2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining

The Good, the Bad and their Kins: Identifying Questions with Negative Scores in StackOverflow

Piyush Arora, Debasis Ganguly and Gareth J.F.Jones
ADAPT Centre, School of Computing
Dublin City University, Dublin, Ireland
Email: {parora,dganguly,gjones}@computing.dcu.ie

***Abstract*—A rapid increase in the number of questions posted on community question answering (CQA) forums is creating a need for automated methods of question quality moderation to improve the effectiveness of such forums in terms of response time and quality. Such automated approaches should aim to classify questions as *good* or *bad* for a particular forum as soon as they are posted based on the guidelines and quality standards defined/listed by the forum. Thus, if a question meets the standard of the forum then it is classified as *good* else we classify it as *bad*. In this paper, we propose a method to address this problem of question classification by retrieving *similar* questions previously asked in the same forum, and then using the text from these previously asked similar questions to predict the quality of the current question. We empirically validate our proposed approach on the set of StackOverflow data, a massive CQA forum for programmers, comprising of about $8M$ questions. With the use of these additional text retrieved from similar questions, we are able to improve the question quality prediction accuracy by about 2.8% and improve the recall of negatively scored questions by about 4.2%. This improvement of 4.2% in recall would be helpful in automatically flagging questions as bad (unsuitable) for the forum and will speed up the moderation process thus saving time and human effort.**

I. Introduction

The driving force of human intellect is the ever increasing desire to discover, learn and know more about different topics and find answers to problems with mutual collaboration. Interactive websites for community based question answering (CQA) provide opportunities to ask questions ranging from critical topics related to health, education and finance to recreational queries for the purpose of fun and enjoyment etc. The wide range of diversity of questions in a general CQA forum can be seen with the following examples, which show the two most viewed questions[1] in StackExchange[2], a hugely popular CQA forum,: a) *Where is the quietest place on Earth ?* b) *What are the conditions in which a creature would evolve more than one brain?* .

Recent years have witnessed an upsurge in the popularity of CQA forums such as the StackExchange, Yahoo! Answers[3], Quora[4] etc. The software programmers' community, called the StackOverflow[5] (SO), is the largest in the StackExchange CQA forum. SO is dedicated to providing programmers with the assistance of a knowledge-base of high quality answers to programming related questions. At the time of writing this paper, the number of questions in SO is over 8 million, the number of answers over 14 million, and the number of users over 3 million. Such large numbers mean that it is of utmost importance to provide an effective quality control mechanism of the user generated content of the forum in order to maintain its integrity [1].

SO allows users to provide their views on the questions and answers by either voting up or down. Voting up is how the community indicates which questions and answers are most useful and appropriate.[6] Voting down, also known as casting downvotes, is how the community indicates which questions and answers are least useful[7] to the community.

It is important for CQA forums to maintain a satisfactory quality level for the questions and the discussions (answers and comments) so as to improve the site reputation and provide better user experience [2]. The aim is to avoid repetitive and unclear questions. In fact, SO prescribes a comprehensive set of guidelines which a newly asked question should adhere to. A *good* question asked on this forum should involve one of the following[8].

- A specific programming problem
- A software algorithm
- Software tools commonly used by programmers
- Practical, answerable problems that are unique to software development.

A question is good if it is presented clearly as well as describes a specific programming problem. However, despite the detailed guidelines, a significant number of questions submitted to SO are of low quality [3]. Moderators delete most poor quality questions with high negative scores (the score of a question is the difference between the number of up votes and down votes). Sometimes the moderators may decide to close[9] an inappropriate question instead of deleting it if the question does not require any further discussion/response by the community. A question can be closed if it is one of these[10]:

- Near or exact duplicate to an earlier question
- Off-topic
- Subjective and argumentative
- Not a real question

[1] Based on number of views in a day
[2] http://stackexchange.com/
[3] https://answers.yahoo.com
[4] https://www.quora.com/
[5] http://stackoverflow.com/
[6] http://stackoverflow.com/help/privileges/vote-up
[7] http://stackoverflow.com/help/privileges/vote-down
[8] http://stackoverflow.com/help/on-topic
[9] which means that no additional answers may be posted to the question
[10] http://stackoverflow.com/help/closed-questions

ASONAM '15, August 25-28, 2015, Paris, France
 ISBN 978-1-4503-3854-7/15/08 $15.00
DOI: http://dx.doi.org/10.1145/2808797.2809318

TABLE I
THREE RELATED QUESTIONS EACH FOR A *good* AND A *bad* SO QUESTION.

