International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 03 | March - 2022 ISSN: 2582-3930

A Review on Feature Selection Techniques for Sentiment Analysis

Kriti Agarwal

Abstract

Sentiment Analysis is the technique of identifying and categorizing emotions in order to examine how people feel about services such as movies, products, events, and politics. It is a widely researched on topic in text mining. This paper presents a review and evaluation results for some feature selection techniques such as TF-IDF, document frequency, word frequency, sparsity reduction and chi square statistics. To test these feature selection techniques, the study used twitter data on stock market and Naïve Bayes Classifier for classification because of its computational simplicity and effectiveness. The implementation of the study has been done in R.

Keywords

Feature Selection Techniques, Chi Square Statistics, Sentiment Analysis, TFIDF, Document Frequency, Word Frequency, Sparsity Reduction

Introduction

Sentiment Analysis is the technique of identifying and categorizing emotions in order to examine how people feel about services such as movies, products, events, and politics. Enterprises benefit from research in the subject of sentiment analysis since they can accurately comprehend users' opinions about their product and make improvements as a result. Natural Language Processing (NLP) is used in Sentiment Analysis to interpret human language in both written and spoken form. NLP is divided into four subtasks that allow a computer to interpret language by evaluating sentence structure and grammar. Summarization, Part-of-Speech (PoS) Tagging, Text Categorization, and Sentiment Analysis are the sub-tasks of NLP.

Machine learning or lexicon-based algorithms can be used to conduct sentiment analysis. The sentiment is calculated using a lexicon-based technique, which takes into account the semantic orientation of the words in the text. To put it another way, the words in the text are divided and given scores. The final score, which indicates the sentence's sentiment, is calculated by adding these scores together. Whereas in machine learning, classification is performed on two sets of documents: trained datasets and test datasets. There is a slew of classifier algorithms that have been trained on emotional samples. Without human input, a machine learns to recognize emotions and categorizes them into negative and positive feelings. In machine learning, one such classifier is the Naive Bayes algorithm.

The Bayes theorem is used to create the Naive Bayes Classifier. It's a type of supervised learning method that's frequently used to solve classification problems. Naive Bayes is used for spam filtration, sentiment analysis, and article classification, among other things. Because it is a probabilistic classifier, it makes predictions based on the probability of an object. Text classification problems with high-dimensional datasets are predicted using Naive Bayes. It's called naive because it believes that the appearance of one characteristic has nothing to do with the appearance of another. The basis of Naïve Bayes' Classification is Bayes' rule, given as:

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

Where,

P(A|B) is Posterior probability,

P(A) and P(B) are class priors,

P(B|A) is conditional probability

Given a set of attributes values X (instances) and class c, the probability of each attribute a_i relative to the class needs to be estimated. Here, we employ the product rule – that is, asssume conditional independence amongst the attribute values $P(a_i|c)$. This gives us the following formulae:

$$P(c|X) = \frac{P(c)\prod_i P(a_i|c)}{P(X)}$$

The prediction task is reduced to:

$$pred_X(c) = argmax\ P(c)\prod_i P(a_i|c)$$

The Confusion Matrix is a table that displays the predictions of a classifier. The matrix is N x N, with N being the number of classes. The row represents the projected class, whereas the column represents the actual class.

The Confusion Matrix compares the actual target values to the predictions of the machine learning model. This section summarizes the categorization model's performance as well as its flaws.

At the intersection of each row and column are the counts of instances that meet the row and column criteria when the model is evaluated by the classifier. For a 2-class problem, matrix is:

	Actual Classes	**Predicted Classes**
	a	b
A	$True\ Positive(TP)$	$False\ Negative\ (FN)$
B	$False\ Positive(FP)$	$True\ Negative\ (TN)$

Table 1.1: Confusion Matrix

In Table 1.1, columns represent the predicted values and rows represent the actual values. The accuracy for a 2-class learning problem is calculated as:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

Accuracy is a unitless statistic that spans from 0 to 1. It can be stated as a percentage ranging from 0 to 100%. For a k-class problem, multi-class problems increase the dimensions of the matrix to k x k. Multi-class accuracy is calculated by dividing the diagonal of the confusion matrix by the sum of all entries in the matrix. When evaluating accuracy for a multi-class, weight each of the k-classes' accuracy by the number of occurrences in that class, then divide by the total number of instances.

