Proceedings of the International Conference on Electronics and Sustainable Communication Systems (ICESC 2020)
IEEE Xplore Part Number: CFP20V66-ART; ISBN: 978-1-7281-4108-4

# *Automated Identification of Disaster News for Crisis Management using Machine Learning and Natural Language Processing*

Jayashree Domala[1], Manmohan Dogra[2], Vinit Masrani[3], Dwayne Fernandes[4], Kevin D'souza[5], Delicia Fernandes[6], Tejal Carvalho[7]
[1,2,3,4,5,6] U.G. Student, [7]Assistant Professor
[1,2,3,4,5,6,7] Department of Computer Engineering, [1,2,3,4,5,6,7] St. Francis Institute of Technology

**Abstract— We are living in unprecedented times and anyone in this world could be impacted by natural disasters in some way or the other. Life is unpredictable and what is to come is unforeseeable. Nobody knows what the very next moment will hold, maybe it could be a disastrous one too. The past cannot be changed but it can act constructively towards the betterment of the current situation, 'Precaution is better than cure'. To be above this uncertain dilemma of life and death situations, 'Automated Identification of Disaster News for Crisis Management is proposed using Machine Learning and Natural Language Processing'. A software solution that can help disaster management websites to dynamically show the disaster relevant news which can be shared to other social media handles through their sites. The objective here is to automatically scrape news from English news websites and identify disaster relevant news using natural language processing techniques and machine learning concepts, which can further be dynamically displayed on the crisis management websites. The complete model is automated and requires no manual labor at all. The architecture is based on Machine Learning principles that classifies news scraped from top news websites using a spider-scraper into two categories, one being disaster relevant news and other being disaster irrelevant news and eventually displaying the relevant disaster news on the crisis management website.**

***Keywords— Crisis, Disaster, Machine Learning, Natural language processing, News, News Classification, Pandemic, Scrapy***

## I. INTRODUCTION

In times of sudden crisis outbreak, the problem in question is mismanagement. Health officials, the government failing to keep up with the pressures of recovery, help to citizens, dismay of overstretching health care systems and possible threat to public order, all becoming increasingly difficult with rise in population and corruption. This can impose a severe threat to national security as disasters in any form primarily impact population health. As a deduction to this point at issue, "Automated Identification of Disaster News for Crisis Management using Machine Learning and Natural Language Processing" is introduced as an affirmative solution to solving uncertainty of public security during the occurrence of a serious calamity or danger of lives. Moreover, in such crucial times, the authorities face numerous issues such as spread of fake news to the common public, fabricated news headlines for attention and money-making fraudulent media outlets and bogus facts spread via social media channels. Such acts cause the underlying doubt in public safety to increase while the correct facts remain veiled and away from public reach. This coupled with everyday crime poses an immense hazard to safekeeping of government working systems. Health and welfare of the populace plays a decisive role in smoothly working with health and non-health sectors responsible for preparing for and responding to emergencies.

A broad goal that needs to be accomplished is tackling this problem with a disaster management software, by use of machine learning, as the core of the project to develop a fully functional algorithm that can help disaster management websites to display genuine disaster relevant news. This being effective by scraping current news from the leading news websites and using machine learning algorithms, the news is classified into disaster relevant news or irrelevant news. Various classification algorithms were implemented like multinomial Naive Bayes, logistic regression, SVM, xtreme gradient boosting and random forest. The logistic regression model gave the best result.

