Text Classification Using Context Polarity

Shaurya Guliani
Department of Graduate and Postdoctoral Studies and Department of Science
Wilfrid Laurier University
Waterloo, Canada
guli7170@mylaurier.ca

Amaan Javed
Department of Graduate and Postdoctoral Studies and Department of Science
Wilfrid Laurier University
Waterloo, Canada
jave4800@mylaurier.ca

***Abstract*—Text Classification is an integral application of Data Analysis allowing segmentation of written text into categories which can prove to be extremely useful in specific industries. Semantics can prove to be extremely important in such classification allowing a better analysis and increased accuracy. Bag-of-Words (BOW) and Term Frequency – Inverse Document Frequency (TF-IDF) are two well-known algorithms which work on text classification with high performance. However, both algorithms work on a lexicon frequency model which doesn't consider the semantic knowledge which can be obtained from the text. This paper proposes a new robust algorithm which could compete with BOW and TF-IDF while taking into consideration the semantic knowledge of the target text hoping to perform better. The new algorithm focusses on using polarity as the standard to understand the semantics associated with the text which is a metric more common in the application of Sentiment Analysis but, uses it with respect to different contexts thereby, making the semantic analysis process more robust on different data. The paper shows the result obtained from models trained on the proposed algorithms and analyzes it with the performance obtained by models trained using BOW and TF-IDF algorithm. Same Models are applied on two datasets – Sentiment Analysis and Span-Ham Detection – to display the proposed model's flexibility while working on datasets of different context. The algorithm is implemented using Python along with Natural Language Toolkit (NLTK) library for semantic analysis and Scikit-learn library for model building.**

Keywords—Text Classification, Semantic Analysis, Polarity, Sentiment, Natural Language Toolkit, BOW, TFIDF, Sentiment, Spam, Ham

I. Introduction

The constantly developing field of Data Analysis has shown the importance of Text Classification as being an application allowing the segmentation of textual content into semantic categories. This modern capability is not only visually astounding but also is an important practicality carried out in various industries to enhance the organization of data as well as its retrieval.

Bag-of-Words (BOW) and Term Frequency - Inverse Document Frequency (TF-IDF) have been the go to algorithms to be used in the application of Text segment classification characterized by their high reliability and performance. These models carrying out an unigram frequency dependent algorithm work astonishingly for a massive corpora of text but, performs while ignoring any semantic value carried out by the text they are analysing which are inherently important in natural language understanding and can provide massively to improve accuracies of such classification models.

The aim of the paper is to facilitate the development of a text classification algorithm which could take into account the semantics associated with the target text during the analysis thereby, providing increased performance and could work on datasets of different context hence, providing flexibility of use. The implemented algorithm competes with the traditional BOW and TFIDF algorithm to fill-in for their lack of semantic understanding and gives a more precise algorithm.

The paper is organized into four sections. Section II, "Related Work", gives a brief account of all the previous work and developments that has been done with the objective to solve the mentioned problem. Section III, "Methodology", gives detailed step-by-step procedure followed to carry out the algorithm and comparison done with other models followed by "Results and Discussion" in section IV which discusses about the outcome and the final verdict regarding the created algorithm. Section V, "Conclusion and Future Scope", summarizes the whole paper into few lines specifying its need in the upcoming technologies and discusses some future ideas which can be worked upon the project.

II. Related Work

A lot of significant work has been done in the field of Text Classification in which the importance of semantic analysis has been showcased. N. Pittaras et al [1] displayed the power of WordNet semantic graph by using Deep Neural Networks (DNN) to extract semantic features in their study. This was achieved by carrying out augmentation of the input through the DNN. The use of semantics was also evident in a study showing its application of Cancer Hallmark analysis which introduced a new approach capable of supervised learning and important topic extraction [2]. Yahui Li et al [3] used a graph representation for text segments to understand semantic relations between lexicons by simplifying the complex structure of text. The task of semantic analysis is complex enough to give rise to complexities, polysemy and synonym problems being the most common of them which were addressed in the study by Shuo Yang et al [4] in which they proposed a different strategy to counter this problem when working on document classification.

