Exploiting User Feedback to Learn to Rank Answers in Q&A Forums: a Case Study with Stack Overflow

Daniel Hasan Dalip,
Marcos André Gonçalves
Dept of Computer Science - UFMG
Belo Horizonte/MG, Brazil
{hasan,mgoncalv}@dcc.ufmg.br

Marco Cristo
Institute of Computing - UFAM
Manaus/AM, Brazil
marco.cristo@ic.ufam.edu.br

Pável Calado
Instituto Superior Técnico/
INESC-ID
Porto Salvo, Portugal
pavel.calado@tagus.ist.utl.pt

ABSTRACT

Collaborative web sites, such as collaborative encyclopedias, blogs, and forums, are characterized by a loose edit control, which allows anyone to freely edit their content. As a consequence, the quality of this content raises much concern. To deal with this, many sites adopt manual quality control mechanisms. However, given their size and change rate, manual assessment strategies do not scale and content that is new or unpopular is seldom reviewed. This has a negative impact on the many services provided, such as ranking and recommendation. To tackle with this problem, we propose a learning to rank (L2R) approach for ranking answers in Q&A forums. In particular, we adopt an approach based on Random Forests and represent query and answer pairs using eight different groups of features. Some of these features are used in the Q&A domain for the first time. Our L2R method was trained to learn the answer rating, based on the feedback users give to answers in Q&A forums. Using the proposed method, we were able (i) to outperform a state of the art baseline with gains of up to 21% in NDCG, a metric used to evaluate rankings; we also conducted a comprehensive study of the features, showing that (ii) review and user features are the most important in the Q&A domain although text features are useful for assessing quality of new answers; and (iii) the best set of new features we proposed was able to yield the best quality rankings.

Categories and Subject Descriptors

H.3.7 [**Information Storage and Retrieval**]: [User Issues]

Keywords

Content Quality Assessment; Q&A Forums; Answer Quality; Learning to Rank

1. INTRODUCTION

The Web 2.0 phenomena has brought deep changes to the Internet, as users are now able not only to consume, but also to produce content. This change gave rise to new ways of creating knowledge. Many web sites, such as collaborative encyclopedias, blogs, and forums, allow users to contribute to their content, thus creating large collaborative knowledge repositories. These repositories are characterized by a loose edit control, which allows anyone to freely edit almost anything. As a consequence, the varying quality of their content has raised much concern.

To deal with this problem, many collaborative sites adopt quality control mechanisms, where the users can indicate the quality and appropriateness of the content and even the reputation of the editors. However, such manual assessment not only does not scale to the current rate of growth and change of these systems, but it is also subject to human bias, which can be influenced by the varying background, expertise, and even a tendency for abuse.

Thus, many services provided by these sites, such as ranking, recommendation, and even the manual quality assessment itself, would benefit from the adoption of automated or semi-automated quality control mechanisms. Motivated by that idea, in this work, we propose new strategies for rating content, based on statistical evidences for quality and reputation. In particular, we focus on Question and Answer Forums (Q&A Forums), given their growing importance as source of specialized information on the web.

In Q&A Forums, a user (asker) can post a question about a certain topic for which he/she receives answers from other users. Normally, any user can label a particular answer as useful or not, while the asker can indicate the one he/she considers the best. Figure 1 illustrates the main elements of a Q&A Forum, here using the particular case of Stack Overflow[1]. As we can see, in Stack Overflow, any user can annotate whether an answer is useful or not, and vote for it favorably (upvote) or not (downvote). The asker can place a mark (a green "tick") on the answer he/she considers the best one.

In forums such as Stack Overflow, the answers are expected to be correct and should be ranked according to their quality. According to the Stack Overflow guide[2], a good answer, besides being correct, should be clear, provide examples, quote relevant material, be updated, and link to more information and further reading. Since "quality" is a subjective feature, it is inferred from the opinion of the asker and from the votes received from the other users. Normally, the answer at the top position is the best one, if it has already been indicated as so by the asker.

While this strategy is effective in deemphasizing the bad quality answers, it is somehow dependent on the asker's selection of the best answer. As a consequence, for many questions, a good quality ranking of the answers is not provided, since in many cases (a) the asker takes much time to choose the best answer, (b) he/she does not choose it at all or (c) after choosing it, other answers are improved to the point of becoming better than the previously selected as the best. In fact, in Q&A Forums there may not be a single

SIGIR'13, July 28–August 1, 2013, Dublin, Ireland.