## II. RELATED WORK

A number of researchers have worked on this type of disaster classification either on twitter datasets or on some particular website of news. The authors Mazhar Iqbal Rana, et al., have mentioned the need of classifying the news based on the headline of news rather than story line description of it. And then the authors have made comparisons of various methods and algorithms that can be used for this classification giving advantages and disadvantages based on points like run time, computation [1]. In another paper by Inoshika Dilrukshi et al., the authors have used twitter dataset having short messages retrieved by setting the threshold of 140 characters message and then used the method of bag-of-words words are used as features and their frequency formed the dataset which was trained using the SVM classifier and getting accuracy above 75% in 11 out of 12 categories of class the disaster was classified [2]. While in the paper written by Chee-Hong Chan et al., the authors have made a classifier for particular domain of only financial related data chosen from Channel News Asia website and then in a fixed set of 10 categories the authors have made a general classifier for classifying into those 10 distinct classes

 

successfully but they couldn't achieve good results for the personal classifier where the user will give his own category and a set of probable keywords which can come under that category. For both general and personal classifiers, the classification was done using SVM model [3]. The authors Dewi Y. Liliana et al., have also used SVM classifier as their model in prediction of Indonesian news into the 4 classes defined by them. The dataset was taken from online news website kompas which consisted of 150 news articles and then it was evaluated with three settings of SVM parameters yielding best accuracy 85% with C parameter between 60-150 and SVM gamma between 1.0 - 2.0[4]. In another paper by Beverly Estephany Parilla-Ferrer et al., the authors have used twitter data for an event of the Habagat flood of Metro Manila which occurred in 2012, and then that data is classified as disaster-informative or not informative tweets which results in SVM classifier with accuracy 80% and Naive Bayes 57%. The dataset used for evaluation had 1563 tweets in English language [5]. Authors Kevin Stowe et al., have collected all the tweets with a particular set of words to extract all news on Hurricane Sandy which had occurred in 2012 from twitter and the dataset collected was 22.2M. An SVM Classifier was used for classifying it in some of the classes and the accuracy achieved was 85% and F1-score of 55% [6]. A paper by authors Koustav Rudra et al., have also used twitter dataset pertaining to 4 events that had occurred and each of these 4 events have 5K tweets each making the dataset size to 20K. An SVM classifier is used on it to achieve an accuracy of 80% [7]. Authors Abeer Abdel Khaleq et al., have also published a paper in which they have worked with twitter dataset covering 3 major hurricanes and by making comparison of various classification algorithms they got accuracy of around 86% with logistic regression [8]. Authors Tim Nugent et al. have taken their dataset by querying Elasticsearch index 2405 news reports from 2012 to 2016. And this dataset was classified into seven disaster types. And there has been use of various classifier algorithms with SVM getting the best result with an F1-score of 77.3% [9]. Also, the authors Alberto Tellez Valero et al., have used a Spanish news corpus collected using various online Mexican newspapers from 1996 to 2004. The model got the F-1 score of 92% for that Spanish news corpus using SVM algorithm [10]. Authors Umid Suleymanov et al., have extracted the data from a Azerbaijani news article having 1,50,000 articles in it. Artificial Neural Network was used to classify the news of this Azerbaijani language to 8 different categories with an accuracy of 89.1% [11]. The authors Abu Nowshed Chy et al., have taken the Bangla news dataset using the crawler for extracting the data from online Bangla news sites. They have used the Naives Bayes classifier for classification of news articles [12].

## III. IMPLEMENTATION

Ideally, disaster relevant news is automatically delivered to crisis management websites, making their work more accurate and quicker in keeping the masses informed about any shortcomings. Thus, a system which automatically scrapes news from the websites, stores it in the database where it undergoes the classifier and the relevant disaster news is displayed on the crisis management website is obtained. To achieve this, the implementation carried on in two modules. Module 1 deals with the automation of scraping and Module 2 deals with the building of the classification model.

### *A. Module 1*

The system is made of an automatic scraper deployed on the server of Heroku, which continuously scraps the data from websites like Times of India, NDTV India and Indian express after every 5 minutes. This scraper scrapes the news from these websites using the following steps and the block diagram of the same is as shown in Fig. 1.

*1) Request for Source Code:*

Scrapy which is the heart of the system makes a request to all those websites like Times of India, NDTV India and Indian express which are fed to the scraper.

