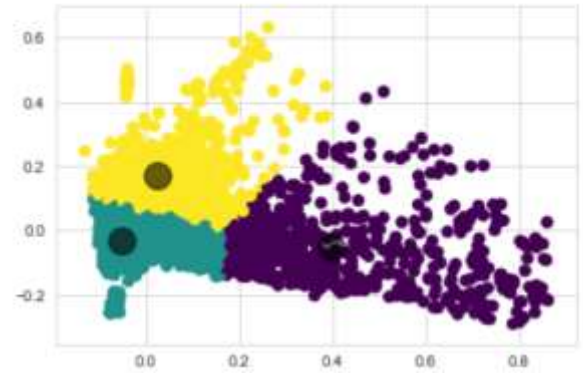

Figure 3 Clusters

Although we had followed the same approach as Liu and Lee[14] when we analyzed the clusters we did not find any association that connects each cluster with our target labels(positive, negative or neutral). We noticed that one of the three clusters always contain spam messages; we dropped this cluster and repeated the processes until we got rid of all the spam messages. Even after removing all the spam messages, we still did not find any association between the clusters and the target labels.

For the lexicon labeling approach, we used VADER (Valence Aware Dictionary and sEntiment Reasoner) package to label the data. VADER is a rule-based model that uses a combination of lexical features and grammatical and syntactical conventions rules[25]. VADER produces the result as an array of negative(neg), neutral(neu), positive (pos) and a compound which is the sum of all the ratings that has been normalized between -1(extreme negative) and +1 (extremely positive). For example, for the email below:

'test successful. way to go!!!'

VADER result is: "{'neg': 0.0, 'neu': 0.441, 'pos': 0.559, 'compound': 0.5859}"

To interpret this result we simply check the compound score if it is greater than 0 it is a positive email, else if it is less than 0 it is a negative email, and if it is equal to 0 it is a neutral email (see the below function).

df['label'] = df['compound'].apply(lambda score: 'pos' if score>0 else 'neut' if score==0 else 'neg')

After labeling with VADER, we split the dataset into positive, negative, and neutral datasets. Each one consists of: 302,408 positive emails, 22,585 negative emails and 29,056 neutral emails. Due to the limitation of technical resources, we took a sample of 3000 emails from each of the newly created dataset. To make sure that the samples are representative, we stratify the dataset before sampling. We use K-means to cluster this new dataset for further analysis and cleaning. Although we removed the spam and the duplicate, we found some of them again.

The final dataset after removing the spam and the duplicate emails contains:

Neutral: 2954 emails, Positive: 2950 emails, and Negative: 2798 emails.

For verification purpose, we manually examine a sample of the labeled data set and find the labeling mostly logical(correct).

F. Developing the Model:

To build the final model, we have used Naïve Bayes and Support vector machine to train the data. We followed the same process in Naïve Bayes and SVM. First shuffled the dataset and then split it to a training dataset and testing dataset with a ratio of 67 to 33.

We developed the model with three classes (positive, negative, and neutral),

The test result with Support vector machine using three classes (positive, negative, neutral) yield an accuracy of 82.8%. Table 1 shows the confusion matrix, while table 2 lists the precision, recall, F1-score and the support measures for the conducted experiments.

We also experiment with removing the neutral class to test if the accuracy will change. The binary classification gives better results than the three classes classification.

With binary classification (positive, negative) the accuracy was increased by almost 3%, we got an accuracy of 85.03 %.

Table 1 Confusion matrix (SVM with 3 classes)

[1] **negative**	[2] **neutral**	[3] **positive**
[4] 685	[5] 116	[6] 123
[7] 53	[8] 871	[9] 26
[10] 92	[11] 84	[12] 822

Table 2 Classification report (SVM with 3 classes)

	Precision	**Recall**	**F1-score**	**support**
Negative	0.83	0.74	0.78	924
Neutral	0.81	0.92	0.86	950
positive	0.85	0.82	0.83	998
Avg/total	0.83	0.83	0.83	2872

Testing the Naïve Bayes classifier with three classes (positive, negative, neutral), achieved 58.14%. Tables 3 and 4 summarize the results.

For the binary classification (positive, negative) the accuracy of the Bayes classifier reaches 73.12%, i.e. increases by almost 15%.

Table 3 confusion matrix (NB with 3 classes)

[13] **negative**	[14] **neutral**	[15] **positive**
[16] 497	[17] 6	[18] 399
[19] 171	[20] 250	[21] 568
[22] 48	[23] 10	[24] 923

Table 4 classification report (NB with 3 classes)

	Precision	**Recall**	**F1-score**	**support**
Negative	0.69	0.55	0.61	902
Neutral	0.94	0.25	0.40	989
positive	0.49	0.94	0.64	981
Avg/total	0.71	0.58	0.55	2872

IV. EVALUTION

This section presents and discusses the finding of the experiments of the SVM and Naïve Bayes model.

