Binary and Multi-Class Classification of the Status of Questions on StackOverflow.com

By Michael Phaneuf and Dylan Edwards

I. Introduction

Stack overflow is one of the most influential community sites in the software engineering industry. There are more than 11 million registered users and over 6500 questions posted every day. With this massive amount of content being generated by the users moderation becomes a significant undertaking, and users can often become confused and frustrated as to why their questions are being closed. This leads to many other posts asking why their original post had been closed. Due to the influence and size of the stack overflow community these types of interactions could hurt the user, and make the stack overflow community less inclusive to new engineers.

In our project we want to create a tool that can be used to grade the quality of stack overflow questions. Closing a question on stack overflow is a serious measure that is taken against questions that are not deemed to be up to the standards of the community due to their lack of clarity, content or other reasons deemed relevant by the moderator. However, the details of the reasons for why a question was closed may be unknown and especially for new users this can be frustrating and make them refrain from interacting with the community in the future. This can often manifest itself as arguments in the comments or between users. For our project we will create a set of classifiers that will be able to determine the quality of a question and if that question were likely to be closed if it were uploaded to the stack overflow forum.

II. Related Work

A. *Why Will My Question Be Closed? NLP-Based Pre-Submission Predictions of Question Closing Reasons on Stack Overflow*[1]

In the past Researchers from MTA-SZTE Research Group on Artificial Intelligence at the University of Szeged, Hungary had published a paper where they created a series of models using data sourced from the Stack overflow Data dump to help classify posts in two categories. The first was the classification between the open versus closed questions and the second series a five-class classification predicting the different closing reasons (off-topic, unclear, too broad, opinion-based).

B. *Why, When, and What: Analyzing Stack Overflow Questions by Topic, Type, and Code*[2]

Miltiadis Allamanis and Charles Sutton used topic modeling to associate programming concepts and identifiers with specific types of questions. They trained three topic models using Latent Dirichlet Allocation, which is a generative model for describing documents as mixtures of topics, with

[1] https://www.researchgate.net/profile/Laszlo-Toth-12/publication/339447769_Why_Will_My_Question_Be_Closed_NLP-Based_Pre-Submission_Predictions_of_Question_Closing_Reasons_on_Stack_Overflow/links/5ee08e2592851cf1386f578c/Why-Will-My-Question-Be-Closed-NLP-Based-Pre-Submission-Predictions-of-Question-Closing-Reasons-on-Stack-Overflow.pdf

[2] https://homepages.inf.ed.ac.uk/csutton/publications/msrCh2013.pdf

each topic containing frequently co-occurring words. They were able to show that the types of questions asked on Stack Overflow do not vary across programming languages. They also presented a method for identifying what question types were mostly associated with particular programming constructs/identifiers.

C. Mining Successful Answers in Stack Overflow[3]

Researchers at Università degli Studi di Bari in Italy investigated how Stack Overflow users can increase the chance of getting their answer accepted. They identified some key success factors for answers such as presentation quality, affect, time, and reputation. They used a logistic regression model in order to compare the significance of different factors. They found that code snippets and reputation were very important factors in getting an answer accepted. They also found that comment positive and negative sentiment were somewhat important.

III. METHODOLOGY

A. Dataset

Our dataset comes from kaggle.com and was uploaded by Stack Overflow.[4] It contains post text and associated metadata at the time of post creation. The state of the post as of July 31st is also included. It contains the following fields (not in this order): Input, PostCreationDate, OwnerUserId, OwnerCreationDate, ReputationAtPostCreation, OwnerUndeletedAnswerCountAtPostTime, Title, BodyMarkdown, Tag1, Tag2, Tag3, Tag4, Tag5, Output, OpenStatus, Additional Data, PostId, PostClosedDate.

B. Baseline Binary Logistic Regression

The first model we construct is our baseline logistic regression model for binary classification. With this model we hope to predict whether a question will be "closed" or not on StackOverflow. We first use the NLTK porter stemmer on our dataset and remove stop words. We then remove punctuation and replace contractions. We then use pandas to clean the data and combine all text into one column while converting open status to an integer, 0 to identify closed questions and 1 to identify open questions. Next we run the SKLearn logistic regression model with bigram BOW vectorizer to create the model's features.

