Dictionary-based Sentiment Analysis on University-Government Issue related tweets

Homer Ian B. Reyes
New Era Univerisity
Quezon City, Philippines
homer.reyes@neu.edu.ph

Jhamille P. Alba
New Era Univerisity
Quezon City, Philippines
jhamille.alba@neu.edu.ph

Malachi Jade S. Roque
New Era Univerisity
Quezon City, Philippines
malachi.roque@neu.edu.ph

Jerald P. Punzal
New Era Univerisity
Quezon City, Philippines
jerald.punzal@neu.edu.ph

***Abstract*— Twitter is one of the best platforms in today's time, to express people's thoughts and voice out their opinion by posting them on social media. Information shared on Twitter is always relevant and up to date, we get our dataset on Twitter by scraping Institutional-government related issue tweets. After extracting and preparing the data, we analyze the sentiment of each tweet using a dictionary-based approach and determine how they feel about the topic issue. After that, we organize and plot the sentiment scores to visualize more clearly the results. We find out that almost half of the tweets are neutral, nonetheless, positive tweets are more than negative tweets.**

Keywords—Sentiment Analysis, dictionary-based, Twitter, lexicon.

I. INTRODUCTION

This study focuses on gathering information about Institutional-government related issue tweets from Twitter. Social media platforms like Twitter are good sources of information; news, opinions, and views for, different perspective responses The researchers gathered data from Twitter using an application called Twint - Twitter Intelligence Tool the researchers gathered English and Tagalog tweets about Institutional-government related issues from Twitter. The sentiment analysis approach is utilized to measure the feeling that a text conveys to the reader. This approach is applied to this study to determine the sentiment of the tweets. Knowing what individuals are talking about and understanding their issues and conclusions is profoundly profitable to businesses, directors, political campaigns. And it's truly difficult to physically examined through such expansive volumes and compile the themes. Thus, is required an automated algorithm that can examine through the content archives and automatically output the sentiment of the topics discussed.

II. RELATED LITERATURE

Data preprocessing is an important task that prepares the text for further process and analysis. These techniques simply bring the document to a format that is easily understandable, predictable, and analyzable by the machine.

A. Stopwords Removal

These are words in any form of language that is common and does not give value or meaning to a sentence. This can be safely removed and still the thought of the document remains understandable. Only the keywords that form the topics are extracted.

- To get better and decent results, S.P. Paramesh [1] proposed in the automation of IT tickets that the data must be cleaned. They used the removal technique and used the standard English Stop words list. Thus removing unnecessary words before analyzing the data and making the results more accurate.
- Daša Munkova [2], proves that removing stop words has a crucial role in the quantity of extracted rules in text processing on their study in transaction/sequence model. It also proved the significance of the quality of extracted rules in the case of paragraph sequence identification.
- Filtering text usually occurs before analyzing in further stages. Common filtering is stop words removal. These are the common words that appear on the data that does not have much meaning (e.g. prepositions, conjunctions, etc.). Likewise, words that are common and rarely used also possibly have no significant relevance and can be removed from the documents[3,4].
- Ravi Lourdusamy, Stanislaus Abraham [5] stated that stop words do not provide the grasp of the context and they can be safely removed from the textual data without ruining the meaning. Challenges in the process of stop word filtering are first, difficulty in constructing a list of stop words that is appropriate on the given dataset because of its inconsistency between different textual sources, and secondly, their high frequency of occurrences pose difficulty in processing the data.
- According to M.F. Porter [6], keywords in the text are the most important part of the documents, unlike stop words it is considered a division of natural language. It makes the text look heavier and noisy, it is also less important for analysts. Only keywords are important and measured in any text mining applications.
- Vairaprakash Gurusamy et al. [7], analyzed the importance of pre-processing in text mining, information retrieval, and natural language processing. The paper examines and shows the need for text pro-processing in NLP systems and one of

the pre-processing methods they proposed is stop word removal.

