Towards an Accurate Prediction of the Question Quality on Stack Overflow using a Deep-Learning-Based NLP Approach

László Tóth[1], Balázs Nagy[1], Dávid Janthó[1], László Vidács[1,2] and Tibor Gyimóthy[1,2]
[1]*Department of Software Engineering, University of Szeged, Hungary*
[2]*MTA-SZTE Research Group of Artificial Intelligence, University of Szeged, Hungary*

Keywords: Question Answering, Q&A, Stack Overflow, Quality, Natural Language Processing, NLP, Deep Learning, Doc2Vec.

Abstract: Online question answering (Q&A) forums like Stack Overflow have been playing an increasingly important role in supporting the daily tasks of developers. Stack Overflow can be considered as a meeting point of experienced developers and those who are looking for a solution for a specific problem. Since anyone with any background and experience level can ask and respond to questions, the community tries to use different solutions to maintain quality, such as closing and deleting inappropriate posts. As over 8,000 posts arrive on Stack Overflow every day, the effective automatic filtering of them is essential. In this paper, we present a novel approach for classifying questions based exclusively on their linguistic and semantic features using deep learning method. Our binary classifier relying on the textual properties of posts can predict whether the question is to be closed with an accuracy of 74% similar to the results of previous metrics-based models. In accordance with our findings we conclude that by combining deep learning and natural language processing methods, the maintenance of quality at Q&A forums could be supported using only the raw text of posts.

1 INTRODUCTION

Over the last few years, online question answering (Q&A) forums such as Stack Overflow have become the essential repositories of knowledge for software engineers. Since the information gathered on the portal is tied to professional questions originated from practitioners or hobbyist programmers, these posts provide an effective and also practically applicable support to both novice and experienced professionals (Correa and Sureka, 2013; Ponzanelli et al., 2014).

To preserve the professionality and quality of Q&A forums, participants must follow the rules established by the community. One such fundamental rule is that the questions posted on the portal should be relevant, unambiguous and comprehensible. This means that every question being submitted onto Stack Overflow has to be related either to specific development issues or methods such as using a particular API, algorithmic problems, tools or methodology. Questions that cannot be answered appropriately and straightforwardly, e.g., open-ended, off-topic, or chatty questions, which may reduce the quality of the portal, should not be supported.

The main purpose of Stack Overflow is to solve well-defined technical issues, therefore, subjective posts or questions raising never-ending, opinion-related discussions should also be avoided. In addition, questions need to be formulated in a simple and explicit way so that their purpose is obvious to the potential respondents. Questions that are not met the above criteria will be closed or, in some cases, even deleted[1] from the site by moderators or experienced members with distinguished privileges (Correa and Sureka, 2014). At the same time, high-quality questions and answers are rewarded by the community by upvoting their score, which, consequently, increases the user reputation of the question poster or the respondent. Eventually, at a certain reputation level, exclusive rights will be granted to these actively contributing users.

The posting frequency at Stack Overflow is rather high, with an average of over 8,000 new questions being sent by the users on workdays and almost 10,000 in total during the weekend[2]. This extreme level of activity makes it very difficult and inefficient to manually review every single question sent onto the portal.

[1]https://stackoverflow.com/help/asking
[2]https://sostats.github.io/last30days/

Tóth, L., Nagy, B., Janthó, D., Vidács, L. and Gyimóthy, T.
Towards an Accurate Prediction of the Question Quality on Stack Overflow using a Deep-Learning-Based NLP Approach.
DOI: 10.5220/0007971306310639
In *Proceedings of the 14th International Conference on Software Technologies (ICSOFT 2019)*, pages 631-639
ISBN: 978-989-758-379-7

The workload of the review process could be reduced by applying a classification tool which can identify questions that do not satisfy the aforementioned criteria.

Automating the quality-based classification of questions submitted onto Q&A portals has arisen the interest of researchers in recent years (Correa and Sureka, 2014; Ponzanelli et al., 2014; Arora et al., 2015; Yao et al., 2013; Baltadzhieva and Chrupała, 2015). The various procedures applied by the researchers have produced moderately good results with an accuracy of 73% (Correa and Sureka, 2013) or with precision from 62.1% to 76.2% (Ponzanelli et al., 2014). Previous investigations indicate that the reliable classification of a question based on its quality is a challenging task.