Another common practice is to take the predicted class probabilities and scale them such that they sum up to 1. This is referred to as normalizing the predictions so they can be interpreted as percentages.

One of the drawbacks of Naive Bayes is that in real-world datasets, the assumption of completely independent conditional probabilities is frequently inaccurate, resulting in poor performance. As a result, while utilizing Naive Bayes, feature interaction should be taken into account. The performance of the Nave Bayes classifier can be improved in a variety of ways. One of the ways is feature selection. By deleting irrelevant, noisy, or redundant information, feature selection enhances the classifier's performance while reducing runtime and memory requirements.

Feature selection reduces the number of input variables while creating a predictive model. Reducing the number of input variables reduces the model's computation cost and, in some cases, increases its performance. A large number of variables can slow down model creation and training. Furthermore, dealing with such a large number of variables needs a large quantity of memory. Because of the large number of input variables, which contain aspects that are unrelated to the target variable, the performance of some models may decrease.

There are two types of feature selection methods: supervised and unsupervised. The target variable is ignored in the unsupervised feature selection technique. They use correlation to eliminate unnecessary variables. The target variable is used to remove irrelevant variables in supervised feature selection.

We compared multiple feature selection strategies for Sentiment Analysis in this article, including Sparsity Reduction, TF-IDF, Document Frequency, Word Frequency, and Chi-Square Statistics. To execute the experiments, Naive Bayes Classifier was employed, and the dataset was a set of Twitter data regarding the stock market.

Review of Related Studies

TF-IDF (Term Frequency – Inverse Document Frequency)

Term Frequency is the number of times a term appears in a document divided by the total number of words. It is given by the following formula:

$$TF(t) = \frac{Number\ of\ times\ term\ t\ appears\ in\ a\ document}{Total\ number\ of\ terms\ in\ the\ document}$$

Because each sentence is varied in length, a word may appear more frequently in a longer sentence than in a shorter one. As a result, the total number of times a word appears in a document is divided by the total number of words.

The document is described using term frequencies. To put it another way, the more a term is used, the more it defines the document. However, phrases like "the" and "and," which contain no substantial information, appear frequently, contradicting the preceding assumption. As a result, the classifier's performance suffers because of presence of these terms. To resolve the problem, stopwords should be removed from the dataset. Popular words might also be filtered, and surface topic terms could be multiplied by term frequency and inverse document frequency.

The Inverse Document Frequency (IDF) is a metric that counts how many times a term appears in a document. If it is discovered to be commonly utilized across papers, it is given a lesser weight. The prominence of key words can be considerably improved by removing terms with lower weights.

Document Frequency

Document Frequency is a common text classification technique that uses a basic word reduction technique. It is straightforward to build and is used in feature selection for large-scale corpora because of its linear complexity.

Document Frequency refers to the number of documents in the dataset that contain the phrase. The word that is taken into account for subsequent processing is the one that appears in sufficient documents. For example, the word 'sensex' is considered a feature in our dataset if it appears in at least 5 pages. Terms having a Document Frequency of less than a certain threshold are removed to reduce space and work for the classifier while also increasing accuracy.

Reduction in Sparsity

Data features with a lot of zero values are known as sparse data. Vectors of one-hot-encoded words, for example, or categorical data counts. The tendency of sparse features to enhance the space and temporal difficulties of models is a common challenge. The model will fit the noise in the training data if there are too many features. This is referred to as overfitting. Models that have been overfitted are unable to generalize to newer data. This has a negative impact on the model's predictability.

The problem of sparsity is addressed using a variety of approaches. The removal of sparse features, which can create noise and raise the model's memory requirements, is a frequent strategy.

Chi Square Statistics

There are two variables in feature selection. One relates to the frequency of occurrence of feature t, while the other refers to the likelihood of occurrence of category C. We primarily look at whether t and C are independent in text classification. If they're independent, the feature can't be used to identify whether or not a text belongs in category C. However, determining whether t and C are independent is difficult in practice. As a result, Chi Square Statistics is used to describe the applicability of the method. A bidirectional queue is used to represent a textual feature t and a category C, it is shown in table 1.2,

	C	¬C	Total
t	A	B	A+B
¬t	C	D	C+D
Total	A+C	B+D	A+B+C+D

Table 1.2: Feature and Category

The higher the chi square score for category C, the relevancy between feature t and category C is greater. When the score is 0, the feature t and category C are independent.