Polarity has also been used previously with regards to understand the semantics of a text segment, though mostly in the field of sentiment analysis. Research by Wen-tau Yih et al [5] introduced a novel representation of the semantic WordNet in which antonyms were placed on the opposite sides of a sphere inducing some polarity regarding the context into them. An interesting discussion was published by Leimin Tian et al [6] which talked about the importance of individual modalities and multimodal information which indirectly conveys the two important aspects of sentiment: polarity and intensity. A semantic approach discussed by Zahra Ayeste and

Samaira Noferesti [7] based on domain knowledge for polarity shift gave an interesting way of detecting the change in polarity using supervised machine learning using extracted features.

III. METHODOLOGY

The algorithm is implemented using the Python Programming Language along with NLTK, Scikit-learn Pandas and Numpy libraries. NLTK plays an important role in pre-processing the text to be fit for semantic analysis and also provides with the SentiWordNets corpora which is essential to extract the sense from a token under consideration. Scikit-Learn provides different vectorizers to carry out the BOW and TF-IDF algorithms, models to be used for data fitting and performance metric which allows the measurement of performance achieved by models trained on the proposed algorithm. Pandas and Numpy libraries provides with functions facilitating data handling and processing.

A. Natural Language Processing

NLP is an area of artificial intelligence that deals with natural language interaction between computers and humans. The goal of NLP is for computers to be able to read, interpret, and respond to human language in a useful way. Correcting spelling problems in text, understanding and processing search queries, and interpreting and responding to voice commands are some of the basic applications of NLP. NLP is a difficult field to master. Among the difficulties it faces are the following: Ambiguity and setting: Handling human language's complexity and nuances, such as idioms and sarcasm. Dealing with a variety of dialects, slangs, and language styles [8].

B. Natural Language Toolkit

The Natural Language Toolkit (NLTK) is used for natural language processing (NLP) in Python. It is a free and open-source library. Its applications include classification, tokenization, stemming, tagging, parsing, and semantic reasoning. NLTK requires Python 3.7 or above. It simplifies hard processes like tokenization, which divides text into sentences or words, and part-of-speech tagging, which is necessary for syntactic analysis. The combination of NLTK with machine learning techniques enables a variety of applications, ranging from sentiment analysis to text classification. This function is especially useful in the age of big data, when such analyses can reveal insights into client preferences, market trends, and other areas [9].

C. SentiWordNets

SentiWordNet is a natural language processing (NLP) application. It is a revised version of the WordNet database, which contains a huge number of English words organised into sets of synonyms known as synsets. Its primary use is to aid in sentiment analysis, the act of evaluating whether a piece of writing is good, negative, or neutral. SentiWordNet assigns three scores to each word: positivity, negativity, and objectivity. These ratings show whether a word is more likely to be used in a good, negative, or neutral context. It is commonly used for activities such as analysing the sentiment of product reviews, detecting the tone of social media posts, and understanding consumer feedback [10].

The algorithm is created in a way to work as a sentiment analyser which could classify a text into binary classes: positive or negative. However, can work in different contexts as well by only incorporating a few changes. The algorithms work on polarity scores of important tokens which are calculated using a list searching and sliding window negation handling technique which will be discussed further.

The following steps are carried out to implement the algorithm focusing on making it work as a sentiment analyser:

- Data Collection
- Data Pre-processing and Feature Extraction
- Model Training
- Model Evaluation

1) Data Collection

For the project, a Kaggle Dataset is used which consists of 4000 hotel reviews each labelled with their respective sentiments – positive or negative. We will also be using two text files which consist of words – one for positive words (2006 words) and one for negative words (4783 words). The text files are populated by using the research paper: [11] These words will play the role of deriving the polarity and sentiment of particular words. A list of 44 sense-inverting words such as 'not', 'wasn't', etc. is also collected which will be used for sense inversion, if any. The list is retrieved from the research paper: [12]

