Auto Tagging, Final Report

Ömer Faruk Yaşar
21527577
omeryasar224@gmail.com

Yahya Koçak
21527189
yahyakocak49@gmail.com

Umut Piri
21783872
piriumut97@gmail.com

***Abstract*—Stack Overflow is one of the most popular online programming question and answer websites related to computer programming worldwide. Users on this site must provide tags for their submissions. It facilitates accurate classification and efficient search of high-quality labels. However, the tagging process is distributed and not coordinated because of their understanding of users' posts, English skills, and preferences. Automatic tag recommendation is becoming more and more important for information sites today.**
In this article, we propose a language representation model called BERT (Bidirectional Encoder Representations from Transformers). BERT is designed to pre-perform deep bidirectional displays from the unlabeled text, conditioned together in both left and right contexts on all layers. We tried to evaluate BERT over the StackOverflow database.

I. Introduction

StackOverflow is a user-oriented question and answer site about computer programming. This site allows users to tag their questions to make it easier for others to find their questions and to get answers faster. The subject of the question asked should be tagged with their respective fields. A problem can be related to more than one topic, that's why the user can put 5 tags each question. Users can subscribe to the tags they are interested or specialize in and can answer the questions asked or follow the topics.

StackOverflow site has enabled its users to make a maximum of five labels to a post. Users are recommended to use existing tags when tagging, but they are also allowed to create new tags. Therefore, the set of labels is infinite. Multiple tags with the same meaning have been created, allowing new and inexperienced users to add new tags. For example, even if "natural-language-processing" and "nlp" tags have the same meaning, they are available as separate tags. Therefore, it would be very useful to have a system that can automatically tag questions or suggest tags to the user.

We have proposed a system to predict tags in StackOverflow automatically for given text. There are many similarities between the articles in the same label, these similarities are used to predict tags for the unknown labeled articles. Our purpose is to make predicts as accurate as possible.

This paper mentions the following. Second part, similar studies and methods used in these studies are mentioned. In the third part, we explained the contents of the StackOverflow data we used and the preprocesses we made before using the data were mentioned. In the fourth part, the methods we used and the results we received were evaluated. Finally, the results of our studies and the conclusions we draw from these results are mentioned.

II. Releted Works

There exist many proposed approaches to auto-tagging and the associated topic text categorization. Some of them are listed below:

Deep Pyramid Convolutional Neural Networks for Text Categorization [1] paper of Rie Johnson and Tong Zhang proposes a low coomplexity word level deep pyramid convolutional neural network (DPCNN) architecture for text categorization that can efficiently represent long range associations in text.They have used three key features, the first one is that downsampling without increasing the number of feature maps. Secondly, shortcut connections with pre-activation and identity mapping for enabling training of deep networks.Third, text region embedding enhanced with unsupervised embeddings for improving accuracy.

Development of System for Auto-Tagging Articles, Based on Neural Network [2] Basis of the project is processing articles from popular web forums using neural networks. This project uses data providers such as web scraping and APIs for open resources. After data acquisition part, tag classifier run on gathered data. Paragraph2Vec model is used which has a similar architecture with word2vec. This model takes into account the contextual document as well as contextual words.

A Hybrid Auto-tagging System for StackOverflow Forum Questions [3] Smrithi Rekha V., Divya N. and Bagavathi Sivakumar P. are proposed a different approach to auto tagging problem, which is a hybrid model. They have used two different algorithms to get better results, algorithms they used are programming language detection system and question classification system. The dataset provided from Stackoverflow is used in this project, which have around 50,000 questions with tags.

A Discriminative Model Approach for Suggesting Tags Automatically for Stack Overflow Questions [4] There are three main steps in this proposed approach, converting questions into vectors, training a discriminative model and suggesting tags for a given question. Each question converted into a high dimensional vector space, where each dimension corresponds to a unique word from the documents and term frequencies are used to information retrieval term weighting scheme to claim the importance of a term in a question. 1.3 million questions are used to train the model.

Predicting Tags for StackOverflow Posts [5] Clayton Stanley and Michael D. Byrne developed an ACT-R which is insipired by Bayesian probabilistic model that can predict hashtags. The strength of association between post words and tags was calculated using one million posts. A logistic regression statistical technique is used to calibrate model parameters and optimize performance. Tag is predicted according to highest activation among possible tags.

III. Dataset

A. About Dataset

The data set we use is the data set that contains 10% of the questions and answers from the Stackoverflow programming question and answer website. The questions are all about computer programming and can be posted by anyone. There are 1,264,216 questions in the data set and there are Id, OwnerUserId, CreationDate, ClosedDate, Score, Title, Body for each question. Each question is tagged by the author with the most represented tags of the publication. Sample tags are programming languages (java, python, MySQL, C) as well as general topics (databases, algorithm, arrays). There are 3750994 tags in total and 37035 unique tags in total. Tags have two properties, id and tags and id links back to questions. The 5 most repeating tags are 'c ', 'java', 'javascript', 'android', 'python'. The chart is below.

