Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 159 (2019) 428–436

Procedia Computer Science

www.elsevier.com/locate/procedia

23rd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

A New Method to Identify Short-Text Authors Using Combinations of Machine Learning and Natural Language Processing Techniques

Biveeken Vijayakumar[a], Muhammad Marwan Muhammad Fuad[b]*

[a]*School of Computing, Electronics amd Mathematics, Coventry University, UK, vijayakb@coventry.ac.uk*
[b]*School of Computing, Electronics amd Mathematics, Coventry University, UK, ad0263@coventry.ac.uk*

Abstract

Identifying authors by their style of writing is a very challenging task. This problem has several applications, one of which is to identify fake online reviews written by spam accounts. The existence of such fake reviews degrades the credibility of the whole review collection, hence these fake reviews should be identified and removed. This process, however, needs to be automated since it is impossible to perform it manually in large review collections. Current authorship identification approaches identify authors based on large-scale texts such as documents. For this reason, these methods do not scale well to short texts such as online reviews that have limited features to learn from. This paper introduces a new method of author identification in short texts using combinations of machine learning algorithms and natural language processing techniques. The experiments we conducted on Yelp reviews gave promising results.

© 2019 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0/)
Peer-review under responsibility of KES International.

Keywords: Author identification, fake reviews, machine learning, natural language processing, Yelp.

1. Introduction and Related Work

With the rise of social media, review sites such as Yelp have become influential in aggregating opinions on products and services online. For example, Yelp had an average of 103 million unique visitors and more than 177 million reviews in the fourth quarter of 2018 [1]. As a result, review sites rely on providing accurate and truthful reviews for

* Corresponding author. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000 .
E-mail address: ad0263@coventry.ac.uk

1877-0509 © 2019 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0/)
Peer-review under responsibility of KES International.
10.1016/j.procs.2019.09.197

their customers. Consequently, individuals and companies may commit review fraud by creating fake reviews in the aim of raising or damaging the reputation of a product or service on the review sites [2]. This can be done by users with multiple spam accounts on these sites. Therefore, being able to accurately and effectively identify fraudulent reviews is important for these sites in order to uphold the reputation of the reviews provided by them. The task of author identification can be used in this domain to identify if two or more spam accounts belong to the same author and help remove those accounts and the reviews associated with them.

Author identification (AI) is the task of identifying the author of a given document from a closed set of N authors, given a collection of documents whose authorship is already known [3]. Several methods already exist for AI. Most of these methods, however, are based on large-scale texts such as documents and books with low numbers of authors [4]. Applying AI tasks on social media data is relatively new and this has been limited to blog post data. Online reviews are different from documents, blog posts, and books in that reviews are generally much shorter in length, so these current AI methods do not scale well down to short texts such as online reviews, which have limited linguistic features to learn from. Additionally, in blog posts and books, the texts are typically limited in topic domains whereas in online reviews, the topic domain can vary depending on the service being reviewed by the same author.

As an example of AI, when spam reviews are posted on a large scale, multiple spam reviews by the same author will be posted across multiple review accounts. This is so that the author can avoid being identified as fraudulent. In this case, when one review is identified as a fake and spam review, AI can be used to detect the other spam reviews posted by the same author on multiple review accounts. This paper will study this approach of AI and additionally how AI could possibly be used for identifying spam reviews instead of the traditional approach of identifying them, which uses labelled spam reviews. The later is not realistically possible as we cannot identify spam reviews on a large scale.

Luykx and Daelemans [5] highlight that most of the previous work into AI uses datasets with a small number of authors and a large amount of data per author. To challenge this, Luykx and Daelemans use a larger dataset of 145 authors who have written essays (large-scale text). They use a support vector machine (SVM) classification model and numerous natural language processing (NLP) techniques such as lemmatizers, part-of-speech tags, and chunkers to evaluate the effect of increasing the number of authors used. They found that increasing the number of authors caused a significant drop in accuracy. In a real word context, such as online reviews, there is a smaller amount of data per author and a larger number of authors. Therefore, implementing an approach that is scalable to a large number of authors and a smaller size of text is important.

Stamatatos [6] and Koppel et al [7] have previously explored the importance and the impact of feature extraction in AI and have identified multiple features that can be used to characterise an author. This includes lexical features (character and word n-grams), stylistic features (stop words, stemming), and structural features (sentence length, word length).