A. Discussion of the labeling process:

This work aims to automatically label the dataset. We compared K-mean labeling with lexicon labeling. Liu and Lee[14] used K-mean to label the Enron dataset, we followed the same approach but when analyzing the clusters we did not find any association that connects each cluster with our target labels(positive, negative or neutral). The only explanation for this is that Liu and Lee may have used a sample from the dataset which could be clustered to positive, negative, and neutral, while we used the whole dataset and different random samples from the dataset but in all the cases we did not have the target labels. Besides the data that they used was very unbalanced.

For the lexicon labeling we used VADER package, VADER performs better than other lexicons. We had manually examined the labeling of VADER, and most of the result was correct.

B. Discussion of the results:

Both the Naïve Bayes and SVM model performed better in the binary classification (positive, negative) than the multiclass classification (positive, negative, neutral). The accuracy increased in Naïve Bayes model by almost 25.7% from 58.14% to 73.12%, and in the SVM model it increased by nearly 2.69% from 82.8% to 85.03%. Although the accuracy increases in the binary classification, however in this application the neutral class is still important.

In the multiclass classification, the accuracy of the Naïve Bayes model is 58.14%, which is almost like random guessing. For that, we decided to continue testing and developing with the SVM model.

From the confusion matrix and the classification report (see Tables 1 & 2), we observe that positive and neutral emails have better recall and F1-score than the negative emails, but in precision, they almost got the same result.

We believe that the model classifies negative email wrong to positive or neutral due to the negation problem. This problem was not accounted for at the cleaning (removing stop words) and feature extraction process (TF-IDF). For example, "I do not believe he sent that file" after removing the stop words and performing TF-IDF for feature extraction the word "not" will be removed in the cleaning process or not consider in the TF-IDF phase.

C. Comparison of models:

The closest related work to our study was presented by Liu and Lee[14]. They proposed a framework for sentiment analysis for the Enron email dataset. For feature extraction, they used (TF-IDF) for feature extraction, K-means algorithms for labeling the dataset, and SVM for sentiment classification. Their approach is relatively similar to the one we followed to build the model, except we used lexicon (VADER) for labeling the dataset.

They got an accuracy of 97.7%, and we got an accuracy of 82.80%. However, Liu and Lee's model failed to classify negative emails; they got 0.0% precision, recall, and F-measure for negative emails. They mentioned that 188 out of 200 emails classified as neutral their explanation was incompleteness and limitations of the data cleansing.

D. Reusability Evaluation:

To test the reusability of the model, we test the model in Hillary Clinton email dataset. The dataset was released by the state department and consists of 7945 emails. We used a pool of 3 experts to label the data set manually. We sent 15 random emails our three experts to label the emails as follows: 0 for neutral, 1 for positive and -1 for negative.

In case of disagreement between the experts we used the majority opinion as the final label. In rare cases where the 3 experts give 3 different labeling, we applied our trained SVM classifier to determine the final label.

Comparing the labeling of the classifier with human labeling, we got 8 out of 15 with the same labeling, which means the accuracy of 54%. We believe this poor result is because we did not perform any preprocessing for the emails. Although the result is low, this experiment proves that human labeling is not consistent; it gives a different result from person to another person.

V. CONCLUSION, LIMITATION, AND FUTURE WORK

In this paper, we present a hybrid sentiment analysis model for the Enron Email dataset. This work aims to find the best technique for labeling the dataset automatically and avoid manual labeling. We built the model in two phases; phase one was labeling the dataset automatically. In phase two we built a classifier from the labeled dataset. For the labeling phase, we compare the lexicon (VADER) labeling and K-mean labeling. In K-mean labeling, we did not find any association that connects the clusters with our target labels (positive, negative, or neutral). However, lexicon (VADER) labeling gave us a reliable result. We used this labeled dataset to train the classifier, we used TF-IDF for feature extraction, then compared Naïve Bayes and Support vector machine (SVM) techniques we got an accuracy of 58.2% and 82.1% respectively. To evaluate our classifier, we compared with Liu and Lee [14](the closest related work to our project). Liu and Lee got an accuracy of 97.7%, and we got an accuracy of

82.80%. However, Liu and Lee's model worked with an extremely unbalanced data set and failed to classify negative emails; they got 0.0% precision, recall, and F-measure for negative emails. We got 83% precision, 74% recall, and 78% F-measure. Their explanation was incompleteness and limitations of the data cleansing.

In this paper, we have two main limitations. Firstly, in the preprocessing phase, we could not remove the signature from all the emails. We used *emailParser* library and *Talon* library to remove the signature from the body text, however in most of the emails, it was not removed. We believe the reason is that most of the email senders write their signature in the same line of the body text, besides their email address is different from their signature and most of the techniques that remove the signature compare the email address with the final block in the body text.

Secondly, we did not handle the negation problem. Therefore, the classifier confuses *negative* emails for *positive* or *neutral* as the cleaning (removing stop words) and feature extraction process (TF-IDF) phase drops the negation. For example, "I do not believe he sent that file" after removing the stop words and performed TF-IDF for feature extraction the word "not" will be removed in the cleaning process or not consider in the TF-IDF phase.