ACM 978-1-4503-2034-4/13/07 ...$15.00.

[1] http://www.stackoverflow.com
[2] http://meta.stackoverflow.com/questions/7656/how-do-i-write-a-good-answer-to-a-question

Figure 1: Example of a question in the Stack Overflow Q&A Forum (*Is there a name for this: '->'*), for which one, out of seven answers, is shown. The figure also illustrates the tools users can use to indicate how good are the answers.

unique "best answer", with several of them bringing useful information to the asker (and others interested in the question). Moreover, most strategies for automatic quality assessment found in the literature expect that the answer to be ranked has already received votes from the users [32, 28]. Thus, they are unable to assess the quality of answers to new or unpopular questions, which often do not contain such information.

Since such questions would largely benefit from ranking algorithms based on automated quality assessment strategies, in this work, we propose a learning to rank approach (L2R) to rank answers in Q&A Forums according to their quality. Unlike previous work, instead of directly estimating answer quality, we try to estimate the *answer rating*, that is, the feedback a user would give regarding the quality of an answer. More specifically, we adopt an approach based on Random Forests and represent query and answer pairs using eight groups of features. Some of these features are used in the Q&A domain for the first time. Using the proposed method, we were able to outperform a state of the art baseline with gains of up to 21% in NDCG, a metric used to evaluate rankings. We also conducted a comprehensive study of the features showing that, unlike what was previously observed for collaborative encyclopedias [10], user and review features are the most important in the Q&A domain. Further, text features, which are very simple to compute, are useful for assessing quality of new answers, which did not had enough time to acquire many reviews, and the set of new features we proposed was able to yield even better quality rankings.

This paper is organized as follows. In Section 2, we present previous related work. In Section 3, we present our method and the set of pieces of evidence used to evaluate answer quality. In Section 4, we present the experiments performed to evaluate our method and the results obtained. Finally, in Section 5, we present our conclusions and directions for future work.

2. RELATED WORK

Many previous studies are related to the problem of automatic assessment of quality in Q&A Forums. These can be classified according to three distinct objectives: (1) find the best answer; (2) rank the given answers; and (3) assess the quality of the question. Since works in group 3, such as [19, 3], are less related to ours, in the following paragraphs, we discuss only groups 1 and 2.

Works in group 1, which address the problem of finding the best answer to a given question, generally follow a straightforward classification strategy. A set of questions, for which the asker has already selected the best answer, is used for training. The answers are represented using a particular set of features and a classifier is applied, to label each answer as "best" or not, according to those features. Studies in this group have suggested the use of features related to expertise [2, 34], content length [2], grammar errors [2], question topics [2], user information [28], and comments [1]. From these, we highlight the work of [28], which proposed nine different features related to the answers content and to user information, to predict the best answer in Yahoo Answers. The authors learned the best answers through a classifier based on Logistic Regression [7]. They identified features related to the users answering and asking the question are good indicators of the best answers. Similarly, the authors in [1] proposed new features related to the answer and its follow-up comments. They used an Alternating Decision Tree method [12] to classify the best answers. As a result, they achieved accuracy levels of 84% to 87% in the samples used. Furthermore, they found out that length features are not correlated with best answers for the datasets used and a feature based on the rating of the answer can be a good predictor of the best answers. Since [1] is the most recent work and the method in [28] can be easily be adapted to rank answers, we used them as our baselines.

Works in group 2, which address the problem of ranking answers, focus on matching questions to answers, using some sort of similarity measure. Examples of works in this group are the work of [31], who used an L2R method with only relevance functions as evidence; by [16], who proposed a ranking model which takes into account an answer quality estimate; and [32], which explored user expertise in the ranking. This last work deserves a more detailed description since, among those we studied, it achieved the best performance. The authors in [32] argue that a user can have different expertise levels for different topics. Thus, they proposed quality-aware methods to rank answers. First, they learn good answers by using a manually annotated corpus, where answers are identified as good or bad. Then, they use this information, combined to relevance features, to calculate an expertise value that is used to rank the answers. Their intuition is that a good answer will be provided by a user that has provided good answers to similar questions in the past. To evaluate their method, they performed a manual annotation of a set of answers regarding their relevance ("relevant" or "not relevant"). By using the proposed expertise features, along with traditional relevance features, they were able to outperform all the previously described work. Because of that, we adopt this work as our third baseline.