The focus of our present study is the quality-based classification of questions uploaded to Stack Overflow based on their linguistic characteristics. The purpose of this work is two-fold. On the one hand, we present an elegant and accurate solution to the quality-classification problem that can aid Stack Overflow moderators in automating the detection of low-quality questions posted to the portal. On the other hand, a further aim of our model is to support the questioners to pre-test the quality of their question prior to submission. With these goals in mind, we used natural language processing (NLP) methods in combination with deep learning techniques to classify questions by their quality that is assumed to be the main viewpoint in the final decision about closing or deleting them. Using this approach we achieved an accuracy of 74% and a precision of 75%.

In summary, this paper provides the following contributions:

- We present a novel model based on deep learning and natural language processing methods for classifying questions using their linguistic and semantic features solely.
- Applying this model, we empirically demonstrate its usability in the classification of the questions posted on Stack Overflow.

2 STACK OVERFLOW

Stack Overflow was launched in 2008 by Joel Spolsky and Jeff Atwood (Atwood, 2018). The purpose of the site is to support both professional and enthusiast developers in their everyday labor activities. It is important to emphasize that the site was created to support the work of actual practitioners, so, initially, it was not meant to support mentoring the beginners, participating in the solution of student assignments, or providing a platform for either educational or informal conversations. Since the beginning, the operability and the quality of the portal have been primarily maintained by experienced users, who devote their free time to volunteering.

To preserve the professionalism and quality of the portal, both questions and answers are subjected to a review process after their submission done by experienced users and moderators. This is a manual task relying on a voting system. In particular, privileged users, those, who have 3,000 or more reputation points can cast a vote to close a question, and five such votes result in closing the question. A previously closed question can be deleted from the site if it receives at least three votes from members within the 10,000 or higher reputation range.

Reputation points and different badges reflect both the experience and the activity of the users on Stack Overflow. These achievements can be earned by asking good quality questions or providing good quality answers both receiving high scores from the community. Higher reputation allows more elevated privileges such as the right to vote for closing or deleting a post.

There are several reasons for a question to be judged for closing. Stack Overflow Help Center provides tips for users about what kind of questions are considered appropriate and inappropriate. The most important criterion for a good quality question is that it should be both clearly answerable and subject specific. The Help Center of Stack Overflow also provides additional suggestions for style expected in comments. According to these suggestions, the questions and answers should be clear and understandable, and the friendly tone towards other members is also expected. This latter expectation is reflected in the reputation points of the members, since users who are socially more interactive and open to new ideas tend to receive more positive votes as found by Bazelli et al. (Bazelli et al., 2013). As a result, positive attitudes play a prominent role in evaluating the questions or answers.

To get a quick idea about the magnitude of the problem caused by poor quality posts, we examine the cardinality of closed or deleted questions, and the changes of these values in the past years. We note that there are reasons, e.g., closing because of duplication, or off-topic, not necessarily related to the quality. In most cases, identifying these posts is relatively easy to automatize.

Figure 1 presents the temporal distribution of questions, questions with accepted answers, and the number of deleted and closed questions during the

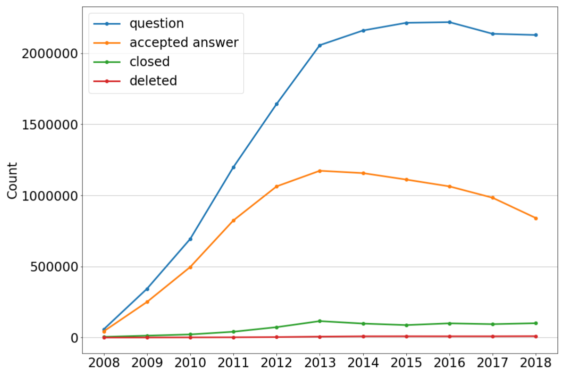

Figure 1: Temporal distribution of the Stack Overflow Questions.

lifetime of Stack Overflow. As it can be seen the number of questions along with the ones with accepted answer has been dramatically increased to year 2013, while the change in the closed and deleted questions remained moderate and more-or-less constant. It is also worth to mention that the number of posts with accepted answers has been decreasing since 2014.

