Automatic Tagging of Stack Overflow Questions Using BERT Word Embeddings and Deep Learning

Bikash Thapa*, Alexandra Ballow*, Hannah Senediak*, Bonita Sharif†, Alina Lazar*
*Youngstown State University, Youngstown, Ohio, USA
†University of Nebraska - Lincoln, Lincoln, Nebraska, USA
{bthapa01,alballow,hesenediak}@student.ysu.edu,bsharif@unl.edu,alazar@ysu.edu

ABSTRACT

Question-and-answer (QA) websites like Stack Overflow require users to attach up to five tags when they submit a question. However, users may assign tags that are not relevant to the question. A better approach would be to recommend to users the most appropriate tags for their question and let them choose. The goal of this project is to combine newly developed natural language representations together with deep learning algorithms to improve the prediction accuracy of tags for Stack Overflow questions. We used word representations generated by word2vec and a Convolutional Neural Network (CNN).

CCS CONCEPTS

• **Computing methodologies → Dimensionality reduction and manifold learning**; *Cluster analysis*; *Feature selection*;

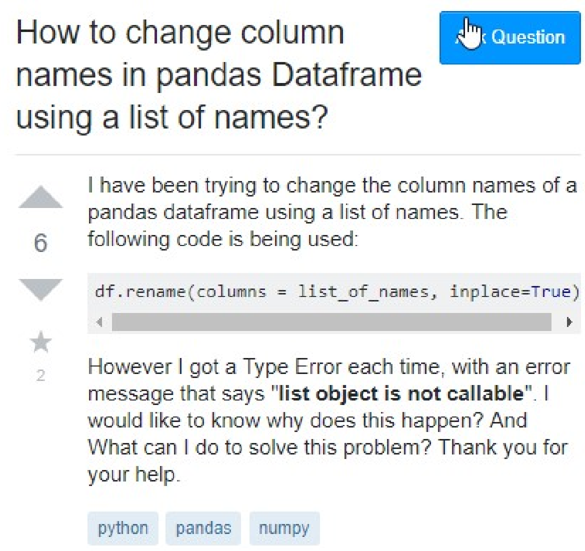
KEYWORDS

Classification, Deep Learning, Word Embedding

ACM Reference format:
Bikash Thapa*, Alexandra Ballow*, Hannah Senediak*, Bonita Sharif†, Alina Lazar*. 2019. Automatic Tagging of Stack Overflow Questions Using BERT Word Embeddings and Deep Learning. In *Proceedings of Grace Hopper Celebration, Orlando, FL. USA, October 2019 (GHC 2019),* 3 pages.
https://doi.org/10.475/123_4

1 INTRODUCTION

The Stack Overflow website is a growing resource among coders everywhere. Software developers use it to post questions and answers related to programming and computer science problems. Over the years the website has slowly evolved into a large free repository of knowledge. Questions such as seeking input on some efficient and time-saving methods of coding a particular program, getting help on solving various bottlenecks in coding are commonly seen. Given the availability and size of the dataset many researchers from fields such as information retrieval, text mining and machine learning have been working with this textual dataset to solve different problems [1]. Because it is used so much, usability and especially searching is a great concern. One way to help users find what they are looking for is to use tags. Currently, tags are created by users, however this can be tedious and can produce inaccuracies. Automatically suggesting tags to users could stop these problems [10].



How to change column names in pandas Dataframe using a list of names?

I have been trying to change the column names of a pandas dataframe using a list of names. The following code is being used:

```
df.rename(columns = list_of_names, inplace=True)
```

However I got a Type Error each time, with an error message that says "**list object is not callable**". I would like to know why does this happen? And What can I do to solve this problem? Thank you for your help.

python pandas numpy

Figure 1: Stack Overflow Sample Question with Three Tags

Tags, as shown in Figure 1, have been used to identify online postings not only by Stack Overflow but also by most social media sites including Facebook, Twitter and Instagram. Tags and hashtags are very useful and can help with query-based like searches to retrieve information faster and more accurately. Especially, in the technical domain for "question and answer" sites such as Stack Overflow or Quora is it essential that the tags chosen during the post creation provide a correct representation of the submitted post.

When users submit questions on Stack Overflow they need to attach at least one and up to five tags to their question. These tags broadly identify the programming language talked about, the problem type in discussion and maybe some other fine-grained categories the question belongs to. The tags associated to each question help with information retrieval or user queries. For example, it may be very useful when users try to identify duplicate questions or related questions to a particular problem. Several approaches [2, 6, 9, 10] for automatically generating tags from short questions or text have been recently developed. The Stack Overflow dataset is perfect for this problem as it provides the ground truth (as the author of the question adds these tags). The problem with this

GHC 2019, October 2019, Orlando, FL. USA

ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/123_4

multi-label, multi-class classification approach is that it is hard to reach good accuracy.

Our main goal for this project is to use newly developed natural language representations together with deep learning algorithms to improve the prediction accuracy for automatically generating Stack Overflow question tags. Our main research question is:

- What are the best deep learning CCN architectures and hyperparameters setups that can be successfully used to make tag predictions?

