MJRET

OPEN ACCESS

Multidisciplinary Journal of Research in Engineering and Technology

www.mjret.in

eISSN:2348-6953

Predicting Tags for Stack Overflow Questions Using Classifier

Nidhi Vyas, Jagriti Mishra, Mrunal Metkar, Vaishali M. Barkade
Department of Computer Engineering
JSPM's RajarshiShahu College of Engineering, Pune, India

Abstract :*The adequacy of any online educational related platform is mostly dependent on the user's experience that he/she experiences on using that platform. Hence it is the fundamental requirement to develop a system that takes care of user's interest into account while posting content on the platform online . Online platforms like Reddit, Quora, Geeks for Geeks, and Stack Exchange have large amount of data in the form of questions and answers that are posted by the users.Now-a-days very large-scale datasets are available by such websites that can be used in the form of input for designing a system based on tag prediction. We have used stack_overflow_tag_prediction dataset. Each question in Stack Overflow generally contains four segments ID, Title,Body and Tags as illustrated in Fig.1. With the help of text in the title and body our system predicts the tags for the particular question. The predicted tags are vital for the actual working of Stack Overflow.*

Keywords*: Tf-idf, Tags, text classification, feature extraction*

1. INTRODUCTION

Now-a-days the convolution and intensity of software is growing hastily and on daily basis various programming languages, tools are arriving in the market. For software development various mixture of platforms, technologies and tools are required. Because of which software professionals need to enhance and broaden their skill sets accordingly as per current the trend in software development. Currently, there is a rapid rise of many online communities for software development, that not only help software developers in conforming new technologies but along with that they provide them support for their software relevant problems. The online communities are basically divided into three groups - Tutorial sites (w3schools.com), Code sharing websites (github.com), Question-Answer (Q&A) based communities(stackoverflow.com). User queries along with their explanation are provided by different Online Question-Answer communities like peakinternet1, codeproject2, superuser3,StackOverflow4 (SO) etc, which on the other hand lowers the load of developers by giving them identical sort of illustration and bug fixing . Stack Overflow is one of the largest Question-Answering website and it is widely used globally by software developers for programming advice. A tag prediction system is maintained by Stack Overflow website that helps the programmers to resolve their queries with the help of suggested tags . Stack Overflow structure is mentioned in the below snapshot i.e shown in Fig 1. In this Fig 1, the programmer posted a question in which a question related to value error and type error in Python is asked. In the below post there are three sections of a question i.e., Question Title, Question body and tags. In question body the code or description about error is mentioned. Here, 3 tags are predicted by the SO website i.e. 'python', 'type error', 'value error'. When the predicted tags are viewed by the programmer they get more insights about the error which are faced by them. Thus, those predicted tags proves to be useful and helps in resolving gaining

in- depth knowledge about the particular error. The concept for predicting tags is known as tagging and the overall system that predicts tags is called Stack Overflow tag prediction system. Tagging is also useful for - 1. Displaying posts related to a question asked by user that may have already been answered thus enhancing the experience. 2. Grouping questions about similar topics together for users.