Note that a characteristic shared by all the methods in group 2 was the use of discrete quality taxonomies as ground truth. By doing this, they ignore the fact that, among the good answers (and among the bad answers), some are better and some are worse. To avoid this, Sakai et al. [27], extended previous efforts by using a continuous scale to evaluate answers in Q&A Forums. The proposed solution, however, requires an expensive manual annotation.

Our method is closer to those in group 2: ranking answers. However, unlike the previous methods, we do not require explicit quality rating annotations. Instead, we use the number of positive and negative votes (rating) available on a different set of questions as an implicit quality assessment. This assessment can then be used to train an L2R method, which can later be applied on new questions, even if their answers were not voted yet. Furthermore, as in [27], we use a continuous scale for answer quality. We also improve on previous proposals by studying a new set of topic-based features and textual features which were previously used to assess the quality of collaborative encyclopedia content.

3. ASSESSING ANSWER QUALITY

In this work, we adopt a point-wise Learning to Rank (L2R) method to assign a quality score for the answers in a Q&A Forum. In this section, we present the chosen L2R method and the answer and query representations used.

3.1 Learning to Rank with Random Forests

A successful approach for the task of web search result ranking is to treat it as a supervised machine learning problem [22]. In this approach, each query-document pair (d, q) is represented by a set of features and annotated with a numerical score indicating how relevant document d is to query q. The features used are usually related to the similarity between the contents of document d and the query q. A set of such pairs can then be used to train a machine learning algorithm to predict the relevance of other non annotated pairs.

L2R methods are classified into three categories: point-wise, pair-wise, and list-wise [22]. Point-wise strategies can be viewed as regression approaches that predict relevance by minimizing a loss function. Differently, pair-wise approaches learn, for any given two documents, if one is more relevant than the other. Finally, list-wise approaches iteratively optimize a specialized ranking performance measure, such as NCDG (cf. Equation 2).

From the many algorithms proposed in literature, a pointwise approach using Random Forests (RF) [6] has been shown consistently effective in several real world benchmarks [22]. Among its advantages, we cite its insensitivity to parameter choices, its resilience to overfitting, and its high degree of parallelization. In this work we adopt this method as our learning strategy.

The RF algorithm is summarized in Algorithm 1. Let $D = \{(x_1, r_1), ..., (x_n, r_n)\}$ be a set of query-document pairs x_i and their associated relevance ratings $r_i \in \mathcal{R}$. Each pair x_i is an f dimensional feature vector that incorporates statistics about queries and documents. For example, one feature could be the similarity between question and answer according to BM25 ranking function. The RF method will train a predictor $T(.)$ such that $T(x_i) \sim r_i$ and the ordering of values according to $T(.)$ is similar to that obtained according to r_i.

Algorithm 1 Random Forests Algorithm

Input: $D = \{(x_1, r_1), ..., (x_n, r_n)\}, K : 0 < K \leq f, M > 0$
Output: $T(.)$
1: **for** $i \leftarrow 1$ to M **do**
2: $\quad D_t \leftarrow sample(D)$
3: $\quad h_t \leftarrow BuildDecisionTree(D_t, K)$
4: **end for**
5: $T(.) \leftarrow \frac{1}{M} \sum_1^M h_t(.)$
6: **return** $T(.)$

The main idea of RF is to apply a decision tree regression algorithm to M subsets of D and average the results. As we can see in Algorithm 1, first, a sample D_t is extracted with replacement from D (line 2). Then, a decision tree is built for this sample using $K \leq f$ randomly chosen features (line 3). This process is repeated M times (line 1) and, at the end, the results are averaged (line 5). This process reduces overfitting by using different data sets from the same underlying distribution. Single trees are built independently from others, thus making RFs inherently parallel.

In this paper we use the RF implementation provided by the RankLib L2R tool[3]. As values for M and K, we use the default parameters of that implementation. In particular, as the decision tree regression algorithm we use *Multiple Additive Regression Trees*, an implementation of the Gradient Tree Boosting method [13], provided by RankLib. Note that this algorithm requires a validation set to define the optimal number of boosting iterations.