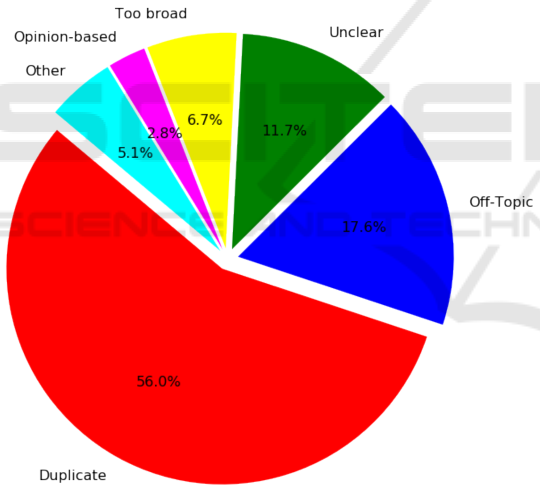

Figure 2: Reasons for closing questions.

Table 1: Reason for closing questions.

Reason of closing	Id of the reason	Number
duplication	(101,1)	466748
off-topic	(102,2)	146440
unclear	(103,4)	97713
too broad	(104)	56089
opinion-based	(105)	23665
other	(3,7,10,20)	42748

The distribution of the reason for closing can be examined using Table 1 and Figure 2. Please note that there was a change in the question closing policy at Stack Overflow in June 2013, therefore some of the currently used closing reasons has multiple Ids. The category labelled as *other* contains all reasons that were abolished in 2013 and are no longer in use.

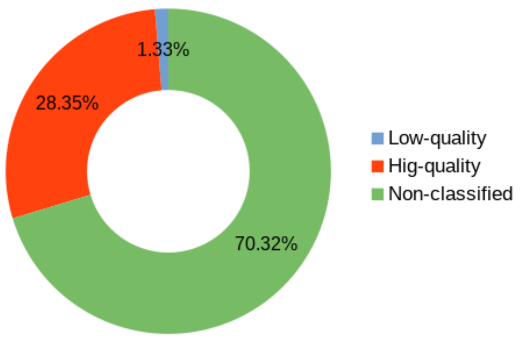

Figure 3: Distribution of the questions based on quality.

We focused on the quality-based binary classification of Stack Overflow questions in our experiments; therefore, the unequivocal definition of the high-quality and low-quality questions is indispensable for our purpose. To this end, we followed the definition established by Ponzanelli et al. (Ponzanelli et al., 2014): We consider a question to be of high-quality if it has at least one accepted answer, its score is higher than zero and it is neither closed nor deleted. A poor-quality question has a score less than zero, and it is either closed or deleted in its final state. According to these quality definitions, the distribution of the questions on 21 April 2019 is as follows (see also Figure 3):

- 236,154 low-quality questions;
- 5,031,820 high-quality questions;
- 12,478,772 non-classified question.

Based on the above quality definitions, a little more than 70% of the questions cannot be classified. There are two main reasons of this issue: *i)* a vast number of questions has zero scores (8,062,685), and *ii)* although many questions have positive scores, they do not have any accepted answers (3,226,399) making our classification impracticable.

3 PROCESS DESCRIPTION

Our investigation focuses on the classification of Stack Overflow questions by their quality via utilizing the method of natural language processing combined with a deep learning approach. We used only the textual information extracted from the raw text of question titles, bodies and tags. Since these are all known by the questioner during the post creation,

