International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/CSEIT217668

SEO Ministration : Content Related Tag Suggestion Using One vs Rest

Dyapa Sravan Reddy[1], Lakshmi Prasanna Reddy[1], Kandibanda Sai Santhosh[1], Virrat Devaser[2]

[1]School of Computer Science and Engineering, Lovely Professional University, India

[2]Assistant Professor, School of Computer Science and Engineering, Lovely Professional University, India

Article Info
Volume 7, Issue 6
Page Number: 245-253

Publication Issue :
November-December-2021

Article History
Accepted : 02 Dec 2021
Published : 10 Dec 2021

ABSTRACT

SEO Analyst pays a lot of time finding relevant tags for their articles and in some cases, they are unaware of the content topics. The current proposed ML model will recommend content-related tags so that the Content writers/SEO analyst will be having an overview regarding the content and minimizes their time spent on unknown articles. Machine Learning algorithms have a plethora of applications and the extent of their real-life implementations cannot be estimated. Using algorithms like One vs Rest (OVR), Long Short-Term Memory (LSTM), this study has analyzed how Machine Learning can be useful for tag suggestions for a topic. The training of the model with One vs Rest turned out to deliver more accurate results than others. This Study certainly answers how One vs Rest is used for tag suggestions that are needed to promote a website and further studies are required to suggest keywords required.

Keywords : Search Engine Optimization, Content Tags, Machine Learning, Data-set, Selenium, Beautiful Soup, Text Analysis, LSTM, One vs Rest, One vs One.

I. INTRODUCTION

Search Engine Optimization is the method of leveling up the quantity and quality of traffic to any website with the help of organic search engine results.
Usually, a search engine is embedded with a crawler that puts together all the content that could be ever found on the internet. It reverts all 1's and 0's to Search Engine to build indices. This index is fed into an algorithm that strives to match all data with your query. Numerous factors are considered by the Search Engine's Algorithm to make indices. According to survey responses by 128 SEO Professionals in 2013, Social Metrics such as tweeted tags and their quality and quantity contribute around 7.24% of total factors and page-level link features contribute around 19.15% (metadata and Header Tags) [1]. There are different types of tags usually like Header tags range from H1 to H6.

The World Wide Web Consortium concludes that every HTML report ought to have an identity detail within the head section of the Web Page. They additionally state that the identity detail must be used to become aware of every person's page content [2].

Every website on the web must have a unique identity and it is provided by meta tags. A unique HTML tag that gives data about the web page is a Meta tag. Meta tags don't render on a webpage so, they are invisible to the clients. Meta tags assist search engines to categorize a web page for a user query. This tag plays a vital role in SEO. It is hard to forget that Meta tags are an incredible option to make a website an enraged success. Every website owner who uses meta tags can control how their web pages result via search engines like Google. Meta tag key phrases assist search engines like google to index a web page. Website proprietors may subsequently utilize meta tags to characterize applicable keywords or to determine a theme for their Webpage [3].

So, SEO analysts need to understand the entire content in the website to write relevant keywords or tags which usually takes more amount of time. Sometimes the analyst may not understand what the topic is about and sometimes it may take a longer time than expected.

To ease this process, this paper proposes a machine learning model that suggests tags that can be taken as reference and further be improvised with some keyword tags or can be used directly. This study has analyzed and implemented LSTM and One vs Rest.
Comparison of these methods to address the issue is been discussed in-depth, in this paper. Before moving into methods, let's gain some knowledge about the dataset.

II. Background and Terminology

Dataset Preparation:

The collection of records with common fields is known as a dataset. The dataset is divided into two types they are, training dataset and testing dataset. The model is fed with a training dataset initially and later validated using a testing dataset. In this way, one can conclude if the model is accurately interpreting the given data [4].
The model uses the data to learn and improve continuously. To make the model predict accurately, one should provide a high-quality dataset. There is very little chance to get a high-quality dataset without any missing, duplicate, incorrect and irrelevant data on the internet. So, the easiest solution to achieve a high-quality dataset is to prepare it. In this study, the dataset is created by us using web scraping tools. This study needs not only a high-quality dataset but also a real dataset which makes the model interpret the real-world data and predict accurately. So, let's see how the dataset is prepared.
The requirements of the dataset are it should have two attributes they are content and content tags. The main aim is to extract the values of these attributes from web pages like blogs and medium articles to prepare a real dataset.
The data used in this study is pulled from medium articles. Before initiating extraction of the content from web pages, we need the website link of that webpage. So, the initial step is extracting all web links from medium articles related to technology. In this study, the dataset contains data related to technologies only because it's not so easy to extract all types of content from all over the internet, so for study purposes, only data related to technologies is extracted. This can be achieved using selenium.