[3] `http://people.cs.umass.edu/~vdang/ranklib.html`

3.2 Question-Answer Representation

The idea of L2R can be straightforwardly used in the Q&A domain. Here, questions can take the role of queries and answers can take the role of documents. Unlike the traditional web search task, however, in this case answers are provided by users specifically to address the given questions, which makes more uncommon the presence of not relevant answers. Thus, instead of a measure of relevance, we are now interested in predicting a measure of answer quality. Likewise, statistical features used to represent questions and answers should mainly reflect this notion of quality, instead of similarity.

In this section, we will present the features used to represent the question-answer pairs. These features try to capture the quality of the answer either directly, through textual features, such as style and structure, as well as indirectly, through non-textual features, such as author reputation and relatedness to the question. In total, we studied 186 features (98 textual and 88 non-textual). Of these, 89 features have never been previously used in the Q&A domain.

To simplify our analysis, all features were organized into groups, according to the characteristics they try to capture. Thus, the non-textual features were divided into *user*, *review*, and *user-graph* features. Textual features were divided into *structure*, *length*, *style*, *readability*, and *relevance* features. All groups are described in the following.

3.2.1 User Features

The intuition behind user features is to indirectly infer the quality of the answer by examining the user who posted it. More specifically, we are interested in features related to the user profile or its behavior, captured from events such as (1) post of questions and answers; (2) suggestion of edits in questions and answers; (3) post of comments to questions and answers; and (4) gain of merit ratings and badges for questions and answers. In Table 1, we present all the features computed for each answer using its user information. This table, as all the others in this section, shows for each feature (a) its name, (b) its description, and (c) a reference to the first work using it in the Q&A domain, or 'New' if it was not used before.

Although most of the features in Table 1 are self-explanatory, some require a more detailed explanation, provided in the following paragraphs. In the table, we refer to a question for which the best answer was already selected as *solved question*.

As argued in [32], users can have different expertise levels for different topics. Thus, another important source of information is the category of the questions being answered. This can be obtained through the *tags* (e.g. "html", "C++", "database") assigned by users to the questions. We refer to the set of categories (tags) of the Q&A pair being predicted as $\mathcal{T}$. Let $Q_{\mathcal{T}}(u)$ be a vector with the number of questions posted by user u to each category in $\mathcal{T}$, $A_{\mathcal{T}}(u)$ be a vector with the number of answers posted by user u to each category in $\mathcal{T}$, and $QA_{\mathcal{T}}(u)$ be a vector with the number of questions and answers posted by user u to each category in $\mathcal{T}$. Features questions entropy (*u-etat*), answers entropy (*u-etqt*), and questions and answers (*u-etpt*) correspond to the entropy calculated over vectors $Q_{\mathcal{T}}(u)$, $A_{\mathcal{T}}(u)$ and $QA_{\mathcal{T}}(u)$, respectively.

Some user features are based on user rankings. For instance, given a user u and a list of the users sorted in decreasing order according to the number of answers they posted ($R_{answers}$), feature

u-rknq is simply the rank of u in $R_{answers}$. Users are also ranked according to (a) the number of questions they posted ($R_{questions}$), (b) the number of answers they posted whose categories are in $\mathcal{T}$ (R_{acat}), (c) the number of questions they posted whose categories are in $\mathcal{T}$ (R_{qcat}), (d) the total rating received by asking questions (R_{rask}), (e) the total rating received by answering questions (R_{rans}), (f) the total rating received by asking questions whose categories are in $\mathcal{T}$ (R_{racat}), and (g) the total rating received by answering questions whose categories are in $\mathcal{T}$ (R_{rqcat}).

Table 1: User features.