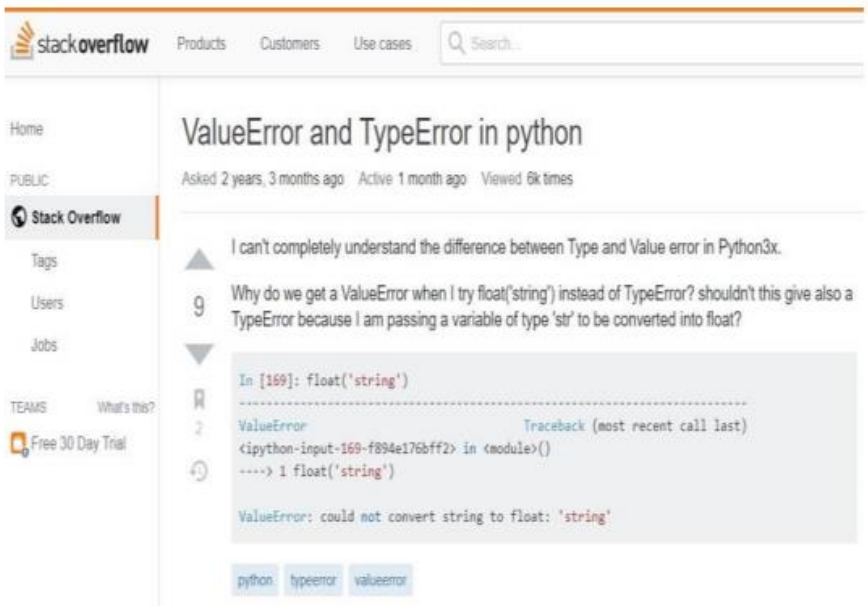

Fig. 1 Snapshot of Stack Overflow website

Tags proves to be important when grouping similar topics together and it must be done with the system which guesses on own the tag just on the basis of content of body which the user gives as input in order to extract relevant data. Thus, we have developed a system that helps in tagging as manual process of tagging is time consuming and hectic also, thus it enhances the experience of the website for the user. To ease the work of programmer in getting his query resolved it is important to build an automatic system which predicts the tags for the posted questions. A model known as Stack Overflow Tag Prediction System is presented in this paper that predicts the tags for the questions that are posted on SO website. The input data of system is taken in the form of attributes as- Question body, Question title and tags. For input data (csv file) is used from the Kaggle website. The model is trained with the help of data mentioned using classification approach and it predicts maximum 5 tags, minimum of 1 tag. This paper also shows the results on the basis of two classifiers applied namely, SGD(Stochastic Gradient Descent) and Logistic Regression.

2. LITERATURE SURVEY

Taniya Saini and ShubhamTripathi in their paper[1] they implemented different classifiers for the prediction of tags of the questions that are present on online platforms. SVM, Naive Bayes and Unique feature Extraction were the classification techniques they used along with tf-idf count of the posts that helped them to select the features, tokens from the posts that helped to predict the topic that was relevant to the post.

Steven Bird and Edward Loper[2] have proposed NLTK. NLTK i.e., Natural Language Toolkit provides us with a simple, uniform framework for assignments, extensible, demonstrations, and projects. It is easy to learn, along with documentation it is simple to use. It is widely used in teaching and research.

ThrostenJoachims[3] introduces support vector machine for text categorization. In this paper, the text categorization is done using Support vector machines algorithm. A model named Tagassist which does automatic tag suggestion to the user's post.

Sanjay C. Sood, Sara H. Owsley, Kristian J. Hammond, Larry Birnbau [4] in their paper they showed that Tag Assist was able to predict the relevant tags for new blog posts. To increase the precision without reducing the recall value they referred novel tag compression algorithm and case evaluation component. The blogging community will be effectively benefited from such system that can propose relevant tags.

Jos´eR.Cede~noGonz´alez, Juan J. Flores Romeroy, Mario Graff Guerreroz and Felix Calder´onx [5] in their work they showed that good results can be achieved by using a simple classifying system, that was based on scientific libraries that were available publicly. By this proposal many advantages were available that were not present in other schemes. One such advantage is that the tuning of the classifiers can be done with respect to the tag they try to predict. That means if there is a high probability that a certain tag will appear in some position, then it will be easier to train the classifier with certain bias, which was not possible to get with developments.

Clayton Stanley and Michael D. Byrne [6] publisheda paper regarding, tag prediction of stack overflow posts. In their work they developed a Bayesian probabilistic model that was ACT-R inspired, and was used to predict the hashtags that was used by authors in their post.

GrigoriosTsoumakas and IoannisKatakis [7] work is based upon multi-label classification. The sparse related literature was organized into a structured presentation on which they performed comparison that was based upon certain multi-label classification methods experimental results.

Andrew Kachites McCallum [8] worked on Multi-Label Text Classification with a Mixture Model that was trained by EM. In his work, Bayesian classification approach was used. A mixture model was used to represent multiple classes that comprised of a document. The classes responsible for generating the document and not each word related to it was represented using labeled training data. So, he referred to the concept of EM to fill that missing values for learning both the word distribution in each class mixture component and distribution over mixture weights.

Jeffrey Pennington, Richard Socher and Christopher D. Manning [9] worked on the concept of GloVe: Global Vectors for Word Representation. In their work they proposed a model which resulted in a global log bilinear regression model. In that the advantages of the two major model families were combined, those are local context window methods and global matrix factorization. In addition to this the statistical information is leveraged efficiently in a large corpus just by training only on the non-zero elements in a word-word co-occurrence matrix, instead of training entire sparse matrix or individual context windows.

