International Journal of Scientific Research in Science and Technology
Print ISSN: 2395-6011 | Online ISSN: 2395-602X (www.ijsrst.com)
doi : https://doi.org/10.32628/IJSRST52310240

Autonomous Tagging of Stack Overflow Questions

B. Ravinder Reddy[1], K. Bhadri Prabhav[2], P. Hemanth Sagar[2], P. Rahul[2]
[1]Assistant Professor, Department of CSE, Anurag University, Hyderabad, T.S., India
[2]Department of CSE, Anurag University, Hyderabad, T.S., India

ARTICLEINFO

Article History:

Accepted: 05 March 2023
Published: 22 March 2023

Publication Issue
Volume 10, Issue 2
March-April-2023

Page Number
263-269

ABSTRACT

Educational resources like question-and-answer websites like Stack Exchange and Quora are growing in popularity online. A large number of these gatherings depend on labeling, which includes a part marking a post with a suitable assortment of subjects that depict the post and make it more straightforward to find and sort. We give a multi-name order framework that naturally distinguishes clients' requests to upgrade the client experience. A straight SVM and a carefully selected portion of the researched highlight set are used to create a one-versus-rest classifier for a Stack Overflow dataset. By utilizing a subsample of the initial data that is restricted to 100 labels and at least 500 events of each label throughout the data, our characterization framework achieves an ideal F1 score of 62.35 percent.

Keywords – Tagging autonomously, stack overflow.

I. INTRODUCTION

As a way to learn, question-and-answer forums have become increasingly popular since online education became available. Stack Trade, Quora, and Massive Open Online Courses (MOOCs) like Coursera and OpenEdX are examples of different models. Despite the growing amount of content available on these forums, there is currently no automated method for actually assembling and ordering the information so that it can be presented to customers in a reasonable manner. A forum query's subject could be automatically deduced and tagged. The client experience on internet based gatherings can be further developed by a framework that naturally decides the subject of an inquiry by: 1) gathering inquiries concerning comparative subjects for clients to peruse; and 2) showing clients' presents that are connected on an inquiry they are entering, since their inquiry may as of now have been replied on the discussion. To make it more straightforward to bunch comparable posts, a few discussions, as Quora, expect individuals to unequivocally submit labels connected with their inquiries. Then again, physically naming a post is a problem for clients and damages the client experience in general. An innovation that can naturally construe post labels is what we recommend. We give a multi-mark order framework that relegates discussion themes labels naturally to this end. Our

classifier is constructed and tested on a number of Stack Overflow queries.

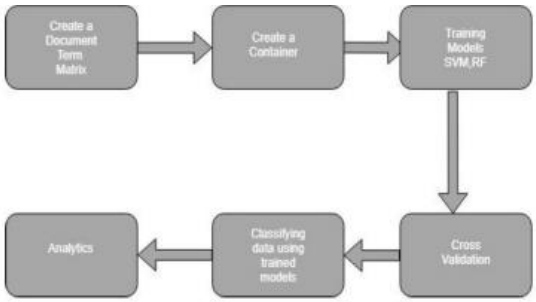


Fig.1: Example figure

Information sharing platforms have become increasingly popular for Q&A sessions. OpenEDX, Quora, StackOverflow, and Reddit are a few examples. Despite the fact that the amount of information available on these websites has increased multiple times, no automated method exists to classify the data. The majority of these websites force users to tag their questions, making it difficult to ask questions. Data can be ambiguous because users might not correctly classify the problem. As an approach to proficiently order data, mechanizing the labeling system would be useful. A framework that permits independent labeling might further develop the client experience by sorting out data into discrete normal subjects. Another benefit is that the client can be given inquiries that are pertinent to his concern, which might assist him with finding an answer all the more rapidly and successfully. A framework for responsive stages that consequently doles out questions labels is depicted in the article. To naturally relegate labels to requests made in any discussion, we propose a grouping methodology in light of a Document-Term matrix [16]. StackOverflow queries were used to develop and validate the classifier.

