QUALITY CLASSIFICATION OF QUESTION ON STACK OVERFLOW

Nguyen Thi Phuong Thao[1†], Nguyen Thi Nguyet[1†], Bui Nguyen Phuong Linh[1†] and Ho Thanh Duy Khanh[1†]

[1]VNUHCM - University of Information Technology,Viet Nam.

Contributing authors: 20521936@gm.uit.edu.vn; 20521689@gm.uit.edu.vn; 20521527@gm.uit.edu.vn; 20521445@gm.uit.edu.vn;

[†]These authors contributed equally to this work.

Abstract

Community Question Answering (CQA) is the field of computational linguistics that deals with problems derived from the questions and answers posted to websites and it have a growing popularity as a way of providing and researching of information. Crowdsourced knowledge is a resource for users yet it can raise concerns about the quality of the shared content. As recognizing good question that can improve the CQA services and the user's experience, the study focuses on question quality instead. Using dataset of questions and answers posted to the Stack Overflow website, we have analyzed and conducted quality classification of questions. In addition to taking advantage of the natural language processing capabilities of neural network Deep Learning models such as LSTM, Bi-LSTM, Distil-BERT, we also apply some Machine Learning classification models like Logistic Regression, Multinomial Naïve Bayes, Decision Tree, Random Forest. Then we compare all the models and give the best model to help classify the quality of question. Initially, the result were obtained with the Distil-BERT model with the highest accuracy of 91.80%.

Keywords: LSTM, Bi-LSTM, Distil-BERT, Logistic Regression, Multinomial Naïve Bayes, Decision Tree, Random Forest

1 Introduction the problems

The Internet and social networks make it easy for us to capture information from a variety of sources, allowing users to exchange and share information effectively, transcending geographical and time constraints. Because of this, there is now a lot of information posted on the Internet every day and the advent of CQA websites helps to provide an interface for users to exchange and share knowledge. The user asking a question lacks knowledge of a specific topic and searches for an expert to provide the desired knowledge. Ensuring the quality of questions is essential in Q&A communities. A question of high quality is assumed to attract more visitors and more answers in short time. In the long run, the activity and popularity of a Q&A community increases if the quality of the shared content is high. But, a lot of times questions are downvoted and never answered. This is mostly deemed due to the poor quality of question asked such that the reader is unable to frame a suitable answer. To avoid this, it is suggested that a user reads and rereads their own question before posting it on the platform but there is not a metric on Stack Overflow that suggests you whether your questions are poor-quality or high-quality. Because of that and with the proliferation of deep learning techniques, this decade can be said to the golden age for Natural Language Processing (NLP) so in this project, we will try to work on this problem by going to analyse and classify the quality of questions using three Deep Learning techniques include LSTM, bi-LSTM, Distil-BERT. We also try four Machine Learning models: Logistic Regression, Multinomial Naïve Bayes, Decision Tree and Random Forest. Finally, we compare the results and analyse the information gained from it. As a result, users will know before posting a question whether they need to improve or modify the question.

2 Introduction deep learning and machine learning methods

In this study, in order to classify the quality of questions collected on Stack Overflow, we use three deep learning models:

- LSTM:

Long Short Term Memory networks– are a special kind of RNN, capable of learning long-term dependencies. LSTMs are explicitly designed to avoid the long-term dependency problem. All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer. LSTMs also have this chain like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting in a very special way.

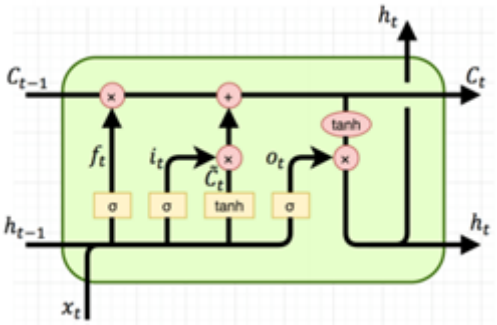

Fig. 1 Describe the LSTM model

• Bi-LSTM:

It consists of an Embedding layer as its input. Then we have two bi-directional LSTM layers stacked. The LSTM layers use around 64 hidden neurons. Also, the first LSTM layer returns a sequence that can be directly fed into the second layer. These design decisions were made keeping in mind that the model should reach at least an accuracy of 80%. The final layer is a dense layer using a soft-max activation function so that the output is in a probabilistic format.