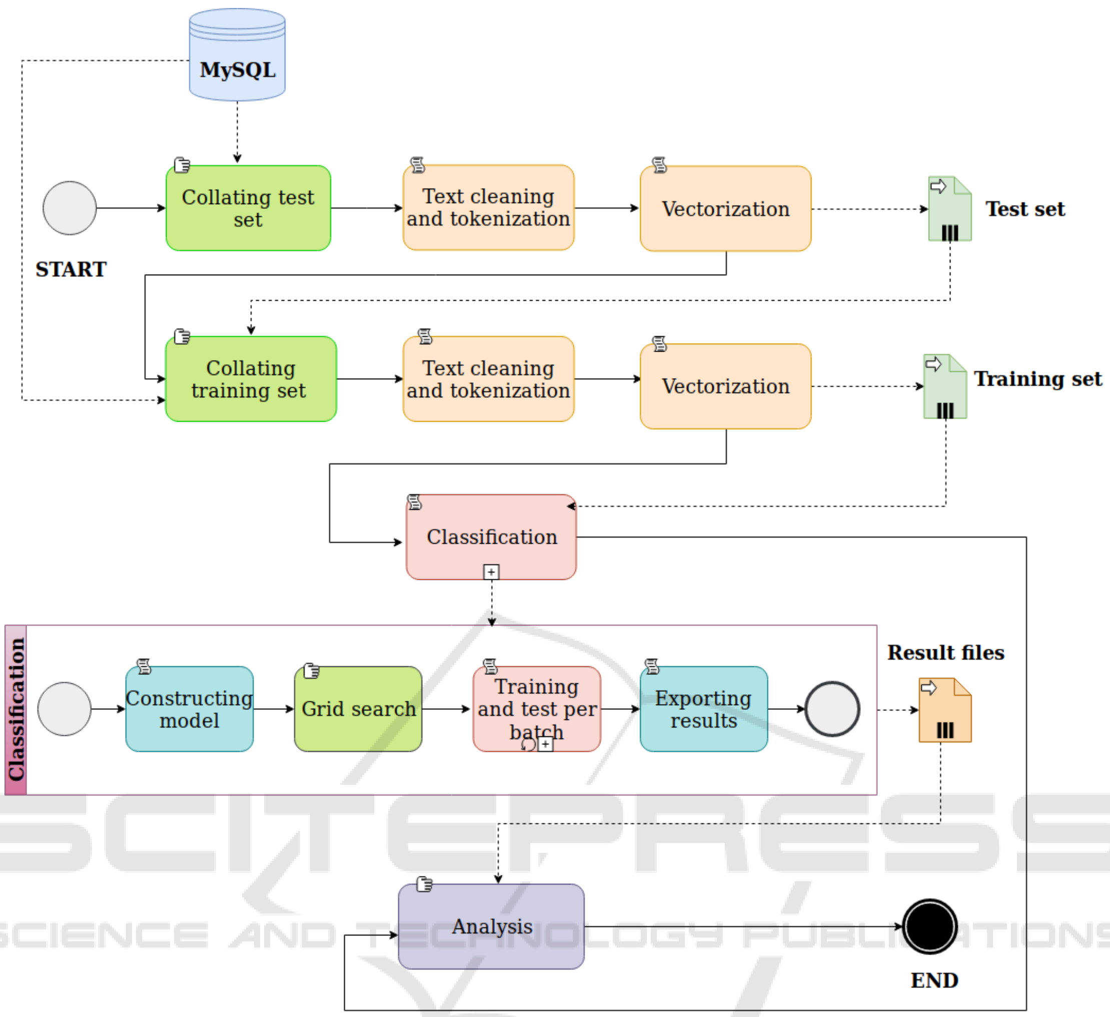

Figure 4: Workflow of the experiments.

consequently, the method presented here can also be used to support the user to pre-test their questions before posting it to the site.

Our classifier is constructed using Keras[3] library for deep learning computations with Tensorflow[4] back-end (Gulli, 2017). The model consists of one *Gated Rectifier Unit (GRU)* (Irie et al., 2016) followed by four *Dense* layers, one *Flatten* layer as well as a final *Dense* layer. The first *GRU* layer contains the same number of units as the input dimension. The input and also the output of the *GRU* layer is a three-dimensional tensor `(batch_number, timesteps, number_of_features)`. As the whole post was represented as a single vector, the `timesteps` parameter was set to one. The `number_of_features` parameter was set to the representation dimension of the post, title, and tags.

The number of neurons in the *Dense* layers is decreasing from layer to layer. The size of the first *Dense* layer – similarly to the previous *GRU* layer – is equal to the input size. In the subsequent three *Dense* layers, the number of neurons is continuously halved relative to the previous layer. The inputs as well as the outputs of these layers are both three-dimensional tensors, where the third dimension corresponds to the number of neurons in the actual layer.

The purpose of the *Flatten* layer is to transform the three-dimensional input tensor into two-dimensional form before the classification.

The last *Dense* layer contains only two neurons that perform the classification.

The above specified structure of the neural network was selected based on results from earlier classification experiments. However, the real power of the *GRU* layer was not exploited here because of the

[3]https://keras.io/

[4]https://www.tensorflow.org/

chosen representation of the posts. In particular, the information that the sequence of words can provide, that is, the memory properties of the *GRU* unit was not used in our case.

We applied the batch normalization process (Ioffe and Szegedy, 2015) to decrease the variation of the distribution of inputs in the internal layers of the deep neural networks. This method has numerous benefits such as stabilizing the network, accelerating the learning and reducing the dependence of gradients on the scale of the parameters.

We utilized a regularization technique called dropout to avoid overfitting (Srivastava et al., 2014). This regularization technique deactivates the neurons with a given probability provided by a hyperparameter called `dropout_rate`, during the learning process.