Selenium:

Selenium is a web-based automation tool that works efficiently for web scraping. Selenium has a web driver which enables several features used to drive to the required web page and scrape different elements content from the web page, according to our needs. Using a selenium web driver, one can extract data from multiple web pages, group the data, and save it.[5]

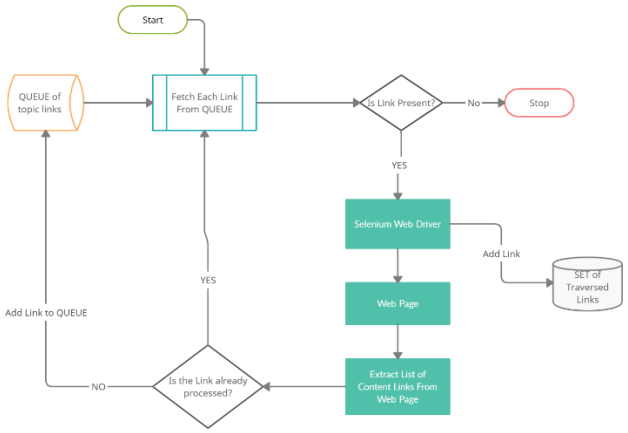

Fig no. *(1) Flow Diagram of Using Selenium to extract website links of content.*

Let's see how the dataset is created by using selenium in this research. Initially, some topic links are chosen like data science, machine learning, etc from medium articles and stored in a queue. Next, from the queue, each link is sent to the selenium web driver to fetch the web page and extract the list of content links from the web page. Before sending the link to the web driver, it is added to a set to track the traversed links. After extracting the content links, each link is verified and added to the queue only if it is not traversed yet. This cycle continues until the queue becomes empty. Now, the website links are ready. The leftover part is with scraping these links and extracting our necessary content. This can be achieved by using beautiful soup.

Beautiful Soup:

Beautiful soup is a python package that is used widely to extract data from any HTML or XML page. This package offers simple searching and navigation methods that help in the easy extraction of content from web pages.[6]

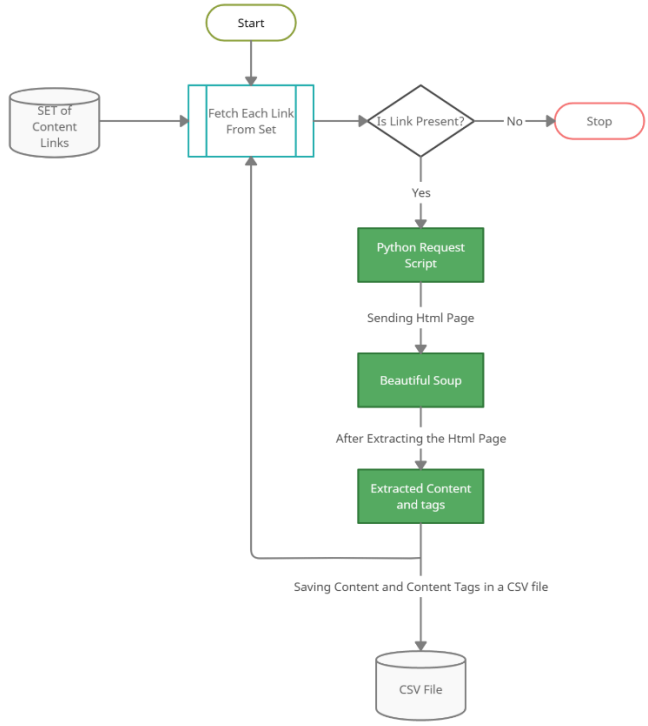

Fig no. *(2) Flow Diagram of Using Beautiful Soup to extract content and content tags from a webpage.*

Let's see how the dataset is created by using Beautiful Soup in this study. In the first step, each link from the set is sent to the python request script which makes a get request to the link that returns HTML content. This HTML content is then passed to the beautiful soup which returns an object. This object has simple methods like find with which one can extract the content and content tags from the web page and store them in a CSV file. This cycle continues until the set of links becomes empty. So now the dataset is ready which contains content and tags.