- Kathiravan and N.Haridoss's [8] purpose of removing the stop words is to extract the content words from the given data and it can control the increase of term space's dimensionality. Stop words removal is typically used in preprocessing tasks in various NLP problems.
- Xue, X. and Zhou, Z. proved that [9], most of the words frequently used in the text are useless in Information Retrieval (IR) and data mining. These words are called 'stop words', which repeatedly occur on the data that carry no information (e.g. pronouns, prepositions, conjunctions). It shows that in the English language, there are about 400-500 stop words in total that are listed. The first step during preprocessing is to remove these Stop words, which have proven as very important.

B. *Stemming*

Stemming is a preprocessing procedure that reduces all words with the same root into their similar form, it cuts the word of its derivational and inflectional suffixes.

- The stemming algorithm focuses on acquiring the root of a word. They are language-dependent and were first introduced by Julie B. Lovins [10] in 1968.
- Vijayarani et al., (2015) presented an exhaustive study on pre-processing techniques. This research was used to bring out three classifications of the stemming algorithms, viz., truncating, and statistical and mixed methods [11].
- Anjali Ganesh Jivani [12] The purpose of stemming is to cut off the different grammatical forms or word forms of a word. This is its adjective, adverb, noun, verb, etc. Its objective is to remove or reduce the words that may have inflectional forms into their common base form, and occasionally derivationally related forms of a word are included in this.
- C.Ramasubramanian et.al [13] from ANNA University looks into the disadvantages of one of the stemming algorithms called MF Porter's algorithm. He discussed the drawback of the existing approach along with the process to overcome the disadvantages. Stemming doesn't always provide good results as the word loses its meaning. Stemming is an aggressive word shortening technique that aims to bring a word to its base root. The suffixes are chopped off to bring it to the base root while the semantic meaning of all the different forms remains the same.
- Root word or stem word is identified by using this method from the given inflection word. For example, the words visit, visited, visiting all can be stemmed from the word "visit". it is used to remove prefixes and suffixes and it gives effective and exact matching stems to save time and memory space.[14],[15] and it is used to improve the effectiveness of retrieval and reduces the size of indexing files.[16],[17].
- Stemming converts textual data to their root forms, which contains a great deal of language-dependent linguistic knowledge. The theory behind stemming is that all of the words with the same stem or root word mostly describe similar or relatively close meanings in the text. For example, the words, looks, looking, and looked can be stemmed from the words 'look'. In the present work, the Porter Stemmer algorithm[6].

C. *Tokenization*

Another technique that is essential for Natural Language Processing is splitting the entire text document into smaller units which are called a word. Tokenized data helps analyze the relationship between words and can be interpreted in different meanings.

- Jonathan J Webster and Chunyu Kit [18], used this method of breaking and separating a string of words into pieces which are called tokens. This method also segregates the symbols and characters such as punctuation marks that can be safely removed. This list of tokens is then used for further language processing.
- Vijayarani et al.,[19] have determined the capability of tokenization tools two measures viz., limitations and outcome are used in a comparative study. Based on their study the authors come to an end that the NLP.net tokenizer is the best among the tools studied.
- Vikram Singh et al.,[20] from the National Institute of Technology, Haryana, used essential pre-processing techniques for Information Retrieval Systems. The study focuses on tokenization and stemming algorithms which is one of the most common and simple yet effective to enhance the data that has been used.
- onit Singh [21] evaluates state-of-the-art systems for IE tasks such as NER, conference resolution, temporal expression extraction, and relation extraction. The author highlighted the fact that if the errors are ignored in text preprocessing stages they will get propagated to information extraction tasks in later stages. It just shows that the tokenization process is just splitting the collection of words from the given data, which the result becomes the input to the next stages of processing of data mining [8].

D. *Part-of-Speech Tagging*

Tagging every word in a sentence can be considered useful to indicate the parts of speech such as noun, pronoun, verb, adverb, adjective, preposition, conjunction, and interjection. Depending on the context.