We chose the ReLU (Rectified Linear Unit) activation function for the inner layers. This function helps to avoid the vanishing gradient problem which is related to commonly applied activation functions such as the sigmoid and tangent hyperbolic in inner layers of the deep neural networks (Glorot et al., 2011). For the last layer, which performs the classification, we chose the softmax function. The first layer (GRU) uses tangent-hyperbolic as the activation function. As a loss function, the Binary Cross-Entropy was selected.

For training the model, we applied the Nesterov Stochastic Gradient Descent method with momentum (Sutskever et al., 2013). Using this technique, the value of the loss function is considered not only in the actual position but also a little ahead in the direction defined by the hyperparameter `momentum`.

For evaluation purpose, we applied three metrics, the precision, the recall, and the accuracy. The metrics were calculated using the weighted average method. Using this method, we focus on the overall performance, considering the number of samples of the different classes.

For testing the model, we chose a random subset from the Stack Overflow questions posted up to until mid September 2013. This period was selected because we intended to compare the performance of our model with that of Ponzanelli et al. (Ponzanelli et al., 2014) who used the Stack Overflow data dump from September 2013. That dump is not available anymore; therefore, we selected the specific period using the dump created on the 4th of March 2019. The test set was established to contain two disjunct classes: high-quality and low-quality questions according to the criteria mentioned in the previous section. Questions that were edited by their originator after posting were excluded. In the present study the same filtering conditions were applied as used earlier by Ponzanelli and co-workers, therefore the direct comparison of our results with the previous ones is possible.

The test dataset was compiled using balanced subsets of high-quality and low-quality questions. The full test set contains 110,547 posts from both subsets. The final working test set was defined as taking 30% of the full set as examples (66,328 posts) similarly to the study of Ponzanelli and colleagues.

Our model was trained using the whole question dataset without any time restrictions applying the same selection criteria for the two classes as we did for the test set. Questions occurring in the test set were excluded from the training set; hence for the evaluation of the model, we used questions that were never seen by the model before.

As our objective is to decide on the potential rejection of a question to be posted on Stack Overflow using only the linguistic features, the training and test sets contain only the question body, the title and the tags attached to the given post. The quality label 0 or 1 for low- or high-quality questions, respectively, along with the unique Id of the posts were also included in the training set. The Id was used only for post identification during the preprocessing, and it was removed before the actual training.

The whole process of our experiments is presented on Figure 4.

A cleaning procedure was first applied on the raw text of question bodies, titles, and tags. In particular, the code blocks, HTML tags, and non-textual characters were removed. Tokenization was performed using the `nlp` method available in the language tool spaCy[5].

The cleaned dataset was then transformed to document vectors obtained from spaCy using the language model `en_core_web_md`, which is a general linguistic model based on text from blogs, news, and comments. The model provides word vectors and document vectors which are based on the Word2Vec representation (Mikolov et al., 2013). The drawback of this approach is that the document vectors in spaCy are computed by the mean of the word vectors of the given document, which is in our case the post itself including its body, title, and tags. The dimension of vectors used in this representation is 300.

Considering that many special words and technical phrases, e.g., built-in function names, occur in the posts, that have no vector representation in spaCy's vocabulary, we constructed a 200-dimensional Doc2Vec model (Le and Mikolov, 2014) with the Gensim library[6] using Stack Overflow questions. This model is based on the Word2Vec method

[5]https://spacy.io/

[6]https://radimrehurek.com/gensim/apiref.html

with the extension of the feature vector with a new document-related vector. To improve the accuracy of our model, the code blocks, preformatted text, and numbers were not removed but changed into particular placeholder tokens. In this way, the semantics of the document can be preserved without considering the unnecessary details. Some specific characters such as #, + (see C#, C++), or punctuation were also retained during the preprocessing.

Using a random subset of the transformed input obtained with spaCy, a grid-search procedure was performed to find the optimal hyperparameters for our deep learning model, and the first experiment series were executed. Then, a second series was also executed after another grid-search procedure using the input set obtained with our Doc2Vec representation of Stack Overflow questions. We present the results of these experiments in the next section.

4 RESULTS AND DISCUSSION

As mentioned in the previous section, we have performed two series of experiments. First we applied a vectorization process using document vectors provided by spaCy. Based on the grid-search we set the `learning_rate` to 0.175 and the `momentum` value to 0.9. The `dropout_rate` was set to 0.2, the `batch_size` to 1000 and the number of epochs to 150. These latter parameter values were selected based on previous experiences and test calculations with the model.