	Content	Content_Tags
0	['Sign in', 'Zhi Li', 'May 30, 2019·8 min read...	['machine-learning', 'nlp']
1	['Sign in', 'Kung-Hsiang, Huang (Steeve)', 'Ma...	['machine-learning', 'data-science', 'deep-lea...
2	['Sign in', 'Lukas Frei', 'May 30, 2019·5 min ...	['data-visualization', 'data-science', 'python...
3	['Sign in', 'Deborah Kewon', 'May 30, 2019·4 m...	['data-science', 'pyspark']
4	['Sign in', 'Yitong Ren', 'May 30, 2019·4 min ...	['machine-learning', 'neural-networks', 'mathe...

Fig no. *(3) Picture of Dataset of 5 records*

Data Pre-processing:

Data pre-processing is one of the crucial tasks to be accomplished before making an ML model. It is the

process of preparing the data and making it a perfect fit for an ML model

Generally, when a dataset is downloaded or created, the dataset may contain some noises, missing values, and unformatted data which cannot be directly used in an ML model. So, the data is cleaned and formatted according to the model requirements that helps in increasing the efficiency and accuracy of an ML model [7].

For text pre-processing firstly, stop words and repeated words are being removed. Then the text is tokenized or stemmed. For this nltk library is used and to convert these strings into vectors there are a plethora of algorithms, out of which TF-IDF is being used in this study.

TF-IDF

Inverse Document Frequency is represented as IDF and Term Frequency is represented as TF. TF determines the count of a word in a document and IDF determines the word count across different documents. TF-IDF reads whole documents and develops word vectors by assigning weight to each word.

Inverse Document Frequency for a word is calculated by taking the logarithm of the value obtained by formula:

The total number of sentences / Total number of Sentences containing that word.

Then Word Vector is made by multiplying both TF and IDF [8].

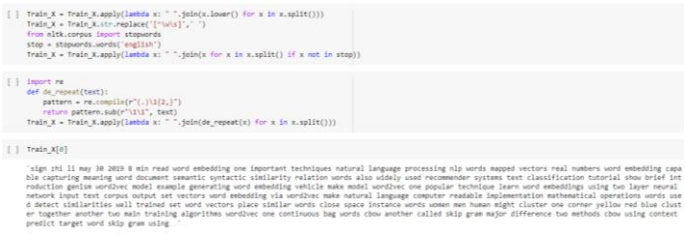

Fig no. *(4) Removing stop words and repeated words in a Text.*

Fig no. *(5) Usage of TF-IDF for developing word vectors by using fit function.*

III. Methodology

To obtain tags out of any data one needs to perform text analysis and text classification on pre-processed data. After which one can use some multilabel classification models to classify this text into output's named as tags. Out of some available multilabel classification models this research used LSTM and One vs rest.

LSTM:

Long Short-Term Memory and Recurrent Neural Network are shortened as LSTM and RNN. LSTM is a special kind of RNN. They can cope up with Long term dependencies in a better way than basic Recurrent Neural Networks. LSTM is majorly used to handle situations where RNN do not play their part.

WHY NOT RNN?

Recurrent Neural Network takes both preceding layers output and current input as input. RNN stores this information for a short period. Few Widespread applications of RNN are music composition, speech processing, text analyzing, non-Markovian and control. To obtain the current layer's output sometimes data that is given as input to previous layers, a long time ago (long term dependency) is needed. Since RNNs remember information only for a shorter period, it fails to handle long-term dependencies. Also, in the case of RNN's, there is no control over how much information and what information is to be carried forward to other layers and how much information is to be forgotten.

Moreover, the major drawbacks of RNN are the vanishing gradient problem and exploding gradient problem. In the vanishing gradient problem, while backtracking the loss(gradient) in RNN's, the value is

reduced as it passes through layers and finally it becomes almost negligible, resulting in inundation of weights in initial layers, consequently producing the same output as before. In exploding gradient problem, the gradients are high in value due to the large value of components, due to which the weights get updated beyond optimal value [9].