Question	**Score**
Why does this code using random strings print "hello world"?	857
Related Question List	**Score**
How to generate a random alpha-numeric string	435
Python random string generation with upper case letters and digits	318
Why does the use of Random with a hard-coded seed always produce the same results?	12
Question	**Score**
How to send 100,000 emails weekly?	-147
Related Question List	**Score**
Send a daily mailing list of 50,000 mails	-1
What PHP mail library can I use to send hundreds of e-mails daily via Gmail?	0
PHP How to send 100 emails at once by PHP?	-1

- Too localized (less likely to be useful to others)
- Noise or pointless.

Table I illustrates the question quality variation of SO with an example of a very highly rated question and a very lowly rated one. The inappropriate question shown in Table I is closed by the moderators and has been classified as a "not constructive" question. The forum expects answers to be supported by facts, references, or expertise; but this particular question may potentially lead to debate, arguments, polling, or extended discussion. The good question in our example, on the other hand, is associated with a precise information need. This type of question is considered to add value to community knowledge.

The motivation for our work is that an automated process of alerting moderators on creating inappropriate questions can potentially be helpful in reducing the manual effort for quality maintenance of the CQA forums. In this paper, we attempt to automatically predict whether a question will receive a negative net score (i.e. whether it will receive more down votes than up votes) by treating this problem as a binary (two class) classification problem. We refer to such questions as inappropriate (bad) for the forum, while the ones with positive net scores are considered suitable (good). We follow the classification approach using votes as gold truth as they reflect the community feedback that whether a given question or answer is most or least useful to the forum. The central theme of our work is to automatically identify such inappropriate questions as soon as they are posted on to the forum.

A major challenge in this task is the *cold start problem*, i.e. a new question needs to be classified (automatically) despite the lack of community feedback such as votes, comments or answers, associated with it. Although for a new question, this community feedback information is not readily available, it is possible to use information from previously asked questions that are similar in content and theme to the new one, and potentially improve the question classification.

We hypothesize that content features extracted from similar questions asked previously can be used to enrich the features of the current question, which in turn can potentially help to better predict its quality. In principle, this is somewhat similar to document expansion in information retrieval (IR), where a short document is expanded with the textual content from other documents in order to improve its informativeness and retrievability [4]. However, we emphasize that the our end objective in our case is to expand the text of the current document with text from other similar documents in order to *improve its classification accuracy*. We focus only on content features, since this makes the approach easily extensible and scalable to other CQA forums such as Ubuntu Forums[11], Yahoo Answers[12] etc., where features such as votes, favourite count, user reputation etc., are not present, but we can easily avail of the textual information.

Contributions of this paper. This paper proposes a novel method for improving the question quality prediction accuracy of a CQA forum by making use of content extracted from previously asked similar questions in the forum. We investigate various document (questions previously asked in the forum) and query (current question to be classified as good or bad) representation alternatives and different retrieval models for retrieving the set of similar questions. We show that the performance of the question classification tasks depends on how effectively we retrieve this set of similar questions.

The rest of the paper is organized as follows. In Section II, we review relevant prior art and comment on the differences of these with our own work. Section III presents our approach to improve question quality prediction. Section IV describes the characteristics of the dataset used in our experiments and discusses the experimental settings. Section V presents and discusses the results of our experiments. Finally, Section VI concludes the paper with directions for future work.

II. RELATED WORK

In this section we discuss some of the previous work done on CQA forums. Prior work related to CQA can broadly be categorized into two types:

- selecting the best answer to a question [5], [6], [7], or ranking the answers to a question [8]; and
- predicting the quality of a question [3], [9], [10], [11], [2], [12], [13], [14].

We have divided the related work into two parts, first discussing the answer selection and second focusing on question quality prediction. The authors in [5] developed a regression model for predicting the quality of an answer on the Yahoo! Quest dataset. They used a combination of content-based and community feedback based features such as the length of a question's subject and body, number of answers and comments, length of the content of answers, references within answers, ranks of answers, and information from an answerer's profile such as his reputation points, number of (best) answers entered by him etc. A logistic regression model was then trained on these features to predict a model for the answer scores. More recent work which selects the best answer of a question is described in [15], where the best answer is selected using a

[11]http://ubuntuforums.org/
[12]https://answers.yahoo.com

classifier. In addition to the features used in [5], [15] also takes into account the *topic entropy* and *topical reputation* features of users, which respectively refer to the distribution of a user's post across different tags (treated as topics) and a measure of a user's reputation with respect to a particular tag.