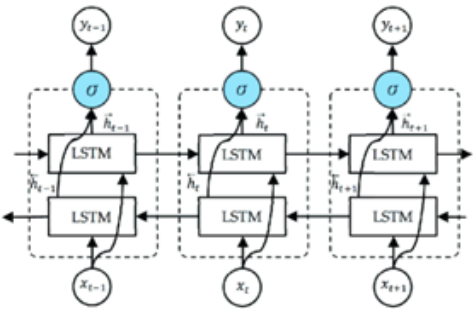

Fig. 2 Describe Bi-LSTM Model

•Distil-BERT

It is a small, fast, cheap and light transformer model based on the BERT architecture. Knowledge distillation is performed during the pre-training phase to reduce the size of a BERT model by 40%. To leverage the inductive biases learned by larger models during pre-training, the authors introduce a triple loss combining language modelling, distillation and cosine-distance losses. The architecture of the proposed feature extraction model based on DistilBERT is shown in Figure 1.2 DistilBERT receives as an input X which represents a tweet from the dataset (word sequence). The inputted

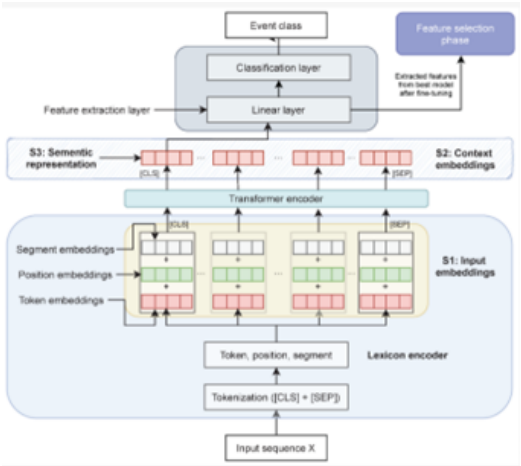

Fig. 3 The proposed feature extraction model.

sequences to DistilBERT are converted to a set of embedding vectors where each vector is mapped to each word in a sequence (S1). DistilBERT uses the transformer encoder to learn the contextual information for each word. The transformer encoder uses a self-attention mechanism to generate the contextual embeddings (S2). The extracted contextual embeddings for each word are concatenated into a single vector to represent the semantic information presented in the tweet (S3). S3 is the input of a fully connected layer that outputs a vector of size d where d is the number of neurons. Later, a classification layer is placed at the end of the feature extractor model to fine-tune the pre-trained DistilBERT on the event detection task and predict the corresponding event class for each inputted sequence (tweet). In what follows, we detail the model fine-tuning and feature extraction processes.

In addition, we also apply four machine learning classification algorithms including:

•Logistic Regression:

Estimates the probability of an event, such as voted or not voted, based on a given data set of independent variables. Since the outcome is a probability, the dependent variable is limited to between 0 and 1. In logistic regression, a logit transform is applied on the odds - that is, the probability of success divided by the probability of failure lose.

•Multinomial Naive Bayes:

Multinomial Naive Bayes algorithm is a probabilistic learning method that is mostly used in Natural Language Processing (NLP). The algorithm is based on the Bayes theorem and predicts the tag of a text such as a piece of email or newspaper article. It calculates the probability of each tag for a given sample and then gives the tag with the highest probability as output.