C. BERT Binary

Next we built another model for binary classification, this time utilizing BERT word embeddings. We use hugging face, an open source library for transformer based models. First we load all data using pandas and import the hugging face tokenizer. We use this to tokenize and pre-process the data by concatenating all text columns and converting "open status" to an integer just as we did with our previous model. We use hugging face's pre-trained BERT model and fine tune it to our training data. Finally we use the hugging face trainer module to define the training data, eval data and metrics that will be used to evaluate our model to define the training data, evaluation data and metrics that will be used to evaluate our model such as precision and recall.

D. Baseline Multi-class Logistic Regression

The third model we built was our baseline logistic regression model for multi-class classification. With our multi-class models we hope to predict both the "open status" of questions and the

[3] https://www.researchgate.net/profile/Nicole-Novielli/publication/276353348_Mining_Successful_Answers_in_Stack_Overflow/links/55575fb508ae980ca60e1bcb/Mining-Successful-Answers-in-Stack-Overflow.pdf

[4] https://www.kaggle.com/c/predict-closed-questions-on-stack-overflow/data

reason for which closed questions may have been closed. This would allow us to take potential new questions and make suggestions on how they can be improved. To construct this model we follow the same pre-processing steps as the previous two models in regards to our data. We then turn the “category” column into an integer between 0-4 to represent the reason for a question being closed. We remove all “open” questions. We then run the SKLearn logistic regression model with the multi-class option using a bigram BOW vectorizer to create the model features. We get metrics such as F1 score using SKLearn metrics.

E. BERT Multi-class

Our final model was another multi-class classification model, but this time using BERT word embeddings. We use hugging face, an open source library for transformer based models. We again follow the same pre-processing steps in the previous models and use the hugging face auto tokenizer to tokenize our data. For this model we use pytorch to fine-tune a pertained BERT model because this allows for extra configurability, which makes it easier to use this model for multi-class classification. This allows us to tell the last layer of the model to expect four features instead of just two for number of classes. We then use the pytorch trainer module to define the training data, evaluation data, and metrics we use to evaluate our model such as precision and recall.

IV. Results and Analysis

First we compare the results of our binary classification models using precision.

	Logistic Regression	**BERT**
Precision	0.98	0.975

Clearly both models perform very well. After further analysis this makes sense logically. Classifying whether a question will be closed or not should be a relatively easy task as this is not very subjective. A question that get closed and removed from the site clearly has something wrong with it and is drastically different enough from a quality question that it gets completely locked. There is no gain to making the model more complex by using BERT word embeddings since the baseline logistic regression model is about as good as it can get.

Next we compare our multi-class classification models using F1 score.

	Logistic Regression	**BERT**
F1 Score (Micro)	0.504	0.535
F1 Score (Macro)	0.378	0.374

These models both perform much worse than the binary classification models. Following the same logic as for why binary classification is so easy, it makes sense that multi-class will be much more difficult. The reason a question was closed or is a “bad” questions is much more subjective than just whether a question should be closed or not. Different people may have different views on how a question could be made better. The BERT model did have a slightly better micro F1 score, however both models had very similar performance.

V. Threats to Validity

There are a few threats to the validity of our results. There are much fewer “closed” questions than open ones in the dataset, so our training data was limited in size, which makes our results somewhat less reliable. We also only ran our models using the same dataset. To truly test the validity of our results we would need to collect additional data and replicate our results.

Essentially, our results would be made much stronger through replication.

VI. Conclusion

We set out to create a model that could predict whether a question on Stack Overflow would be "closed" and why it might be "closed". We built four total models and found that determining the "open status" of a question is much easier than determining why a closed question may have been closed. This makes sense since whether a question is bad or not is much less subjective than the reason for it being bad. Ultimately we found that our baseline models performed just as well as our BERT models, which is normal. This tends to happen if your data is well structured or if the differences between classes is easily identifiable. This is the case for us since we have a very well organized dataset to work with and the differences between closed and open questions is easy to identify. There is a great amount of room for additional research and improvement upon our results. A more extensive dataset of "closed" questions could be collected and our multi-class model could be improved upon. There is no doubt that using some additional, or different, machine learning techniques or models could result in much better performance. Ultimately a model that could accurately determine the quality of a question and how poor questions could be improved would be a great help to the Stack Overflow community.