Our work is related more to question quality prediction than answer rating. A major difference between the answer rating task and the question quality prediction task is that while the former is free to use community feedback features extracted from posts, the latter is more restrictive in nature due to the practical assumption that community feedback is not available for a newly posted question. Despite this argument, most of the previous reported work in question quality prediction does utilize some of this information, for example:

- by making use of the number of answers and the sum of scores of answers to a question etc. [9];
- applying the number of question views, answer age, number of comments etc. as features in their classifier [15];
- classifying SO questions into six categories such as factoid, definition, opinion etc. by using statistical features such as length, part-of-speech (POS) tags etc. from the answers [16].

The authors in [2] developed a model for predicting question quality using only the content of the question. Along with the unigram features they explored global and local topic modeling over the question text. They tried different combinations of textual and topic modelling features and found that a combination of content and local topic modelling performs best for the question quality prediction. The work reported in [2] is somewhat similar to our work. In our work, we go beyond unigrams and explore word n-grams (n up to 3) extracted from the textual content of a question.

Apart from this there have been several studies focussing on different aspects of StackOverflow posts – nature of StackOverflow questions [17], reasons for unanswered question [18], rerouting of questions [19], experts identification [20], understanding the edits made in the question [21], analyzing the reputation management for the StackOverflow forum [22].

Instead of extracting information from answers to the current question and thereby violating the practical constraint (i.e. for a new question, the answers are not readily available), we in our work rely on extracting questions that are similar in content to it. Given a question we aim to find similar questions in the forum that have been asked previously. Using the content features from the question and similar question we predict the quality of the new question. The major difference between our work and all previous approaches, introduced in this section, is that none of them investigated the effect of including features from similar questions, to predict whether a question is good or bad. To the best of our knowledge, this is the first work investigating the role of similar questions for deciding the quality of a new question.

III. Proposed Methodology for Question Classification

In this section, we describe the details of our proposed methodology. In particular, we treat the prediction of whether a new question is likely to receive a net negative score as a binary classification task. Since the focus of our work is to investigate the usefulness of document (current question) expansion for improving question quality prediction, we make use of text based features, similar to [2]. Following the classical text classification approach, we use the multinomial Bayes model [23] of word n-grams counts as features extracted from the SO questions.

The intuition behind using the textual feature is that the language and vocabulary of the question can indicate its quality (i.e. good or bad). The hypothesis is that the two classes of questions, i.e. good and bad, should have considerably distinct characteristic word distributions. For example, in SO, the good questions should contain program code snippets, whereas the bad questions would seldom have them.

To motivate the idea of question expansion, we argue that the text of a current question may not have sufficient information to accurately classify it. The classification effectiveness can potentially be improved if the text from other questions, similar to the current one, can be used to train a text classifier, e.g. Naive Bayes.

Let D be the current question that needs to be classified into one the classes $C_1 \dots C_k$. For our problem, k=2, i.e. there are two classes. The posterior probability of the class of a question D, denoted by $P(C_i|D)$, is given in terms of the priors as shown in Equation 1.

$$P(C_i|D) = \frac{P(D|C_i).P(C_i)}{\sum_k P(D|C_k).P(C_k)} \tag{1}$$

The class priors for a document, i.e. the $P(D|C_i)$ values, are estimated using the maximum likelihood estimates (MLE) computed over its constituent terms, or more generally speaking the n-grams. It is a common practice to employ additive smoothing to assign non-zero MLE estimates for unseen terms in a class [23], as shown in Equation 2.

$$P(D|C_i) = \prod_{t \in D} P(t|C_i) = \prod_{t \in x} \frac{n(t, C_i) + 1}{\sum_{t'} n(t', C_i) + k} \tag{2}$$

From Equation 2, it can be seen that for a text classification problem involving short documents, the MLE class priors may be unreliable due to the small $n(t, C_i)$ values.

This problem of short documents in the context of IR, has been shown to result in poor retrievability [4]. To address this problem in IR, short text documents, such as microblogs [4] or spoken documents [24], can be expanded by making use of the content from other similar documents in the collection.

In the context of our problem, we propose to use document expansion for improving the MLE estimates for the class priors, which in turn may potentially lead to better classification effectiveness. We treat the current document (SO question) as the query and retrieve a ranked list of previously asked questions from an indexed collection of questions. The top N documents from this retrieved set are used as the neighbourhood of $N(D)$ for expanding D. The term weight of a term w in the vocabulary of $N(D)$ is set to its normalized term frequency, i.e the ratio of raw term frequency to document length.