2) Data Pre-processing and Feature Extraction

The initial and the most important phase of pre-processing consists of analysing the hotel revies and give a polarity score to them. The 'Review Text' field is processed to remove any single quotes ('), double quotes (") or apostrophes (') as these are the important punctuations which might hinder the functioning of the program. After this, all the stop words are removed using NLTK and the punctuations are removed using the string library's translator function. NLTK POS-Tagging and Tokenization is then used to separate the review text into separate words and do polarity prediction of words which come out to be adverbs, verbs, nouns and adjectives after tokenization. These words scanned through the negative and the positive word lists. If found in the negative list, a score of -1 is given to word or 1 is given if found in the positive list. If the word is not found in both the lists, SentiWordNets library of NTLK is used to predict the sense of the word. If found, the sense is passed through the same process of list searching as explained. The static window technique is then applied on the word and if an inversion word found, the polarity score's sign is changed. The static window technique of size n (n should be odd) looks for $\frac{n-1}{2}$ words before and after the target word and if an inversion word is found after scanning the negation word list, the polarity of the target word is inverted. For this project, the value of n for static windows is taken to be five. Finally, when all the polarities of words in the sentence is predicted, all the scores are summed up and divided by the number of tokens whose polarities were calculated to give each sentence a final score. If any Null value for the final score is present for any of the text segments, the mean of the present values is used to handle it.

3) Model Training

Before training the model, the polarity scores are scaled using Standard Scaler from the Scikit-learn library which scales the data in a range of 1 variance and near 0 mean. This helps in increasing the accuracy and fitting of data in the model to make the training easier. The whole data is divided into 70% training and 30% training. 6 different models are used for the fitting the sentiment analysis dataset namely –

Logistic Regression, Naïve Bayes, Multi-Layered Perceptron Neural Network, Support Vector Machine, Entropy Decision Tree, Gini Decision Tree, 5-Nearest Neighbour and 3-Nearest Neighbour.

3.1)Support Vector Machine (SVM):

The support vector machine is an efficient and adaptive algorithm which is frequently uses for classification of text, as well as sentiment analysis. This process involves erecting a hyperplane in a multi-dimensional room to clearly categorize data dots, hereby opinions. SVM is powerful because it can manage huge feature spaces which are typical for text data and successful in binary classification problems such differentiation of favourable and unfavourable sensations. SVM is especially relevant when considering linear separable data sets whose performance may get improved with kernel-based tricks for non-linearity. [13]

3.2) MLP (Multilayer Perceptron) Neural Network:

Many people may argue that sentiment analysis can be handled by multi layered perceptions (MLP) because it is one type of feed forward artificial neural networks that have greatly proved their capability in modelling the complex relations of data to come up with meaningful results. A multilayer perceptron is made up of several layers of nodes that are connected through weighted and biased edges, thereby enabling it to identify deep representation of data. For this reason, MLP is useful in sentiment analysis by extracting the complex features from the words' embedding and the surrounding. They are ideal for studying nuances within language and its relation to context within large and disparate corpuses of texts. [14]

3.3) K-Nearest Neighbours (KNN):

The KNN algorithm is one of the simple but powerful algorithms that can be used for both classification as well as regression issues. KNN classifies a textual document according to the sentiments exhibited by its 'K' nearest texts in sentiment analysis. Instance-based learning works best with a well-defined pattern of similar emotional expression in the data set. Nevertheless, KNN may encounter difficulties when handling high-dimensional data common in text mining and thus makes the process of feature selection and dimensionality reduction mandatory for efficient use of KNN. [15]

3.4) Naive Bayes Classifier:

Text classification, specifically sentiment analysis, is commonly addressed by a well-known algorithm known as Naive Bayes which is based on Bayes' theorem. It has also proved very effective, especially in analysing large datasets making it popular and easy to implement it. The Naïve Bayes assume independence among features, which, when it comes to text, implies whether some words or phrase are present or absent. This statement may be considered somewhat simplistic, but it frequently performs quite adequately in sentiment analysis especially when the data set involved is huge and the features represent the feelings which they carry. [16]

