Stack Overflow Question Quality Classification Using Deep Learning Techniques

Shinn Taniya and **Cindy Lay**
Department of Computer Science, Harvey Mudd College, Claremont, CA
(staniya, clay)@g.hmc.edu

Abstract

Community Question Answering (CQA) forums like Stack Overflow play an important role to support developers of all experience levels. Thus, it is essential to establish an automatic quality control metric to filter high-quality questions better than current manual moderation methods. In this paper, we apply different natural language processing and deep learning techniques to classify high-quality questions based on linguistic features and assigned tags. Using random forests, we evaluate question features most influential to the quality of the posts. In accordance with our findings, we conclude that an approach that combines deep learning and natural language processing methods serves as an accurate solution to the automated quality classification problem for Stack Overflow. We found that bi-directional LSTM and CNN had higher accuracies than BERT although BERT had higher precision and recall. Furthermore, we found that when evaluating the dataset using sentiment analysis, Neural Network Classifcation had an accuracy of about 46% while our Random Forest Classifier had an accuracy of about 51% and found tags to be the most influential feature to predicting post quality.

1 Introduction

Community Question Answering (CQA) forums such as Stack Overflow have become crucial in supporting the daily tasks of software engineers. Regardless of one's level of programmatic experience, Stack Overflow is a helpful resource as it provides effective, practical, and relevant support.

However, given that CQA websites are structured such that anyone can create a post, there is a wide variety of question quality. To preserve the professionality and quality of Stack Overflow, it is essential to consider a proper quality control metric in which the questions posted on the website are relevant, unambiguous, and comprehensible. (Tóth et al., 2019) Submitted questions should be related to specific development issues or methods. More subjective questions or questions that lead to endless, opinionated discussions should not be supported. Furthermore, questions that fail to be clear or concise should also be avoided.

Although Stack Overflow does have a quality control infrastructure in which posts that do not meet established criteria are closed or deleted by moderators or experienced members with distinguished privileges, given that the posting frequency at Stack Overflow has an average of over 8000 new questions, it is impractical to manually review the quality of every post. (Tóth et al., 2019)

Hence, automating the quality-based classification of Stack Overflow questions has become the interest of researchers in recent years [(Agichtein et al., 2008); (Li et al., 2012); (Barua et al., 2014); (Bazelli et al., 2013); (Tóth et al., 2019); (Kavuk and Tosun, 2020)]. The focus of our present study is the quality-based classification of Stack Overflow questions based on their linguistic characteristics and the tags associated with each post. The study will involve two parts: the first is to apply different text classification techniques (BERT, bi-directional LSTM, CNN) to study their performance using solely the raw body text of posts to predict question quality. The second is to continue off of work done by Bazelli et al. (Bazelli et al., 2013) that incorporates sentiment analysis on the raw text of the data to represent text data numerically. Then, neural net classification and Random forest classification are applied and Random Forest is used to rank the three features: Title, Body, Tag in terms of their feature importance.

2 Related Work

With the proliferation of deep learning techniques ranging from neural networks to pre-trained language models, there exists a multitude of text classification research done for quality analysis and classification of Stack Overflow questions. However, many of the previous research implemented basic machine learning algorithms or applied complex pre-trained language models such as BERT to investigate the performance of a single model to classify question quality using only linguistic characteristics. For example, in understanding how to apply deep learning techniques to analyze CQA forums, Tóth et al.'s (Tóth et al., 2019) research was educational as it (Tóth et al., 2019) focused on the quality-based, binary classification of questions uploaded to Stack Overflow based on their linguistic characteristics. Toth et al. reports that the semantics of the posts can be caught using a specific Doc2Vec representation. This way, the classification can be performed solely on textual information.

We could cite hundreds of other text classification work which we drew inspiration from including the work done by Agichtein et al. (Agichtein et al., 2008) in investigating methods for exploiting community feedback of CQA websites to automatically identify high-quality content and the work done by Barua et al. (Barua et al., 2014) in applying Latent Dirichlet Allocation (LDA) to analyze the main topics present in Stack Overflow developer discussions and to gain insights into the development community. To further continue their studies in assessing the performances of various deep learning techniques to understand the nature of questions on Stack Overflow, we will try approaches such as BERT, bi-directional LSTM, and CNN and compare the results using statistical measures.

However, to also try a unique approach, we focused on this dataset because we hoped to tackle quality-based classification through investigating the feature importances of unique attributes that are not commonly included in Stack Overflow datasets.