There are users who answer the questions. The answers of these users are kept in the dataset. 2.014.516 answers are kept in the dataset. Answers include ID, body, creation date, score, owner ID, and Paret Id for each of the answers to these questions. The Parent Id column is linked back to the Questions.

In our database, each question has 1 to 5 tags. The distribution of these labels is as in the graph.

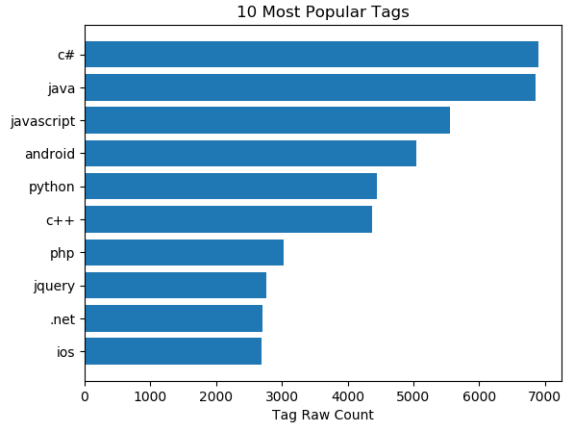

Fig. 1: Most Common Tags

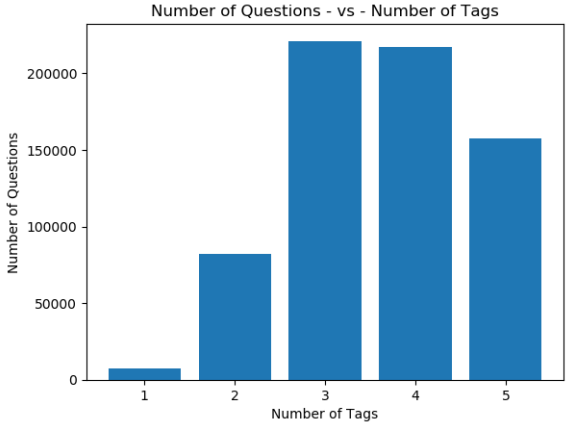

Fig. 2: Number of tags

B. Dataset Preprocessing

To make our data set more useful, it is necessary to go through some processes. The first process of these is through the tags. The tags of the same questions are indicated in different lines in our data set. We need to combine them. To do this, we used the 'groupby' method. The tags of the questions were kept in one place after this process.

Questions consist of two seperate parts that we interested in; head and body parts. To benefit both of them while trying to classifying them in to correct tags we merge these parts in single column and drop the other one. To have meaningul and correctly asked questions we eliminate questions with lower than score 5. So we by doing that we hope to increase quality of our data. As we mentioned there are more than 9000 independent tags but they are not distributed evenly so we worked on most popular 25 tags.

If done correctly, text preprocessing can help improve the accuracy of NLP tasks. For this purpose, we made some text preprocessing operations. We remove stop words and special characters from title and body. Our data set contains html tags, which creates deficiencies in terms of meaning and integrity. That's why we also removed the html tags and lowered the characters.

Lemmatization is Text Normalization techniques used to prepare text, words and documents for further processing

in Natural Language Processing. Thus, we will clear our data and get more meaningful sentences.

IV. Model

BERT(Bidirectional Encoder Representations from Transformers) is state-of-art language representation model which designed to pre-train deep bidirectional representations and create connection between left and right context in all network. By doing that, the BERT model that we pre-train can be tuned with just one additional classification layer to variety of tasks such as; question answering, text classification, machine translation with state-of-art performances.

BERT is trained with huge amount of unlabeled text data including all Wikipedia context(2.5 million words) and Book Corpus(800 million words). Since BERT models are pre-trained with large text corpus, understanding of our model about the language become deeper. So It's make BERT useful for all NLP problems. To solve our problem we use pretrained BERT[6] model and add one dropout and one dense layer which appropriate for our classification task. BERT is a multilingual transformer based model that has achieved state-of-the-art results on various NLP tasks. BERT is a bidirectional model that is based on the transformer architecture (Fig 4), it replaces the sequential nature of RNN (LSTM GRU) with a much faster Attention-based approach.Transformers consists two parts; encoder and decoder.Encoder consist of 6 identical layers and each layer has two sub layer. The decoder has also 6 identical layers and each layer has three sub layer. In BERT[6] paper they introduce two distinct model according to the layer size. BERT-base with 110m parameters and 12 layers and BERT-large with 24 layers and 340m parameters. We fine tuned BERT-base model to our problem with adding classifier layer to end of the BERT-base model and train it with the stackoverflow data for few epochs we are going to mention details about hyperparameters in next sections.