3.5) Decision Trees:

Decision Trees are a non-parametric supervised learning technique suitable for classification and regression tasks. They break down complex decision-making processes in sentiment analysis into decisions which are made based upon features extracted from text data. The trunk symbolizes the overall sentiment categorization, while the branches represent different paths of decision-making that eventually culminate in the corresponding sentiment classification. The principle is simple and easily understood. Yet, decision trees run a great risk of overfitting with large text corpora that must be carefully adjusted or even trimmed. [17]

3.6) Logistic Regression:

Despite being named otherwise, the logistic regression is commonly used in binary classification problems, including sentiment analysis. For instance, it estimates the likelihood that a datum is associated with a certain class thus suitable for classifying text as either conveying a positive or a negative sentiment. Logit regression is particularly efficient if the relationship between the independent variables (extracted from text) and the dependent variable sentiment, is nonlinear but may be approximated by a logistic model. [18]

4) Model Evaluation

The testing set of the data is finally used to predict ratings/sentiments using the 8 trained models specified in the previous section and their evaluation is done based on F1-Score and Accuracy. Similar evaluations are also done with the BOW and the TF-IDF algorithm to compare the results achieved.

The created algorithm holds the ability to work on dataset with different contexts with same precision. To achieve this, the same approach as discussed above is used with different lists for polarity searching. For example, the same algorithm is also carried out on a spam-ham detection dataset. The algorithm remains the same with a difference that the list consisting of positive polarity words in replaced with a list carrying words which are more likely to occur in a ham message and the list consisting of negative polarity words is replaced with a list carrying words which are more likely to occur in a spam messaging. The dataset used for this purpose is also extracted from kaggle which consists of a total of 5572 messages, each labelled as either ham or spam.

The above discussed approach uses only the score obtained by polarity calculation of each text segment. However, this is not enough on its own to be better than BOW and TF-IDF algorithm. To become useful, this model can combine it's understanding of semantics with TF-IDF's and BOW's power of performing on massive corpora of text by using both, polarity score and frequency vector (as extracted by TF-IDF, BOW or both) to become even better than the latter. These observations and results are discussed in the next section.

IV. RESULTS AND DISCUSSION

Multiple algorithms including the proposed one - Context Polarity, the traditional ones – BOW and TF-IDF and combinations of both were trained across multiple machine learning models as discussed in the previous section to work on two datasets of different context – Sentiment Analysis and Spam-Ham Detection . Table 1 and Table 2 displays the results of these experimentations followed by a detailed analysis which would justify the power of the proposed model.

ALGORITHM	METRIC (%)	Logistic Regresssion	MLP Neural Network	Naïve Bayes	Support Vector Machine	KNN - 9	KNN - 5	Decision Tree - Gini	Decision Tree - Entropy
Context Polarity (CP)	Accuracy	90.5834	90.4167	90.8334	91.0001	90.5834	90.0834	87.6667	87.6667
	F1 - Score	90.5834	90.4167	90.8334	91.0001	90.5834	90.0834	87.6667	87.6667
TF-IDF	Accuracy	87.5833	91.75	80.9167	88.1667	87.6667	87.6667	84.9167	84.3333
	F1 - Score	93.1994	95.3058	89.0796	93.4922	93.2172	93.1734	91.1405	90.8382
BOW	Accuracy	92.4167	92.4167	80.0000	88.0000	84.0000	84.5000	86.3333	85.8333
	F1 - Score	95.5631	95.5761	88.6148	93.2458	91.2249	91.4443	91.9608	91.5758
CP + TFIDF	Accuracy	92.9167	92.9167	80.6667	93.3333	91.5833	91.5833	88.5000	89.5000
	F1 - Score	95.9350	95.9271	89.1386	96.1575	95.1231	95.1419	93.2617	93.8776
CP + BOW	Accuracy	92.9167	92.0834	80.8333	91.8333	85.8333	86.6667	89.8333	89.0000
	F1 - Score	95.8957	95.4567	89.2221	95.2473	92.2937	92.6874	94.0661	93.5484
CP + BOW + TF-IDF	Accuracy	92.9167	92.6667	80.7500	91.7500	85.9167	86.6667	88.4167	88.4167
	F1 - Score	95.8957	95.7733	89.1803	95.1965	92.3356	92.6874	93.1963	93.2294