Tomas Mikolov, Kai Chen, Greg Corrado and Jeffrey Dean [10] in their paper worked on finding the Word Representations in Vector Space efficiently. The quality of their wordrepresentation is measured using word similarity task, and results are further compared based on best performing techniques on different types of neural network.

3. IMPLEMENTATION

The implementation of the model was mainly divided into 4 stages, where every stage of the implementation dealt with a specific input and output. Fig 2. helps in getting a clear idea about how the implementation is done step by step. Below are the 4 stages/step-by-step implementation process of the model which were implemented.

A. Data Pre-Processing

- Input : stackoverflow_tag_prediction.csv
- Output : Pre-processed dataset

B. Splitting of dataset

- Input : Pre-processed dataset
- Output : train.csv and test.csv

C. Classifier Applied

- Input : train.csv
- Output : Trained model

D. Predicting Tags (for test data)

- Input : test.csv, trained model
- Output:Tags for Testing data

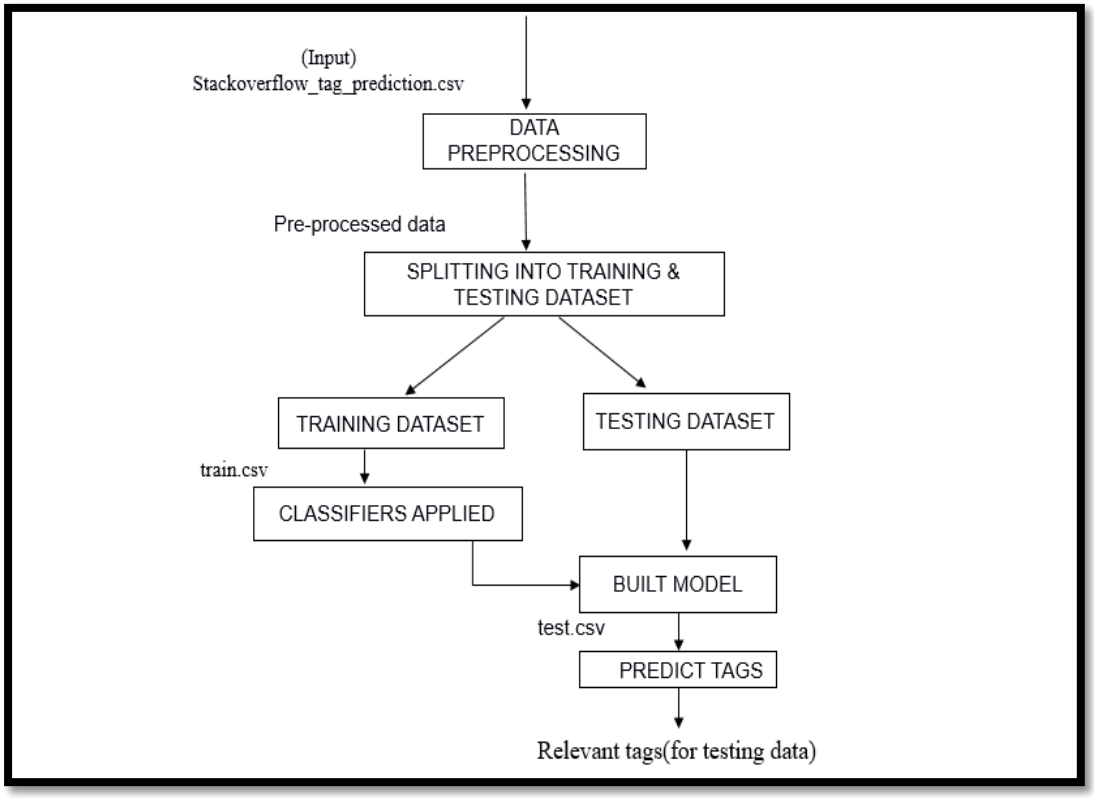

Fig. 2 System Architecture

Stage 1: Data Pre-Processing

The model implementation of the model starts with the dataset loading and pre-processing. The dataset named 'Stackoverflow_tag_Prediction.csv' is taken from the kaggle website. The dataset has following attributes: Question_id, Body, Question_title and tags, where tags for the questions are to

be predicted. Before pre-processing and analysing, dataset is loaded into a SQLite3 database and following data pre-processing and analysis steps like removing duplicates, lemmatization, stemming, vectorization, determining frequently occurring tags, plotting word-cloud of tags.