Feature	Description	Ref.
u-dayc	n. of days since register	New
u-lac	n. of days since last access	New
u-bdg	n. of merit badges	New
u-aca	avg n. of comments per answer	New
u-xca	max n. of comments per answer	New
u-mca	min n. of comments per answer	New
u-acq	avg n. of comments per question	New
u-xcq	max n. of comments per question	New
u-icq	min n. of comments per question	New
u-aqc	n. of comments posted to answers and questions	New
u-se	n. of suggested edits	New
u-ase	n. of suggested edits approved	New
u-rse	n. of suggested edits rejected	New
u-edt	n. of edits made on answers	New
u-apq	avg answers posted per question	[2]
u-xpq	max answers posted per question	New
u-ipq	min answers posted per question	New
u-ap	n. of posted answers	[2]
u-qp	n. of posted questions	[2]
u-sela	n. of posted questions already solved	[2]
u-rknq	Rank position in $R_{questions}$	New
u-rkna	Rank position in $R_{answers}$	New
u-avqt	avg n. of questions posted in the categories of $\mathcal{T}$	New
u-xqt	max n. of questions posted in the categories of $\mathcal{T}$	New
u-mqt	min n. of questions posted in the categories of $\mathcal{T}$	New
u-aat	avg n. of answers posted in the categories of $\mathcal{T}$	New
u-xat	max n. of answers posted in the categories of $\mathcal{T}$	New
u-mat	min n. of answers posted in the categories of $\mathcal{T}$	New
u-arkat	avg rank position in R_{acat}	New
u-mrkat	max rank position in R_{acat}	New
u-xrkat	min rank position in R_{acat}	New
u-arkqt	avg rank position in R_{qcat}	New
u-xrkqt	max rank position in R_{qcat}	New
u-mrkqt	min rank position in R_{qcat}	New
u-etpt	Questions and answers entropy	[1]
u-etat	Questions entropy	New
u-etqt	Answers entropy	New
u-t3q	n. of categories user is a top-3 answerer	[32]
u-t3a	n. of categories user is a top-3 asker	[32]
u-t3qn	*u-t3q* / n. of categories user answered	New
u–t3an	*u-t3a* / n. of categories user asked	New
u-rtq	Total rating received by asking questions	[2]
u-rta	Total rating received by answering questions	[2]
u-artqt	avg rating received in the categories of $\mathcal{T}$	New
u-xrtqt	max rating received in the categories of $\mathcal{T}$	New
u-mrtqt	min rating received in the categories of $\mathcal{T}$	New
u-artat	avg rating received in the categories of $\mathcal{T}$	New
u-xrtat	max rating received in the categories of $\mathcal{T}$	New
u-mrtat	min rating received in the categories of $\mathcal{T}$	New
u-srtqt	Tot. rating got by asking questions in cats. of $\mathcal{T}$	[1]
u-srtat	Tot. rating got by answering questions in cats. in $\mathcal{T}$	[1]
u-rkq	Rank position in R_{rask}	New
u-rka	Rank position in R_{rans}	New
u-arkta	avg rank position in R_{racat}	New
u-xrkta	max rank position in R_{racat}	New
u-mrkta	min rank position in R_{racat}	New
u-arktq	avg rank position in R_{rqcat}	New
u-xrktq	max rank position in R_{rqcat}	New
u-mrktq	min rank position in R_{rqcat}	New

3.2.2 User Graph Features

User Graph Features features try to capture the expertise level of the users who answer questions by examining their relationships. While these features could be classified as *user features*, we decided to study them separately since they are particularly demanding to obtain. More specifically, we created a graph G where each node represents a user and an edge from user u to user v indicates that u answered a question posted by v. This graph was initially proposed in [34], and later used in [2], to estimate the expertise of a user, a method named as *ExpertiseRank*. ExpertiseRank is the PageRank value computed over G' (the transposed of G) [23]. As in [2], in addition to the actual PageRank value over G' (*ug-er*), we also use as feature the PageRank over G (*ug-pr*) and compute the HITS algorithm to create the authority and hub features (*ug-hu*, *ug-au*) [17].

3.2.3 Review Features

In collaborative digital libraries such as Wikipedia, features related to the reviewing process have been used with success to estimate the maturity level of the content [10]. In general, the more the content was reviewed, the best its quality. Similarly, in most Q&A Forums, such as those hosted by Stack Exchange, users can edit answers in order to improve them. In fact, they are encouraged to fix mistakes, include examples and further reading sources, etc. Thus, we believe that, as in the case of Wikipedia, these features may be a useful estimate of how much effort was invested in an answer. Table 2 describes the individual review features we have studied.