IV. Experimental Setup

In this section, we first describe the characteristics of the StackOverflow dataset used in our experiments. We then describe our experimental settings to retrieve the neighbourhood set $N(D)$ of a question D for the classification.

A. Dataset Description

We used a recent StackOverflow data dump (released in 2014) from StackExchange platform[13] for our experiments. This data dump has questions ranging from year 2008–2014 thus spanning a period of almost over 6 years.

TABLE II
Input Data statistics

All Questions	7990787
Answers	13684117
Question with accepted answers	4596855
Question with no answers	921222
Total Posts	21674904

Overall, the data dump has 21.67 million posts consisting of 7.9 million questions and 13.68 million answers. (see Table II). Each post is an XML file comprising of different textual fields such as the title, body, tags etc. in addition to other community feedback based fields such as scores, number of views, favourite count etc. For our experiments, we only use the textual field so as to make our method general enough to be applicable for other text-only CQA forums[14].

In Figure 1, we show the year wise distribution of questions with score > 0, score < 0 and score $= 0$[15]. The figure shows a steep increase in the number of negatively scored questions, thus making the automatic moderation of such questions an important problem to be addressed. Moreover, the quality of questions has deteriorated in the last couple of years, as can be seen from Figure 1, that the number of questions with zero scores have outnumbered the questions with positive scores between year 2012 and 2013.

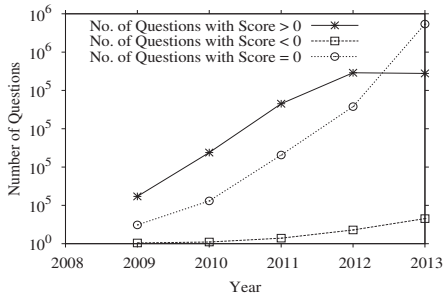

Fig. 1. Per-year frequency of positively, negatively and zero scored questions.

Despite the rapid increase in the number of negatively scored questions as shown in Figure 1, there is still a large difference between the number of questions that receive positive scores and those which receive negative ones, as shown in Table III which indicates that the data set is quite heterogeneous in nature.

TABLE III
Question score and view statistics.

Category	Questions	
	Total	Views $> 1K$
Score Negative	380800	30163
Score Zero	3829686	383053
Score Positive	3780301	1315731

B. Selecting a subset for the classification experiments

In this section, we describe how we chose the training and the test sets for our question classification experiments. Analogous to [2], we reason that the scores received by questions with less than 1000 views are statistically unreliable, and hence should not be used to train a supervised classification model. It can therefore be assumed that instead of relying on the score of a question alone, one should take into account a combined contribution from both the scores and the views as a quantitative measure for question quality.

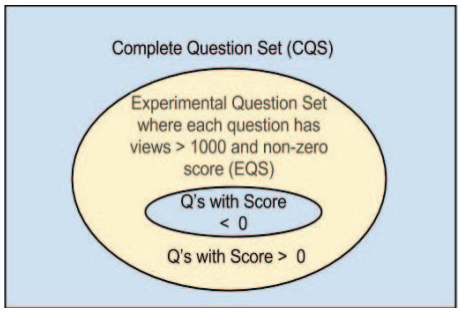

Fig. 2. Experimental Data Selection

For the purpose of our experiments, we created a subset, which we call EQ, to denote the set of questions with at least a thousand views. Note that to obtain meaningful comparisons, our intention is to experiment only on those questions that have received considerable user attention and enough views (at least a thousand) to be up-voted or down-voted. This subset is shown in Figure 2.

In the set EQ of questions, we divide the questions into two classes a) positive, and b) negative. The positive class in this case refers to questions which eventually get positive net scores, whereas the negative ones are those which receive negative net scores. Moreover, for our experiments we discard questions which have zero scores because such questions can be considered as neutral, i.e. neither good nor bad. The distribution of questions in the EQ set can be seen from the first and third rows of Table III (zero score questions shown in Table III are not a part of the EQ set).

We use only the documents in the EQ set for our experiments. Each document in this set, comprised of the title, body and the tags, is indexed with the help of Lucene[16], an open-source retrieval framework, implemented in Java. The textual

[13] https://archive.org/details/stackexchange

[14] e.g. http://ubuntuforums.org/

[15] We do not show the statistics for the years 2008 and 2014 because they are incomplete (not covering full 12 months)

[16] http://lucene.apache.org/core/4_4_0/

content of each field is passed through the *EnglishAnalyzer* Lucene analyzer, which performs the standard steps of stop-word removal and Porter stemming. We then use the Lucene indexed EQ set for retrieval of similar questions given a current question.