- S.P. Paramesh [1] used POS Tagging as one of his methods in cleaning data as he proposed that the automation of IT tickets the description data must be cleaned to get better results.
- Daša Munkova et al. [2] used POS Tagging for the classification of words in his Transaction/sequence Model to determine and identify word sequences in

paragraphs. POS enhances the “word” and its “context” with a huge volume of information about itself and its neighbors. It helps conclude possible knowledge about the neighboring words and syntactic structure weaving around the text.

- POS Tagging is the process of assigning a POS marker to each word in an input text. Since words have more than one POS in a sentence, the aim for tagging is to tag the right words with the right parts-of-speech such as the word ‘book’ can be both a noun and a verb. Thus, POS tagging resolves these ambiguities and chooses the right proper tag for the context [5].

E. Lemmatization

A process to convert the word into its base form or which is called ‘Lemma’ by analyzing the intended meaning of the word instead of its literal meaning.

- Lemmatization considers the morphological analysis of words such as grouping up different inflected forms of a word to be analyzed as an individual [22].
- Lemmatization aims to reach a common stem by minimizing the different forms of words. Lemmatization aims to remove the ‘-s’, ‘-es’, ‘-ed’, and ‘-ing’ suffixes and reverts them to their base form which is called ‘lemma.’ Lemmatization would try to return either ‘see’ or ‘saw’ depending on whether the word token was a verb or a noun [5].
- Mohammad Taher Pilehvar [23] analyzed the effect of text preprocessing techniques on the decisions of a neural text classifier, he stated that the performance of preprocessing techniques such as lemmatization depends on the domains to which the datasets correspond to.

F. Word Sense Disambiguation

It is a process of determining which meaning of the word is activated according to the use of the word in a particular context. Homonymy is defined as the words having the same spelling or same form but having different and unrelated meanings.

- Words can provide different meanings depending on how it is used and structured in a sentence. This may mislead the meaning of the context and get unexpected results. WSD distinguished the different meanings of similar words according to how it was used in the text [24].
- It has been a long-standing research objective for NLP. Word Sense Disambiguation is intended to find the right meaning of the words in a particular context.. .e., groups of words appearing next to each other in the same context have related meanings [25].

G. Change to Lowercase

This step is commonly neglected and ignored but is one of the simplest yet effective steps of preprocessing the text especially in cases where the dataset is significantly scattered within. It has been found that variation in capitalization gives different results.

- Feras Al-Hawari and Hala Barham [26] analyzed and found the accuracy by comparing the prediction accuracy of classification models by training it with data that was not preprocessed and the other was preprocessed and cleaned. One of the techniques used in changing the lowercase. The test revealed that classification models that were implemented had a higher accuracy for a preprocessed and cleaned data with 20 to 30%.

H. Removal of Punctuations

Removing the noise is helpful when doing text analysis on data. Punctuations are one of the examples that make the text noisy, unstructured documents have numerous such occurrences such as exclamation marks, apostrophe, comma, and so on. The machine does not understand it anyway thus, it can be removed.

- Shreedhara K.S [29] has emphasized having efficient text preprocessing. Stop words are removed along with the date, time, and numerical data using regular expressions. Regex is helpful and commonly used whether it is a pattern matching or searching method, it is applied to a sequence of characters to identify and replace the punctuations with a certain rule.

I. Sentence Segmentation

It is a process of dividing a text document into individual meaning sentences or units. It is also called Sentence boundary detection. It is usually split apart by a punctuation mark. Identifying word boundaries is useful for further processing that can be done on each sentence.

- Elizabeth D. Trippe [28] discussed the importance of text preprocessing relating to feature extraction to feature selection in later stages of Natural language Processing. Sentence segmentation, filtering, and stemming are used and give a significant effect on the process.