Hyperparameter	Value
Batch Size	64
Learning Rate	3e-5
Sequence Length	128
Loss Function	SparseCategoricalCrossentropy
Optimizer	Adam
Epoch	3

TABLE I: Hyperparameters used in base model

You can see representation of our architecture at Fig. 4 and hyperparameters that we used in our model at table I and number of parameters at table II. I am going to summarize our architecture; As an input we have encoded stack over flow questions. Then it's fed into TFBertMainLayer this

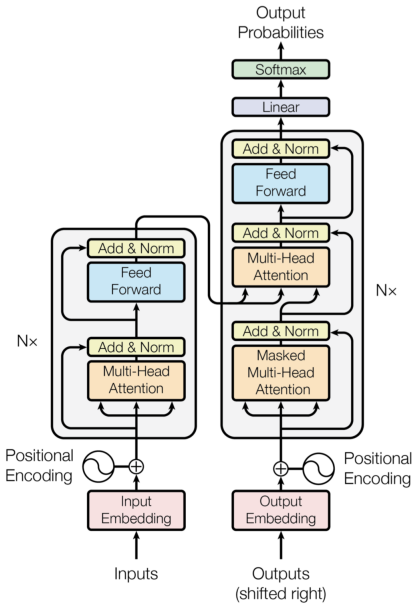

Fig. 3: The Transformer model arcihtecture taken from [7]

is absraction bert architecture that we used. It consist of 12-layer, 768-hidden, 12-heads. After that we have dropout and dense layers to complete our classification task and finally get the logits for each classes and finish the process. Tensorflow-2.0 is used at implementation of model

Layer	Number Of Parameters
Bert Layer	108310272
Dropout Layer	0
Dense Layer	19225

TABLE II: Trainable parameters in each layer

V. Evaluation Metric

As an evaluation metric we picked accuracy and top-k accuracy. We calculate accuracy as (correctly predicted tags/all tags). Calculation similar for top-k accuracy but instead of taking maximum probability that softmax produced, take top-k probability and if one of them match with actual tag we count as correct.

VI. Hyperparameters

Hyperparameters are the parameters that can not learned with training process. Hyperparameters defined before training process and remain unchanged. It is important to optimal hyperparameter set for your model to have optimal results from it so in this section we are going to share different experiments and it's results with different hyperparameters set. The hyperparameters that we changed are , sequence length, batch size, learning rate and number of epochs. At first we try to find optimal epoch number for model. To

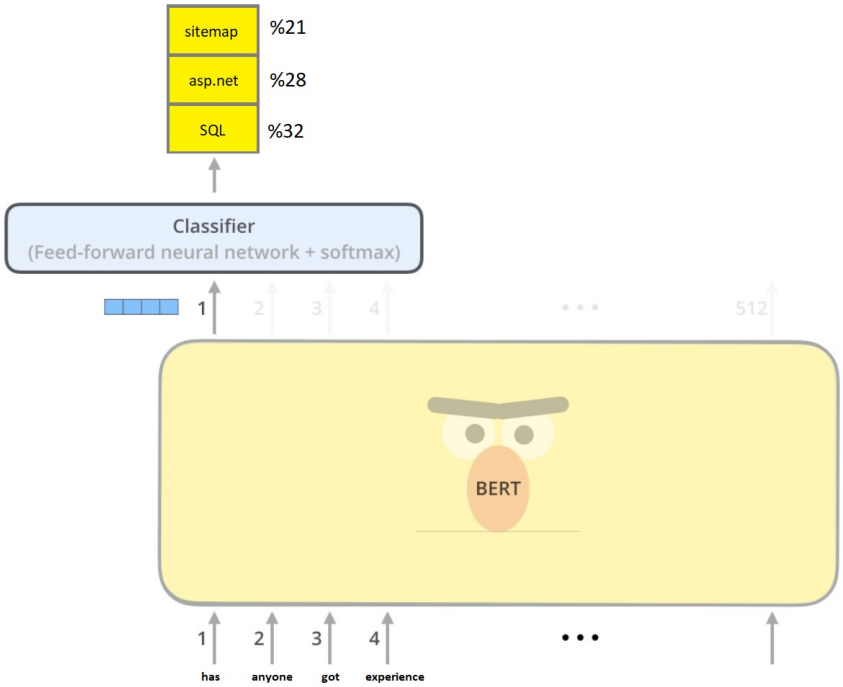

Fig. 4: Tag Classification Architecture

do that, We train the model with 10 epochs and plot out training loss and validation accuracy (Fig. 5) and when we looking it we can see that our model converge at 6'th epoch so we choose epoch number as 6. In further experiments we keep epoch number as 6. You can observe the effect of hyperparameters effect on the accuracy that we get from table **??**. We choose the hyper parameters that optimize our model's accuracy. Epoch=6 , learning rate = 5e-5, batch size = 32, sequence length = 128.