The main benefits of this approach are 1) to automatically identify the keywords in short text and 2) to determine what the best NLP and code representations and deep learning architectures and parameters.

2 RELATED WORK

In the last couple of years many deep learning algorithms were developed and provided better performance especially for computer vision problems like object classification in still images. Deep learning methods are considered special sub-category of machine learning methods, and part of the artificial intelligence field. They were developed based on the classical neural networks idea and are capable of learning data representations very well at multiple levels of abstractions, given the multiple layers of the network structure. Deep learning has been successfully applied to many fields as computer vision, speech recognition and translation, natural language processing and many others. For this project, deep learning can bring several benefits in learning natural language representations in order to improve the prediction of tags.

Collobert and Weston [3] were the first to use deep learning for six natural language processing (NLP) tasks that can be seen as tasks assigning labels to words. They applied multitask learning using a single convolutional neural network architecture for part-of-speech tagging and other several tasks, given input sentences. The feature extraction step is integrated into the deep learning architecture which improves the generalization performance.

The research conducted throughout this paper was influenced by many different approaches of past successful models. Looking at how different types of models, like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have been applied to sentence classification in the past, gave a starting point for this project. CNNs are becoming standard models used for various tasks because of their recent success. Models, like the ones proposed by Yoon Kim [5], feed a representation of a sentence through a convolution operator. Those results are then applied through a max pooling function and varying filters to obtain a penultimate layer. Finally, the penultimate layer in passed through a softmax layer to find a probability distribution. Yoon Kim took 4 variations of this basics model to determine the optimal model construction. This study concluded that unsupervised pre-training is an important part of natural language processing.

Knowing which type of model to use, with so many options, can be challenging. Liang investigated three types of multi-label text classification to see which performed the best when modeling a 3 million question database. After looking at a basic fastText model, a CNN based model and a Hierarchical structure based model, it is determined that complex models, like the CNN and HAN perform better than simple models. In addition, they studied a model which combined the CNN and HAN model, which performed better yet. All models studied were able make accurate predictions without using labeled data.

3 DATASETS AND STATISTICS

Over the years the website has slowly evolved into a large free repository of knowledge. Currently, the site receives around 8000 questions per day, and includes 16 million questions, 24 million answers and 66 million comments all available to download in a data dump collection. The data is made publically under the Creative Commons cc-by-sa 3.0 license. Given the availability and size of the dataset many researchers from fields such as information retrieval, text mining and machine learning have been working with this textual dataset to solve different problems [1].

The samples in the training set consists of questions from an existing Stack Overflow dataset extracted from the Stack Overflow data dump. This dataset The dataset includes 85,085 questions and 244,228 unique tags. In Figure 2 we show the top-10 most common tags and their counts for our Stack Overflow dataset. "Python" is shown as the most common tag.

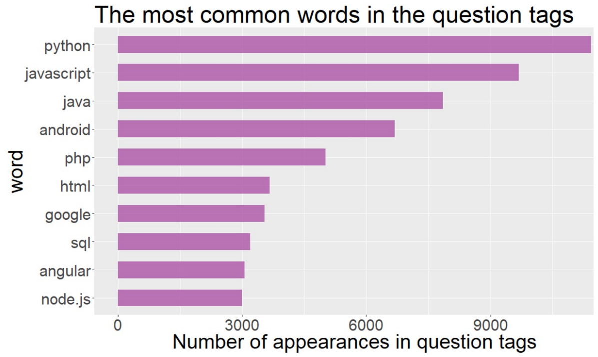


Figure 2: Top-10 Common Tags in the Stack Overflow Dataset

4 METHODS

To create computer generated tags, it is important to understand the information being discussed in the questions and answers posted. To do this, the first step would be to find common words and phrases in the data. Then, to ensure only the important words are kept, stop words (`not`, `the`, `or`, `and`) are removed. Next, using the tokenization process, the title and the body of each question are split into tokens or words. However, there are nuances which are lost in this simple approach. To see a more complete view of the questions and answers it is important to look at them as a whole and as individual elements. Using deep learning, specifically convolutional neural networks [4, 7], it is possible to see themes in the data not implied by the word distribution. Things like the coding platform are vastly important but might only be mentioned once at the beginning. Deep learning makes these discoveries possible.

