Proc. of the 3rd International Conference on Electrical, Communication and Computer Engineering (ICECCE)
12-13 June 2021, Kuala Lumpur, Malaysia

Multi-Class Text Classification: Model Comparison and Selection

1st Waqas Arshad
Faculty of Information Technlogy
University of Cental Punjab
Lahore, Pakistan
waqas.arshad@ucp.edu.pk

2nd Muhammad Ali
Dept. of Computer Science
The Superior University
Lahore, Pakistan
muhammedbwn@gmail.com

3rd Muhammad Mumtaz Ali
School of Information Eng.,
Zhengzhou University,
Henan, China
mumtazalirajput@gmail.com

4th Arifa Javed
School of Information Eng.,
Zhengzhou University,
Henan, China
arifa.javed1521@gmail.com

5th Saddam Hussain
Dept. of Computer Science,
The University of Lahore
Lahore, Pakistan
saddam.hussain@cs.uol.edu.pk

2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE) | 978-1-6654-3897-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICECCE52056.2021.9514108

Abstract—**The objective of text classification is to categorize documents into a specific number of predefined categories. We can easily imagine the issue of arranging documents, not by topic, but rather by and large assessment, e.g. deciding if the sentiment of a document is whether positive or negative. While working on a supervised machine learning problem with a defined dataset, there are many classifiers that can be used in text classification. Utilizing dataset of stack overflow questions, answers, and tags as information, we find that standard machine learning systems completely beat human-delivered baselines. These majorly include Naive Bayes Classifier for multinomial models, Linear Support Vector Machine, Logistic Regression, Word to vector (Word2vec) and Logistic Regression, Document to vector (Doc2vc) and logistic regression, Bag of Words (BOW) with Keras. Our paper is a detailed examination and comparison of accuracies among these algorithms.**

Keywords—Text classification, stack overflow, Machine learning, supervised machine learning, classifiers, Linear Support Vector Machine, Logistic Regression, Word2vec and Logistic Regression, Doc2vc and logistic regression, BOW with Keras.

I. Introduction

Text mining (TM) is gaining more and more importance as the number of electronic documents is increasing, which includes both structured and semi-structured information. Text classification frameworks are combined with some data processing frameworks in order to effectively categorize a document from multiple classes. Currently, common text classification methods are Naive Bayes Classifier for Multinomial Models, Linear Support Vector Machine, Logistic Regression, Word2vec, Logistic Regression, Doc2vc and logistic regression, BOW with Keras. These supervised classification machine learning algorithms are recognized as simple and effective methods for text classification. Our work centers on the comparison of these classification algorithms and the response to the question, "What machine learning model should one use for multi-class text classification?" We have done a thorough comparison among text classification models in order to choose the most appropriate one for a particular problem.

II. Literature Review

Because of the quick advancement of huge information and cloud computing, we could discover a large amount of online information accessible for processing in outgoing years. This led to a massive interest in the classification of text. In this segment, we will complete a survey of these studies and identify the prime objectives of them. Ye Wang et.al [23] compared the performance and robustness of four techniques: one-hot encoding, tf-idf weighting, word2vec and paragraph2vec in feature selection. Whereas, in the classification part, they picked and compare Naive Bayes, Logistic Regression, Support Vector. Machine, K-closest Neighbor and Decision Tree as classifiers after that compare and analysed classifiers. [03] used Decision Tree (DT), Support Vector Machine (SVM), K Nearest Neighbors (KNN), Naïve Bayes (NB) and hidden Markov model (HMM) as classification methods and compare them according to their modifications. [14] pointed towards text classification and sentiment analysis which helps individuals/peoples of different fields to judge the sentiment of the user.

[07] depicted the comparison of three different classifiers including K-Nearest Neighbors, Naive Bayes, and Support Vector Machine. [22] performed analysis of different techniques used in text classification and explained the strong and weak points of those techniques to improve awareness in the field of data mining. [18] gave an introduction of text classification and overview of text classifiers and performed the comparison on these classifiers on the basis of performance, complexity, and principal.