In the first experiment (*Experiment 1.1*), we used only 425,097 random samples from the training set for training and obtained 65.5% for precision, 65% for recall and 65% for accuracy on a test set containing 66,328 samples. For evaluation, we used the `scikit_learn` library for Python. For calculation of the performance metrics (precision, recall, accuracy), we applied the weighted average method provided by `scikit_learn`.

We repeated the experiments by increasing the number of training examples to 764,443 samples (*Experiment 1.2*). The parameters and the structure of the input along with the size of the test set were the same as in the previous experiment. In this case, we obtained 73.3% for precision, 69.2% for recall and 69.3% for accuracy. We repeated the experiment using similarity metrics as additional features for the input (*Experiment 1.3*). Similarity metrics were calculated via the cosine similarity available in spaCy. In particular, we added similarities between body and title, body and tags, as well as title and tags as three new features to the feature space, where each similarity is defined as the cosine similarity of the corresponding document vectors. This similarity feature, however, has no significant effect on the accuracy (precision of 73.1%, recall of 67.8%, and accuracy of 67.8%). One possible explanation of this result is the fact that while each question body, title, and set of tags is represented by a vector of 300 dimensions, each one of the 3 similarity measures (body-title, body-tags, title-tags) is only a scalar value, which could not contribute to the results effectively. This assumption, however, has not yet confirmed.

Table 2: Performance measures of the experiments. See text for experiment details.

Exp. number	**Precision**	**Recall**	**Accuracy**
Experiment 1.1	65.5%	65.0%	65.0%
Experiment 1.2	73.3%	69.2%	69.3%
Experiment 1.3	73.1%	67.8%	67.8%
Experiment 2.1	75.0%	74.0%	74.0%

For the second series of experiments, we applied our Stack-Overflow-specific Doc2Vec representation of the input (*Experiment 2.1*). We used the same test set as before but the training set was expanded and we took 1,031,998 samples for training. Although a new grid-search for optimal hyperparameters were carried out, the values for `learning_rate` and `momentum` from the previous experiments were retained. Interestingly, changing these parameters did not produce better results. With the above settings, we obtained a precision of 75%, a recall of 74% and an accuracy of 74%. The confusion matrix is shown on Figure 5.

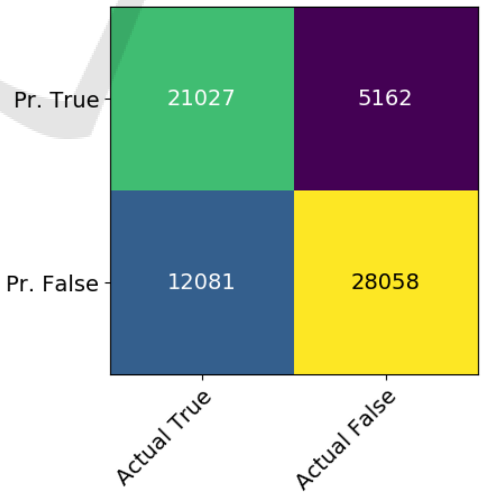

Figure 5: Confusion matrix for *Experiment 2.1*.

Ponzanelli et al. used various metrics for classification. They established three categories of metrics such as Stack-Overflow- or common-metrics, readability-metrics, and popularity-metrics. The last category is related to the questioners, and not the questions. The authors utilized a decision tree as well as a genetic algorithm to train a quality function expressing a measure of quality using different combinations of the metrics. They obtained precision values between 61.2% and 66.3% using the decision tree algorithm in combination with the common- and the readability-metrics. Based on common-metrics using the quality function they obtained a precision of 73.3% regarding the classification of low-quality questions. When only the readability-metrics were used, the results were less accurate; however, applying the popularity-metrics, they obtained exquisite precision on the right tail of the dataset (up to 90.1%). The main conclusion is that popularity-metrics can be considered as a better feature in deciding the quality of a given question.

Our best result of 75% for precision is slightly better than that obtained by Ponzanelli and coworkers using the linguistic-based-metrics. However, if popularity-metrics are also taken into consideration, the model by Ponzanelli et al. clearly outperforms our neural-network-based approach, and we can conclude that features related to user expertise are superior in predicting question quality.

Nevertheless, the precision of 75% and the recall of 74% obtained in this study indicate that natural language processing methods can be applied for prefiltering the questions prior to submission to Stack Overflow.