II. LITERATURE REVIEW

Text categorization with support vector machines: Learning with many relevant features:

[1] T. Joachims et al., Focus on how Support Vector Machines (SVMs) can be used to extract text classifiers from models in this review. It reviews the focal points of learning with text data and gets a handle on why SVMs are reasonable for this task. Experimental data supports hypothetical revelations. SVMs consistently perform well across a wide range of learning tasks and outperform the most effective algorithms currently available. They are additionally completely computerized, which wipes out the requirement for human boundary tweaking.

Web document classification by keywords using random forests:

[3] As indicated by Klassen, M., and Paturi, N.Web et al., Serving client search requests relies heavily on file structure. Accurate classification of online pages is required in order to develop and maintain such directories without the assistance of human specialists. In this study, they look into random forest learning algorithms and page categorization using keywords from documents as characteristics. The underlying results demonstrate that the random forests learning method outperformed other well-known approaches to learning. In spite of the way that the altogether portrayal rates declined as the amount of subjects moved from five to seven, random forests really beat elective strategies.

Multi-label text classification with a mixture model trained by em:

[6] McCallum, A. K. et al. express that, Every perception in the preparation and test sets has a remarkable class name, so ordinary ways to deal with design acknowledgment issues commonly center just around the unilabel order issue. However, considering that a single sample may be assigned a number of classes in many real-world jobs, approaches to the more general multi-label problem must be investigated. They explain how the multinomial (Naive Bayes) classifier used in this study can be used to categorize text by looking at the methods described in our previous work. The results are displayed on the

Reuters-21578 dataset, and our suggested method yields satisfactory outcomes.

Efficient estimation of word representations in vector space:

[7] K. Chen, T. Mikolov, G. Corrado, and J. Dean, among others for computing consistent vector portrayals of words from incredibly enormous informational indexes, proposed two particular model designs. In a word similitude challenge, the nature of these portrayals is checked out, and the outcomes are contrasted with the best calculations from an earlier time that depended on various types of neural networks. Advancing top notch word vectors from a 1.6 billion word informational collection takes under a day, and they see critical additions in precision at a fundamentally diminished computational expense. On their test set, they also show that these vectors perform exceptionally well when evaluating syntactic and semantic word similarities.

Glove: Global vectors for word representation:

[9] Pennington, J., R. Socher, and C. D. Monitoring looked at the current methods for learning how words are represented in vector space. Although the origin of these normalities has remained a mystery, these methods have been successful in using vector math to detect fine-grained semantic and syntactic consistency. They examine and explain the model qualities required for word vector normalities like these to happen. As an immediate consequence of this, a shiny new worldwide logbilinear relapse model is made that consolidates the upsides of two critical model families that have been reported: methods like global matrix factorization and nearby setting window Their model successfully uses measurable data via preparing just on the nonzero parts of a word cooccurrence lattice rather than the whole meager framework or explicit setting windows in an enormous corpus. The model produces a vector space with critical base, as exhibited by its 75% execution on a new word relationship task. It additionally performs better compared to named element acknowledgment models and likeness challenges.

III. METHODOLOGY

To make it simpler to bunch comparable posts, a few discussions, as Quora, expect individuals to unequivocally submit labels connected with their inquiries. Then again, physically marking a post is a problem for clients and damages the client experience in general.

Disadvantages:

1. clients' weight
2. reduces overall user satisfaction

An innovation that can naturally gather post labels is what we propose. We give a multi-mark order framework that doles out gathering points labels consequently to this end. Our classifier is constructed and tested on a number of Stack Overflow queries.

Advantages :

1. We had enough diverse training data to learn statistics for the appropriate tags because of this.

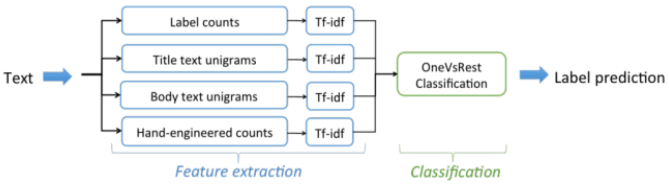


Fig.2: System architecture

MODULES:

We developed the modules listed below in order to carry out the aforementioned project.