Table 1 : Results after various algorithms were used across multiple models for Sentiment Analysis

ALGORITHM	METRIC (%)	Logistic Regresssion	MLP Neural Network	Naïve Bayes	Support Vector Machine	KNN - 9	KNN - 5	Decision Tree - Gini	Decision Tree - Entropy
Context Polarity (CP)	Accuracy	86.9019	89.4138	84.9282	86.9019	88.8755	88.3971	89.3540	89.3540
	F1 - Score	92.9920	94.1293	91.8446	92.9920	93.8532	93.5719	94.1176	94.1176
TF-IDF	Accuracy	95.0956	98.2655	90.9090	97.3086	93.2416	91.0885	96.3516	95.9330
	F1 - Score	97.2757	99.0212	94.6251	98.4873	96.2914	95.1670	97.9286	97.6870
BOW	Accuracy	97.8468	97.7870	89.6531	97.3086	89.1148	90.8492	96.1124	96.2918
	F1 - Score	98.7713	98.7376	93.7205	98.4688	94.0832	94.9786	97.7792	97.8767
CP + TFIDF	Accuracy	94.1387	97.6674	90.4904	97.4282	93.8995	91.7464	95.9928	94.8564
	F1 - Score	96.6824	98.6519	94.2076	98.5136	96.5163	95.3409	97.6646	96.9993
CP + BOW	Accuracy	97.4880	97.5478	91.0287	97.8468	88.8157	90.1913	95.9330	95.7535
	F1 - Score	98.5636	98.5992	94.5887	98.7671	93.9147	94.6229	97.6775	97.5743
CP + BOW + TF-IDF	Accuracy	97.3086	98.0263	89.3540	98.1459	89.2344	90.1315	94.9162	95.5143
	F1 - Score	98.4667	98.8725	93.6246	98.9394	94.1482	94.6060	97.0740	97.4217

Table 2 : Results after various algorithms were used across multiple models for Ham-Spam Detection

The results obtained through different models and algorithms when applied on the Sentiment Analysis dataset are shown in Table 1. Analyzing the results, it is clearly evident that even though the proposed Context Polarity algorithm, which only uses the polarity scores received after processing the text, gives a satisfactory value of F1-Score across all models with a high of 94.9105% for the Support Vector Machine, is still less than the best performance obtained by the TF-IDF and BOW algorithm trained on the MLP Neural Network with an F1-score of 95.3058% and 95.5761% respectively. A similar pattern is also seen when analyzing the results given by the Spam-Ham Detection dataset as displayed in Table 2 where the TF-IDF and BOW algorithms work massively well using the MLP Neural Network model with an F1-score of 99.0212% and 98.7376% respectively meanwhile, the proposed Context Polarity algorithm is only able to achieve an F1-score of 94.1293% using the same model. This shows that for the sentiment analysis dataset in which the need of semantic understanding is important, the proposed model can give satisfactory result being able to perform as well as BOW and TF-IDF however, for a dataset like spam-ham analysis which are more word-frequency oriented, it is very difficult to compete with the high performance obtained by the traditional methods. Hence, it is safe to say that the proposed algorithm can understand the semantics of a text segment it is analyzing but to get its performance at par with other methods, combined algorithms are required.