Comparative Study of Feature Selection Methods for Sentiment Analysis

S.No	Year	Title	Methodology Used	Dataset Used	Accuracy
1.	2019	Two new feature selection metrics for text classification [6].	Relevance Frequency Feature Selection and Alternative Accuracy2 metrics	Reuters Dataset 20 Newsgroup Dataset Ohsumed Dataset	On 20 Newsgroup: RFF: 64.65% Acc2: 69.65% Reuters: RFF: 76.15% Acc2: 87.42% Ohsumed: RFF: 65.33% Acc2: 69.8%
2.	2009	Feature selection for text classification with Naïve Bayes [11].	Multi Class Odd Ratio (MOR) Class Discriminating Measure (CMD)	Reuters21578 and the Chinese text classification corpus given by Ronglu Li	Reuters 21578: 85.62% Chinese test classification: 72.59%

3.	2007	Feature Selection Methods for Text Classification [3].	Feature Selection Strategies: Subspace Sampling, Weight-Based Sampling, Uniform Sampling, Document Frequency, Information Gain.	TechTC-100, 20-Newsgroups, and Reuters-RCV2	For TechTC-100 dataset, Document Frequency achieved the highest accuracy. For 20-Newsgroups, the accuracy steadily increased with increase in number of features. For Reuters-RCV2, Information-Gain and Document Frequency performed much better.
4.	2018	Influence of Word Normalization and Chi-squared Feature Selection on Support Vector Machine (SVM)	Stemming and Lemmatization in addition of Chi-Square Feature selection	BBC Dataset	Stemming: 95.05% Lemmatization: 93.24%

		Text Classification [4].			
5.	2018	Grooming Detection using Fuzzy-Rough Feature Selection and Text Classification [32].	Fuzzy-rough feature selection technique	PAN'13 Author Profiling data set	Binary classification: 73.00% Multi-Class Classification: 73.11%
6.	2013	Fast and Accurate Sentiment Classification using an Enhanced Naïve Bayes Model [25].	Negation Handling, Word n-grams and feature selection by mutual information	IMDB movie review dataset	Increased the accuracy to 88.80% from original 73%.
7.	2015	A Text Mining Application of Emotion Classification of Twitter's Users Using Naïve Bayes Method [13].	Pre-processing, processing and validation. Testing performed using 10-fold cross validation.	The data has been extracted from Twitter using the Twitter API.	The accuracy achieved was 83%.
8.	2015	Feature Extraction or Feature Selection for Text Classification: A Case Study on Phishing Email Detection [14].	feature selection techniques - Chi-Square and Information Gain Ratio feature extraction techniques – Principal	Data set is prepared by collecting a group of e-mails from the publicly available corpus of legitimate and	Accuracy was highest for Latent Semantic Analysis with 97%.

			Component Analysis and Latent Semantic Analysis	phishing e-mails. Then the e-mails are labeled as legitimate and phishing correspondingly.	
9.	2010	Feature Selection for Text Classification Based on Gini Coefficient of Inequality [19].	Gini Coefficient of inequality is applied on the datasets then they are normalized using Information Gain, Mutual Information and Chi-squared Statistics.	Reuters-21578, 7-Sectors WebKB, Open Directory Project	Using Mutual Information, Information Gain and Chi-square methods, the accuracy improved by 28.5%, 19%, 9.2% respectively.
10.	2018	A Chi-square Statistics Based Feature Selection Method in Text Classification [31].	Information Gain and Chi-Square Statistics.	The data set used in this paper is comment corpus about computer and book, which contains 8K positive and negative categories.	Accuracy: Information Gain: 92.50%. Chi-Square Statistics: 95.03%

Experiments and Results

Dataset

The dataset used in this study is a corpus of stock market-related tweets. Kaggle's regular libraries were used to obtain the data. The data source is given in [1]. The dataset contains 5791 tweets, each of which is labelled with a positive or negative value of 1 or -1. There are two columns in the csv file: text and sentiment. The R environment has been used to implement the code. The code is provided in [2].