- Exploration of data: This module will be used to enter data into the system.
- Processing: Using this module, we will read data for processing.
- Splitting the data into train and test: Data will be divided into train and test with this module.

- Creation of models: determining the accuracy of the model and algorithms
- User registration and login: Those who use this module can register and log in.
- User input: The use of this module will provide prediction input.
- The final prediction was shown.

IV. IMPLEMENTATION

Ngram features and label counts make up the vast majority of our feature collection. How we select and fine-tune pipeline feature extractors and classifiers is the subject of this section. Starting with a list of features related to the number of times an imprint appears in the title and body of the inquiry (referred to as "name counts") provides the highest level of precision among the free options we evaluated. We start with mark builds up to construct our whole list of capabilities, then, at that point, add and voraciously change include extractors each in turn (by three-crease cross approval). To identify states that are associated with particular labels, we include body and title ngrams. Tuning hyperparameters for ngrams and mark counts requires selecting binarization settings (Bernoulli versus Multinomial counts), IDF, and standard settings for the subsequent TFIDF transformer. Because it matters a lot to each mark, the count of each expression is reweighted using TFIDF [5, or Term Frequency Inverse Document Frequency]. Binarization and counter cut-off value selection are also necessary for Unigram tuning. We find that bigram features limit our ability to use a larger dataset and have no effect on performance. Since a client is bound to sum up and classify the request in the title, we likewise found that expressions in the title are more demonstrative of marks. By taking into account unmistakable expression weightings, we find that extricating ngrams for the body and the title freely further develops execution. Separating raw ngrams for the inquiries' code part, if any exist, dials back because of huge substance contrasts. Consequently, we only extract ngrams from the text portions of each question and separate the text and code components.

SVM: The Support Vector Machine, which is also a managed learning model, was used as another classifier. In addition, it examines the results of grouping and relapse examinations. The qualities in our preparation information can be categorized as one of two classifications. From the preparation information, a SVM preparing technique makes a model that is utilized to characterize test information into one of two classes. The portion, piece boundaries, and edge boundary C all assume a part in deciding SVM's presentation, and it likewise forestalls overfitting. SVM models support parts, so we might in fact show connections that are not direct. It also lasts longer because it maximizes margin. Linear classifiers known as Support Vector Machines offer theoretical assurances of high prediction accuracy to classifiers.