To dismiss these problems RNN's are modified by adding a new unit because of which scaling factor is always maintained as constant equal to one. This new unit called LSTM (Long Short-Term Memory) is embedded with numerous gates.

LSTM WORKING:

The hidden layers in LSTM are embedded with gated cells. LSTM has 4 layers. These layers interact with each other in such a way that the output and cell state are simultaneously produced. There are 3 logistic sigmoid gates and 1 tanh gate. These sigmoid and tanh gates are used to put a threshold on the amount of information that is directed along with the cell. These gates decide what information should be passed and what information should be forgotten. Generally, output falls between 0-1 where '0' represents 'reject all' and '1' represents 'include all'

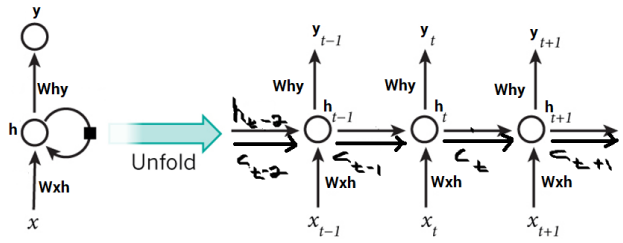

Fig no. *(6) Image explaining Hidden layers of LSTM which is created by* Sinha, A. (2020, 05 11). Retrieved from geeks for geeks: https://media.geeksforgeeks.org/wp-content/uploads/20200304231821/img46.png

Let $h_{(t-1)}$, $C_{(t-1)}$ and x_t be three inputs of LSTM cell and h_t, C_t be the outputs as shown in the figure. The hidden state is represented as h_t and, the cell state or memory is denoted by C_t and, the data point is represented by x_t. There are two inputs for the first sigmoid layer respectively $h_{(t-1)}$ and x_t. Here preceding cell's hidden state is represented by $h_{(t-1)}$. Since its output selects what part of the information from the cell preceding it to be accepted, it is known as Forget gate. The output falls between 0 and 1(both inclusive). The previous cell state is represented by $C_{(t-1)}$. Output is point-wise multiplied with $C_{(t-1)}$ [10].

CONVENTIONAL LSTM:

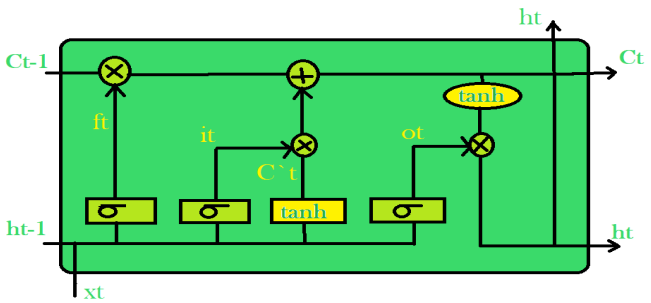

Fig no. *(7) Image explaining Conventional LSTM which is created by* Sinha, A. (2020, 05 11). Retrieved from geeks for geeks: https://media.geeksforgeeks.org/wp-content/uploads/20200304231933/LSTM_cell.png

Input gate is the second sigmoid layer. It determines the part of the information that is to be sent to the cell. $h_{(t-1)}$ and x_t represents two inputs which are taken by them. C_t represents a new vector created by the tanh layer. tanh layer and sigmoid layer together decide which part of information a cell stores. The amount of information to be added to the cell state is determined by their point-wise multiplication. The output from forgot gate is multiplied with the state of the previous cell and summed up with the result to obtain the current cell state C_t. Consequently, the output of the cell is determined by sigmoid and tanh layers. The part of the cell state that is present in the output is decided by the sigmoid layer and output is shifted to fall between [-1,1] by tanh layer. The output obtained from two layers undergo pointwise multiplication to give the cell's output.

Since recognition for LSTM is skyrocketing many modifications have been added to conventional Long Short Term Memory architecture to dumb down the complexity of a cell's internal structure. Peephole connections were invented by Gers and Schmidhuber. The Gate Layers are allowed to know information about cell states in all the moments in this. In other advancements, LSTMs replaced two separate gates with a single gate. This single gate is the combination of input gate and forgets gate. Another advancement in LSTM comes with the embedment of the Gated Recurrent Unit to improve design complexity and reduce the number of gates.