C. Question Expansion Settings

In Section III, we presented our strategy for document expansion for text categorization from a formal view-point. In this section, we describe the experimental settings for question expansion in the context of the SO question quality prediction task that we investigate.

Our objective is to improve question classification accuracy, for which we study the following.

- Which field (title alone, or title with body) should be used for formulating the queries for retrieving similar questions?
- Which fields from the indexed questions should be considered for computing the retrieval similarities?
- What retrieval model should be used?
- What is the size of the neighbourhood, i.e. the number of similar questions that should be used for expanding a question?
- Which n-gram features should be used to train a text classifier on the expanded questions?

Given a new question, we treat it as a query and conduct a search over the indexed SO question collection (as described in Section IV-B). Since a SO question consists of different fields, we experimented with various field selection based query formulation and document retrieval strategies, as described below (also shown in Figure 3):

- **T_T**: Use only the title of the current question as a query to search in only the title field of the indexed questions.
- **T_TB**: Use only the title of the current question as a query to search in the title and the body fields of the indexed questions.
- **TB_TB**: Use both the title and body of the current question as a query to search in the title and body fields of the indexed questions.

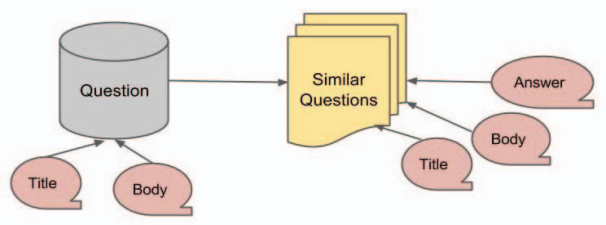

Fig. 3. Given a new question, extraction of similar questions and the fields associated with it

Using all different settings with respect to the query and search field, we extract k similar questions (where we vary k such that $k \in \{1, 3, 5, 9\}$). While extracting k similar questions from the ranked list of retrieved questions, we apply a filter to make sure that only previously asked questions are considered for question expansion so as to cater for the real use-case scenario.

We employed several retrieval algorithms to perform the search for related questions. In particular, we used the BM25, LM and tf-idf retrieval models. The set of similar questions, obtained with various combinations of the retrieval models, was then used to expand the current question. We experimented with selectively adding text from the similar questions for expansion, i.e. in one variant we included just the title and body of the similar questions, whereas in the other along with title and body we also included answers to the similar question for expansion. Initial experiments showed that including answers of the similar questions appear to always degrade the results as compared to performing expansion without including answers. Expansion using answers most likely results in a drift from the main content and focus of the question which in turn leads to lower prediction accuracy. Consequently, we focus only on adding title and body of the similar questions during the expansion phase for our investigations. In addition to the retrieval models, we also experimented with different values of n while considering the n word-gram features for the text classification.

It is worth mentioning here that we do not intend to measure the retrieval effectiveness directly in terms of standard IR metrics, such as MAP or nDCG, because firstly it is not the core objective of our work, and secondly due to the fact that computing such metrics requires a set of manually assesed relevant documents for each query, which in our case is not available. We do however quantify the usefulness of extracting information from similar questions by reporting improvements in the classification effectiveness.

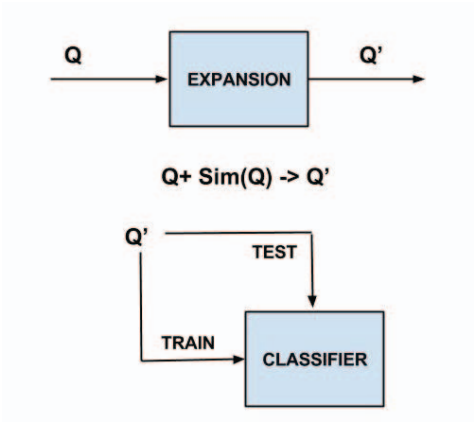

Fig. 4. Flow chart describing Question Expansion and Classification steps

D. Train/test a text classifier with the expanded questions

In Figure 4, we illustrate the information flow of our approach. First we perform question expansion as discussed in Section IV-C, then we divide the expanded set of questions (Q') into two parts – training and test. We learn a classifier model using the training set and evaluate the performance on the test set.