Features *r-count*, *r-aage*, *r-qage*, *r-sepu*, and *r-aepu* were previously proposed for assessing quality in collaborative digital libraries [10]. The general intuition behind them is that a content that received many edits has likely improved over time. In Q&A Forums, additional features with the same goal can be extracted. For instance, in Stack Exchange forums, a user can comment questions and answers and suggest edits to the author of an answer, who can accept them or not. This is a way of suggesting content improvements and providing additional information. From such comments we extracted the features *r-ase*, *r-qse*, *r-suc*, *r-qas*, *r-aas*, *r-aas*, and *r-ars*. Additionally, general information about the comments, such as *r-acc*, *r-qcc*, and *r-au* are good indicators of community engagement.

We also derived features that capture the question history by means of its answers. These are *r-ab* and *r-naq*. These features are important since they can indicate controversial topics and questions that are hard to answer.

3.2.4 Structure Features

These features attempt to describe the answer contents organization, analyzing the use of images, separation into sections, links, and HTML formatting tags. Table 3 describes the computed features.

Features *ts-ic*, *ts-sc*, *ts-ssc*, *ts-sssc*, *ts-asl*, *ts-mxsl*, *ts-misl* and *ts-ssl* were previously used in quality assessment of digital encyclopedias [24]. They are based on the idea that a good answer, specially if it is long, is organized into sections (and subsections), and contains images to improve understanding.

Several features use hints from the HTML source that indicate highlighting of concepts and ideas (*ts-boi*, *ts-quo*, and *ts-cod*), organization (*ts-lt* and *ts-lit*), and reliability by means of quotation

Table 2: Review features. Symbols *#sug* and *#ans* stand for the number of suggested edits and the number of answers, respectively.

Feature	Description	Ref.
r-count	Review count	New
r-aage	Answer age	[1]
r-qage	Question age	New
r-uedi	n. of users who edit the answer	New
r-aepu	Average n. of edits per user	New
r-sepu	Standard deviation of edits per user	New
r-ase	#sug to the answer	New
r-qse	#sug to the question	New
r-suc	n. of users who suggested edits to an ans. or qst.	New
r-qas	#sug approved by the asker	New
r-aas	#sug rejected by the asker	New
r-aas	#sug approved by the answer author	New
r-ars	#sug rejected by the answer author	New
r-ab	#ans posted before this answer	New
r-naq	#ans posted to the question	[2]
r-qcc	n. of comments posted to the question	[28]
r-acc	n. of comments posted to the answer	[28]
r-au	n. of users who commented the answer	New

of and reference to other sources (*ts-quo*, *ts-miq*, *ts-maq*, *ts-avq*, *ts-sdq*, *ts-xlc*, *ts-ilc*, and *ts-urf*).

Finally, we also used features to capture organizational hints specific of the Q&A domain, such as the use of code snippets in the answers (*ts-cod*, *ts-mic*, *ts-mac*, *ts-avc*, and *ts-sdc*). These are particularly important features in our study case since in Stack Overflow the asker is often searching for programming solutions.

3.2.5 Length Features

Length features are one of the most successful indicators of quality in collaborative encyclopedias [10]. This motivated us to test them in the domain of Q&A Forums. The general intuition behind them is that a mature and good quality text is probably neither too short, which could indicate an incomplete topic coverage, nor excessively long, which could indicate verbose content. We use three length features: word, sentences and character count (*tl-wcount*, *tl-scount*, and *tl-ccount*). These features were first used in Q&A forums by [2].

Table 3: Text Structure features.

Feature	Description	Ref.
ts-ic	Image Count	New
ts-sc	Section Count (n. of HTML H1 tags)	New
ts-ssc	Sub-section Count (n. of HTML H2 tags)	New
ts-sssc	Sub-sub-section Count (n. of HTML H3 tags)	New
ts-asl	Average section length	New
ts-mxsl	Maximum section length	New
ts-misl	Minimum section length	New
ts-ssl	Section length standard deviation	New
ts-boi	Italic plus Bold tag count	New
ts-pc	Paragraph Count	New
ts-quo	n. of quoted blocks	New
ts-cod	n. of code snippets	New
ts-lt	n. of lists	New
ts-lit	n. of list items in the text	New
ts-avc	Average code length	New
ts-mac	Maximum code length	New
ts-mic	Minimum code length	New
ts-sdc	Code length standard deviation	New
ts-avq	Average quoted text length	New
ts-maq	Maximum quoted text length	New
ts-miq	Minimum quoted text length	New
ts-sdq	Quoted text length standard deviation	New
ts-xlc	n. of links to external sources	[28]
ts-ilc	n. of links to other query/answer in the forum	New
ts-urf	n. of interactions with other forum users	New

Table 4: Text Style features. Symbols '%p' and '#p' stand for *percent of phrases* and *number of phrases*, respectively. KLD($\mathcal{D}$) stands for the Kullback-Leibner divergence [18] of a language model for dataset $\mathcal{D}$.