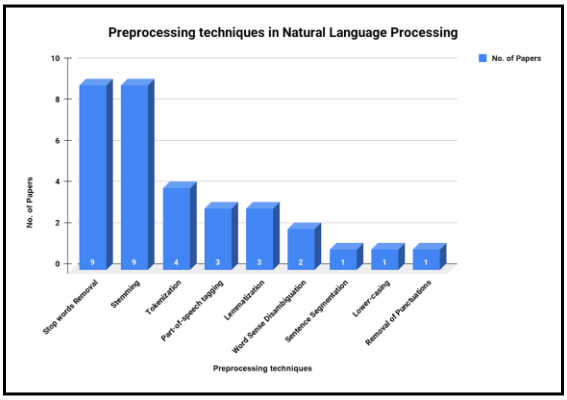

Fig. 1. Preprocessing techniques

“Fig. 1” shows that out of nine of the pre-processing techniques, the most used methods are both stop words removal and stemming, with both methods having been used by nine (9) different papers; while sentence segmentation, lower-casing, and removal of punctuation are the least used pre-processing techniques with all three only have one (1) paper used in their methods.

III. METHODOLOGIES

A. *Materials Used*

Python: A high-level interpreted programming language created by Guido van Rossum and first released in 1991. Python is a simple yet powerful language that has excellent functionality for processing linguistic data like Natural Language Processing.

Jupyter Notebook App: It is an open-source web application that can be used to create documents that contain live code, equations, visualizations, and text. Notebook documents are both human-readable documents containing the analysis description and the results as well as executable documents that can be run to perform data analysis. This is the main Integrated Development Environment (IDE) that we used for our study.

Twitter Intelligence Tool (TWINT): Twint is an advanced Twitter scraping tool written in Python that allows for scraping tweets from Twitter without using Twitter's API [29]. Twint is responsible for scraping the data we used in this study. The tweets we gathered are University-Government accord related tweets. The issue was started on January 21, 2021. Since that date, we collected a total of 575 tweets.

B. *Data Pre-processing*

Handling contractions: Negation are words that affect the sentiment orientation of other words in a sentence. Examples of negation words include not, no, won't, shouldn't, wouldn't. etc. To address this problem we use the contractions module to handle shortened words which are very common negations.

Removing Punctuations and Symbols: The tweets we gathered are unstructured and it has noisy text and that does not give importance to this study. The main objective is to extract only the tweet itself and disregard the other information like hashtags, mention, and links. We used the Regular Expression (Regex) module to remove the unwanted punctuations and symbols on the data.

Lower-case: We utilized lowercasing since capital letters give different results, lowercasing the text makes it easier to handle other preprocessing methods like stop words removal, negation, and more.

Tokenization: It is essentially splitting a phrase, sentence, paragraph, or an entire text document into smaller units, such as individual words or terms. Each of these smaller units are called tokens. We used tokenization to split the tweet into individual words.

Stop words removal: Common and meaningless text are safely removed in the context. Stop words ISO is the most comprehensive collection of stop words for multiple languages. We used this module to collect English and Tagalog stop words and use it as a basis on removing the stop words.

C. *Sentiment/Opinion Dictionary*

Dictionary-Based is a computational approach to measure the feeling that a text conveys to the reader. In the simplest case, sentiment has a binary classification: positive or negative. This method relies heavily on a predefined list (or dictionary) of sentiment-laden words. This approach is applied to this study to determine the sentiment of the tweets.

We created two dictionaries that can be used as a justification for determining the sentiment of the tweets, which are positive and negative dictionaries. Each dictionary is composed of English and Tagalog words. The Tagalog sentiment lexicon was generated via graph propagation based on a knowledge graph. It contains both positive and negative lexicons for 81 languages including Tagalog [30]. The English sentiment lexicon has a list of around 6800 words in total both English positive and negative sentiment words. This list was compiled over many years by Minqing Hu and Bing Liu [31,32]. Together with these lexicons, it will suffice to analyze the sentiment of the tweets.