•Decision Tree:

Decision Tree is a structured hierarchical tree used to classify objects based

on sequences of rules.
•Random Forest:
Random Forest is a data structure applied to machine learning that grows a large number of random decision trees that analyse sets of variables. This type of algorithm enhances the ways in which technology analyse complex data.

3 Description of the dataset

3.1 Implementation process

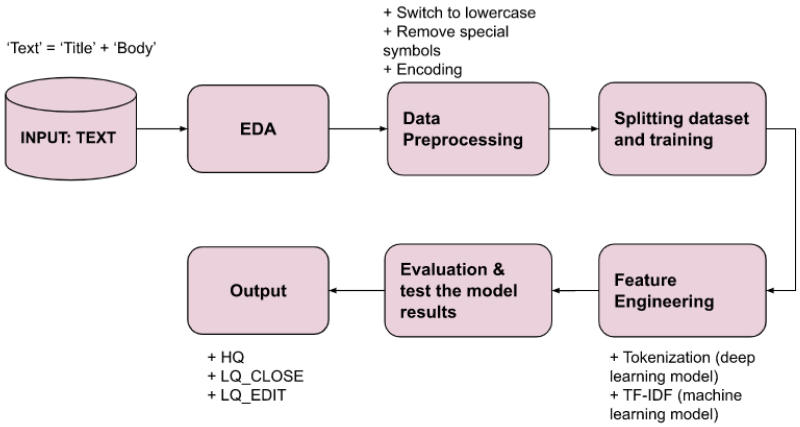

Fig. 4 Table of summarizing the research process

3.2 Information about datasets

The dataset used for this project is the Stack Overflow questions dataset found and collected by Moore. This dataset can be found on Kaggle.
https://www.kaggle.com/datasets/imoore/60k-stack-overflow-questions-with-quality-rate
https://stackoverflow.com

3.3 Information about attributes

The dataset consists of over 60,000 data samples that are collected from the Stack Overflow website. These questions were asked in a time period ranging from January 1st 2016 to January 1st 2020. The dataset includes 2 data files: train.csv (45,000 samples), valid.csv (15,000 samples). The dataset consists of 6 features: unique question ID, a question title, main body or content of the

question, tags representing the important words (keywords) in the question, creation date of the question as well as the class/label of the question. The label itself consists of 3 classes:

• High-Quality (HQ): questions that receive a score a more than 30 from the community and is not edited a single time by anyone.

• Low-Quality Edited (LQ_EDIT): questions that receive a negative score and multiple edits from the community.

• Low-Quality Closed (LQ_CLOSE): questions that were immediately closed by the community due to its extremely poor quality. These questions are sorted according to their question ID. Also, the main content or text of the questions are in the HTML format and the dates are in the UTC format.

4 Results

Algorithms		Score	Precision	Recall	F1-score	Accuracy
Deep Learning	LSTM	HQ	0.76	0.8	0.78	0.84775
		LQ_EDIT	0.79	0.74	0.76	
		LQ_CLOSE	1	1	1	
	Bi-LSTM	HQ	0.7	0.9	0.79	0.83608
		LQ_EDIT	0.86	0.61	0.71	
		LQ_CLOSE	1	1	1	
	Distil-BERT	HQ	0.87	0.88	0.87	0.918
		LQ_EDIT	0.88	0.88	0.88	
		LQ_CLOSE	1	1	1	
Machine Learning	Naïve-bayes	HQ	0.71	0.87	0.78	0.7225
		LQ_EDIT	0.66	0.66	0.66	
		LQ_CLOSE	0.84	0.64	0.72	
	Logistic Regression	HQ	0.87	0.84	0.85	0.82267
		LQ_EDIT	0.76	0.76	0.76	
		LQ_CLOSE	0.84	0.87	0.85	
	Decision Tree	HQ	0.75	0.74	0.75	0.7355
		LQ_EDIT	0.66	0.64	0.65	
		LQ_CLOSE	0.8	0.82	0.81	
	Random Forest	HQ	0.82	0.78	0.8	0.7865
		LQ_EDIT	0.7	0.74	0.72	
		LQ_CLOSE	0.84	0.84	0.84	