The method we present in this paper combines these different features in the attempt to identify an optimal model for AI on short texts such as online reviews. Our method addresses author identification on short/noisy review texts and on large author sets where we are trying to identify the likelihood of a review belonging to a review class. We combine a variety of natural language processing techniques with different machine learning methods to achieve this. A dataset of online Yelp reviews is used to test our method. This paper will be limited to exploring lexical features only, as the paper specifically explores AI on short text. Semantic features are not explored, as they would require larger texts in order to successfully identify any meaningful patterns.

This paper is organized as follows; in Section 2 we introduce our method and its different machine learning and natural language processing components. We present the experiments we conducted to validate our method in Section 3. We conclude with Section 4.

2. The Proposed Method

Our proposed method for AI is broken down into four components: 1- Data Preprocessing, for preprocessing the reviews, 2- Feature Representation, for extracting features from the preprocessed reviews, 3- Natural Language Processing, for adding context to the reviews, and 4- Machine Learning Classification, to classify the author of a review.

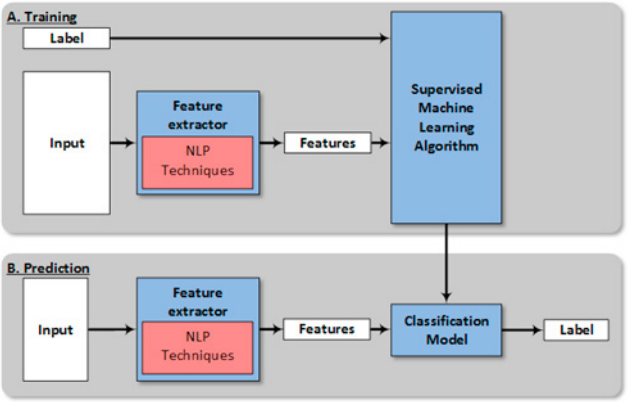

Fig. 1. Overview of our author identification method

The classification of a review consists of a training stage and a prediction stage. In the training stage, training data consisting of multiple reviews is preprocessed and passed onto a feature extractor where multiple NLP techniques are applied to the reviews and features are generated. These features are then used to train a machine learning classifier. This classifier is then used in the prediction stage to predict the author (label) of new reviews (inputs). Fig. 1 shows our AI method.

2.1. Data Preprocessing

Data preprocessing is done with the aim of normalizing the data in the dataset and reducing the number of features in the feature set. This in turn makes it suitable for fitting the data onto all types of classification models as the time complexity of fitting the data onto each model is reduced. The different preprocessing steps used are described as following:

- *Lowercase normalization:* to prevent multiple versions of the same word, all the words despite their casing are normalized to lowercase form so they can all be counted collectively.

- *Separation of punctuation:* the majority of punctuation is conjoined with words in a sentence, therefore multiple variations of the same word form again. However, punctuation can be a vital feature in AI. Therefore, punctuation is separated from the words instead of removing it.

- *Removal of 1 count occurrences:* If a word appears only once across the whole dataset, this word will most likely not have an impact on the classification model, so, such words are regarded as irrelevant and subsequently removed.

To give an idea about the importance of preprocessing, for the dataset we used in the experimental section, the number of unique features was reduced from 192,077 (100%) to 98,577 (55.99%). The details of feature reduction are shown in Fig. 2.

2.2. Feature Representation

An author can typically display a preference for certain words/phrases more than other authors. To capture the writing style of an author and help differentiate between different authors, the n-gram model is used. The n-gram model is a tokenization model where the value of N corresponds to the number of consecutive features (words) that are tokenized together into a single token [8]. Consequently, the dependency between consecutive words in a sentence can be captured.

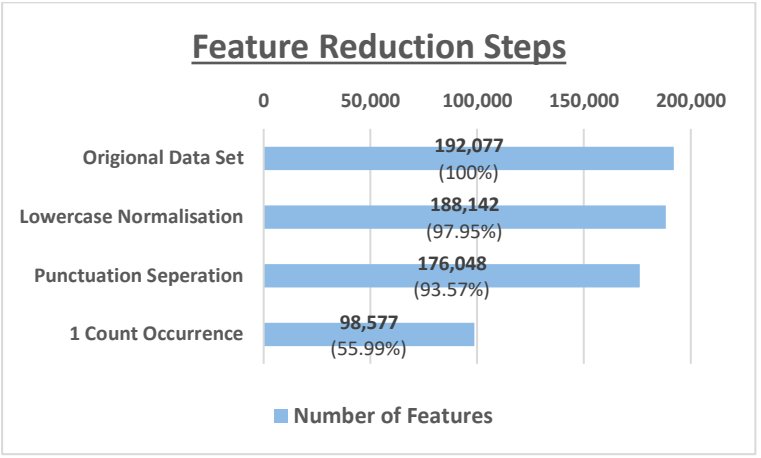

Fig. 2. Effect of feature reduction after applying data preprocessing steps.