Stage 2: Splitting of dataset

The output of module 1 was the pre-processed data in the form of a sqlitedb file which was then splitted into train and test set. The training set was provided to the classifiers for the model-building and testing set was used later for the accuracy checking of the model. The dataset was divided in two parts in the ratio 80:20, i.e 80% of the dataset is for the training purpose and 20% of the dataset for the testing purpose at the later stage.

Stage 3: Classifier Applied

The dataset was labeled where tags i.e. 4th attribute is the target attribute/label for the prediction. Being supervised machine learning problem, we had used following 2 classifiers : OneVsRest +SGD classifier and Logistic Regression. For the 1st classifier OnevsRest is used as our model predicts multiple tags and not binary tags. OnevsRest is used so that, for a specific tag to be predicted that specific tag is 1 and all other tags are 0. SGD (Stochastic Gradient Descent) is used for the optimization purpose. Logistic regression suits well for multi-labeled classification, hence we have used Logistic regression as the 2nd algorithm for the model building.

Stage 4: Predicting tags

Tags are predicted for the testing dataset, where the maximum number of tags predicted are 5 and minimum number of tags predicted is 1 and the average number of tags predicted are 3. After testing the accuracy, it was found that the accuracy for the Logistic Regression classifier was more than OnevsRest+ SGD classifier. For the initial stage we had tested for 0.5 million questions out of 7 million amd accuracy for Logistic Regression classifier and OnevsRest+SGD classifier was 25% and 23% respectively.

4. OUTCOME

The outcome for the model implementation was as below, where we can check the accuracy of the model for the testing set.

TF-IDF with 0.5 million datapoints:

Model	Vectorizer	Accuracy	Hamming loss	Precision	Recall	Micro f1
SGDClassifier(loss=log)	TF-IDF	0.23682	0.002779	0.7222	0.3263	0.4495
LogisticRegression	TF-IDF	0.25108	0.00270302	0.7172	0.3672	0.4858

Fig. 3 Result

5. CONCLUSION

Thus we have implemented ML model using Logistic Regression and SGD. Hence prediction with minimum and maximum number of possible tags with high accuracy metrics as precision along with recall was the motive and Logistic Regression model provided better results.

ACKNOWLEGEMENT

It is our proud privilege to release the feelings of our gratitude to several persons who helped us directly or indirectly in conducting our project work. We express our heartful indebtedness and owe a deep sense of gratitude to our faculty guide Prof. Vaishali M. Barkade for her guidance and support.

The study had indeed helped us to explore more knowledge related to our domain that is MACHINE LEARNING as well as topic which we have decided.

REFERENCES

[1]. T. Saini and S. Tripathi, 2018 "Predicting Tags for Stack Overflow using different classifiers".

[2]. E. Loper and S. Bird, "Nltk: The natural language toolkit," in Proceedings of the ACL-02 Workshop on Elective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics, ETMTNLP '02. Association for Computational Linguistics, 2002, pp. 63– 70.

[3]. T. Saini and S. Tripathi, 2018 "Predicting Tags for Stack Overflow using different classifiers".

[4]. T. Joachims, 1998. Text categorization with support vector machines: Learning with many relevant features.

[5]. S. Sood, S. Owsley, K. J. Hammond, and L. Birnbaum, "Tagassist: Automatic tag suggestion for blog posts," ICWSM, 2007.

[6]. R. C. no Jos´ e, J. J. G. alez, M. G. F. Romeroy, F. Guerreroz and C. on, 2015.

[7]. Clayton Stanley and Michael D Byrne. 2013. Predicting tags for stackoverflow posts. In Proceedings of ICCM 2013

[8]. Tsoumakas, G. and Katakis, I. (2007). Multi-label classification: An overview. Int J Data Warehousing and Mining, 2007:1-13. McCallum, A. K. (1999). Multi-label text classification with a mixture model trained by em. In AAAI 99 Workshop on Text Learning.

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