*2) Response of Source Code Returned to Scraper:*

Scraper gets the source code from these websites. After it receives the source code it checks whether the source code is a Java-Script Source code or a HTML source code. This step is necessary because Scrapy, the main framework of the system, can only interpret the HTML source code.

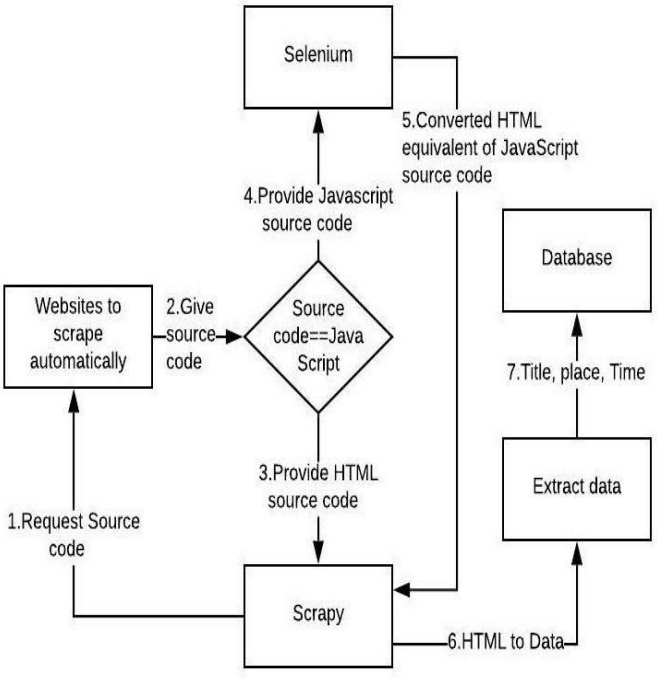


Fig.1. Block Diagram of the Module 1

*3) If Source Code is HTML, it goes to Scrapy:*

The Scrapy module will take the input as the HTML source code response which is received from the web page as input and give us the required data as the output by use of the Xpath.

*4) If Source Code is JavaScript, then it goes to Selenium:*

Selenium is a framework which will help parse the JavaScript to HTML. Once the scraper has detected that the source code is in JavaScript, it is passed on to Selenium for processing.

*5) Selenium Outputs HTML Source Code which is fed to Scrapy:*

Selenium will output the HTML converted source code from the JavaScript source code received. This HTML equivalent source code is provided to the Scrapy for the processing of data mentioned using the Xpath.

*6) HTML Source Code Yields Data:*

Using XPath, the model can specify which part of data from that web page needs to be extracted, which can also be multiple data specified using the multiple XPath of that web page.

*7) Store the Extracted Data into the Database:*

The data extracted like the Title of the news headline, Place of the disaster occurrence, Time of the occurrence. All these fields need to be added to a database that connect to the Scrapy module. In this case, it needs to be added to the Heroku PostgreSQL Database.

So, this whole process is repeated and if it detects any new data using the above steps, it is stored in the Database. And since this system is deployed onto the Heroku the entire system is automated as it repeats this cycle on its own through the cloud resources.

*B. Module 2*

The process of building a machine learning model for the purpose of classifying the news as relevant disaster and irrelevant disaster news has multiple steps. Every step is crucial and decides the future of the text classifier model. These steps have to be performed in the given order and the same is represented as a block diagram in Fig. 2.

*1) Data Collection:*

For the classification model, the source of data collection are news websites, as these are the most frequently updated resources with a lot of trustable information. The news websites that are selected for scraping are Times of India, NDTV India and Indian express. A Scrapy tool is used to gather this information. Spiders are deployed on the websites and data is extracted. Scrapy is preferred due to its dynamic