In unigrams (1-gram), individual words in a piece of text are tokenized into individual features. In bigrams (2-gram), two consecutive words are tokenized together as one token. For example, the sentence "I would recommend this place." will be tokenized as the following {"I", "would", "recommend", "this", "place."} in unigrams. Whereas in bigrams, the same sentence is tokenized as following {"I would", "would recommend", "recommend this", "this place."}.

In our method only unigrams, bigrams and a combination of both are evaluated, since the time taken to train the classification model exponentially increases as the number of n-grams increases. This is due to the increase in the number of different features being used to train the classifier as the number of n-grams increases. Therefore, it is vital that the n-grams be limited for a trade-off in time so that the model can be used in a real-world application.

2.3. Natural Language Processing (NLP) Techniques

NLP techniques help classifiers by providing them with context of each review. As a result, they help gain a better understanding of the stylistic and linguistic features of each author. In this paper three different NLP techniques are applied. These are described below.

2.3.1. Stemming

Stemming is used to consolidate different variations of a word that share the same stem to one common root form [9]. For example, the words "likes", "liking", "likely", and "liked" will all be simplified to their root form "like". Consolidating all the variations of a word with the same root form ensures that the frequency of the root form is calculated collectively and hence this increases the probability of the root form occurrences. The root form can now have a meaningful impact on the classification model. The stemming algorithm that is used for the classification in our method is the *Snowball* [10] stemming algorithm provided by NLTK [11].

2.3.2. Lemmatization

Lemmatisation is similar to stemming as they both involve consolidating multiple variations of a word to one common root form. However, lemmatisation considers the context and lemma of a word when reducing and aggregating the words [9]. As a result, it does not discriminate words that have slightly varying meanings. For example, the words "went", "going", "gone", and "go" will become the word "go" even though the word "went" consists of different characters to "go" which is something the stemming algorithms do not do. The lemmatisation algorithm used in our will be the *WordNet Lemmatizer* [12] provided by NLTK [11].

2.3.3. Stop Words Removal

Stop words are the collection of the most common words in the English language. These words do not have meaningful impact on text classification [13]. An example of these stop words includes “the”, “it”, "are", etc. By including these words as features, they can cause misclassifying pieces of text. Therefore, these features are removed as tokens. For our method, the list of 128 stop words provided by NLTK [11] is used.

2.4. Machine Learning Classifiers

There are three different machine learning methods that are known to perform well in text classification tasks. We thus use them in our AI method. These classifiers are described in the following;

2.4.1. Support Vector Machines (SVM)

SVM is a large-margin linear classification algorithm. The algorithm trains a model with the aim of finding a separating hyperplane vector $\overrightarrow{w}$ between two different classes that maximizes the separation distance [14]. Derivation of the hyperplane vector $\overrightarrow{w}$ can be formalised as the following:

$$\overrightarrow{w} := \sum_i \gamma_i a_i \vec{r}_i, \quad \gamma_i \geq 0 \tag{1}$$

Where γ_i is calculated by solving dual-optimization problems and $a_i \in \{1, -1\}$, with 1 and -1 representing two different classes. Review vectors $\vec{r}_j$ are called support vectors when corresponding $\gamma_j > 0$. This is due to $\vec{r}_j$ being the review vectors that influence the hyperplane vector $\overrightarrow{w}$. For this reason, in order to determine the class, it would be to simply test which side of the hyperplane $\overrightarrow{w}$ each review $\vec{r}_j$ falls under.

SVM used in this paper is a binary classifier where an SVM is trained to classify between each pair of classes (candidate author) in order to perform multi-class classifications [15]. This approach of training SVM is called one-vs-all, where each new text (review) is passed through each SVM and the most likely class is chosen.

In our method, we train SVMs with a linear kernel rather than complex polynomial kernels as the computation time for polynomial kernels was far too long and infeasible. This is due to the high number of classes (4,521 authors) being trained.