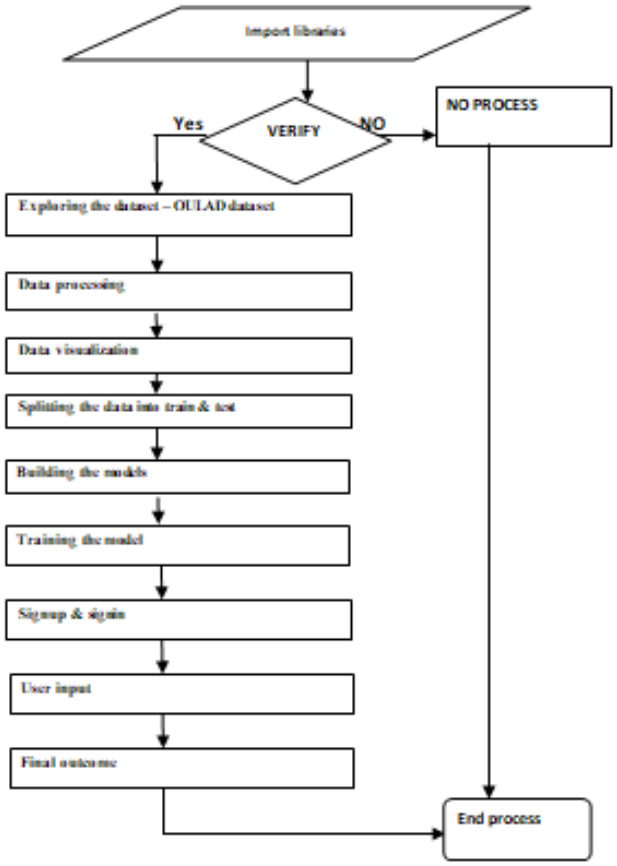


Fig.3: Dataflow diagram

V. EXPERIMENTAL RESULTS

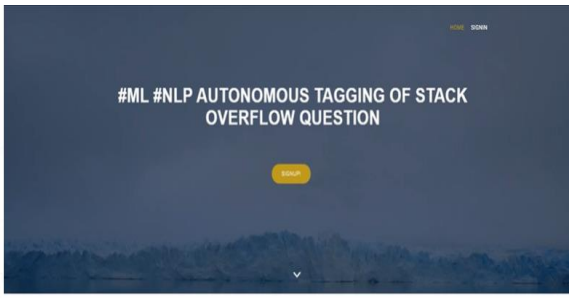


Fig.4: Home screen

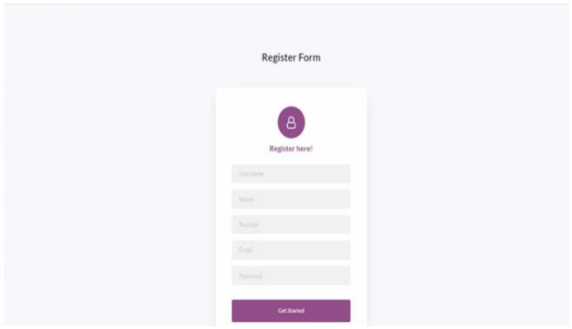


Fig.5: Register screen

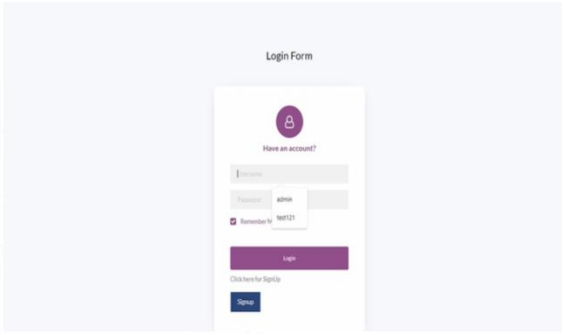


Fig.6: Login screen

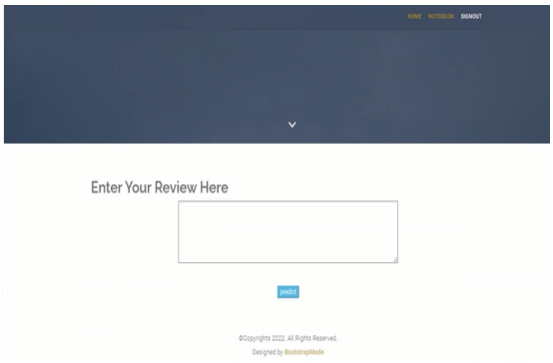


Fig.7: Main page

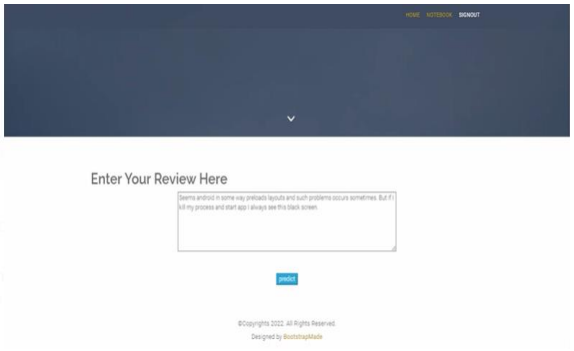


Fig.8: User input

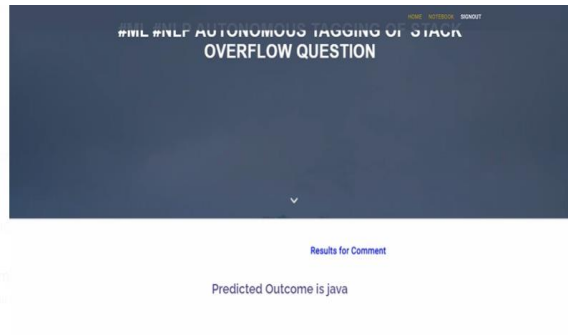


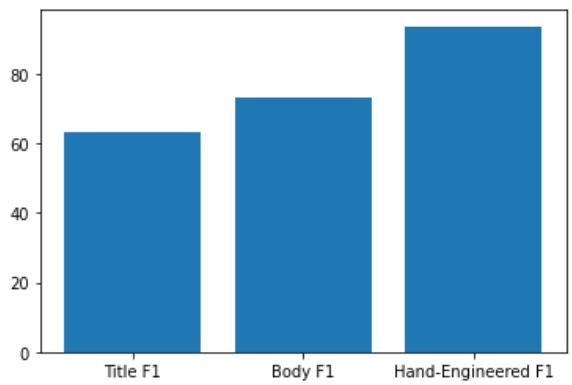
Fig.9: Prediction result

Performance Metrics:

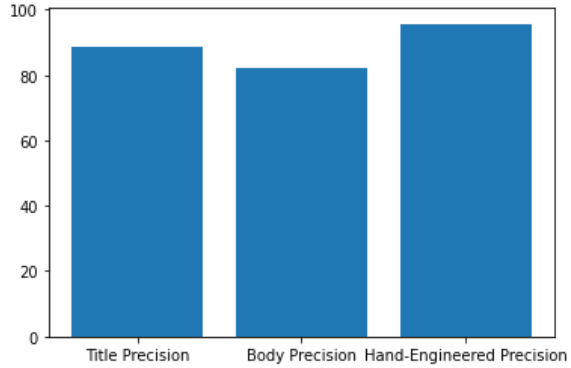
During the crucial phase of model evaluation, the model's operation is demonstrated. Execution estimates in light of the chaos structure are utilized to assess the exhibition of the chose models on the test dataset. A confusion matrix indicates whether a model's characterisation on the test set is True Positive (TP), True Negative (TN), False Positive (FP), or False Negative (FN). zero, true (one), or false (one)

Performance Comparison Graphs:

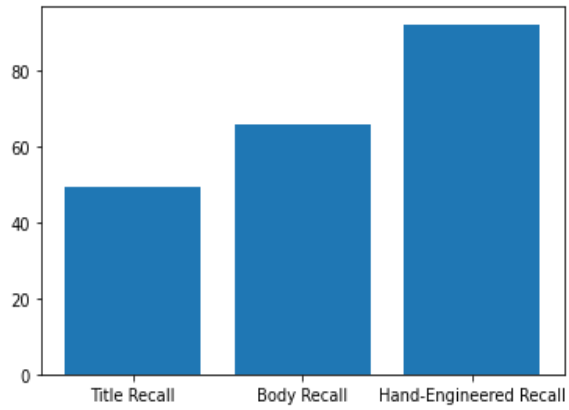
F1-score:



Precision:



Recall:



VI. CONCLUSION

We utilize a one-versus rest order plan to naturally characterize the subjects of inquiries posed to on internet based discussions. Our grouping method achieves an ideal F1 score of 62.35 percent for a subset of the dataset with 100 of the most well-known labels and a preparation set of at least 500 examples of each tag by employing a straight SVM model and carefully selected highlights. To manage the significant change in our classifier, we will need to use highlight choice strategies like principal component analysis (PCA) in the future. Using word-dispersed portrayals, we likewise plan to research extra angles. We could prepare a brain language model on a corpus by, for instance, using the arrangement of marks and the arrangement of k words that are most semantically connected to each name as the ngrams jargon. We tried word2vec on the StackOverflow dataset, but we were unable to generate word vectors with significant nearest neighbors. Pre-prepared vectors and extra tweaking are probable ways of working on this methodology. We accept that displaying label connection and considering the progressive construction of the labels could further develop execution given the hardships of classifying wide labels like windows. Non-direct classifiers like Gaussian portion SVMs and brain organizations, as non-linearity might make it conceivable to more readily isolate information, may likewise be subjects of our examination.