Sentiment analysis of stock market tweets can be used to create a prediction model for determining and analyzing the relationship between public opinion and stock prices. This can aid in making price forecasts in the future. Positive tweets about a firm may motivate others to invest in its stock, resulting in an increase in the stock price of that company. During implementation, the company's name was also divided into two categories: negative and positive. Based on the emotions of the tweet, this result can be used to predict whether the company's stock will grow or decline in the future.

Evaluation and Analysis

The Naive Bayes Classifier was used in our research. The features obtained were 15328 when the data was trained using the Naive Bayes Classifier without pre-processing the data. As a result, we obtained a 5791 x 15328 matrix. The vector grew to 463.5 MB in size when the data was separated into training and test datasets, and it reported an error. The model's operation came to a halt. The R environment was unable to handle such a huge dataset.

Some usable data could be acquired by pre-processing the data by deleting stopwords, punctuation marks, and numerals. The number of features extracted was 9184, and the accuracy was 70.12%. To tackle the problem of zero probability, Laplace Smoothing was used on the model. The accuracy was improved by 2.67 percent as a result of this.

The accuracy gained was 74.00% when words with frequencies fewer than 3 were removed from the training dataset. The minimum frequencies were increased to 5 and 10 in the next two experiments, yielding accuracy of 73.83% and 71.67%, respectively. As a result, it was discovered that as the minimum frequency was increased, the classifier's accuracy declined as the number of features reduced.

The sparsity was lowered to 0.99 to reduce the problem of overfitting. There were 112 features that were kept. This resulted in a 68.65% accuracy. This was a surprising outcome because it was thought that reducing sparsity would improve accuracy.

The dataset was subjected to the TF-IDF method of feature selection, which yielded an accuracy of 70.12% (equivalent to the standard classifier) at cutoff 1.1. The features retained after using the TF-IDF approach were 4516, compared to 9184 in the standard classifier. In the process, some documents were lost, leaving 4971 records to be saved. As a result, feeding the classifier less data provided the same results as the normal classifier.

The Chi Square Test was performed on the dataset. The features were assigned a weight in association with its relation with a target term. In our study we used Chi Square Test to determine the relationship between the term 'aap (Advanced Auto Parts Inc.)' and the other terms in the dataset. The terms that are closely related with ‘aap’ can provide more insight into how firm aap's stocks will fare in the future. In Chi-Square Test, if the score is higher that means the terms are closely related.

In our study the features having a weight of less than 0.1584780 were removed. This resulted in 2289 features being preserved and an accuracy of 68.65%, which is lower than the standard classifier, was obtained. Again, the classifier was evaluated using features that had a weight of less than 0.04918127. This resulted in a 70.12% accuracy with 2289 features. This demonstrates that some characteristics are critical for a classifier to get more accurate results.

The classifier was trained with Document Frequency with word lengths ranging from 5 to 15 and number of documents it appeared in ranging from 5-90, yielding an accuracy of 65.45% with 945 features.

Accuracy Comparison of Different Methods

Method	Accuracy
Pre-Processed Data	70.12%
Laplace = 1	72.79%
Minimum Frequency = 3	74.00%
Minimum Frequency = 5	73.83%
Minimum Frequency = 10	71.67%
Sparsity = 0.99	68.65%
TF-IDF cut off = 1.1	70.12%

Chisq Score < 0.1584780	68.65%
Chisq Score< 0.04918127	70.12%
Document Frequency = 5-90 and Word Length = 5-15	65.45%