Now, given a dictionary of words associated with positive and negative sentiment, the sentiment of a text is calculated as follows:

Count the number of positive and negative words in the tweet. We calculate the difference to define the sentiment of the text. The formula we used is *Absolute Proportional Difference* whereas, the absolute difference of two real numbers, describes the distance on the real line between the points corresponding to (P) positive lexicon and (N) negative lexicon [33].

$$\textit{Sentiment} = (P—N) / \textit{Number of words}$$

IV. RESULTS

After analyzing and determining the sentiment of the tweets. The researchers plot the results using table, graph, and word cloud to visualize the outcome.

TABLE I. SENTIMENT ANALYSIS

Sentiment Scores				***Total No. of Tweets***
Polarities	***Positive***	***Neutral***	***Negative***	
No. of Tweets	173	266	136	575

Table 1 shows the result of the dictionary-based sentiment analysis on extracted tweets. Among the 575 tweets, we find out that it has 173 positive tweets, 136 negative tweets, and the remaining are neutral regarding the topic.

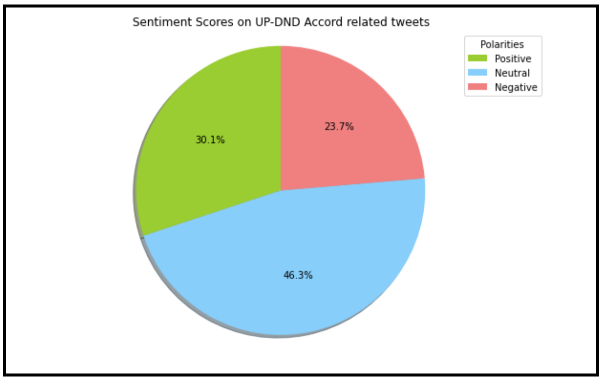

Fig. 2. Sentiment Analysis Scores

“Fig. 2” shows that 30.1% of the tweets were positive tweets, on the other hand 23.7% are negative tweets. But the most noticeable portion is that 46.3% of the collected data are determined as neutral tweets.

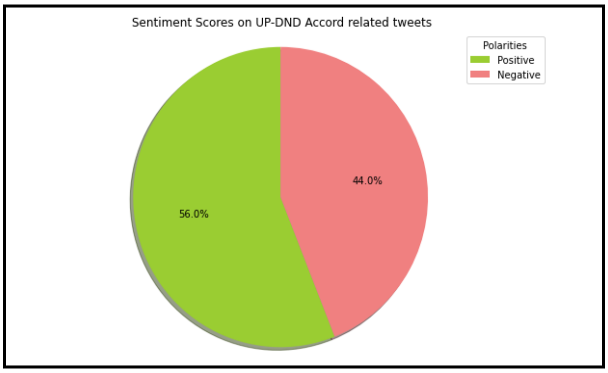

Fig. 3. Positive and Negative Polarities

“Fig. 3” shows that the gap between positive and negative sentiment is 12%. Therefore, more positive responses that was expressed on social media, which is on Twitter.

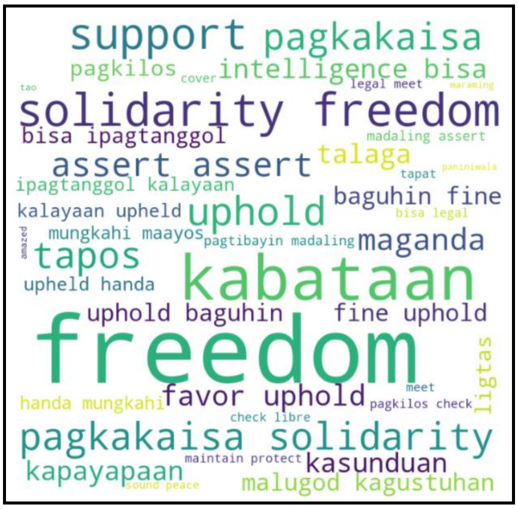

Fig. 4. Positive Word Tweets

“Fig. 4” shows the most frequently used positive words on tweets. The most common words used are freedom, kabataan, and solidarity which is relevant on the issue topic.