VII. FUTURE WORK

As a result, adding more well-known tags to our prediction algorithm would be our next step. Furthermore, to acquire a more deeper perception of the dataset, we expect to explore refined strategies like deep learning. Deep gaining approaches will without a doubt separate more data from the dataset than present factual strategies.

VIII. REFERENCES

[1]. Prof. Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features.

[2]. Kaggle (2013). Facebook recruiting III - keyword extraction. https://www.kaggle.com/c/facebook-recruiting-iii-keyword-extraction.

[3]. Klassen, M. and Paturi, N. (2010). Web document classification by keywords using random forests. In Networked Digital Technologies, volume 88 of Communications in Computer and Information Science, pages 256–261. Springer Berlin Heidelberg.

[4]. Loper, E. and Bird, S. (2002). Nltk: The natural language toolkit. In Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics, ETMTNLP '02, pages 63–70, Stroudsburg, PA, USA. Association for Computational Linguistics.

[5]. Manning, C. D., Raghavan, P., and Sch¨utze, H. (2008). Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA.

[6]. McCallum, A. K. (1999). Multi-label text classification with a mixture model trained by em. In AAAI 99 Workshop on Text Learning.

[7]. Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. CoRR, abs/1301.3781.

[8]. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikitlearn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830.

[9]. Pennington, J., Socher, R., and Manning, C. D. . (2014). Glove: Global vectors for word representation. Conference on Empirical Methods in Natural Language Processing.

[10]. Schapire, R. E. and Singer, Y. (2000). Boostexter: A boosting-based system for text categorization.

[11]. Thorsten Joachims. Transductive inference for text classification using support vector machines. 99:200–209, 1999.

[12]. Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. 14(2):1137– 1145, 1995. 5

[13]. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Ima-genet classification with deep convolutional neural networks. pages 1097–1105, 2012.

[14]. Yann LeCun, YoshuaBengio, and Geoffrey Hinton. Deep learning. nature, 521(7553):436, 2015.

[15]. Andy Liaw, Matthew Wiener, et al. Classification and regression by randomforest. R news, 2(3):18–22, 2002.

[16]. Yutaka Matsuo and Mitsuru Ishizuka. Keyword extraction from a single document using word co-occurrence statistical information. International Journal on Artificial Intelligence Tools, 13(01):157–169, 2004.

[17]. S. Nashat, A. Abdullah, S. Aramvith, and M. Z. Abdullah. Original paper: Support vector machine approach to real-time inspection of biscuits on moving conveyor belt. Comput. Electron. Agric., 75(1):147–158, January 2011.

[18]. SuphakitNiwattanakul, JatsadaSingthongchai, EkkachaiNaenudorn, and SupachanunWanapu. Using of jaccardcoeffi-cient for keywords similarity. In Proceedings of the International MultiConference of Engineers and Computer Scientists, volume 1, 2013.

[19]. Payam Refaeilzadeh, Lei Tang, and Huan Liu. Crossvalidation. pages 532–538, 2009.

[20]. Sebastian Schuster, Wanying Zhu, and Yiying Cheng. Predict-ing tags for stackoverflow questions. CS229 Projects, Stanford university, 2013.

Cite this article as :

B. Ravinder Reddy, K. Bhadri Prabhav, P. Hemanth Sagar, P. Rahul, "Autonomous Tagging of Stack Overflow Questions", International Journal of Scientific Research in Science and Technology (IJSRST), Online ISSN : 2395-602X, Print ISSN : 2395-6011, Volume 10 Issue 2, pp. 263-269, March-April 2023. Available at doi : https://doi.org/10.32628/IJSRST52310240
Journal URL : https://ijsrst.com/IJSRST52310240