The problem of tag prediction can be considered a multi-label classification problem. A deep learning algorithm will be trained

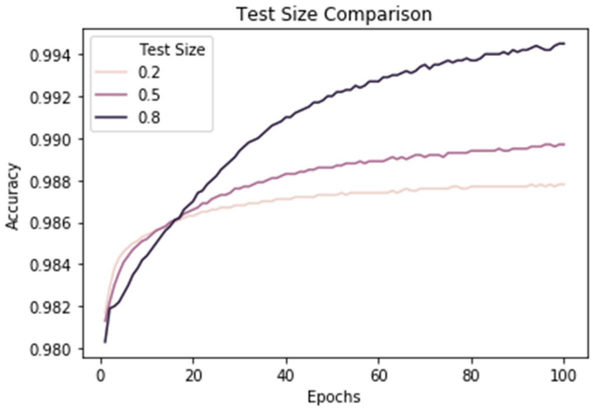


Figure 3: Testing Different Test Dataset Sizes

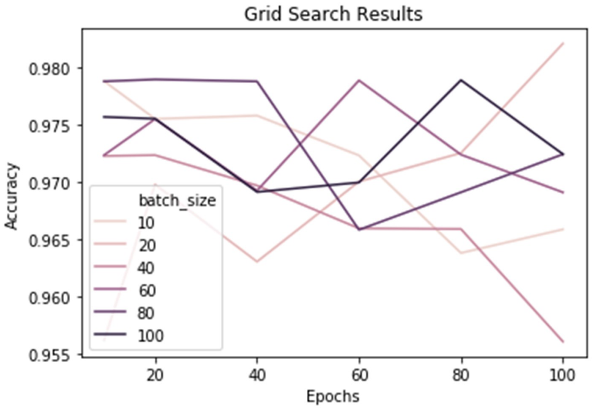


Figure 4: Testing Different Parameter Combinations

using the question's title, sentences extracted from the question's text, and lines of code in order to predict the top five tags.

The first step in our approach will consist in representing the Stack Overflow posts as word embeddings. These representations are generally generated by creating a matrix of what words are similar in a large corpus. Several good pre-trained word embeddings have been recently developed. The most simple and popular model word2vec, also called the skip-gram model, learns word vectors using a neural network predictive model [8]. The second step consists of running experiments using the word and code embeddings generated during the first step as inputs for multiple setups of convolutional neural networks (CNN) for the classification task. CNNs, which were originally invented for computer vision problems, build on layers for convolving filters and have proved to improve on other methods when it comes to solve document classification problems [5]. However, these models require researchers to specify an exact model architecture and set precise values for hyperparameters, including the filter region size and the regularization parameters. This can be accomplished only by designing and running multiple sets of experiments.

Overall these techniques provide several different ways for short texts and programming codes to be processed and classified based on a set of training data. To train and test these models, the dataset will be divided into a training set and a testing set using seveal ratios. From each question we will predict five tag labels and we will calculate the prediction accuracy. For this project we are using only the most common 100 tags.

5 RESULTS

First, the dataset was split into train and test data with different proportions of train and test data and the results were compared as shown in Figure 3. We can conclude that the accuracy increase is correlated with the training dataset size. A test size of 0.2 has a maximum accuracy of 98.8%, while a test size of 0.5 has a maximum accuracy of 99.0%. Finally, a test size of 0.8 increases the maximum accuracy to 99.5%. Also, a grid search was performed to tune the hyperparameters. The best accuracy result as shown in Figure 4 was obtained when the epochs was set to 100 and batch size was set to 20. The accuracy for these parameters was 98.2%.

6 CONCLUSIONS

A process for predicting tags for unseen questions has been developed. We used both the title and the body of the Stack Overflow questions to generate word embeddings based on word2vec. Using this list of words it is possible to generate statistics such as the longest and shortest passages as well as the vitally important most used words. In addition to the statistics, a convolutional neural network (CNN) model was created and successfully run on this dataset. The encoded text is then input into the CNN. The parameters are then tuned on the data to maximize model accuracy up to 99.5%.

In future we will explore other word embeddings and CNN configurations for this multi-label problem. We also plan to compare the results obtained with CNN with results from classical classification algorithms such as: linear support vector machines (SVM), NaÃŕve Bayes, and random forests.

REFERENCES

[1] Arshad Ahmad, Chong Feng, Shi Ge, and Abdallah Yousif. 2018. A survey on mining stack overflow: question and answering (Q&A) community. *Data Technologies and Applications* 52, 2 (2018), 190–247.

[2] Stefanie Beyer and Martin Pinzger. 2015. Synonym suggestion for tags on stack overflow. In *Proceedings of the 2015 IEEE 23rd International Conference on Program Comprehension*. IEEE Press, 94–103.

[3] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*. ACM, 160–167.

[4] Yaser Keneshloo, Tian Shi, Naren Ramakrishnan, and Chandan K. Reddy. 2018. Deep Reinforcement Learning For Sequence to Sequence Models. *CoRR* abs/1805.09461 (2018). arXiv:1805.09461 http://arxiv.org/abs/1805.09461

[5] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014).

[6] Marek Lipczak and Evangelos Milios. 2010. Learning in efficient tag recommendation. In *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 167–174.

[7] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent Neural Network for Text Classification with Multi-Task Learning. *CoRR* abs/1605.05101 (2016). arXiv:1605.05101 http://arxiv.org/abs/1605.05101

[8] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[9] Avigit K. Saha, Ripon K. Saha, and Kevin A. Schneider. 2013. A discriminative model approach for suggesting tags automatically for stack overflow questions. In *Mining Software Repositories (MSR), 2013 10th IEEE Working Conference on*. IEEE, 73–76.

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