2.4.2. Multinomial Naïve Bayes (MNB)

MNB is a probabilistic classifier, which makes the naïve supposition that each feature is conditionally independent of other features given the class of the feature [16]. Therefore, the classification model does not consider the sequence of words but only the occurrences of the words in a text. The model is based on the Bayes rule and in the context of author identification is summarized as the following where the author class a with the highest likelihood is chosen as the author of the review using feature vectors $f_1, f_2, \ldots f_n$:

$$P(a \mid f_1, f_2, \ldots f_n) = \frac{P(f_1, f_2, \ldots f_n \mid a)\ P(a)}{P(f_1, f_2, \ldots f_n)} \tag{2}$$

where $P(f_1, f_2, \ldots f_n)$ is computed by summarizing the count of each feature f appearing in r reviews divided by the total number of reviews. The likelihood of each author is calculated as following:

$$P(a) = \frac{number\ of\ reviews\ written\ by\ author\ a}{total\ number\ of\ reviews} \tag{3}$$

To train the MNB classifiers, *Laplace Smoothing* of 1.0 is used and the option to learn prior class probabilities. This is to prevent the probabilities of zero being generated in the case of a feature in a review not appearing in the training dataset.

2.4.3. Maximum Entropy (ME)

ME is another probabilistic classifier. However, his classifier is different from MNB is that it does not make the naïve supposition that features are conditionally independent of one another. As a consequence, when using a feature-based model such as n-grams, the features can be iteratively added without the problem of having overlapping features [17]. To calculate the likelihood of a class (author), the *Principle of Maximum Entropy* is used. This is where the distribution that is closest to the uniform distribution is chosen [18]. Hence, no further assumption is made than what is already given by the training data.

For example, if our training data consists of N pairs $\{(r_1, a_1), \dots, (r_N, a_N)\}$ where $a \in A$ is the set of author classes and $r \in R$ is the set of reviews that are denoted as a vector of word occurrences, then the likelihood for the maximum entropy model can be formalized as:

$$P(a|r, \mu) = \frac{exp(\sum_i \mu_i f_i(a,r))}{\sum_a exp(\sum_i \mu_i f_i(a,r))} \quad (4)$$

Where μ is the weight vector that is calculated by maximizing the conditional probability of each feature $f_i(a, r)$ of each review r.

3. The Experiments

Three different experiments were conducted using different combinations of machine learning classifiers and natural language processing (NLP) techniques. The first experiment consisted of identifying baseline classification models for AI. For this experiment, each machine learning classifier was combined with either a unigram vectorization model or both a unigram and bigram vectorization model combined. For the second experiment, the unigram baseline models were combined with the numerous NLP techniques identified in this paper with the aim of improving the accuracies of the baseline models. The accuracies of the optimal combination of unigrams and the NLP techniques are shown in Table 3. In the third and final experiment the combined unigram and bigram baseline models were then combined with the numerous NLP techniques. The accuracies of that experiment are shown in Table 4.

Cross Validation is a method for evaluating the accuracies of classification models [19]. To evaluate the different author identification models, *K-Fold Cross Validation* is used. This is the process of splitting the dataset into k equal sized partitions, with a training set being assigned $k - 1$ partitions and a testing set being assigned 1 partition of the dataset. The classification model is trained using the training set and later tested on the test set to calculate the accuracy for the model. This process is repeated k times where each different partition is used as a testing set and the rest as a training set. An average of all the accuracies calculated across these iterations is then generalised to a single accuracy estimation. By using this validation approach, more data is used to train the models. Additionally, it ensures that all the data in the dataset is used for both training and testing without causing any bias towards the accuracy results. In our experiments, 10- fold cross validation was used.

In order to choose the optimal combination of NLP techniques for each classification model, forward selection is used. Forward selection is the process of starting with no NLP techniques in the classification model and selectively adding the individual NLP technique that has the most significant accuracy improvement to the model [20]. The procedure used to do this is described in Algorithm 1 (Fig. 3).

3.1. Yelp Review Dataset

For the purpose of performing AI on online reviews, the Yelp Review dataset [21] consisting of restaurant and hotel reviews is used. This dataset contains 6,685,900 unique reviews written by 1,637,138 different authors across

```
Algorithm 1: NLP Forward Selection
Input: nlpTasks, model
Procedure
    chosenNlpTasks = [] ;
    bestScore = null ;
    stopCondition = false ;
    while stopCondition = false do
        kfoldScores = [] ;
        tempBestScore = null ;
        for task in nlpTasks do
            tempNlpTasks = chosenNlpTasks + task;
            train model with tempNlpTasks using KFold Cross Validation;
            save score in kfoldScores;
        choose bestTask that maximise kfoldScores;
        choose tempBestScore that maximise kfoldScores;
        if tempBestScore ≤ bestScore then
            stopCondition = true;
        else
            bestScore = tempBestScore;
            add bestTask to chosenNlpTasks;
            remove bestTask from nlpTasks;
    return chosenNlpTasks ;
```

Fig. 3. The NLP forward selection algorithm.