We use a multinomial Naive Bayes (NB) classifier with additive smoothing [23]. Text classification on the question text itself constitutes the standard baseline approach. Additionally, we expand every question in the training set to train an NB classifier with modified prior estimates; the difference arises

due to the modified term weights in the expanded documents. In addition, we perform document expansion for each test set instance.

We conduct our experiments using the scikit-sklearn[17] tool. We perform 10 fold cross-validation over our dataset comprising of an equal number of positive and negative question samples to avoid over-fitting. In all our experiments, results are presented using a total of 2000 random samples from the EQ set with equal number of positively and negatively scored questions averaged over 10 sampling phases. Such random sampling and averaging of the results to avoid random effects in the reported results is quite common in classification tasks with a large number of instances, see e.g. [3].

For all our experiments, in addition to accuracy, we also report the precision, recall and the F-measure with respect to the negative class (i.e. the negatively scored questions). The reason we report these metrics with respect to the negative class is that in this prediction task, it is rather undesirable to predict a truly inappropriate question as an appropriate one. Since only the bad questions should raise an alarm to the moderators, such misclassifications of bad questions as good ones would not trigger manual intervention. On the other hand, misclassifications in the other direction, i.e. reporting an appropriate question as inappropriate, can be corrected by the moderators since they would receive the notifications in such cases.

E. Baselines

In this section, we describe the various baseline approaches that we compare our method against. In particular, the baselines used in our experiments are outlined as follows.

- BL_NB: This is the standard text classification approach using the text of the questions (title and body) using the NB classifier.
- BL_KNN: This is the standard K-NN approach of classification, in which a test instance is assigned the class into which most of its neighbours belong. In our case, this neighbourhood comprises of the top most similar questions to a current question. For example, in the three neighbourhoods of a question, if two are positive and one is negative then the current test question is assigned a positive class.
- BL_SCORE: This baseline aggregates the scores of the similar questions and assigns a positive class to the current test question if the aggregate is positive, and assigns it a negative class otherwise.

The first baseline, BL_NB, does not use information from similar questions for classification. The second and the third baselines use the information from similar questions for classification. While the first of them, BL_KNN, relies on the question class, the second, BL_SCORE, relies on the assigned scores to predict the classification output.

V. Results

In this section, we report the results of the experiments with our proposed approach in comparison to the baselines outlined in Section IV-E.

[17] http://scikit-learn.org/stable/

TABLE IV
Classification effectiveness of the baselines, where k = 3 for BL_KNN and BL_SCORE .

Run Name	n-gram	10-fold CV Measures			
		Accuracy	Precision	Recall	F-measure
BL_NB	1	**71.60**	0.802	**0.616**	0.697
BL_NB	2	71.25	0.832	0.580	0.684
BL_NB	3	69.65	0.861	0.514	0.644
BL_KNN	-	51.50	0.928	0.039	0.074
BL_SCORE	-	51.60	0.880	0.037	0.071

First, in Table IV, we report the classification results achieved with our baselines. BL_NB, i.e. the standard text classification approach with NB classifier, turns out to be most effective baseline. The accuracy values, which we get with this standard approach of text classification, are satisfactory. In fact, the results are comparable to those reported in [2]. This confirms that our baseline BL_NB is indeed a strong one.

The other two baselines that rely on the similar questions for classification (where k = 3)[18] do not perform well. The most likely reason for this poor performance is the non-uniformity in the number of positively scored questions in comparison to the number of questions with negative scores. This shows that using the scores of questions in the set of similar questions alone is not a good estimator for predicting the quality of a new question.

TABLE V
Accuracy values obtained with NB classifier on expanded questions, where k = 1 and word n-grams = 1.

Approach	Tf-Idf	LM	BM25
T_T	69.80	69.60	70.45
T_TB	70.75	72.45	**72.70**
TB_TB	71.45	71.65	71.85

Initial experiments revealed that the best combination while searching across different fields is when the query consists of just the *title* of the question and is searched over *title* + *body* fields of the question (which confirms the previous finding in [25]). Initial results with different field combinations are shown in Table V (where k = 1 and word n-grams = 1 i.e. considering only unigram). Consequently for the rest of our experiments we took the best setting and search title of the question in indexed title+body fields.