Table 1.3: Accuracy Comparison of Different Methods

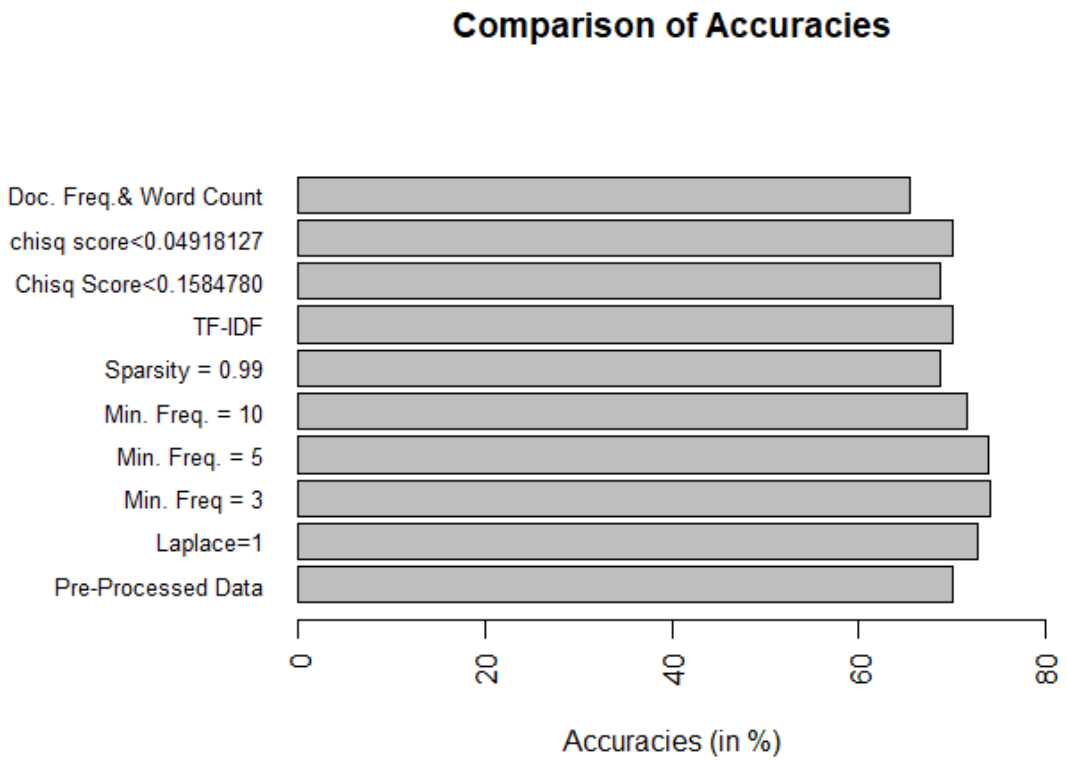

Fig.1: Bar Graph showing Accuracy Comparison of Different Methods

The above graph shows the comparison of accuracies of all the methods used in the experiment. From the graph we can make out that the highest accuracy achieved was 74.00% when the features with less than frequency, 3, were dropped.

Conclusion

Sparsity Reduction, TF-IDF, Document Frequency, Word Frequency, and Chi-Square Statistics were all investigated in this research. Extensive testing revealed that Word Frequency was the most accurate method,

achieving the maximum accuracy for minimum frequency, 3. This implies that the classifier requires a large number of features to function properly.

To acquire more accurate findings, we can explore implementing context-based sentiment analysis, using hashtags as features, employing bigrams or n-grams for feature selection, and handling negation handling.

References

[1] https://www.kaggle.com/yash612/stockmarket-sentiment-dataset

[2] https://github.com/kritia69/Naive-Bayes-Classifier

[3] Anirban Dasgupta, Petros Drineas, Boulos, Vanja Josifovski, Michael W. Mahoney. "Feature Selection Methods for Text Classification". *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, 2007.*

[4] Ardy Wibowo Haryanto, Edy Kholid Mawardi, Muljono. "Influence of Word Normalization and Chi-squared Feature Selection on Support Vector Machine (SVM) Text Classification". *International Seminar on Application for Technology of Information and Communication (iSemantic), 2018.*

[5] Dimitrios Kotzias, Misha Denil, Nando De Freitas, Padhraic Smith. "From Group to Individual Labels using Deep Features".

[6] Durmus Ozkan, Erdal Kilic. "Two new feature selection metrics for text classification." *Informa UK Limited, trading as Taylor & Francis Group*, 2019.

[7] Evelyn Setiani, Win Ce. "Text Classification Services Using Naïve Bayes for Bahasa Indonesia".

[8] I.Rish. "An Empirical Study of the Naïve Bayes Classifier".

[9] Jason D.M Rennie (2001). "Improving Multi-Class Text Classification with Naïve Bayes".

[10] Jay M Vala, Prem Balani. "A Survey on Sentiment Analysis Algorithms for Opinion Mining". *International Journal of Computer Applications (0975 – 8887)* **Volume 133 – No.9**, January 2016.