192,609 businesses. Some of the authors in this dataset have only posted a small number of reviews, therefore, it is hard to perform AI on those authors as they have inadequate number of linguistic features to capture and learn from. For this reason, the dataset that we use in our experiments comprises only of authors who have posted at least 100 reviews each, which is in line with the fact that fake reviewers tend to post many reviews. As shown in Table 1, there are 4,521 different authors who have posted at least 100 reviews each with 883,737 total reviews posted amongst them. This is still a high number of authors and reviews. To create a balanced dataset of reviews, we use the same number of reviews per author. For authors who exceed 100 reviews, reviews are randomly sampled and removed to only contain exactly 100 reviews per author in the final dataset.

Additionally, the Yelp Review dataset contains supplementary useful information such as “star”, “cool”, and useful ratings along with date posted for each review. This additional information is not used as features in our method, because, although we are testing our method on Yelp Review dataset, we aim to present a method that can be generalized to different AI domains on short-texts.

Table 1. Reviews per author.

Minimum number of reviews per author (x)	Number of Reviews	Number of Authors
100	883,737	4,521
50	1,456,225	12,823
1	6,685,900	1,637,138

Table 2. Accuracy of baseline classifiers.

Machine Learning Classifiers	Accuracy for Unigram (%)	Accuracy for Unigram and Bigram (%)
Multinomial Naïve Bayes	47.1	46.7
Support Vector Machine	84.2	89.0
Maximum Entropy	80.7	81.1

3.2. The Results of Baseline Models

The results obtained using the baseline models combined with either unigram only or both unigram and bigram are shown in Table 2. As we can see from the table, this combination gives high classification accuracy for both SVM and maximum entropy. The classification accuracy for MNB is, however, low. These results are consistent across both types of n-gram models that are tested. For SVM and maximum entropy, there is an increase in accuracy when increasing the n-gram model used from unigram only to both unigram and bigram. The best performing baseline model was SVM with unigrams and bigrams. MNB however performed poorly with an accuracy of 46.7% for unigram and bigram. It was also the only model where the results degraded when we used unigram and bigram instead of unigram only.

3.3. The Results of Improved Models with NLP techniques

In Table 3 we show the results of combining a machine learning classifier with an NLP technique for the case of a unigram, and in Table 4 we show these results for the case of unigram and bigram. The results in the two tables indicate that the NLP techniques had an identical effect in both n-gram models. This is due to the same combination of NLP techniques providing the most optimal results across both n-gram models. However evidently there has been no significant effect on the baseline models apart from lemmatization on SVM. This produced the highest increase from the baseline model of 1.5%.

4. Conclusion

In this paper we presented an approach to perform author identification on short texts where combinations of multiple machine learning classifiers and natural language processing techniques were applied. The experiments we conducted show that a combination of SVM classifier with unigram and bigram vectorization model, and lemmatization give the best performance in author identification with 90.5% accuracy. Additionally, we found that the NLP techniques used in this paper had little effect on the baseline approach to author identification with the exception of lemmatization, and that increasing the number of n-grams used had the most significant effect giving the highest increase in accuracy of 4.8%, which is shown when increasing the n-grams in the baseline SVM models as shown in Table 2.

Despite the number of authors being large and the size of reviews being small, we found that our model was still able to produce good accuracy results.

As a direction for future work, the proposed author identification method can be extended to perform author verification and background author identification.

Table 3. Classification accuracy of improved classifiers for unigrams.

Machine Learning Classifiers	NLP Technique(s)	Accuracy (%)
Multinomial Naïve Bayes	Stop Words + Stemming	47.2
Support Vector Machine	Lemmatization	85.1
Maximum Entropy	Stemming	80.8

Table 4. Accuracy of improved classifiers for unigrams and bigrams.