WHY NOT LSTM FOR OUR PROBLEM?

LSTM requires plenty of resources. Since there are many linear layers present, a large amount of memory is needed. This makes LSTM inefficient hardware-wise. It won't be able to retain information for a longer period. Since ours is a huge amount of data to be analyzed it is not able to classify properly. Dropout is a regularization technique in which some weights are made zero so that neurons become inactive. Implementation of dropout regularization became difficult. So, LSTMs are prone to overfitting in our case.

ONE VS REST:

One vs Rest is a classification algorithm that classifies objects into multiple classes.

CLASSIFICATION:

Classification is the most prevalent task in Machine Learning. Classification algorithms are categorized into 2 types:

(i) Binary Classification Algorithm, &
(ii) Multi-Class Classification Algorithm

(i) Binary Classification Algorithm: This algorithm divides whole data/input into two different clusters/groups.
(ii) Multi-Class Classification Algorithm: In this kind of algorithm, the data/input is divided into more than 2 groups/clusters.

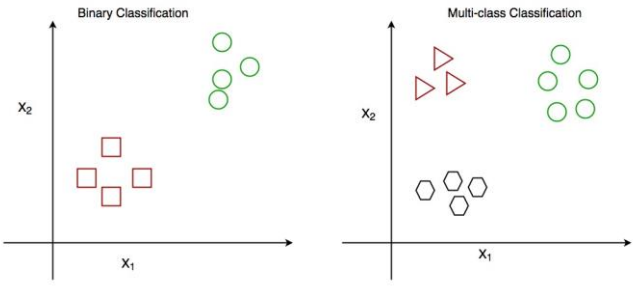

Fig no. *(8) This picture explains Binary and Multi-class classification* which is created by Bora, A. (2020, 07 17). Retrieved from geeks for geeks: https://media.geeksforgeeks.org/wp-content/uploads/20200702103829/classification1.png

Classification algorithms are primarily built to solve Binary Classification problems. The most commonly known algorithms are SVM (Support Vector Machine), Logistic Regressions, and Perceptron Models. The disadvantage of these algorithms is that they can't be used directly for solving multi-class classification problems.

Binary Classification Algorithms can be used to solve multi-class classification problems. To make it possible, the Multi-Class Classification Dataset is divided into different binary classification datasets and fitted the binary classification model on every dataset accordingly. There are two different heuristic methods for this approach. They are:

(i) One Vs One Strategies &
(ii) One Vs Rest Strategies [11].

ONE VS ONE APPROACH FOR MULTI CLASS CLASSIFICATION:

One of the methods to use binary class classification algorithms for multiclass classification is One vs One. Multi-Class classification datasets are being divided into binary classification problems. One vs One method divides the whole dataset into one Data Set for each class Vs with each other class.

An example for Multi-class classification problem is, consider 4 different classes as follows 'apple', 'banana', 'grapes', 'orange'.

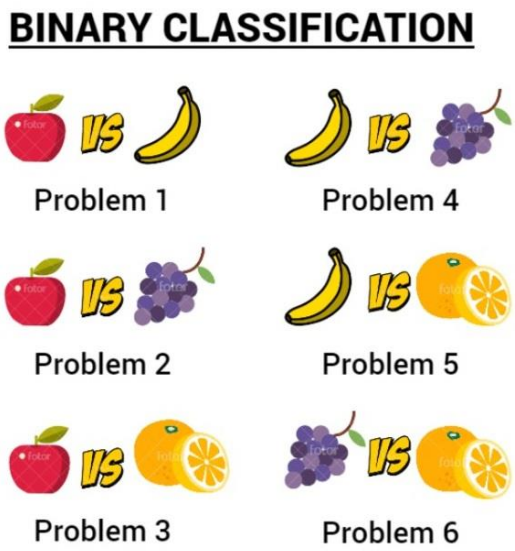

Fig no. *(9) Example of how classes are classified using One vs One approach for multi-class classification.*

No. of Binary Datasets can be calculated by using the below formula:

(NumOfClasses * (NumOfClasses – 1)) / 2

Here One class label is predicted by every Binary Classification model. The model can achieve the most accurate predictions by using the One Vs One approach.