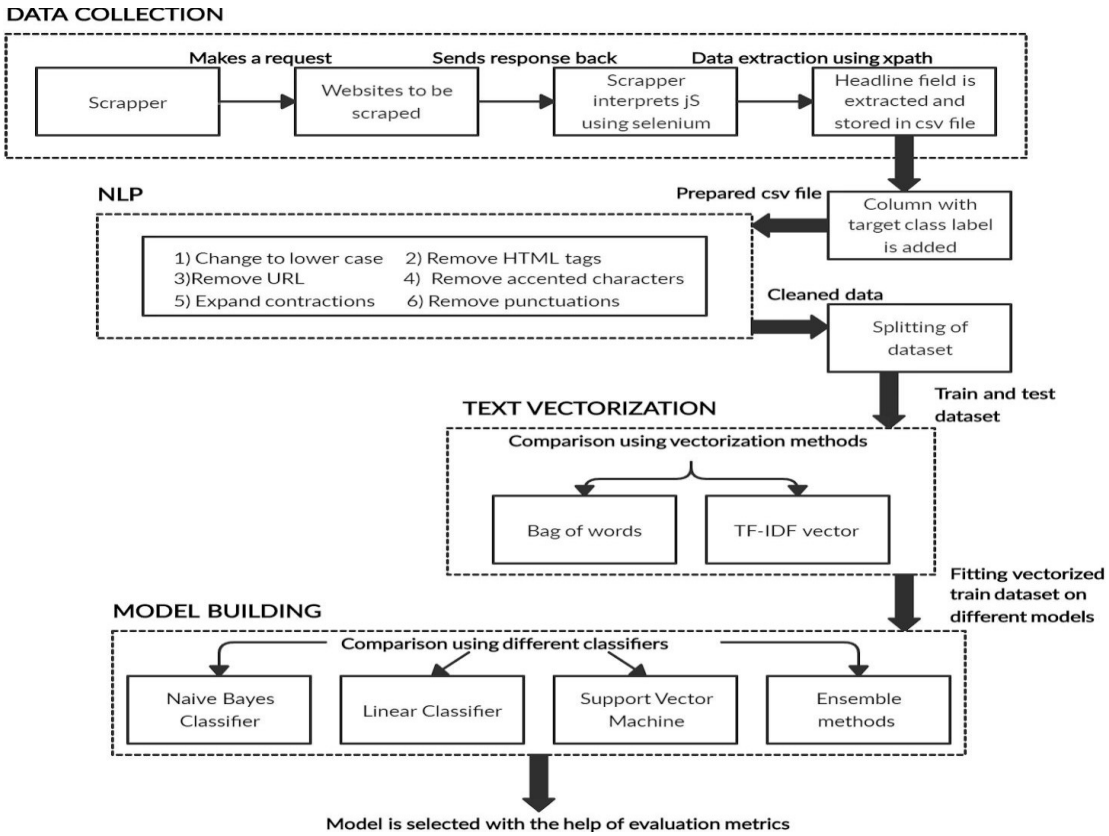


Fig.2. Block Diagram of the Module

Table I. Data preprocessing table

| Sr. No. | Type of Data Preprocessing | Attributes applied | Why was this preprocessing done | What was the output of this Data Preprocessing technique |
|---|---|---|---|---|
| 1 | Change to lower case | Title | Variation in input capitalization gives us different output. For example, India and india are the same but the machine learning model treats it differently. | All the upper case letters have been converted to lower case. |
| 2 | Remove HTML tags | Title | The data we got through web-scraping might contain a lot of noise. HTML tags do not add much value towards understanding and analyzing text so we need to remove the HTML tags. | All the HTML tags are removed. |
| 3 | Remove accented characters | Title | The text corpus may contain accented characters. Hence, we need to convert and standardize it into ASCII characters. For example, conversion of é to e. | All the accented characters are converted according to english language. |
| 4 | Remove URL | Title | The url links are of no use and will not help so we will remove it and replace it with "", eventually just wiping it. | All the URL's are removed. |
| 5 | Expanding contractions | Title | Contractions are shortened versions of words.For example, "do not" to "don't". Converting each contraction to its expanded, original form helps with text standardization. | All the contractions are expanded. |
| 6 | Removing special characters (punctuations, hashtags and @ | Title | Special characters and symbols are usually non-alphanumeric characters, which add to the extra noise in unstructured text. Simple regular expressions (regexes) can be used to remove. | The speical characters like @, # and punctuations are removed. |

attribute of real-time scraping of concurrent pages [13]. Two types of news are scraped: Disaster relevant news scraped from the disaster domain of the news website and other being bogus disaster irrelevant news. Eventually all the scraped news is saved on an excel sheet which has one column 'Title'. The number of rows in the dataset is approximately 11k.