We also note here that there are a few drawbacks of the question classification based on well-defined quality metrics such as the scoring system or the accepted answer measure. Particularly, the scores do not always reflect the real quality because not every question is scored. Furthermore, accepting an answer is up to the question poster, and in many cases, even though a satisfactory and highly upvoted answer exists for a given question, it does not count as accepted because the user ignores to explicitly approve the fact of acceptance.

Another issue is that the quality of the questions is not always in line with the decision of closing. There might be cases where low-quality questions remain opened – especially in the case when the importance of the topic justifies it. Moreover, there might be high-quality questions and they still get closed because of another reason, such as the question is off-topic. Recommendation systems therefore should apply a more sophisticated and improved approach to address these exceptions.

5 RELATED WORK

The nature of the questions and answers at Stack Overflow has been in the focus of research in recent years. One of the lines of these studies is to follow the interest of the developers in different topics and their changes, to examine their relationship with current technological trends. Barua et al. (Barua et al., 2014) have investigated this relationship by applying Latent Dirichlet Allocation (LDA) to discover both the topics of the discussion and the trends of changing the interest of developers. It was found that web and mobile development has become hot topics in recent years. Other researchers have studied the personality traits of the users at Stack Overflow. Bazeli et al. (Bazelli et al., 2013) have investigated these traits based on the reputation of the users. They found that reputation is related to openness and other positive emotions. Ginsca et al. (Ginsca and Popescu, 2013) applied user profiling to determine whether it can be exploited to spot high-quality contributions.

One of the most studied areas is the automatic tagging based on the contents of the posts. NLP methods combined with machine learning were applied to support the appropriate tagging. Beyer et al. (Beyer et al., 2018) have classified the questions into seven categories using a machine learning method. These categories were defined considering the purpose of the questioner. Others such as Schuster et al. (Schuster et al., 2017) or Saini et al. (Saini and Tripathi, 2018) have focused on the prediction of the existing tags based on the words in the particular question.

Classifying the questions based on their quality has become an important research area in recent years. One of the pioneering studies from this area is the paper by Treude et al. (Treude et al., 2011), where the authors investigated whether a question received an accepted answer or remained unanswered. They introduced self-defined question categories and studied the number of posts receiving accepted answers in each category. It was found that some categories such as how-to-like instruction questions have been answered more frequently than other categories such as questions from novice programmers.

Correa et al. (Correa and Sureka, 2013) studied the effects of 19 features extracted from Stack Overflow posts using machine learning methods including Support Vector Machine, Naïve Bayes, Logistic Regression and Stochastic Gradient Boosted Trees, and they have achieved an accuracy of 73% in identifying a closed question. In another series of experiments, Correa and Sureka (Correa and Sureka, 2014) achieved 66% estimation accuracy with the help of Decision Tree and Adaboost classifiers during the investigation of the deleted Stack Overflow posts.

Ponzanelli et al. (Ponzanelli et al., 2014) studied various quality metrics of the questions submitted to Stack Overflow. In their experiments, the Decision Tree classifier and genetic algorithms were used to predict the values of the quality metrics, and precision values ranging from 62.1% to 76.2% were achieved. Yao et al. (Yao et al., 2013) investigated the correlation between the quality of the questions and answers applying a co-prediction method applying both regression and classification algorithms, which have been elaborated for non-linear optimization problems. They found a strong correlation for the quality between questions and answers.

Baltadzhieva and Chrupala (Baltadzhieva and Chrupała, 2015) studied the linguistic features of the questions to assess the extent of the influence of these features on the number of answers and score a question receives. They found that linguistic information improves the prediction accuracy of their models.

6 CONCLUDING REMARKS

Maintaining the quality at Q&A portals like Stack Overflow is a challenging task. While the popularity of these websites in terms of post numbers is increasing, the quality of the posts and questions submitted to them is decreasing. At the same time, checking and removing unsuitable questions place a growing burden on moderators, and efforts to automatize this task have produced moderate results. Therefore, developing procedures that can support the quality check with sufficient efficiency is still highly desirable.

In this paper, we presented a method based on natural language processing and deep learning procedures for classifying questions submitted to Stack Overflow by their quality. As previous studies indicated, natural language processing can provide invaluable support for the classification process. We considered only the linguistic features of the questions, those which are provided by the user during the post creation before the actual submission to Stack Overflow. This approach helps to elaborate methods that can support the user in pre-checking the quality of their questions.