[04] described various and most important text mining techniques which include text pre-processing, classification and clustering. They also briefly explained the application of text mining in bioinformatics and health. [09] compared the information bottleneck method with SVM based text categorization using simple-minded Bag of Words (BoW) representation. [17] illustrated the text classification process using machine learning and statistical techniques such as SVM, K Nearest Neighbor and Naive Bayes method. [25] introduced the framework of machine learning on big data to guide others about its opportunities and challenges. Their framework includes pre-processing, learning and evaluation. [1,2,5,6,8,11,12,13,16,19,20,21,24]'s work is also appreciated.

III. DATA COLLECTION

The Stack overflow (Question & tags) is a pretty enormous data set. As of August 2018, it consists of 9 million users that exceeded 16 million questions by mid-2019. The tags include JavaScript, Java, C#, PHP, Android, Python, jQuery and HTML. We have taken data from Google BigQuery a web service. It allows doing interactive analysis of a massive dataset that fits for billions of commotions. It has over 10 million words with 20 total tags and 2000's post on each tag, (figure 1). As we can see, classes are well composed but in scattered form with a lot of unnecessary detail. That is why we have to look a few posts and tag pairs. Our area of concern in any question or search is the most relevant tagged keyword so that text classification will provide the most expected outcomes. Remember the more steps we add in text cleaning the longer time it will take and become more complex. Steps include to clean specifically defined dataset are Conversion of text into lower case, expel stop words, remove punctuation, remove bad characters and so on. After expelling stop words and texting we have 3million words to work with.
After dividing the dataset, raw data is changed into feature vectors thus the new features are created by using the existing dataset. The features are extracted by Count vectorizer and Tf-idf Transformer. In count vector-matrix notation of a dataset is done in a way that every row represents a word from the corpus, and each column represents a frequency count of a specific term in a specific document. Then they change a normalized matrix to tf-idf representation. Tf-idf is basically used in machine learning and text mining as a weighting factor for features. As the weight increases the word frequency in the document also increases. which means the more time this term occurs in the document but that's offset by the number of times word appear in the entire dataset and corpus this offset helps in removing really common words like the or a that appear frequently in all documents.

Its score is formed by combination of two terms TF and IDF.

TF(t) = Number of times term show supine record / Total number of terms in the archive.

IDF(t)=$\log_e$(Total number of reports / Number of archives with term init).

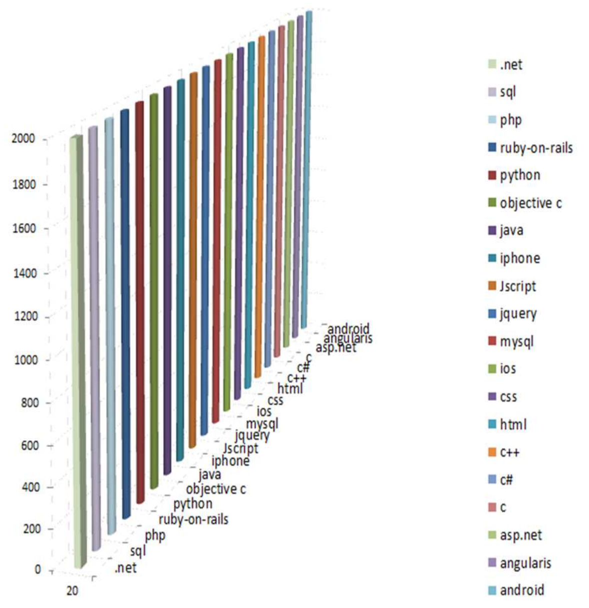

Figure 1. Posts and Tag Groups

IV. EVALUATION MEASURES

A. Accuracy

The accuracy is the ratio of correctly classified instances to the total number of instances in the data set.

B. Precision

The precision is the ratio of true positive / (true positive + false positive). It's innately the aptitude of the classifier not to label as positive a sample that is negative

C. Recall

The recall is the ratio of true positive / (true positive + false negative. It's innately the aptitude of the classifier to discover all the positive samples.