*2) Preparation of Dataset:*

The dataset has to be prepared for preprocessing. The target/output variable column named as 'Label' is added. For each row, the label column is labelled '1' for relevant disaster news and '0' otherwise.

*3) Data Preprocessing:*

The next step is data preprocessing where the unstructured data is converted into a valuable form. This step is performed on the 'Title' column using the concept of NLP (natural language processing). The types of data preprocessing used is shown in table 1.

*4) Splitting of Dataset:*

The dataset is split into the training and the testing datasets in the ratio 80% and 20%.

*5) Text Vectorization:*

In this step, the text data is converted into numerical form [14]. Word Embedding algorithm is a process in which the input text is converted to the equivalent number representation. And in this process for the same text, different number of representations can be obtained. Two different types of word embeddings will be used for this system and the one giving best results will be selected.

*a) Bag of words:*

It's called bag of words because any order of the words in the document is discarded; it only tells us whether a word is present in the document or not. By using the CountVectorizer Function, the text document is converted into a matrix of word.

*b) TF-IDF vector:*

It stands for Term Frequency-Inverse Document Frequency which tells the importance of the word in the corpus or dataset. TF-IDF contains two concepts: Term Frequency (TF) and Inverse Document Frequency (IDF). TF is defined as how frequently the word appears in the document. IDF is based on the fact that less frequent words are more informative and important.

Both these types are implemented on word and character level along with unigram, bigram, trigram and unigram-bigram combinations.

*6) Model building and Selection:*

The final step in the text classification framework is to train a classifier using the features created in the previous step. The following different classifiers are implemented for the system:

*a) Naive Bayes Classifier:*

A Naive Bayes classifier is a probabilistic machine learning model that's used for classification tasks. Multinomial Naive Bayes algorithm will be used because it implements the naive Bayes algorithm for multinomial distributed data, and is one of the classic naive Bayes variants used in text classification [15].

*b) Linear Classifier:*

The linear classifier used for classification in this system is logistic regression. In this classifier one or multiple independent variables relationship is measured with categorical dependent variables by estimation of probabilities with the help of sigmoid or logistic function [16].

*c) Support Vector Machine (SVM):*

The support vector machine (SVM) algorithm is used to find a hyperplane in N-dimensional space (N — The number of

features) that distinctly classifies the data points, i.e. the relevant and irrelevant disaster news [17].

*d) Ensemble method [18]:*

*i. Bagging Models:*

Random Forest models are a type of ensemble models, particularly bagging models and this will be implemented.

*ii. Boosting Models:*

Boosting models are another type of ensemble model part of tree-based models. Xtereme Gradient Boosting Model will be used.

## IV. RESULTS

Evaluation metric plays a critical role for achieving an optimal classifier [19]. The evaluation metrics taken into consideration for obtaining the results are precision, recall, F1 score and confusion matrix. Precision answers the question: What proportion of predicted positives is truly positive. It is used when the prediction is assured, see Equation (1).

$$\text{Precision} = \frac{True\ Positive}{True\ Positives + False\ Positive} \quad (1)$$

Recall answers the question: What proportion of actual positives is correctly classified, see Equation (2).