For example, our dataset allowed us to further investigate Bazelli et al.'s (Bazelli et al., 2013) experiment which used Linguistic Inquiry and Word Count (LIWC) to determine Stack Overflow developers' personality traits by categorizing them based on their reputation. Working off of their conclusion that authors who had posts frequently up-voted expressed significantly less negative sentiment compared to authors of down-voted posts, our work applies sentiment analysis on both the title and body of the posts so that we can numerically categorize bodies of text.

Then, by performing both neural net classification and Random forest classification to analyze feature importance concerning post quality, we sought to validate the work done by Kavuk et al. (Kavuk and Tosun, 2020) which used Synthetic Minority Oversampling Technique (SMOTE) to predict tags on questions. They found that often users incorrectly labeled question tags and questions that were correctly labeled were more likely to be answered. Hence, when generating the rank of the feature importances, we could compare our results to Kavuk et al.'s findings to further discuss if there is an evident correlation between question tags and the question quality.

3 Dataset

The Kaggle Dataset contains 60,000 Stack Overflow questions from 2016-2020, collected from the Stack Overflow website by a data scientist at Kertoft. The dataset consists of the unique question ID, a question title, the main body of the question, tags representing keywords in the question, the creation date of the question as well as the class/ label of the question. The label itself consists of three classes: HQ: High-quality posts without a single edit. LQ_EDIT: Low-quality posts with a negative score, and multiple community edits. However, they remain open after those changes. LQ_CLOSE: Low-quality posts that were closed by the community without a single edit.

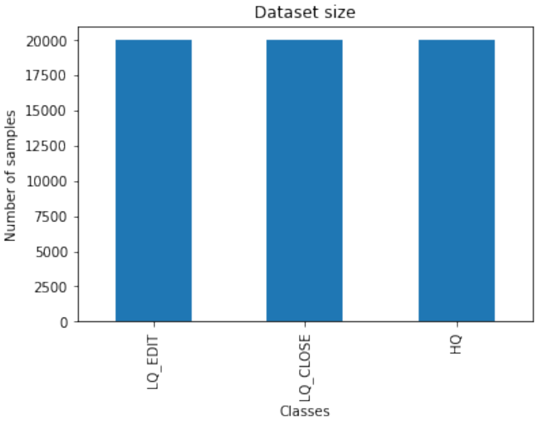


Figure 1: Label Distribution

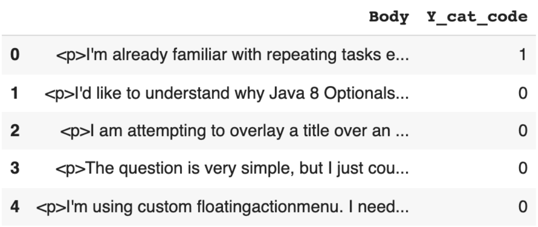
As shown in Figure 1, the dataset is perfectly balanced with each of the three labels having 20,000 samples each.

3.1 Pre-processing

Although Kaggle had already pre-processed the data, for this project, we further processed the data to match each of our two experimental approaches. For the first experimental approach, the title and body are combined into a single string for training purposes. Furthermore, the tags are not important in performing text classification to identify linguistic characteristics so they are removed. The HTML tags are also removed since they don't show any semantic importance to the data.

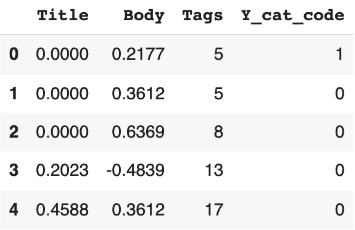
For the second experimental approach, to create a numerical representation of the tags, each unique tag is mapped to an integer. Then, on the assumption that the first tag is the most relevant within the tag set, all other tags other than the first are removed. Furthermore, continuing off work done by Bazelli et al. (Bazelli et al., 2013) that incorporates sentiment analysis scoring to assess post quality, we applied NLTK's Pre-Trained Sentiment Analyzer: Valence Aware Dictionary and Sentiment Reasoner to return the compound sentiment score on the raw title and body texts. The compound score ranges from -1 to 1, where a larger value (signifying growth in conventional mathematical knowledge) represents a more positive sentiment.