4.1 Deep Learning

4.1.1 LSTM

The model achieves 84.52% accuracy on the testing data:
• The model works best in classifying LQ_CLOSE labels with an accuracy rate of nearly 100% and mistaken classification rate to LQ_EDIT label is only 0.125% and there is no mistaken classification to HQ label.
• Rate of primary label classification LQ_EDIT reached 65.65%. It can be seen that the model has difficulty in classifying the LQ_EDIT and HQ labels when the misclassification rate is 34.33%.
• Rate of primary label classification HQ is 88.02%. The rate of misclassification to LQ_EDIT label is 11.98% and 0% to LQ_CLOSE label.

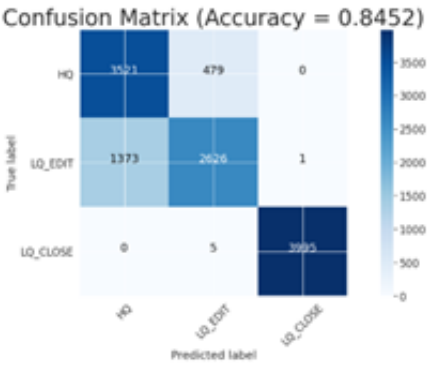

Fig. 5 The confusion matrix of LSTM

4.1.2 Bi-LSTM

The model achieves 84.95% accuracy on the testing data:
• The model still performs best in classifier LQ_CLOSE with almost absolute rate (99,9%) and misclassification rate into LQ_EDIT and HQ labels is very small.
• Rate of primary label classification LQ_EDIT is 69.65% and HQ reached 85.3%. The model had a hard time distinguishing between LQ_EDIT class and HQ class, as 14.7% of the HQ were predicted as LQ_EDIT and 30.325% when flipped.

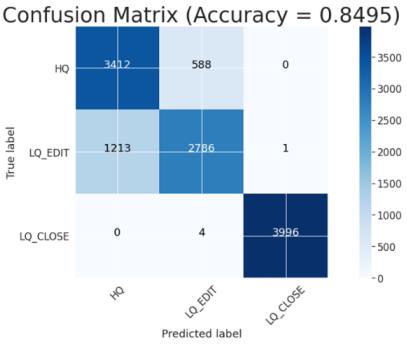

Fig. 6 The confusion matrix of Bi-LSTM

4.1.3 Distil-BERT

The model achieves 91.42% accuracy on the testing data:
• As LSTM and Bi-LSTM, the model shows the best in classifying LQ_CLOSE labels with almost absolute accuracy rate (99.57%) and the rate of misclassification into LQ_EDIT labels is only 0.37% and 0.06% misclassification as HQ label.
• The rate of correct classification of HQ and LQ_EDIT labels is 86.97% and 87.82%. The model has little difficulty in classifying the LQ_EDIT and HQ labels when the misclassification rate of both labels is about 12%.

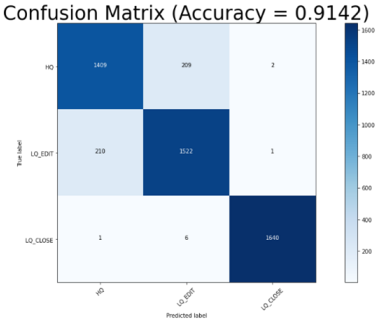

Fig. 7 The confusion matrix of distil-BERT

4.2 Machine Learning

4.2.1 Naïve Bayes

Naïve Bayes has accuracy reaching 73.19%. In particular, the correct classification rate of the HQ label is the highest with 85.38% and only mistakenly classifies 13.35% as LQ_EDIT label and 1.27% as LQ_CLOSE label. The other two labels, LQ_EDIT and LQ_CLOSE, have accuracy rates of 69.80% and

64.64%, respectively. It can be seen that the model has difficulty in misclassifying the label LQ_CLOSE into LQ_EDIT when the misclassification rate is 23.04% and 9.66% when it slipped.