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (2)$$

The F1 score is a number between 0 and 1 and is the harmonic mean of precision and recall, see Equation (3).

$$F1 = \frac{2\ x\ precision\ x\ recall}{precision + recall} \quad (3)$$

For this system, it should ensure that the news is disaster relevant (precision) and also want to capture as many disaster relevant news (recall) as possible. The f1 score manages this tradeoff. Finally, the model which has the highest recall and highest precision therefore giving a high F1 score is selected. Apart from that the model should also have a relatively better confusion matrix, that is it should have minimum false positives (a smaller number of actual disaster relevant news predicted as disaster irrelevant news).

For the dataset size of 11k, the highest values of precision, recall and F1 score is obtained to be 0.89 for the following 2 algorithms:

1. Logistic regression using bag of words at word level (unigram and bigram together)
2. SVM using bag of words at word level (unigram and bigram together)

But the final chosen algorithm is logistic regression because the false positives of logistic regression (FP=44) is less than the false positives of SVM (FP=64). The consolidated results are in table 2.

The final model is tested for its working on a dummy crisis management website as shown in Fig.3 and the relevant disaster news is displayed as shown in Fig.4.


Fig.3. Dummy crisis management website

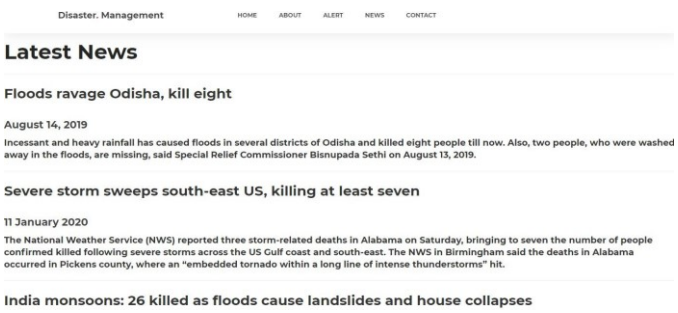


Fig.4. News displayed on dummy website

## V. CONCLUSION AND FUTURE SCOPE

Through this paper, the system is presented for enhancing the acquisition process of disaster data. In particular, the system automatically populates the disaster database by extracting information from trustable online news report sources. The primary research aim is to leverage both relevant data classification and fine-grained classification. The proposed system is entirely based on a machine learning approach and the architecture includes several steps from text extraction to model selection. The experimental results demonstrated the pertinence and potential of this solution, using a training set it was possible to achieve an accuracy of 0.89 and F1- measure of 0.88 in the detection of news about disasters using linear classifier logistic regression algorithm. The algorithm was implemented using the bag of words vectorization where unigram and bigram both were considered.

A future scope to this project would be to further categorize the disaster relevant news into the disaster types like flood, droughts, volcanoes, etc. for better categorized display on crisis management websites.