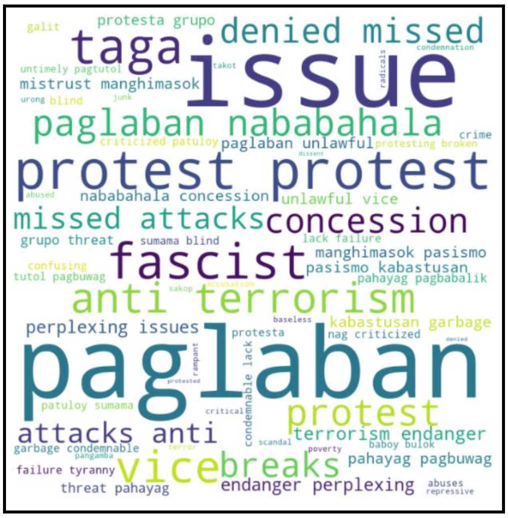

Fig. 5. Negative Word Tweets

“Fig. 5” shows the most frequently used negative words on tweets. The most common words used are *paglaban*, *issue*, and *fascist* which gives significantly bad impression about the issue topic.

V. CONCLUSION

To sum up everything, the text preprocessing techniques form a major contributor in increasing the accuracy of any text-based machine learning algorithm, and out of the 10 preprocessing techniques used, stopwords and stemming stood out the most, having 9 papers using the said techniques for filtering their data.

While sentence segmentation, lower-casing, and removal of punctuations, were at the bottom and were only used once. They were used only depending on the algorithm or experiment of the paper that used this.

Stop words removal had a substantial impact on the quantity and quality of extracted data and provided desired results from the papers that used this. The same goes for stemming, they became mandatory techniques in achieving most of the authors’ desired results.

Natural Language is complicated, and 100% accuracy cannot be expected through any machine learning model by the papers, but machine learning algorithms have so far made a near close prediction of analyzing text.

REFERENCES

[1] S.P. Paramesh, K.S. Shreedhara, “IT Help Desk Incident Classification Using Classifier Ensembles”, ICTACT Journal On Soft Computing, July 2019, Vol: 09, Issue: 04.

[2] Daša Munkova et. al, “Data Pre-Processing Evaluation for Text Mining: Transaction/Sequence Model”, 2013 International Conference on Computational Science, May 2013, p. 1198 – 1207.

[3] Hassan Saif, Miriam Fernández, Yulan He, and Harith Alani, “On stopwords, filtering and data sparsity for sentiment analysis of twitter”, 2014.

[4] Catarina Silva and Bernardete Ribeiro, “The importance of stop word removal on recall values in text categorization. In Neural Networks”, 2003, Proceedings of the International Joint Conference on, Vol. 3. IEEE, 1661–1666.

[5] Ravi Lourdusamy, Stanislaus Abraham, “A Survey on Text Pre-processing Techniques and Tools”, International Journal of Computer Sciences and Engineering, April 2018, Vol: 06, Issue: 03.

[6] M.F. Porter, “An Algorithm for Suffix Stripping”, Program, vol. 14, no. 3, pp. 130-137, 1980.

[7] Vairaprakash Gurusamy and Subbu Kannan, “Preprocessing Techniques for Text Mining”, October 2014.

[8] P. Kathiravan and N.Haridoss, “Preprocessing for Mining the Textual data - A Review”, International Journal of Scientific Research in Computer Science Applications and Management Studies, September 2018, Vol: 07, Issue: 05.

[9] Xue, X. and Zhou, Z. (2009) Distributional Features for Text Categorization, IEEE Transactions on Knowledge and Data Engineering, Vol. 21, No. 3, p. 428-442.

[10] Julie B Lovins, “Development of a stemming algorithm”, MIT Information Processing Group, Electronic Systems Laboratory, 1968.

[11] Vijayarani S, Ilamathi J, & Nithya, International Journal of Computer Science & Communication Networks, Vol 5(1), pp. 7-16, 2015.