For both approaches, the quality label "Y" is numerically categorized under the column “Y_cat_code". The question ID and creation date were effectively removed based on the assumption that they do not influence the quality of the question. Since the text data can not be directly passed to apply deep learning techniques, we create a word embedding for the data by setting a vocabulary and sequence limit. The text is tokenized and converted into sequences of integers that the model can



	Body	Y_cat_code
0	<p>I'm already familiar with repeating tasks e...	1
1	<p>I'd like to understand why Java 8 Optionals...	0
2	<p>I am attempting to overlay a title over an ...	0
3	<p>The question is very simple, but I just cou...	0
4	<p>I'm using custom floatingactionmenu. I need...	0

Figure 2: Linguistic Characteristics Dataset



	Title	Body	Tags	Y_cat_code
0	0.0000	0.2177	5	1
1	0.0000	0.3612	5	0
2	0.0000	0.6369	8	0
3	0.2023	-0.4839	13	0
4	0.4588	0.3612	17	0

Figure 3: Sentiment Analysis Scoring and Tags Dataset

interpret. Reference Figure 2 for experiment approach 1 and Figure 3 for experiment approach 2.

4 Methods

To best determine question quality we decided on two different methods. Our first approach was text classification using Bi-directional Encoder Representation from Transformers (BERT) (Devlin et al., 2018), as well as Bi-directional Long-Short Term Memory (BLSTM) and Convolutional Neural Networks (CNN). Our second method incorporates sentiment analysis on text attributes of the data to transform the data into numerical categories such that neural net classification and Random forest classification can be applied. Furthermore, through the second approach, Random Forests will be used to generate a ranking for dataset attributes based on their feature importance. Comparing these two approaches, we investigate which most effectively classifies high-quality stack overflow questions.

4.1 Linguistic Characteristics Based Approach

For our first approach, for the deep learning approaches to be compared fairly, we set up the models so that the number of trainable parameters is close to each other. The following is the models studied in this project and how they were set up:

1. BERT: Using our pre-processed dataset, we leverage that each post is linked to a range of post qualities. Since each row of posts holds a different form from the text source, we need to clean each part of the data to apply a proper <start> and <end> portion to note the post text. Importing version two of the pre-trained uncased BERT Model on TensorFlow Hub, we tokenized the words using the official TensorFlow BERT model asset. Provided that each line of the dataset is composed of the raw body text and its label, we process the text to BERT input features: Input Word Ids, Input Masks, Segment Ids. The output of BERT for our classification task will be a pooled output of shape [batch_size, 768] with representations for the entire input sequences. Configuration parameters: maximum length of input sequences is 150 tokens, training batch size of 32 samples, and an adam optimizer with a learning rate of 2e-5. Although we wanted to increase the maximum input sequence length to match the others, BERT without training has over 110 million parameters, resulting in our lack of memory resources.

2. Bi-directional LSTM: There are two bi-directional LSTM layers stacked and the model consists of an Embedding layer as its input. The LSTM layers use around 64 hidden neurons whereas the first LSTM layers return a sequence that can be directly fed into the second layer. The final layer is a dense layer using a soft-max activation function to ensure that the output is in a probabilistic format. Configuration parameters: Adam optimizer with a learning rate of 1e-4, training batch size of 32 samples, the maximum length of input sequences is 360 tokens. The learning rate is reduced depending on the progress of the validation loss.

3. CNN: Consists of a single convolutional layer. The input layer contains an embedding layer and has the same properties as the previous one. The pre-processed data is flattened, resulting in a similar dense layer to the Bi-directional LSTM final layer. Configuration parameters: Adam optimizer with a learning rate of 1e-4, training batch size of 32 samples, the maximum length of input sequences is 360 tokens. The learning rate is reduced depending on the progress of the validation loss.

4.2 Sentiment Analysis Scoring and Tags Based Approach

Using our pre-processed dataset (Figure 3 for reference), we first convert the dataframe to a numpy float64 array. Then, the data is permutated such that different data values exist in the training and testing sets for each iteration.

1. Neural Net Classification: To keep the feature values in the -1 to 1 range, we standardize feature values by removing the mean and scaling to unit variance. Using the multi-layer perceptron classifier provided by sklearn, we train the classifier with the following configuration: a hidden layer size of (9,9), a hyperbolic tan activation function for the hidden layer, the stochastic gradient descent solver for weight optimization, and a constant learning rate of 0.1.

2. Random Forest Classification: Picking random data points from our training set, we build a decision tree associated with these data points. To optimize the performance of the model, cross-validation is used to split the training set into model-building and model-validation subsets. Test different numbers of decision trees and depths by iterating through the number of decision trees between 50 and 300, and a depth between 1 and 20. After establishing the optimal number of

decision trees and the depth, re-build the model and test the model against the test set. Then, using the feature_importances attribute of the Random Forest classifier, determine which features contribute most to the quality classification task.