Table II. Results table

| | Model Used | Multinomial Naive Bayes | | | Logistic regression | | | SVM | | | Random forest (Bagging Model) | | | Xtereme Gradient Boosting Model (Boosting Model) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Evaluation Metrics | Precision | F1 Score | Recall | Precision | F1 Score | Recall | Precision | F1 Score | Recall | Precision | F1 Score | Recall | Precision | F1 Score | Recall |
| Vectorizaton Method | Level / n-gram used | | | | | | | | | | | | | | | |
| Bag of Words | Word level – Unigram | 0.86 | 0.86 | 0.86 | 0.88 | 0.88 | 0.89 | 0.87 | 0.87 | 0.87 | 0.88 | 0.87 | 0.88 | 0.88 | 0.87 | 0.88 |
| | Word level – Bigram | 0.79 | 0.68 | 0.65 | 0.85 | 0.82 | 0.84 | 0.86 | 0.84 | 0.85 | 0.85 | 0.83 | 0.85 | 0.84 | 0.79 | 0.82 |
| | Word level – Trigram | 0.73 | 0.4 | 0.41 | 0.83 | 0.77 | 0.81 | 0.83 | 0.78 | 0.82 | 0.83 | 0.77 | 0.82 | 0.82 | 0.72 | 0.79 |
| | Word level – Unigram and Bigram | 0.85 | 0.84 | 0.84 | 0.89 | 0.88 | 0.89 | 0.89 | 0.89 | 0.89 | 0.87 | 0.86 | 0.87 | 0.88 | 0.87 | 0.88 |
| | Character Level – Unigram | 0.72 | 0.72 | 0.75 | 0.76 | 0.73 | 0.78 | 0.58 | 0.66 | 0.76 | 0.81 | 0.77 | 0.81 | 0.77 | 0.77 | 0.79 |
| | Character Level – Bigram | 0.81 | 0.8 | 0.8 | 0.83 | 0.83 | 0.84 | 0.82 | 0.83 | 0.83 | 0.85 | 0.81 | 0.84 | 0.85 | 0.84 | 0.85 |
| | Character Level – Trigram | 0.86 | 0.85 | 0.85 | 0.87 | 0.87 | 0.87 | 0.85 | 0.85 | 0.85 | 0.87 | 0.85 | 0.86 | 0.88 | 0.87 | 0.88 |
| | Character Level – Unigram and Bigram | 0.85 | 0.84 | 0.83 | 0.86 | 0.86 | 0.86 | 0.85 | 0.85 | 0.85 | 0.87 | 0.84 | 0.86 | 0.87 | 0.87 | 0.88 |
| TF-IDF | Word level – Unigram | 0.87 | 0.83 | 0.85 | 0.87 | 0.85 | 0.87 | 0.89 | 0.89 | 0.89 | 0.88 | 0.87 | 0.88 | 0.87 | 0.86 | 0.87 |
| | Word level – Bigram | 0.85 | 0.78 | 0.82 | 0.83 | 0.73 | 0.8 | 0.85 | 0.82 | 0.84 | 0.85 | 0.82 | 0.85 | 0.83 | 0.78 | 0.82 |
| | Word level – Trigram | 0.83 | 0.75 | 0.8 | 0.82 | 0.7 | 0.78 | 0.83 | 0.77 | 0.81 | 0.83 | 0.77 | 0.82 | 0.82 | 0.72 | 0.79 |
| | Word level – Unigram and Bigram | 0.84 | 0.77 | 0.82 | 0.82 | 0.71 | 0.79 | 0.84 | 0.79 | 0.83 | 0.84 | 0.81 | 0.84 | 0.83 | 0.78 | 0.82 |
| | Character Level – Unigram | 0.82 | 0.67 | 0.77 | 0.75 | 0.71 | 0.77 | 0.58 | 0.66 | 0.76 | 0.81 | 0.77 | 0.81 | 0.77 | 0.77 | 0.79 |
| | Character Level – Bigram | 0.78 | 0.71 | 0.78 | 0.83 | 0.82 | 0.84 | 0.84 | 0.83 | 0.84 | 0.84 | 0.8 | 0.83 | 0.84 | 0.84 | 0.85 |
| | Character Level – Trigram | 0.83 | 0.78 | 0.81 | 0.87 | 0.86 | 0.87 | 0.88 | 0.87 | 0.88 | 0.87 | 0.85 | 0.87 | 0.86 | 0.86 | 0.87 |
| | Character Level – Unigram and Bigram | 0.82 | 0.74 | 0.8 | 0.87 | 0.86 | 0.87 | 0.88 | 0.87 | 0.88 | 0.87 | 0.84 | 0.86 | 0.87 | 0.86 | 0.87 |

## REFERENCES

[1] M. I. Rana, S. Khalid, and M. U. Akbar. "News classification based on their headlines: A review". 17th IEEE International Multi Topic Conference, Karachi, pp. 211-216. IEEE, 2014

[2] I. Dilrukshi, K. De Zoysa, and A. Caldera. "Twitter news classification using SVM". 8th International Conference on Computer Science & Education, Colombo, pp. 287-291. IEEE, 2013