The combined algorithms, which use both – polarity score of the Context Polarity algorithm and the feature vector of BOW or TF-IDF (or both), are seen to be working extremely well for both the datasets. In Table 1, it can be clearly seen that the combination: CP + TF-IDF, is able to achieve the best running F1-score of 96.1575% when trained on the Support Vector Machine classifier. Even when compared to the performance of BOW and TF-IDF when trained on MLP Neural Network, the F1-score of the Cp + TF-IDF when trained on the same model is higher than the latter (95.9271%). It is evident now that the combination of CP+TF-IDF is a better performer than the traditional algorithms where the importance of semantics is present. Even for a dataset which is dependent on word frequency like Ham-Spam detection, the combination algorithms can perform comparable to if not better than the traditional algorithms. It can be seen in table 2 that the best performance for a combination model is obtained by the CP+BOW+TF-IDF algorithm with an F1-score of 98.9394 when trained on the Support Vector Machine classifier as compared to the f1-score achieved by the TF-IDF algorithm when trained on the MLP Neural Network model.

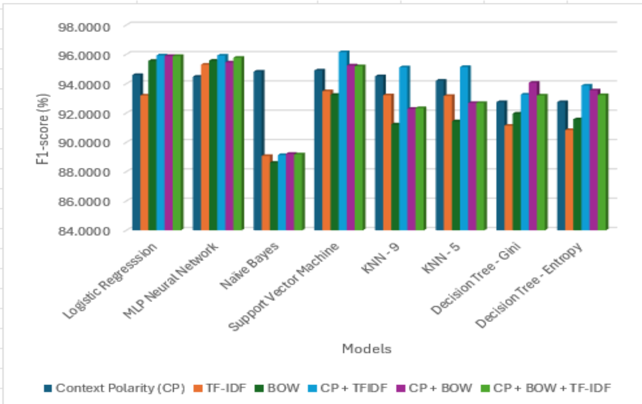

Fig 1: Comparison between algorithms when trained on different models and applied on the Sentiment analysis Dataset.

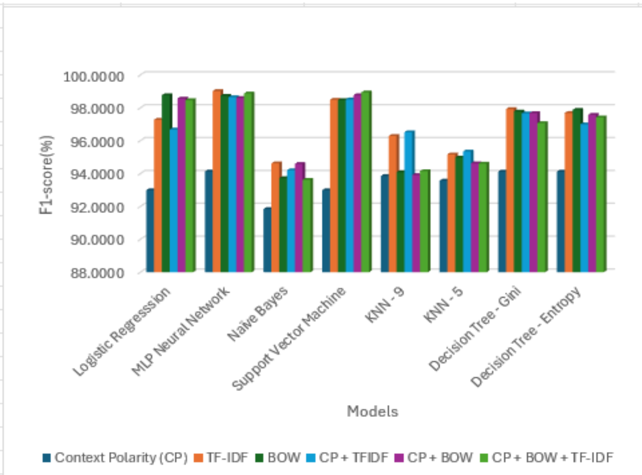

Fig 1: Comparison between algorithms when trained on different models and applied on the Spam-Ham Detection Dataset.

Figure 1 and Figure 2 gives a fair comparison showcasing the power of combined Context Polarity models as compared to the traditional models. Even though, the performance provided by models using only context polarity was comparatively less than the performance provided by BOW and TF-IDF, it was found that it has the ability to understand semantics very well can effectively be combined with the traditional methods to become even better than the latter in few cases by having both, the power to understand semantics as well as the ability to handle massive corpora of words.

V. Conclusion and Future Scope

This paper proposed the idea of a new text classification algorithm which focusses on understanding the semantics of a text competing with the traditional algorithms like Term frequency – Inverse Document Frequency and Bag-Of-Words which lack this understanding. The proposed algorithm functions by giving a polarity score to the text in analyses related to the context it is working on hoping to enhance the accuracy and performance with which text classification can be done. The study showcases the application of the proposed model on sentiment analysis and spam-ham detection datasets while comparing the performances achieved with that of BOW and TF-IDF.

The results achieved indicated that the proposed algorithm, which achieves satisfactory results, is not alone sufficient to perform better than the performance and reliability of BOW and TF-IDF which work on word-frequencies especially for datasets where the importance of semantic understanding is less. However, a combination of the two different algorithms can achieve this even showcasing better performance than the traditional algorithms in few scenarios.