D. F1-Score

The F-1 score taken as a weighted harmonic mean of the precision and recall, where it reaches the finest value at 1 and poorest at 0.

V. CLASSIFIERS

A. Naive Bayes Classifier (Multinomial Models)

When it comes to text categorization, Naïve Bayes classifier is most appropriate to deal with discreet features. Naïve Bayes classifier belongs to the family of probabilistic classifiers and uses Bayes theorem to predict an unknown class. It has high accuracy rates and efficient computational speed on large data sets. In multinomial models' integers features count are required normally.

TABLE I. RESULTS OF NAÏVE BAYS CLASSIFIER

Classes	Precision	Recall	F1-Score
.net	0.73	0.65	0.7
Sql	0.65	0.63	0.65
rub-on-rails	0.72	0.86	0.79
Python	0.69	0.49	0.59
Php	0.65	0.88	0.78
objective-c	0.78	0.76	0.78
Mysql	0.76	0.73	0.75
Jquery	0.71	0.5	0.6
Javascript	0.56	0.64	0.61
Java	0.62	0.64	0.64
Iphone	0.63	0.82	0.72
Ios	0.62	0.58	0.61
CSS	0.83	0.58	0.69
Html	0.93	0.85	0.9
C++	0.88	0.82	0.86
C#	0.69	0.76	0.73
C	0.78	0.78	0.79
Asp.Net	0.86	0.91	0.9
Angularjs	0.93	0.88	0.91
Android	0.65	0.83	0.74
Avg	0.73	0.73	0.74
Accuracy = 0.7295			

B. Linear Support Vector Machine

SVM is known for its high accuracy as compared to other classifiers (eg. decision tree). Linear SVM is mostly appropriate for multi-class text classification and preferable to be used on large datasets. It performs classification and separates data classes by drawing a hyperplane with the largest margin in multidimensional space. We used scikit-learn (Pipeline, SGDC classifier, Tf-IDF Transformer). SDGC can act as a multi classifier. If loss = hinge, it behaves as Linear SVM otherwise If loss=log it will act as logistic regression. In SGD Classifier, we set parameters as (loss='hinge', Penalty='l2', Alpha=1e-3). The loss function is used defaults to `hinge', which gives a linear SVM. The penalty is the standard regularization for linear SVM models.

TABLE II. RESULTS OF SVM

Classes	Precision	Recall	F1-Score
.net	0.76	0.88	0.83
Sql	0.70	0.67	0.69
rub-on-rails	0.73	0.87	0.79
Python	0.70	0.64	0.68
Php	0.69	0.94	0.81
objective-c	0.80	0.89	0.85
Mysql	0.81	0.67	0.74
Jquery	0.76	0.40	0.53
Javascript	0.71	0.58	0.65
Java	0.73	0.67	0.71
Iphone	0.80	0.79	0.81
Ios	0.81	0.55	0.66
CSS	0.76	0.78	0.78
Html	0.84	0.92	0.89
C++	0.83	0.95	0.89
C#	0.80	0.79	0.80
C	0.80	0.86	0.84
Asp.Net	0.86	0.94	0.91
Angularjs	0.86	0.94	0.91
Android	0.83	0.85	0.85
Avg/Total	0.78	0.78	0.78
Accuracy = 0.77916			

C. Logistic Regression

Logistic regression also belongs to the family of probability classifiers and known as the simplest classification algorithm. It is appropriate where the dependent variable is binary 1(True, success) 0(false, failure). Logistic regression deals with probability to measure the relation between dependent and independent material. Word embedding's are very useful as they try to preserve syntactical information. The previous methods we use for feature extraction from the text were count vectorizer and tf-idf they only count words but do not save any syntactical information.