As our results indicate, the semantics of the posts can also be caught by using a specific Doc2Vec representation. In this way, the classification can be performed relying solely on the textual information. This result, however, might further be improved in more than one ways. For example, the Doc2Vec representation can be enhanced by a more careful preprocessing technique. Additionally, introducing the LSTM network in our model is also expected to improve the accuracy of the classifier. These modifications are our next objectives along with the prediction of the possible actual closing reason via a multi-case classifier.

ACKNOWLEDGMENT

This research was supported in part by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002). The Ministry of Human Capacities, Hungary grant 20391-3/2018/FEKUSTRAT is acknowledged.

REFERENCES

Arora, P., Ganguly, D., and Jones, G. J. F. (2015). The Good, the Bad and their Kins: Identifying questions with negative scores in StackOverflow. In *Proc. of the 2015 IEEE/ACM - ASONAM '15*, pages 1232–1239.

Atwood, J. (2018). What does Stack Overflow want to be when it grows up?

Baltadzhieva, A. and Chrupała, G. (2015). Predicting the Quality of Questions on Stackoverflow. In *Proc. of the International Conference Recent Advances in Natural Language Processing*, pages 32–40.

Barua, A., Thomas, S. W., and Hassan, A. E. (2014). What are developers talking about? An analysis of topics and trends in Stack Overflow. *Empirical Software Engineering*.

Bazelli, B., Hindle, A., and Stroulia, E. (2013). On the Personality Traits of StackOverflow Users. In *2013 IEEE International Conference on Software Maintenance*, pages 460–463.

Beyer, S., Macho, C., Pinzger, M., and Di Penta, M. (2018). Automatically classifying posts into question categories on stack overflow. In *Proc. of the 26th Conference on Program Comprehension*, pages 211–221.

Correa, D. and Sureka, A. (2013). Fit or unfit: Analysis and prediction of 'closed questions' on stack overflow. In *Proc. of the First ACM Conference on Online Social Networks*.

Correa, D. and Sureka, A. (2014). Chaff from the wheat: Characterization and modeling of deleted questions on stack overflow. In *Proc. of the 23rd International Conference on World Wide Web*, pages 631–642.

Ginsca, A. L. and Popescu, A. (2013). User profiling for answer quality assessment in Q&A communities. In *Proc. of the 2103 workshop on Data-driven user behavioral modelling and mining from social media*, pages 25–28.

Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In Gordon, G., Dunson, D., and Dudík, M., editors, *Proceedings of the Fourteenth International Conference on Artificial Intelli-*

gence and Statistics, volume 15 of *Proceedings of Machine Learning Research*, pages 315–323, Fort Lauderdale, FL, USA.

Gulli, A. (2017). *Deep learning with Keras : implement neural networks with Keras on Theano and TensorFlow*.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France.

Irie, K., Tüske, Z., Alkhouli, T., Schlüter, R., and Ney, H. (2016). LSTM, GRU, highway and a bit of attention: An empirical overview for language modeling in speech recognition. In *Proc. of the Annual Conference of the INTERSPEECH*, volume 08-12-Sept.

Le, Q. V. and Mikolov, T. (2014). Distributed representations of sentences and documents. *CoRR*, abs/1405.4053.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119.

Ponzanelli, L., Mocci, A., Bacchelli, A., and Lanza, M. (2014). Understanding and Classifying the Quality of Technical Forum Questions. In *14th International Conference on Quality Software*, pages 343–352.

Saini, T. and Tripathi, S. (2018). Predicting tags for stack overflow questions using different classifiers. In *2018 4th International Conference on Recent Advances in Information Technology*, pages 1–5.

Schuster, S., Zhu, W., and Cheng, Y. (2017). Predicting Tags for StackOverflow Questions. In *Proc. of the LWDA 2017 Workshops: KDML, FGWM, IR, and FGDB*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Sutskever, I., Martens, J., Dahl, G., and Hinton, G. (2013). On the importance of initialization and momentum in deep learning. In Dasgupta, S. and McAllester, D., editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1139–1147, Atlanta, Georgia, USA.

Treude, C., Barzilay, O., and Storey, M.-A. (2011). How do programmers ask and answer questions on the web? In *Proc. of the 33rd ICSE*, page 804.

Yao, Y., Tong, H., Xie, T., Akoglu, L., Xu, F., and Lu, J. (2013). Want a good answer? ask a good question first! *CoRR*, abs/1311.6876.