Machine Learning Classifiers	NLP Technique(s)	Accuracy (%)
Multinomial Naïve Bayes	Stop Words + Stemming	46.9
Support Vector Machine	Lemmatization	90.5
Maximum Entropy	Stemming	81.3

References

[1] The Yelp Blog: About. http://www.yelp.com/about (2018). Accessed 15 Mar 2019

[2] Luca, Michael, Georgios Zervas. (2013) Fake it till you make it: Reputation, competition, and Yelp review fraud. Harvard Business School NOM Unit Working Paper (14-006).

[3] Ying Zhao and Justin Zobel. (2007) Searching with style: authorship attribution in classic literature. In Proceedings of the thirtieth Australasian conference on Computer science - Volume 62, ACSC '07, pages 59–68, Darlinghurst, Australia, Australia. Australian Computer Society, Inc.

[4] Keˇselj, V., Peng, F., Cercone, N., Thomas, C. (2003) N-gram-based author profiles for authorship attribution. In: Proceedings of the Conference of Pacific Association for Computational Linguistics, PACLING, vol. 3, pp. 255–264

[5] Kim Luyckx and Walter Daelemans. (2008). Authorship attribution and verification with many authors and limited data. In Proceedings of the 22nd International Conference on Computational Linguistics Volume 1. Association for Computational Linguistics, pages 513–520.

[6] Efstathios Stamatatos, Martin Potthast, Francisco Rangel, Paolo Rosso, and Benno Stein. (2015). Overview of the PAN/CLEF 2015 Evaluation Lab. In Josiane Mothe, Jacques Savoy, Jaap Kamps, Karen Pinel-Sauvagnat, Gareth J.F. Jones, Eric SanJuan, Linda Cappellato, and Nicola Ferro, editors, Experimental IR Meets Multilinguality, Multimodality, and Interaction. 6th International Conference of the CLEF Initiative (CLEF 15). Springer, Berlin Heidelberg New York, pages 518– 538.

[7] Schwartz, R., Tsur, O., Rappoport, A., & Koppel, M. (2013). Authorship Attribution of Micro-Messages. In EMNLP (pp. 1880-1891).

[8] Koppel, M., Winter, Y. (2014) Determining if two documents are written by the same author. J. Assoc. Inf. Sci. Technol. 65, 178–187

[9] Vimala Balakrishnan and Ethel Lloyd-Yemoh. (2014) Stemming and lemmatization: a comparison of retrieval performances. Lecture Notes on Software Engineering, 2(3):262.

[10] Porter, M. (2001) "Snowball: A Language For Stemming Algorithms". Snowball.Tartarus.Org. http://snowball.tartarus.org/texts/introduction.html.

[11] Natural Language Toolkit — NLTK 3.4 documentation. https://www.nltk.org/ (2019). Accessed 15 Mar 2019

[12] WordNet | A Lexical Database for English. Wordnet.princeton.edu. https://wordnet.princeton.edu/ (2019). Accessed 15 Mar 2019

[13] Saif, H., Fernandez, M., He, Y., & Alani, H. (2014). On stopwords, filtering and data sparsity for sentiment analysis of Twitter. InProc. 9th language resources and evaluation conference (LREC). Reykjavik, Iceland

[14] B. Pang, L. Lee, and S. Vaithyanathan, (2002) "Thumbs up?: sentiment classification using machine learning techniques," in Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. Association for Computational Linguistics, pp.79–86

[15] Argamon, Shlomo, Marin Saric, and Sterling S. Stein. (2003) Style mining of electronic messages for multiple authorship discrimination: First results. In Proceedings of the 2003 Association for Computing Machinery Conference on Knowledge Discovery and Data Mining (ACM SIGKDD), pages 475–480.

[16] Kendall, Maurice G, Alan Stuart, and J. K Ord. (1994) Kendall's Advanced Theory Of Statistics. London: Hodder Arnold.

[17] Go, A, R Bhayani, and L Huang. (2009) "Twitter Sentiment Classification Using Distant Supervision". https://cs.stanford.edu/people/alecmgo/papers/TwitterDistantSupervision09.pdf.

[18] K. Nigam, J. Lafferty, and A. Mccallum. (1999) Using maximum entropy for text classification. In IJCAI-99 Workshop on Machine Learning for Information Filtering, pages 61–67.

[19] P. Refaeilzadeh, L. Tang, H. Liu, (2009) Cross-validation, Encyclopedia of Database Systems, pp. 532–538.

[20] I. Guyon, A. Elisseeff, (2003) An introduction to variable and feature selection, Journal of Machine Learning Research 3. 1157–1182.

[21] "Yelp Dataset". (2019) Yelp.Com. https://www.yelp.com/dataset.