4.2.1 Evaluation Metrics

For any deep learning model, achieving a 'good fit' on the model is crucial. To evaluate the performances of each of the models, we will be using three statistical metrics: accuracy, precision, and recall. In our dataset, given that each quality label is of equal importance (reference Figure 1), we believe classification accuracy is the most effective. Other statistical measures we are considering are precision and recall as precision allows us to identify a measure of result relevancy, while recall allows us to measure the number of truly relevant results that the model returns.

5 Results and Discussion

5.1 Linguistic Characteristics Based Approach

As discussed in the Methods section, we are using statistical measures to compare the performances of each of the techniques for the linguistic characteristics-based approach. The results are shown in Table 1. To discuss the accuracy of each of the models first, we see that CNN achieved the highest accuracy, whereas BERT was the least accurate. To understand these results a significant factor to consider is that the Stack Overflow data set does not contain simple English sentences but various terms and words relating to programming and its frameworks. Since one of the limitations of BERT is the lack of ability to handle long text sequences and due to our experimental setup which limited the maximum length of input sequences to 150 tokens to control the use of memory resources, it seems logical that BERT was less accurate than the other two models.

Comparing the accuracies of bi-directional LSTM and CNN, we can first turn to Fig 4 to compare the accuracy and loss on both the models for training as well as the validation set. On the top figure, the blue line represents the bi-directional LSTM model and the green line represents the CNN model where the strong line is for training accuracies and the dotted line is for the validation accuracies. The same applies to the bottom figure representing losses. We see that although two models train well from zero epochs, after the third or fourth epochs, we see that the models start over-fitting on the training dataset. To understand why CNN had an accuracy about 1% higher than that of bi-directional LSTM, we can hypothesize that it is due to the nature of the two models. Bi-directional LSTM is used to process and make predictions given sequences of data whereas CNN is designed to exploit spatial correlation in data. Hence, CNN excels at learning the spatial structure in input data. In comparison, bi-directional LSTM excels at situations where prediction depends not only on the previous input but also the future input, which although applicable to this project, is probably less effective than understanding a spatial structure.

	Techniques		
	BERT	LSTM	CNN
Accuracy	61.54%	65.88%	66.80%
Precision	96.12%	66.34%	67.00%
Recall	96.20%	65.66%	66.67%

Table 1: Performance measurements of techniques for linguistic characteristics based approach

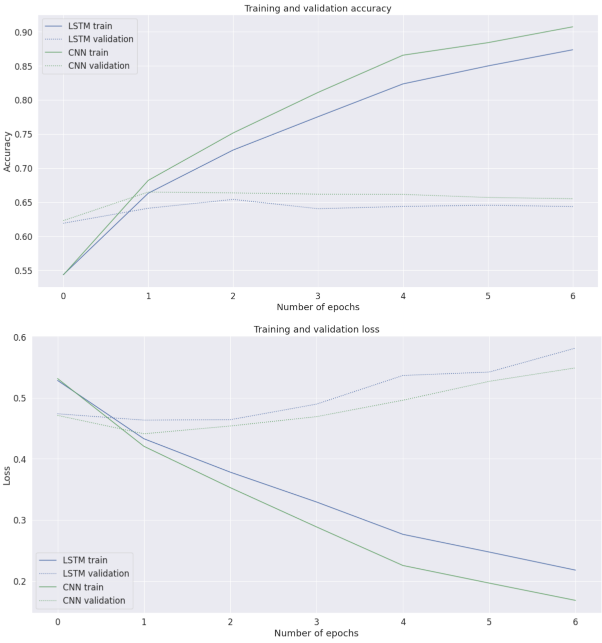


Figure 4: LSTM and CNN Accuracy and Loss

Observing the other performance measurements, although BERT was the least accurate, we see that it has significantly higher precision and recall. This would signify that BERT is fairly accurate in classifying the labels correctly as high precision relates to a low false positive rate and high recall relates to low false negative rate, but since we know that BERT is the least accurate, it is probable that BERT may be susceptible to other types of error.

5.2 Sentiment Analysis Scoring and Tags Based Approach

In our approach using Neural Network Classification, we saw that the log prediction error is 1.03287, and without using cross-validation to tune hyperparameters, after training, the model had 5475 out of 12000 correct labels with a 45.62% accuracy. This result is not surprising as although we are using a deep neural network, as our general deep learning model is unable to learn complex patterns within the data unlike the bi-directional LSTM or CNN models. Furthermore, the neural network model we use is unable to capture sequential information in the input data which the other deep learning architectures we discuss is capable of.