We now report the results obtained with the NB classifier on the expanded questions. To systematically explore the parameter space, we first report the results obtained with different retrieval methodologies on different field combinations for query formulation and retrieval. The results are shown in Table V. The results obtained with the tf-idf approach for question expansion are lower than the baseline. However, the results obtained with standard retrieval models, i.e. LM and BM25, perform better than the baseline BL_NB, which shows that

[18] k = 3 performs best as compared to higher values of k for BL_KNN and BL_SCORE

document expansion can play a crucial role in improving the effectiveness of text classification, for short text.

Next, we investigate the number of questions that one may use as the set of similar questions for expansion. Since, the tf-idf method for retrieving the set of similar questions does not work well, we do not use this method of retrieval in the next set of experiments, where we aim to optimize the number of questions used for expansion. The results for this set of experiments are presented in Table VI.

TABLE VI
INVESTIGATING THE NUMBER OF SIMILAR QUESTIONS (k) AND THE WORD n-GRAMS FOR EXPANSION.

Parameters		n-grams		
k	IR	1	2	3
1	LM	72.45	73.30	71.65
1	BM25	72.70	73.25	71.25
3	LM	72.95	72.95	72.30
3	BM25	71.80	**73.65**	72.90
5	LM	72.85	71.85	72.50
5	BM25	73.05	73.40	73.00
9	LM	72.30	73.40	73.50
9	BM25	73.00	73.00	73.15

Out of the different word n-grams used as text classifier features, it can be seen from Table VI that using $n = 2$, i.e. word unigrams and bigrams performs the best. Table VI also shows that the best results are obtained with 3 documents for expansion.

In addition to classification accuracy, we now report the class specific precision and recall values obtained with the best baseline and our proposed approaches. We compare the results of BL_NB with that of the best settings obtained with our method (where $k = 3$, BM25 and word n-gram = 2, bold-faced in Table VI).

TABLE VII
CLASS (GOOD OR BAD QUESTION TYPE) SPECIFIC PRECISION/RECALLS.

Approach	Class	Precision	Recall	F Measure
BL_NB	+ve	0.688	0.848	0.760
	-ve	0.802	**0.616**	0.697
Expansion	+ve	0.702	0.843	0.766
($k = 3$, BM25)	-ve	0.804	**0.642**	0.714

The class specific precision/recall values are shown in Table VII. In the context of CQA question quality prediction, the most important observation (bold-faced in the table) is the increase in the recall of the negative class, i.e. bad questions asked on the SO forum. This suggests that our proposed document expansion method for text classification is able to correctly identify more instances of negatively scored questions. This is important because identifying such questions more effectively makes the automatic moderation process easier.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we addressed the problem of question quality prediction in a CQA forum. In particular, we experimented with StackOverflow questions, where we attempted to automatically predict questions that are likely to receive a net negative score. This method can potentially be applied for automatic moderation of question quality in StackOverflow or other such CQA forums. In this paper, we make use only of the text based features using a Naive Bayes classifier for addressing this problem of quality prediction. The use of text features alone ensures that our proposed method of question quality classification can be applied for other more loosely structured CQA forums as well.

A problem with the text based features is that they does not work quite well for short texts. Our proposed approach relies on a document expansion method applied on the StackOverflow questions in order to improve the classification effectiveness. For the expansion, we make use of the text of the current question by executing it as a query to retrieve a ranked list of other previously asked *similar* questions from the forum. Our method is able to improve classification accuracy by about 2.8%. Most importantly, our proposed method is able to improve the recall of negatively scored questions by about 4.2%, which implies that for more such questions an alarm can be raised for the moderators.

In future, we would like to study whether incorporating additional feedback information from similar questions such as scores, favourite vote can improve the quality effectiveness for CQA. Further, we would like to investigate the questioner's information and previous interactions for quality prediction.

ACKNOWLEDGMENT

This research is supported by Science Foundation Ireland (SFI) as a part of the CNGL Centre for Global Intelligent Content at DCU (Grant No: 12/CE/I2267).

REFERENCES

[1] L. Mamykina, B. Manoim, M. Mittal, G. Hripcsak, and B. Hartmann, "Design Lessons from the Fastest Q&A Site in the West," in *CHI '11*, pp. 2857–2866.

[2] S. Ravi, B. Pang, V. Rastogi, and R. Kumar, "Great Question! Question Quality in Community Q&A," in *Proc. of ICWSM '14*, 2014.

[3] D. Correa and A. Sureka, "Chaff from the wheat: characterization and modeling of deleted questions on stack overflow," in *Proceedings of WWW '14*, 2014, pp. 631–642.

[4] M. Efron, P. Organisciak, and K. Fenlon, "Improving retrieval of short texts through document expansion," in *Proceedings of the SIGIR '12*, 2012, pp. 911–920.