[11] Jingnian Chen, Houkuan Huang, Shengfeng Tian, Youli Qu. "Feature selection for text classification with Naïve Bayes". *Expert Systems with Applications 36 (2009) 5432–5435.*

[12] Liangxiao Jiang, Dianhong Wang, Zhihua Cai, and Xuesong Yan. "Survey of Improving Naive Bayes for Classification". *Springer-Verlag Berlin Heidelberg* 2007.

[13] Liza Wikarsa, Sherly Novianti Thahir. "A Text Mining Application of Emotion Classification of Twitter's Users Using Naïve Bayes Method".

[14] Masoumeh Zareapoor, Seeja K. R. "Feature Extraction or Feature Selection for Text Classification: A Case Study on Phishing Email Detection". *I.J. Information Engineering and Electronic Business, 2015, 2, 60-65.*

[15] Mitali Desai, Mayuri A. Mehta. "Techniques for Sentiment Analysis of Twitter Data: A Comprehensive Survey". *ICCCA*, 2016.

[16] Nader Mohamed. "Real-Time Big Data Analytics: Applications and Challenges". *IEEE*, 2014.

[17] Rafael B. Pereira, Alexandre Plastino, Bianca Zadrozny, Luiz H. C. Merschmann. "Categorizing feature selection methods for multi-label classification". *Springer Science+Business Media Dordrecht*, 2016.

[18] Sang-Bum Kim, Kyuong-Soo Han, Hae-Chang Rim and Sung Hyon Myaeng (2006). "Some Effective Techniques for Naïve Bayes Text Classification".

[19] Sanasam Ranbir Singh, Hema A. Murthy, Timothy A. Gonsalves. "Feature Selection for Text Classification Based on Gini Coefficient of Inequality". *Proceedings of the Fourth International Workshop on Feature Selection in Data Mining*, PMLR 10:76-85, 2010.

[20] Sonali Vijay Gaikwad, Archana Chaugule, Pramod Patil Department. "Text Mining Methods and Techniques". *International Journal of Computer Applications (0975 – 8887)* **Volume 85 – No 17**, January, 2014.

[21] Tak-Lam Wong, Wai Lam (2009). "An Unsupervised Method for Joint Information Extraction and Feature Mining Across Different Web Sites".

[22] Tianda Yang , Kai Qian, Dan Chia-Tien Lo, Ying Xie and Yong Shi , Lixin Tao. "Improve the Prediction Accuracy of Naïve Bayes Classifier with Association Rule Mining". *IEEE*, 2016.

[23] T.K.Das , P.Mohan Kumar. "BIG Data Analytics: A Framework for Unstructured Data Analysis". *IJET*, 2013.

[24] T. Nasukawa T. Nagano. "Text Analysis and Knowledge Mining System". *IBM SYSTEMS JOURNAL*, **VOL 40, NO 4**, 2001.

[25] Vivek Narayan, Ishan Arora, Arjun Bhatia. "Fast and Accurate Sentiment Classification using an Enhanced Naïve Bayes Model".

[26] Walaa Medhat, Ahmed Hassan, Hoda Korashy (2014). "Sentiment Analysis Algorithm and Applications: A Survey".

[27] Weiyuan Li, Hua Xu. "Text-based Emotion Classification Using Emotion Cause Extraction". *Elsevier Ltd.*, 2013.

[28] Wei Zhang , Feng Gaoa. "An Improvement to Naive Bayes for Text Classification". *Elsevier Ltd.*, 2011.

[29] Youngja Park, Roy J. Byrd. "Hybrid Text Mining for Finding Abbreviations and their Definitions".

[30] Young Gyo Jung, Kyung Tae Kim, Byungjun Lee, Hee Yong Youn. "Enhanced Naïve Bayes Classifier for Real-Time Sentiment Analysis with SparkR".

[31] Yujia Zhai, Wei Song, Xianjun Liu, Lizhen Liu, Xinlei Zhao. "A Chi-square Statistics Based Feature Selection Method in Text Classification". *IEEE*, 2018.

[32] Zheming Zuo, Jie Li, Philip Anderson, Longzhi Yang, Nitin Naik. "Grooming Detection using Fuzzy-Rough Feature Selection and Text Classification". *IEEE, 2018*.