In our approach using Sentiment Analysis, we weighted the Title, Body, and Tag equally and scrambled the data for a different train and test split each time. Figure 3 shows how we have the sentiment score for our Title and Body as well as categorizations for the Tags and Post Quality Categories. The Random Forest that we built had a depth of 13 and 150 trees total. After training, we had 6155 out of 12000 correct labels with a 51.29% accuracy. We calculated the Feature Importance to see which features most contributed to predicting a high-quality question. Among our random forest, the features Title had 18.40% importance, Body had 29.90% importance and Tags had the highest importance of 51.69%. Thus, tags were most successful in predicting the quality of a question.

Interestingly, the Tag of a post was the most influential feature rather than the Title or Body in our Random Forest Classifier. For this experiment, among the list of tags for each post, we chose the first one–the most relevant to the post–to analyze. Since tags were the most influential this could mean that certain tags, or more popular tags, indicated higher quality posts. If we had found a way to analyze the entire lists of tags associated with each post, perhaps a larger number of tags may have correlated to higher quality posts. This outcome was surprising since based on Bazelli's work, (Bazelli et al., 2013) our initial thoughts were that posts with a higher sentiment would be higher quality, yet the Title and Body were only 18% and 30% importance respectively. It is important to note our accuracy was only 51% so it is possible with further experimentation and a better-trained model that different features may also be influential in question quality. To an extent, this matches Kavuk et al.'s (Kavuk and Tosun, 2020) findings as our results imply that correctly labeled question tags contribute greatly to the question quality.

5.3 Larger Impact and Future Work

We recognize due to the nature of English content Stack Overflow, our model performs best on English which excludes large sections of the programming community. For one example, India has a large community of programmers where this quality based metric would be unable to perform or in other Community Question Answering Forums in different countries. In the future, we look to extend this work to different languages to create a more inclusive metric.

6 Conclusion

Our goal was to create an automatic quality control structure to determine the question quality of Stack Overflow Questions. We found both approaches of Linguistic Characteristics and Sentiment Analysis Scoring and Tags Based Approach to be useful to determining Question Quality in different ways. When evaluating Linguistic Characteristics we found that CNN and bidirectional LSTM had higher accuracies than BERT with CNN having a slightly higher accuracy than bidirectional LSTM though BERT had higher precision and recall. When evaluating the dataset using Sentiment Analysis, Neural Network Classification had an accuracy of about 46% while our Random Forest Classifier had an accuracy of about 51% and found tags to be the most influential feature to predicting post quality. In future work, we would like to explore our Sentiment Analysis models further to achieve higher accuracies and have a more confident understanding of which features are the most influential to high quality questions. The programs used for our research is contained here.

References

Eugene Agichtein, Carlos Castillo, Debora Donato, Aristides Gionis, and Gilad Mishne. 2008. Finding high-quality content in social media. WSDM '08, page 183–194, New York, NY, USA. Association for Computing Machinery.

Anton Barua, Stephen W. Thomas, and Ahmed E. Hassan. 2014. What are developers talking about? an analysis of topics and trends in stack overflow. *Empirical Softw. Engg.*, 19(3):619–654.

Blerina Bazelli, Abram Hindle, and Eleni Stroulia. 2013. On the personality traits of stackoverflow users. In *2013 IEEE International Conference on Software Maintenance*, pages 460–463.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Eray Mert Kavuk and Ayse Tosun. 2020. Predicting stack overflow question tags: A multi-class, multi-label classification. In *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*, ICSEW'20, page 489–493, New York, NY, USA. Association for Computing Machinery.

Baichuan Li, Tan Jin, Michael R. Lyu, Irwin King, and Barley Mak. 2012. Analyzing and predicting question quality in community question answering services. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12 Companion, page 775–782, New York, NY, USA. Association for Computing Machinery.

László Tóth, Balázs Nagy, Dávid Janthó, László Vidács, and Tibor Gyimóthy. 2019. Towards an accurate prediction of the question quality on stack overflow using a deep-learning-based nlp approach. In *Proceedings of the 14th International Conference on Software Technologies*, ICSOFT 2019, page 631–639, Setubal, PRT. SCITEPRESS - Science and Technology Publications, Lda.