TABLE III. RESULTS OF LOGISTIC REGRESSION

Classes	Precision	Recall	F1-Score
.net	0.77	0.82	0.8
Sql	0.64	0.64	0.65
rub-on-rails	0.76	0.8	0.79
Python	0.63	0.62	0.64
Php	0.81	0.85	0.84
objective-c	0.81	0.83	0.83
Mysql	0.76	0.75	0.76
Jquery	0.58	0.57	0.58
Javascript	0.6	0.58	0.6
Java	0.69	0.61	0.66
Iphone	0.79	0.82	0.81
Ios	0.69	0.71	0.71
CSS	0.77	0.77	0.78
Html	0.9	0.9	0.91
C++	0.9	0.9	0.91
C#	0.77	0.76	0.78
C	0.82	0.82	0.83
Asp.Net	0.96	0.93	0.94
Angularjs	0.95	0.93	0.95
Android	0.84	0.84	0.85
Avg/Total	0.73	0.77	0.75
Accuracy = 0.7710			

D. Word2Vec

Word2vec is a word embedding model made by Google. In word2vec every word has a vector. Initially, it assigns a random vector or one-hot vector (representation where only one bit in the vector is 1). It maps the word with similar meaning to the same vector representation. Its architecture based on algorithms which are a continuous bag of words and skip gram.

TABLE IV. RESULTS OF WORD2VECT

Classes	Precision	Recall	F1-Score
.net	0.65	0.7	0.68
Sql	0.41	0.41	0.42
rub-on-rails	0.69	0.76	0.73

Python	0.54	0.49	0.52
Php	0.72	0.79	0.76
objective-c	0.67	0.7	0.7
Mysql	0.64	0.6	0.63
Jquery	0.43	0.38	0.41
Javascript	0.55	0.51	0.54
Java	0.62	0.58	0.61
Iphone	0.69	0.7	0.71
Ios	0.59	0.6	0.61
CSS	0.64	0.64	0.65
Html	0.72	0.75	0.75
C++	0.75	0.77	0.77
C#	0.52	0.51	0.52
C	0.6	0.6	0.61
Asp.Net	0.64	0.66	0.66
Angularjs	0.81	0.82	0.82
Android	0.59	0.56	0.59
Avg/Total	0.62	0.63	0.63
Accuracy = 0.62680			

E. Logistic Doc2Vc

Doc2vc extends the genism word2vc class. Size of the sliding window, dimension of representation, etc. are like or almost any parameters which can change with word2vc model are easily adjustable. The difference occurs in the parameter when it relates to the training method used by models. Doc2vc architecture based on algorithms is distributed memory (DM) and distributed bag of word (DBoW).

TABLE V. RESULTS OF DOC2VC

Classes	Precision	Recall	F1-Score
.net	0.8	0.83	0.82
Sql	0.67	0.64	0.66
rub-on-rails	0.82	0.89	0.86
python	0.62	0.64	0.64
php	0.8	0.83	0.82
objective-c	0.83	0.85	0.85
Mysql	0.86	0.8	0.83
jquery	0.71	0.7	0.72
Javascript	0.68	0.62	0.66
Java	0.72	0.67	0.7
Iphone	0.83	0.81	0.83
Ios	0.67	0.66	0.67
CSS	0.8	0.76	0.79
Html	0.88	0.9	0.9
C++	0.89	0.94	0.92
C#	0.79	0.79	0.8
C	0.8	0.82	0.82
Asp.Net	0.92	0.93	0.94
Angularjs	0.92	0.95	0.95
Android	0.83	0.84	0.84
Avg/Total	**0.79**	**0.79**	**0.79**
Accuracy = 0.7954			

F. Bow with Keras

Keras is a deep learning neural network API in python which provides a clean and convenient path to run on top of Theano and Tensor-flow. It was specially developed to implement deep learning models. It developed on four basic building blocks (modularity, minimalism, extensibility, and python).