Another possible way to achieve this is for every possible pair of classes, N(N-1)/2 binary discriminant function must be used. It is called a One vs One classifier. According to a majority vote each point is classified amongst the discriminant functions. Support vector machines (SVM) and related kernel-based algorithms work best for this approach. It is mostly accepted since the size of the training dataset is not directly proportional to the performance of the kernel methods. To bring down this effect the subsets of training data can be used [12].

ONE VS REST APPROACH FOR MULTI CLASS CLASSIFICATION:

One of the methods to use binary class classification algorithms for multi-class classification is One vs Rest. Multiple binary classification problems are obtained by splitting a single multi-class dataset. And then on every binary classification problem a binary classifier is trained to predict outcomes.

For instance, consider a multi-class classification problem with classes 'a', 'b', 'g'. Three binary classification problems can be made from the problem. It can be explained as below.

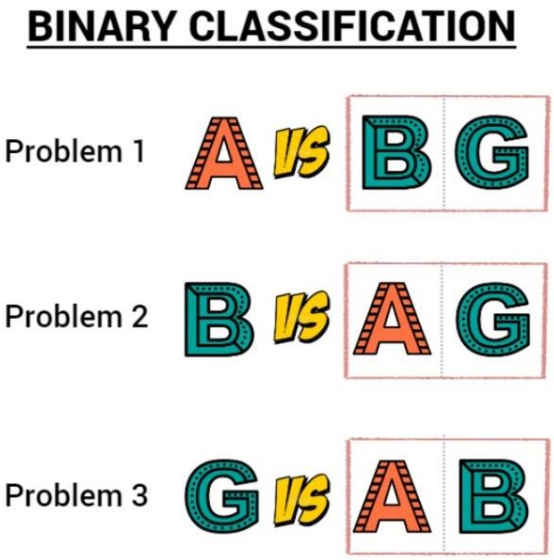

Fig no. *(10) Example of how classes are classified using One vs Rest approach for multi-class classification.*

One vs Rest is the suitable method to use for the above kind of classification where C binary classifiers are trained on fc(x). I*f the data is obtained from class c then it is treated as positive else, if it is obtained from other classes then it is treated as negative.*

This method requires a class membership probability or a probability-like score predicted by every model. Argmax of output scores (class index with the highest score) is used for the prediction of class [13].

WHY ONE VS REST AND WHY NOT ONE VS ONE:

One vs Rest can be natively used for Logistic Regression for multi-class classification. It can also be used for other classifiers that natively support multi-class classification. Since the problem requires generating multiple tags that are useful for SEO analysts. If One VS One is used, it would be able to classify the inputs into only one of the available output classes thereby achieving Multi-class Classification. But the solution to our problem requires generating tags that can be classified into

multiple output classes simultaneously i.e., Multi-label classification.

One vs Rest Practical Implementation:

In the implementation OnevsRestClassifier Class is imported from sklearn.multiclass. Previously TfidfVectorizer is imported from the feature_extraction class in sklearn to make vectors out of the preprocessed text. Now the vectorized data obtained is fed to the model using the fit function. Now the model is trained and is ready to predict the output. Outputs are predicted using predict function.

```
from sklearn.linear_model import LogisticRegression


from sklearn.multiclass import OneVsRestClassifier

from sklearn.linear_model import SGDClassifier
Sg=SGDClassifier(max_iter=1000)
from sklearn.svm import SVC
svc=SVC()
```

```
clf = OneVsRestClassifier(Sg)
```

```
clf.fit(xtrain_tfidf, Y)
```

```
OneVsRestClassifier(estimator=SGDClassifier(alpha=0.0001, average=False,
                                            class_weight=None,
                                            early_stopping=False, epsilon=0.1,
                                            eta0=0.0, fit_intercept=True,
                                            l1_ratio=0.15,
                                            learning_rate='optimal',
                                            loss='hinge', max_iter=1000,
                                            n_iter_no_change=5, n_jobs=None,
                                            penalty='l2', power_t=0.5,
                                            random_state=None, shuffle=True,
                                            tol=0.001, validation_fraction=0.1,
                                            verbose=0, warm_start=False),
                    n_jobs=None)
```