[12] Anjali Ganesh Jivani, “A Comparative Study of Stemming Algorithms”, International Journal of Computer, Technology and Application, Volume 2, ISSN:2229-6093.

[13] C.Ramasubramanian and R.Ramya, “Effective Pre-Processing Activities in Text Mining using Improved Porter’s Stemming Algorithm”, International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 12, December 2013.

[14] Funchun Peng, Nawaaz Ahmed, Xin Li and Yumao Lu, “Context sensitive stemming for web search”, Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 639-646, 2007. Harman Donna, “How effective is suffixing?”, Journal of the American Society for Information Science, 1991; 42, 7-15 7.

[15] Hassan Saif, Miriam Fernandez,Yulan He, Harith Alani, “On Stopwords, Filtering and Data Sparsity for Sentiment Analysis of Twitter”, The 9th International Conference on Language Resources and Evaluation, At Reykjavik, Iceland, pp.810-817,2014.

[16] Krovetz Robert, “Viewing morphology as an inference process”, Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval, pp.191-202,1993.

[17] Melucci Massimo and Orio Nicola, “A novel method for stemmer generation based on hidden Markov models”, Proceedings of the twelfth international conference on Information and knowledge management, pp. 131-138, 2003.

[18] Jonathan J Webster and Chunyu Kit, “Tokenization as the initial phase in NLP. In Proceedings of the 14th conference on Computational linguistics-Volume 4”, Association for Computational Linguistics, 1992, 1106–1110.

[19] Vijayarani S, & Janani R, “Text mining: open source tokenization tools–an analysis”, Advanced Computational Intelligence 3.1: pp. 37-47, 2016.

[20] Vikram Singh and Balwinder Saini, “An Effective Pre-Processing Algorithm For Information Retrieval Systems”, International Journal of DatabaseManagement Systems (IJDMS) Vol.6, No.6, December 2014.

[21] Sonit Singh, “Natural Language Processing for Information Extraction”, July 2018.

[22] Allahyari, M. et. al., “A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques”, KDD Bigdas, August 2017.

[23] Mohammad Taher Pilehvar and Jose Camacho-Collados, ”On the Role of Text Preprocessing in Neural Network Architectures: An Evaluation Study on Text Categorization and Sentiment Analysis”, Proceedings of the 2018 EMNLP Workshop Blackbox NLP: Analyzing and Interpreting Neural Networks for NLP, pages 40–46.

[24] J. I. Toledo-Alvarado et al., “Automatic Building of an Ontology from a Corpus of Text Documents Using Data Mining Tools”, 2012.

[25] Joe Tekli, “An overview on XML Semantic Disambiguation from Unstructured”, Member, IEEE, 2016.

[26] Feras Al-Hawari and Hala Barham, ”A machine learning based help desk system for IT service management”, Journal of King Saud University – Computer and Information Sciences.

[27] Paramesh S.P and Shreedhara K.S, “Building Intelligent Service Desk Systems using AI”, IJESM 2019; Vol: 1, No: 2, pp: 01-08.

[28] Elizabeth D. Trippe, Krys Kochut, and Juan B. Gutierrez, “A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques”, July 2017.

[29] Cody Zacharias, Twint Project, github.com/twintproject/twint/tree/master/twint.

[30] Chen, Y., & Skiena, S. (2014). Building Sentiment Lexicons for All Major Languages. In ACL (2) (pp. 383-389).

[31] 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.

[32] Bing Liu, Minqing Hu and Junsheng Cheng. "Opinion Observer: Analyzing and Comparing Opinions on the Web." Proceedings of the 14th International World Wide Web conference (WWW-2005), May 10-14, 2005, Chiba, Japan.

[33] William Lowe, Kenneth Benoit, Slava Mikhaylov, and Michael Laver. (2011) "Scaling Policy Preferences From Coded Political Texts." Legislative Studies Quarterly 26(1, Feb): 123-155.