TABLE VI. RESULTS OF BOW

Classes	Precision	Recall	F1-Score
.net	0.76	0.88	0.83
Sql	0.70	0.67	0.69
rub-on-rails	0.82	0.89	0.86
python	0.62	0.64	0.64
Php	0.69	0.94	0.81
objective-c	0.80	0.89	0.85
Mysql	0.86	0.8	0.83
jquery	0.71	0.7	0.72
Javascript	0.71	0.58	0.65
Java	0.73	0.67	0.71
Iphone	0.83	0.81	0.83
Ios	0.67	0.66	0.67
CSS	0.76	0.78	0.78
Html	0.84	0.92	0.89
C++	0.89	0.94	0.92
C#	0.79	0.79	0.8
C	0.80	0.86	0.84
Asp.Net	0.86	0.94	0.91
Angularjs	0.92	0.95	0.95
Android	0.83	0.84	0.84
Avg/Total	0.78	0.78	0.78
Accuracy = 0.78916			

VI. CONCLUSION

The results show machine learning models performs much better as compared to human-generated baselines. We have tried six different techniques, in which Doc2Vc performed best with the accuracy of 80% and word2Vect has the least accuracy of 63%. Bow-with-Keras is also significant with 79% accuracy, only 1% less than Doc2Vc's accuracy. Figure. 2 shows the accuracies comparison of above-mentioned classifiers. Doc2Vc approach has also highest precision, recall and f-measure values which clearly outstands it with the other approaches for Stack-overflow document classification.

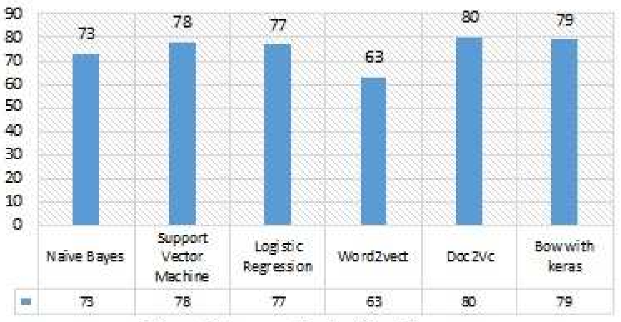

Figure 2. Accuracy levels of classifers

References

[1] C. C. Aggarwal, S. C. Gates and P. S. Yu, "On using partial supervision for text categorization," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 2, pp. 245-255, Feb. 2004. doi: 10.1109/TKDE.2004.1269601.

[2] Aggarwal C.C., Zhai C. (2012) A Survey of Text Classification Algorithms. In: Aggarwal C., Zhai C. (eds) Mining Text Data. Springer, Boston, MA. doi.org/10.1007/978-1-4614-3223-4_6.

[3] Aliwy, A. H., & Ameer, E. H. A. (2017). Comparative study of five text classification algorithms with their improvements. International Journal of Applied Engineering Research, 12(14), 4309-4319

[4] Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., & Kochut, K. (2017). A brief survey of text mining: Classification, clustering and extraction techniques. arXiv preprint arXiv:1707.02919

[5] Androutsopoulos, I., Koutsias, J., Chandrinos, K. V., Paliouras, G., & Spyropoulos,C. D. (2000). An evaluation of naive bayesian anti-spam filtering. arXiv preprintcs/0006013.

[6] Angelova, R., & Weikum, G. (2006). Graph-based text classification: learn from your neighbors. In Proceedings of the 29th annual international acm sigir conference on research and development in information retrieval (pp. 485-492).

[7] Gandhi, V. C., & Prajapati, J. A. (2012). Review on comparison between text classification algorithms. *International Journal of Emerging Trends & Technology in Computer Science*, *1*(3).

[8] L. Douglas Baker and Andrew Kachites McCallum. 1998. Distributional clustering of words for text classification. In Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '98). ACM, New York, NY, USA, 96-103. DOI: https://doi.org/10.1145/290941.290970

[9] Bekkerman, Ron & El-Yaniv, Ran & Tishby, Naftali & Winter, Yoad. (2003). Distributional Word Clusters vs. Words for Text Categorization. Journal of Machine Learning Research. 3. 1183-1208. 10.1162/153244303322753625.