Fig no. *(11) This picture depicts the making and training of a model*

IV. Result and Conclusion

Machine Learning is making numerous advancements in many fields. The motive behind our implementation is to save the time of SEO Analysts. While using LSTM, despite using dropout regularization, we observed overfitting. While using One vs Rest we neither observed underfitting nor overfitting. Since LSTM is overfitting because of limited data available, hereby we conclude that One vs Rest is performing better than LSTM in predicting tags. Since the dataset is made only on a particular domain the major limitation of this implementation is that model cannot predict tags for domains other than computer science. Our future scope is to add more data and make the model more flexible.

```
[62] text=input()
     International Journal of Engineering Applied Sciences and Technology, 2020  Vol. 5, Issue 8, ISSN No. 2455-2143, Pages 207-210  Published Online December 2020 Ir
```

```
k=clf.predict(Tv.transform([text]))
```

```
binarizer.inverse_transform(k)
```

```
[('data-science', 'machine-learning')]
```

Fig no. *(12) input and output of the proposed model.*

In the above figure, text input is given to the model and the model returned tags related to that text and the tags are ‘data-science, ‘machine-learning’. This is how the proposed model works in reality.

V. REFERENCES

[1]. Philis, T. (2013, 09 07). SEO Ranking Factors – Are You Focusing on the Right Tasks? Retrieved from Pleasanton Web Design Blog: https://pleasantonwebdesignblog.com /2013/09/seo-ranking-factors.html

[2]. Html/Training/Metadata. (2011, 01 31). Retrieved from W3C Wiki: https://www.w3.org/wiki/Html/Training/Metadata

[3]. Schachinger, K. (2012, 05 01). How To Use HTML Meta Tags. Retrieved from Search Engine Watch: https://www.searchenginewatch.com/2012/05/01/how-to-use-html-meta-tags/

[4]. Algorithmia. (2020, 03 26). The importance of machine learning data. Retrieved from Algorithmia: https://algorithmia.com/blog/the-importance-of-machine-learning-data

[5]. K. U. Manjari, S. Rousha, D. Sumanth and J. Sirisha Devi, "Extractive Text Summarization from Web pages using Selenium and TF-IDF

algorithm," 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), Tirunelveli, India, 2020, pp. 648-652, doi: 10.1109/ICOEI48184.2020.9142938.

[6]. Nair, Vineeth G. Getting started with beautiful soup. Packt Publishing Ltd, 2014.

[7]. Data Preprocessing in Machine learning. (n.d.). Retrieved from javatpoint: https://www.javatpoint.com/data-preprocessing-machine-learning

[8]. Scott, W. (2019, 02 15). TF-IDF from scratch in python on real world dataset. Retrieved from Towards Data Science: https://towardsdatascience.com/tf-idf-for-document-ranking-from-scratch-in-python-on-real-world-dataset-796d339a4089

[9]. Pulver, A., & Lyu, S. (2017, May). LSTM with working memory. In 2017 International Joint Conference on Neural Networks (IJCNN) (pp. 845-851). IEEE.

[10]. Sinha, A. (2020, 05 11). Understanding of LSTM Networks. Retrieved from geeks for geeks: https://www.geeksforgeeks.org/understanding-of-lstm-networks/

[11]. koustubhk. (2020, 08 28). 1-vs-1 & 1-vs-Rest Classification | SKLearn. Retrieved from Kaggle: https://www.kaggle.com/kkhandekar/1-vs-1-1-vs-rest-classification-sklearn

[12]. Brownlee, J. (2020, 04 13). One-vs-Rest and One-vs-One for Multi-Class Classification. Retrieved from Machine Learning Mastery: https://machinelearningmastery.com/one-vs-rest-and-one-vs-one-for-multi-class-classification/

[13]. Bora, A. (2020, 07 17). One-vs-Rest strategy for Multi-Class Classification. Retrieved from geeks for geeks: https://www.geeksforgeeks.org/one-vs-rest-strategy-for-multi-class-classification/

Cite this article as :

Dyapa Sravan Reddy, Lakshmi Prasanna Reddy, Kandibanda Sai Santhosh, Virrat Devaser, "SEO Ministration : Content Related Tag Suggestion Using One vs Rest", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 7 Issue 6, pp. 245-253, November-December 2021. Available at doi : https://doi.org/10.32628/CSEIT217668
Journal URL : https://ijsrcseit.com/CSEIT217668