Batch Size	LearningRate	SequenceLength	Top4Acc
16	3e-5	128	0.8315
32	3e-5	128	0.8903
32	3e-5	64	0.8764
32	3e-5	32	0.8465
32	1e-5	128	0.8485
32	2e-5	128	0.8845
32	4e-5	128	0.8917
32	5e-5	128	0.9015

TABLE III: Hyperparameter table

VII. EXPERIMENTAL RESULTS

In this section we are going to share some stack over flow question tag predictions that our model perform and their correct labels with table IV and accuracy that we get compared to base model. As you can see from table we

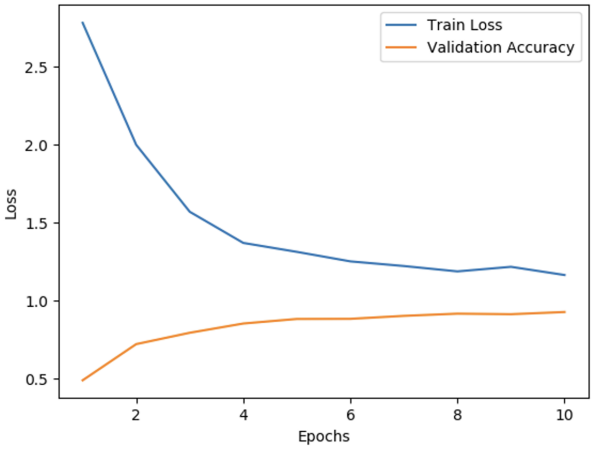

Fig. 5: Epoch-loss graph

improve BERT-base model with optimizing hyperparameters.Our model mostly find correct labels about questions our main problem is predicting correct number of labels.

Base-Model Top-4-Accuracy	Our Model Top-4 Accuracy
0.8913	0.9015

Question	Correct Tags	Predicted Tags
the difference between datagrid gridview aspnet	asp.net	asp.net, .net
how i java webstart multiple dependent native libraries	java	java
how do you secure databaseyml	ruby-on-rails	ruby-on-rails
rofile rail controller action	ruby-on-rails, ruby	ruby-on-rails, ruby
alternative architectural approach javascript client code	javascript	javascript,jquery
aspnet control cannot reference code-behind visual studio 2008	'c#', 'asp.net'	c#, asp.net,.net
how find file exist c# net	c#, .net	c#, .net
how i write iphone app entirely javascript without make web app	javascript, iphone, objective-c	iphone, objective-c
why learn perl python ruby company use c++ c# java application language	c#, java, python, ruby	[c#, java, c++, c
good asp.net c# apps	c#, asp.net, css	c#, asp.net, .net
save restore form position size	.net	.net
what easiest non-memory intensive way output xml python	python	python
how i detect use php machine oracle oci8 andor	php	php
illustrating usage volatile keyword c#	c#,.net	c#,.net
running subversion apache mod python	python	python
do ocunit ocmock work iphone sdk	iphone, objective-c	iphone, objective-c

TABLE IV: Some questions from our dataset and our predictions

VIII. CONCLUSION

In conclusion, We have chance to find work with state-of-art language representation BERT and tuned it to our stackoverflow question tagging problem. As we observe BERT performed really well on this problem.

REFERENCES

[1] Johnson, Rie, and Tong Zhang. "Deep pyramid convolutional neural networks for text categorization." Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2017.

[2] Mukalov, Pavlo, et al. "Development of System for Auto-Tagging Articles, Based on Neural Network." COLINS. 2019.

[3] Rekha, V. Smrithi, N. Divya, and P. Sivakumar Bagavathi. "A hybrid auto-tagging system for stackoverflow forum questions." Proceedings of the 2014 International Conference on Interdisciplinary Advances in Applied Computing. 2014.

[4] Saha, Avigit K., Ripon K. Saha, and Kevin A. Schneider. "A discriminative model approach for suggesting tags automatically for stack overflow questions." 2013 10th Working Conference on Mining Software Repositories (MSR). IEEE, 2013.

[5] Stanley, Clayton, and Michael D. Byrne. "Predicting tags for stackoverflow posts." Proceedings of ICCM. Vol. 2013. 2013.

[6] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

[7] Attention Is All You Need