[10] Berkhin, P., & Becher, J. D. (2002). Learning simple relations: Theory and applications. In Proceedings of the 2002 Siam International Conference On Data Mining (pp. 420-436), doi: 10.1137/1.9781611972726.25

[11] Jingnian Chen, Houkuan Huang, Shengfeng Tian, and Youli Qu. 2009. Feature selection for text classification with Naïve Bayes. Expert Syst. Appl. 36, 3 (April 2009), 5432-5435. DOI: https://doi.org/10.1016/j.eswa.2008.06.054

[12] Inderjit S. Dhillon, Subramanyam Mallela, and Rahul Kumar. 2002. Enhanced word clustering for hierarchical text classification. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '02). ACM, New York, NY, USA, 191-200. DOI=http://dx.doi.org/10.1145/775047.775076.

[13] Inderjit S. Dhillon, Subramanyam Mallela, and Rahul Kumar. 2003. A divisive information theoretic feature clustering algorithm for text classification. J. Mach. Learn. Res. 3 (March 2003), 1265-1287.

[14] Gajjala, Abhiteja, "Multi-Faceted Text Classification using Supervised Machine Learning Models" (2016). *Master's Projects*. 482. DOI: https://doi.org/10.31979/etd.7crd-u5pw.

[15] Anil K. Jain, Data clustering: 50 years beyond K-means, Pattern Recognition Letters, Volume 31, Issue 8, 2010, Pages 651-666, ISSN 0167-8655, https://doi.org/10.1016/j.patrec.2009.09.011

[16] J. Jiang, R. Liou and S. Lee, "A Fuzzy Self-Constructing Feature Clustering Algorithm for Text Classification," in IEEE Transactions on Knowledge and Data Engineering, vol. 23, no. 3, pp. 335-349, March 2011. doi: 10.1109/TKDE.2010.122

[17] Thorsten Joachims. 2001. A statistical learning learning model of text classification for support vector machines. In Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '01). ACM, New York, NY, USA, 128-136. DOI: https://doi.org/10.1145/383952.383974

[18] Korde, V., & Mahender, C. N. (2012). Text classification and classifiers: A survey. International Journal of Artificial Intelligence & Applications, 3(2), 85.

[19] Huan Liu and Lei Yu, "Toward integrating feature selection algorithms for classification and clustering," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 491-502, April 2005. doi: 10.1109/TKDE.2005.66

[20] Bo Pang, Lillian Lee, Shivakumar Vaithyanathan (2002). Thumbs up?: sentiment classification using machine learning techniques. In Proceedings Of The Acl-02 Conference On Empirical Methods In Natural Language Processing-Volume 10 (pp. 79-86). doi: 10.3115/1118693.1118704

[21] Fabrizio Sebastiani. 2002. Machine learning in automated text categorization. ACM Comput. Surv. 34, 1 (March 2002), 1-47. DOI=http://dx.doi.org/10.1145/505282.505283

[22] Thangaraj, M., & Sivakami, M. (2018). Text classification techniques: A literature review. Interdisciplinary Journal of Information, Knowledge & Management, doi.org/10.28945/4066.

[23] Wang, Y., Zhou, Z., Jin, S., Liu, D., & Lu, M. (2017). Comparisons and selections of features and classifiers for short text classification. In Iop conference series: Materials science and engineering (Vol. 261, p. 012018).

[24] Zhang, Wen Zhang, Taketoshi Yoshida, Xijin Tang, A comparative study of TF*IDF, LSI and multi-words for text classification, Expert Systems with Applications, Volume 38, Issue 3, 2011, Pages 2758-2765, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2010.08.066.

[25] Lina Zhou, Shimei Pan, Jianwu Wang, Athanasios V. Vasilakos, Machine learning on big data: Opportunities and challenges, Neurocomputing, Volume 237,2017, Pages 350-361, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2017.01.026.

[26] URL: https://storage.googleapis.com/tensorflow-orkshopexamples/stack-overflow-data.csv