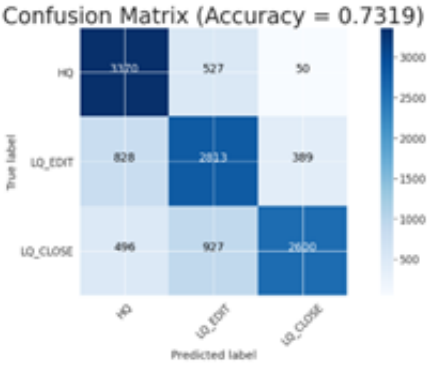

Fig. 8 The confusion matrix of naïve bayes

4.2.2 Logistic Regression

Unlike the Naïve Bayes model, Logistic Regression gives the highest classification accuracy of the LQ_CLOSE label with 86.63%, then the HQ label 83.73% and the lowest LQ_EDIT label 76.45%. The rate of misclassification of the LQ_EDIT label as LQ_CLOSE is quite high with 12.14% and the HQ label being misclassified as LQ_EDIT is also 12.95%. At the same time, Logistic Regression also has the highest accuracy of the four machine learning models when reaching 82.26%.

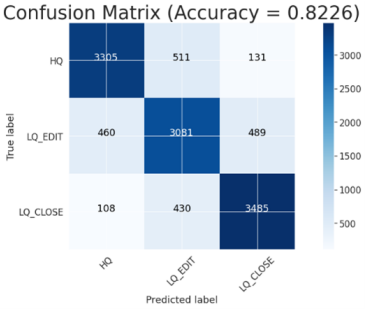

Fig. 9 The confusion matrix of logistic regression

4.2.3 Decision Tree

With algorithmic Decision Tree, the model has an accuracy of 73.7% with the highest correct classification rate still belongs to the LQ_CLOSE label with 82.3%, similar to Logistic Regression, followed by HQ with 73.47% and

LQ_EDIT with 65.33%. It can be seen that the model has difficulty in classifying HQ and LQ_EDIT labels when the misclassification rates of the two comb labels are 21.20% and 20.3%.

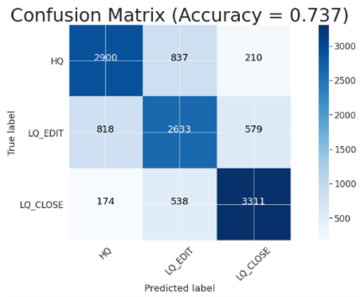

Fig. 10 The confusion matrix of decision tree

4.2.4 Random Forest

Overall, Random Forest is ranked exactly 2nd, just behind Logistic Regression (78.65%). The correct classification rates of the labels LQ_CLOSE, LQ_EDIT and HQ were 82.71%, 75.14% and 78.11%, respectively. In which, the rate of misclassifying the HQ label as the LQ_EDIT label is still quite high with 18.80%.

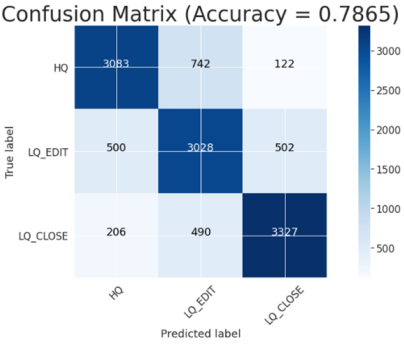

Fig. 11 The confusion matrix of random forest

From the results obtained from the confusion matrix and the Accuracy measure, we can make the following comments:

• Of the three Deep Learning models, we can see that the Distil-BERT model has the best results when it has the highest Accuracy (91.42%) and the correct classification rate of labels is quite high and relatively uniform with the classification rate. Type 2 labels HQ and LQ_EDIT both reach nearly 88% and label LQ_CLOSE is almost absolute (99.9%). Followed by Bi-LSTM and LSTM with Accuracy at 84.95% and 83.52%, respectively, as well as the

confusion matrix of Bi-LSTM giving better results than LSTM when the classification accuracy rate is higher and more uniform.
• With four Machine Learning models, Logistic Regression is the best classification model with Accuracy of 82.26% and almost the highest rate of correct classification of labels. The Random Forest model is also highly appreciated when the accuracy is only behind Logistic Regression and the result of the confusion matrix in the classification has a very low difference between the labels. The Naïve Bayes model has the lowest accuracy and rate of correctly classifying the LQ_CLOSE label.

Types of classification models of Machine Learning simply cannot achieve as good results as Deep Learning models achieve for large and complex data files. The most concrete evidence we can see is that the accuracy of Deep Learning models is about 10% higher than that of Machine Learning models and the classification results of Deep Learning models have classification rates. quite high (most of the correct classification rates are at 61% or more). Especially, the Distil-BERT model is said to be the most suitable when it gives the best classification results out of a total of 7 models.

5 Conclusions and Future Work

With the application of 7 models to the question quality classification problem on Stack Overflow, including 3 Deep Learning models, LSTM, Bi-LSTM, Distil-BERT and 4 Machine Learning models including Naïve Bayes, Logistic Regression, Decision Tree, Random Forest, we have the clearest comparison of the results of the models when operating on the same dataset. It can be concluded that simple Machine Learning models cannot produce as good results as Deep Learning models achieve. Most of the simple machine learning algorithms implemented by others achieve around 76% of accuracies, however neural network Deep Learning models can give an average of 86% accuracy. From the obtained results, the Distil-BERT model is the best model that can develop the application for predicting the quality of the question based on a large amount of previous data, helping users to know before posting the question whether they whether the question needs to be improved or revised. Another interesting direction as also stated by the author of the dataset is to be able to predict the tags when the user has completed typing his/her questions. Some cloud services could be used for such a complex task, which is effectively a multi-class multi-label classification or prediction task. Since this is a dataset that is updated continuously over time, in the future, we will continue to collect, conduct research and apply some other models that give the best results to solve many problems that are and will be raised in the future.

References

[1] Mohini Wakchaure, Prakash Kulkarni, A Scheme of Answer Selection In Community Question Answering Using Machine Learning Techniques, 2019.

[2] Barun Patra, A survey of Community Question Answering, 2017.

[3] Nouha Othman, Rim Faiz, Kamel Smaili, Enhancing Question Retrieval in Community Question Answering Using Word Embeddings, 2019.

[4] Wei Wu, Xu Sun, Houfeng Wang, Question Condensing Networks for Answer Selection in Community Question Answering, 2018.

[5] Hapnes Toba, Zhaoyan Ming, Mirna Adriani, Chua Tat Seng, Discovering high quality answers in community question answering archives using a hierarchy of classifiers Lucas Valetin, Answer ranking in Community Question Answering: a deep learning approach, 2022.

[6] Zhengfa Yang, Qian Liu, Baowen Sun, Xin Zhao, Expert recommendation in community question answering: a review and future direction, 2019.

[7] Hapnes Toba, Zhaoyan Ming, Mirna Adriani, Chua Tat Seng, Discovering high quality answers in community question answering archives using a hierarchy of classifiers

[8] Ryosuke Inoue, Yoshiaki Kurosawa, Kazuya Mera, Toshiyuki Takezawa, A question-and-answer classification technique for constructing and managing spoken dialog system, 2011.

[9] Kajian Li, Chengzhang Qu, A Classified Question and Answer System Based on SpringBoot Integrated Neo4j Graph Database, 2021

[10] Haiying Shen, Guoxin Liu, Haoyu Wang, Nikhil Vithlani, SocialQ&A: An Online Social Network Based Question and Answer System, 2017.