The obvious step in future work is to enhance the preprocessing phase (i.e. remove the signature) and handle the negations problem. Moreover, we intend to enhance the reusability of the model to work with different email datasets. This can be achieved by using a larger sample of the dataset to train the models. In this study we only used 10,000 training samples. We also plan to do more experiments and use deep learnings models.

REFERENCES

[1] F. S. Hillier, *International Series in Operations Research & Management Science Series Editor : customer satisfacton evluation*. 2010.

[2] V. M. Ngo, "Measuring Customer Satisfaction: a Literature Review," *Proc. 7th Int. Sci. Conf. Financ. Perform. Firms Sci.*, no. April 2015, p. Proceedings of the 7th International Scientific Co, 2015.

[3] "Data Never Sleeps 6 | Domo." [Online]. Available: https://www.domo.com/learn/data-never-sleeps-6. [Accessed: 18-Sep-2019].

[4] W. Nawaz, K. U. Khan, and Y. K. Lee, "A multi-user perspective for personalized email communities," *Expert Syst. Appl.*, vol. 54, pp. 265–283, 2016.

[5] E. Cambria, D. Das, S. Bandyopadhyay, and A. Feraco, *Guide to Sentiment Analysis*. 2017.

[6] Y. He and D. Zhou, "Self-training from labeled features for sentiment analysis," *Inf. Process. Manag.*, vol. 47, no. 4, pp. 606–616, 2011.

[7] Y. Ko and J. Seo, "Automatic text categorization by unsupervised learning," pp. 453–459, 2000.

[8] L. M. Chiappe, "P. D. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pages 417–424, Morristown, NJ, USA, 2001. Ass," *Society*, vol. 12, no. 3, pp. 344–350, 2010.

[9] C. Lin, N. P. Road, and E. Ex, "Joint Sentiment/Topic Model for Sentiment Analysis Chenghua."

[10] G. Li and F. Liu, "Sentiment analysis based on clustering: A framework in improving accuracy and recognizing neutral opinions," *Appl. Intell.*, vol. 40, no. 3, pp. 441–452, 2014.

[11] F. Drews, "Pang B, Lee L, Vaidyanathan S. Thumbs up? Sentiment classification using machine learning techniques. In: Conference on empirical methods in natural language processing (EMNLP). Philadelphia, Pennsylvania, USA, 2002, p. 79," *Antike und Abendl.*, vol. 57, no. July, pp. 79–86, 2012.

[12] S. Hangal and M. S. Lam, "Sentiment Analysis on Personal Email Archives," *Proc. CHI 2011 Work. Informatics HCI Des. Theory Soc. Implic.*, 2011.

[13] S. M. Mohammad, Tony, and Yang, "Tracking Sentiment in Mail: How Genders Differ on Emotional Axes," pp. 70–79, 2013.

[14] S. Liu and I. Lee, "Email Sentiment Analysis Through k-Means Labeling and Support Vector Machine Classification," *Cybern. Syst.*, vol. 49, no. 3, pp. 181–199, 2018.

[15] "Enron Email Dataset." [Online]. Available: https://www.cs.cmu.edu/~enron/. [Accessed: 18-Sep-2019].

[16] G. Tang, J. Pei, and W. S. Luk, "Email mining: Tasks, common techniques, and tools," *Knowl. Inf. Syst.*, vol. 41, no. 1, pp. 1–31, 2014.

[17] J. Tang, "Email Data Cleaning," 2005.

[18] L. S. Chen and C. W. Chang, "A new term weighting method by introducing class information for sentiment classification of textual data," *IMECS 2011 - Int. MultiConference Eng. Comput. Sci. 2011*, vol. 1, pp. 394–397, 2011.

[19] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Eng. J.*, vol. 5, no. 4, pp. 1093–1113, 2014.

[20] S. Baccianella, A. Esuli, and F. Sebastiani, "SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining," *Proc. Int. Conf. Lang. Resour. Eval.*, no. November, pp. 2200–2204, 2010.

[21] Y. Xu, G. J. F. Jones, J. Li, B. Wang, and C. Sun, "A study on mutual information-based feature selection for text categorization," *J. Comput. Inf. Syst.*, vol. 3, no. 3, pp. 1007–1012, 2007.

[22] M. S. Simpson and D. Demner-Fushman, *Mining Text Data - Aggarwal-Zhai.pdf*. 2012.

[23] G. Perumal and S. K. Lakshmanaprabu, "Document Clustering Based On Text Mining K-Means Algorithm Using Euclidean Distance Similarity," no. April, 2018.

[24] "Using the elbow method to determine the optimal number of clusters for k-means clustering - bl.ocks.org." [Online]. Available: https://bl.ocks.org/rpgove/0060ff3b656618e9136b. [Accessed: 20-Sep-2019].

[25] E. G. C. J. Hutto, "VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text," *Int. AAAI Conf. Web Soc. Media*, 2016.