Future experimentations could aim on improving the approach to apply context polarity for semantic understanding enabling text classification models to increase their performance. In addition to this, more detailed studies on the combined model's performance on other datasets with different contexts other than spam-ham detection and sentiment analysis would help in providing a better estimate of the algorithm's actual runtime performance. Moreover, text classification is vast field making use of Natural Language Processing and further exploration to discover novel techniques which could help in text understanding would be extremely helpful in making more robust text classifiers. As a whole, this paper gives rise to multiple pathways leading to more complex advancements stressing on the importance of semantic understanding in improving text classification.

References

[1] N. Pittaras, G. Giannakopoulos, G. Papadakis, and V. Karkaletsis (2021), "Text classification with semantically enriched word embeddings," *Natural Language Engineering*, vol. 27, no. 4, pp. 391–425. doi:10.1017/S1351324920000170J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] Gupta, S., Sharaff, A. & Nagwani, N.K (2024). Cancer hallmark analysis using semantic classification with enhanced topic modelling on biomedical literature. *Multimed Tools Appl*. https://doi.org/10.1007/s11042-024-18533-0

[3] Li, Y., Liu, Y., Zhu, Z. *et al* (2023). Exploring semantic awareness via graph representation for text classification. *Appl Intell* 53, 2088–2097. https://doi.org/10.1007/s10489-022-03526-z

[4] Yang, S., Wei, R., Guo, J. (2020). Semantic Document Classification Based on Semantic Similarity Computation and Correlation Analysis. In: Chao, KM., Jiang, L., Hussain, O., Ma, SP., Fei, X. (eds) Advances in E-Business Engineering for Ubiquitous Computing. ICEBE 2019. Lecture Notes on Data Engineering and Communications Technologies, vol 41. Springer, Cham. https://doi.org/10.1007/978-3-030-34986-8_1

[5] Yih, Wen-tau, Geoffrey Zweig, and John C. Platt (2012). "Polarity inducing latent semantic analysis." *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.

[6] Tian, L., Lai, C., & Moore, J. D. (2018). Polarity and intensity: the two aspects of sentiment analysis. *arXiv preprint arXiv:1807.01466*.

[7] Ayeste , Z., Noferesti, S (2022). A semantic approach based on domain knowledge for polarity shift detection using distant supervision. *Prog Artif Intell* 11, 169–180. https://doi.org/10.1007/s13748-021-00267-x

[8] https://medium.com/analytics-vidhya/natural-language-processing-nlp-introdu ction-fe48e9b7ec8d

[9] https://www.nltk.org/

[10] Baccianella, Stefano & Esuli, Andrea & Sebastiani, Fabrizio. (2010). SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining.. Proceedings of LREC. 10.

[11] Minqing Hu and Bing Liu. "Mining and Summarizing Customer Reviews." Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004), Aug 22-25, 2004, Seattle, Washington, USA

[12] Farooq, Umar & Mansoor, Hasan & Nongaillard, Antoine & Ouzrout, Yacine & Qadir, Muhammad Abdul. (2016). Negation Handling in Sentiment Analysis at Sentence Level. Journal of Computers. 12. 470-478. 10.17706/jcp.12.5.470-478.

[13] https://scikit-learn.org/stable/modules/svm.html

[14] https://www.sciencedirect.com/topics/computer-science/multilayer-perceptron

[15] Zhang Z. Introduction to machine learning: k-nearest neighbors. Ann Transl Med. 2016 Jun;4(11):218. doi: 10.21037/atm.2016.03.37. PMID: 27386492; PMCID: PMC4916348.

[16] Chen, H., Hu, S., Hua, R. et al. Improved naive Bayes classification algorithm for traffic risk management. EURASIP J. Adv. Signal Process. 2021, 30 (2021). https://doi.org/10.1186/s13634-021-00742-6

[17] Rokach, L., Maimon, O. (2005). Decision Trees. In: Maimon, O., Rokach, L. (eds) Data Mining and Knowledge Discovery Handbook. Springer, Boston, MA. https://doi.org/10.1007/0-387-25465-X_9

[18] https://christophm.github.io/interpretable-ml-book/logistic.html