[3] CHAN, Chee-Hong, SUN, Aixin, LIM, and Ee Peng. "Automated online news classification with personalization". 4th International Conference on Asian Digital Libraries (ICADL), Research Collection School Of Information Systems. INK, 2001

[4] A. N. Chy, M. H. Seddiqui, and S. Das. "Bangla news classification using naive Bayes classifier". 16th Int'l Conf. Computer and Information Technology, Khulna, pp. 366-371. IEEE, 2014

[5] Dewi Y. Liliana, Agung Hardianto, and M. Ridok. "Indonesian News Classification using Support Vector Machine". World Academy of Science, Engineering and Technology 57. WASET, 2011

[6] Beverly Estephany Parilla-Ferrer, Proceso L. Fernandez Jr., PhD, and Jaime T. Ballena IV, PhD. "Automatic Classification of Disaster-Related Tweets". International conference on Innovative Engineering Technologies (ICIET), Bangkok, Thailand. IIE, 2014

[7] Koustav Rudra, Subham Ghosh, Niloy Ganguly, Pawan Goyal, and Saptarshi Ghosh. "Extracting Situational Information from Microblogs during Disaster Events: a Classification Summarization Approach". *CIKM'15*, Melbourne, Australia. ACM, 2015

[8] Khaleq, Abeer & Ra, and Ilkyeun. "Twitter Analytics for Disaster Relevance and Disaster Phase Discovery: Volume 1". Proceedings of the Future Technologies Conference (FCT), pp.401-417. Springer, 2018

[9] Téllez-Valero, Alberto & Montes, Manuel & Villaseñor-Pineda, and Luis. "Using Machine Learning for Extracting Information from Natural Disaster News Reports". Language Technologies Laboratory, Coordination of Computational Sciences, National Institute of AstroPhysics, Optics and Electronics (INAOE), Puebla, Mexico. CYS, 2009

[10] Kevin Stowe, Michael Paul, Martha Palmer, Leysia Palen, and Ken Anderson. "Identifying and Categorizing Disaster Related Tweets". Proceedings of the Fourth International Workshop on Natural Language Processing for Social Media, Austin. ACL, 2016

[11] Leidner, Jochen L. & Nugent, Timothy & Petroni, Fabio & Raman, Natraj & Carstens, and Lucas. "A Comparison of Classification Models for Natural Disaster and Critical Event Detection from News". IEEE International Conference on Big Data, Boston, USA. IEEE, 2017

[12] U. Suleymanov, S. Rustamov, M. Zulfugarov, O. Orujov, N. Musayev, and A. Alizade. "Empirical Study of Online News Classification Using Machine Learning Approaches". IEEE 12th International Conference on Application of Information and Communication Technologies (AICT), Almaty, Kazakhstan, pp. 1-6. IEEE, 2018

[13] Mohd Rizvi. https://www.analyticsvidhya.com/blog/2017/07/web-scraping-in-python-using-scrapy/

[14] Anita Kumari Singh and Mogalla Shashi. "Vectorization of Text Documents for Identifying Unifiable News Articles". International Journal of Advanced Computer Science and Applications (IJACSA), 2019. http://dx.doi.org/10.14569/IJACSA.2019.0100742

[15] Supervised learning. scikit-learn 0.22.2 documentation. https://scikit-learn.org/stable/supervised_learning.html#supervised-learning

[16] Real world implementation of Logistic Regression. https://towardsdatascience.com/real-world-implementation-of-logistic-regression-5136cefb8125

[17] A friendly introduction to Support Vector Machines (SVM). https://towardsdatascience.com/a-friendly-introduction-to-support-vector-machines-svm-925b68c5a079

[18] Ensemble methods. scikit-learn 0.22.2 documentation. https://scikit-learn.org/stable/modules/ensemble.html#ensemble-methods

[19] Hossin, Mohammad & M.N, Sulaiman. Review on Evaluation Metrics for Data Classification Evaluations. International Journal of Data Mining & Knowledge Management Process. 5. 01-11. 10.5121/ijdkp.2015.5201.