[5] C. Shah and J. Pomerantz, "Evaluating and Predicting Answer Quality in Community QA," in *Proceedings of SIGIR '10*, 2010, pp. 411–418.

[6] Q. Tian, P. Zhang, and B. Li, "Towards predicting the best answers in community-based question-answering services." in *Proceedings of ICWSM '13*, 2013.

[7] J. Jeon, W. B. Croft, J. H. Lee, and S. Park, "A framework to predict the quality of answers with non-textual features," in *Proc. of SIGIR '06*, 2006, pp. 228–235.

[8] D. H. Dalip, M. A. Gonçalves, M. Cristo, and P. Calado, "Exploiting user feedback to learn to rank answers in Q&A forums: a case study with stack overflow," in *Proceedings of SIGIR '13*, 2013, pp. 543–552.

[9] A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec, "Discovering Value from Community Activity on Focused Question Answering Sites: A Case Study of Stack Overflow," in *Proceedings of KDD '12*, pp. 850–858.

[10] B. Li, T. Jin, M. R. Lyu, I. King, and B. Mak, "Analyzing and Predicting Question Quality in Community Question Answering Services," in *Proceedings of CQA '12 Workshop*, ser. WWW '12 Companion, 2012, pp. 775–782.

[11] D. Correa and A. Sureka, "Fit or Unfit: Analysis and Prediction of 'Closed Questions' on Stack Overflow," in *Proceedings of COSN '13*, 2013, pp. 201–212.

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

[13] L. Ponzanelli, A. Mocci, A. Bacchelli, and M. Lanza, "Understanding and classifying the quality of technical forum questions," Univ. of Lugano, Tech. Rep. 2014/02, Jun. 2014.

[14] J. Liu, Q. Wang, C.-Y. Lin, and H.-W. Hon, "Question difficulty estimation in community question answering services," in *Proc. of EMNLP*, October 2013, pp. 85–90.

[15] G. Burel, Y. He, and H. Alani, "Automatic Identification of Best Answers in Online Enquiry Communities," in *Proceedings of ESWC'12*, 2012, pp. 514–529.

[16] H. Toba, Z.-Y. Ming, M. Adriani, and T.-S. Chua, "Discovering high quality answers in community question answering archives using a hierarchy of classifiers," *Inf. Sci.*, vol. 261, pp. 101–115, 2014.

[17] C. Treude, O. Barzilay, and M.-A. Storey, "How do programmers ask and answer questions on the web?: Nier track," in *Software Engineering (ICSE), 2011 33rd International Conference on*. IEEE, 2011, pp. 804–807.

[18] M. Asaduzzaman, A. S. Mashiyat, C. K. Roy, and K. A. Schneider, "Answering questions about unanswered questions of stack overflow," in *Proceedings of the 10th Working Conference on Mining Software Repositories*, ser. MSR '13, 2013, pp. 97–100.

[19] S. Chang and A. Pal, "Routing questions for collaborative answering in community question answering," in *Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on*. IEEE, 2013, pp. 494–501.

[20] X. Liu, W. B. Croft, and M. B. Koll, "Finding experts in community-based question-answering services," in *Proceedings of CIKM '05*, 2005, pp. 315–316.

[21] J. Yang, C. Hauff, A. Bozzon, and G.-J. Houben, "Asking the right question in collaborative q&a systems," in *Proceedings of the 25th ACM Conference on Hypertext and Social Media*, ser. HT '14. New York, NY, USA: ACM, 2014, pp. 179–189.

[22] A. Bosu, C. S. Corley, D. Heaton, D. Chatterji, J. C. Carver, and N. A. Kraft, "Building reputation in stackoverflow: An empirical investigation," in *Proceedings of the 10th Working Conference on Mining Software Repositories*, ser. MSR '13, 2013, pp. 89–92.

[23] J. D. Rennie, L. Shih, J. Teevan, and D. R. Karger, "Tackling the poor assumptions of naive bayes text classifiers," in *Proc. of ICML '03*, 2003, pp. 616–623.

[24] D. Ganguly, J. Leveling, and G. J. F. Jones, "An lda-smoothed relevance model for document expansion: a case study for spoken document retrieval," in *Proc. of SIGIR '13*, 2013, pp. 1057–1060.

[25] J. Jeon, W. B. Croft, and J. H. Lee, "Finding similar questions in large question and answer archives," in *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, ser. CIKM '05, 2005, pp. 84–90.