Feature	Description	Ref.
ty-cpc	n. of words capitalized	[2]
ty-cpe	n. of capitalization errors	[2]
ty-poc	Punctuation count	[2]
ty-pde	Punctuation density	[2]
ty-sde	Space density (n. of spaces / answer length)	[2]
ty-wse	Entropy of the text word sizes	[2]
ty-inn	Information to noise	[30]
ty-nwnt	n. of words not in WordNet	[2]
ty-typo	n. of typos	[2]
ty-slp	Size of the largest phrase	New
ty-lpr	%p where (length - avg. length) >= 10 words	New
ty-spr	%p where (avg. length - length) >= 5 words	New
ty-avc	n. of auxiliary verbs	New
ty-qc	n. of questions	New
ty-pc	n. of pronouns	New
ty-pvc	n. of passive voice sentences	New
ty-cjr	n. of words that are conjunctions	New
ty-nr	n. of nominalizations	New
ty-rpr	n. of prepositions	New
ty-ber	n. of uses of verb "to be"	New
ty-sp	#p starting with a pronoun	New
ty-sa	#p starting with an article	New
ty-scc	#p starting with a conjunction	New
ty-sscc	#p starting with a subordinating conjunction	New
ty-sipc	#p starting with an interrogative pronoun	New
ty-spr	#p starting with a preposition	New
ty-klg	KLD(good answers)	[2]
ty-klt	KLD(good answers of same category)	New
ty-klwid	KLD(Wikipedia discussion pages)	[2]
ty-klwi	KLD(Wikipedia pages classified as "Good")	[2]

3.2.6 Style Features

Style features try to capture the users writing style. The intuition is that good answers should present some distinguishable characteristics related to word usage, such as short sentences. Table 4 describes the features we use. To compute the features marked as 'New' in this table (and the Readability features in Section 3.2.7), we used the Style and Diction software[4].

Feature *ty-cpe* counts what are usually capitalization errors: the first letter of the sentence not being capitalized and the capitalization of letters that are not the first of a word. These features assume that an irregular use of capitalization may indicate a bad quality text. Features *ty-poc* and *ty-pde* try to capture the text quality through the use of punctuation, since an irregular punctuation may also be related to a bad quality text. Feature *ty-inn* measures the proportion of (stemmed) non-stopwords in the text.

We also use some vocabulary features in order to identify typos, similarly to [2]. Feature *ty-nwnt* computes the number of words that are not in the English lexical database WordNet[5]. Feature *ty-typo* counts the number of words present in a list of common misspellings, available from Wikipedia[6].

Another group of features tries to infer the difference between the language model used in the answer and other language mod-

[4] http://www.gnu.org/software/diction/
[5] http://wordnet.princeton.edu
[6] http://en.wikipedia.org/wiki/Wikipedia:Lists_of_common_misspellings

Table 5: Text Readability features.

Feature	Description	Ref.
tr-ari	Automated Readability Index [29]	New
tr-cl	Coleman-Liau [9]	New
tr-fki	Flesch-Kincaid [25]	[2]
tr-fre	Flesch reading ease [11]	New
tr-fog	Gunning Fog Index [14]	[2]
tr-lix	Läsbarhets index [5]	New
tr-sg	Smog-Grading [20]	[2]

els that can be seen as good references. The idea behind them is that an answer is more likely to be written in an inadequate manner if its generating language model is much different from language models which generate good answers. Thus, the feature *ty-klg* compares the language model of the answer to the language model of a group of answers considered good (i.e. the top 100 answers according to their rating, obtained from a sample of Stack Overflow different from the one we use for evaluation in Section 4.1). Feature *ty-klt* is similar but using only answers in the same categories of the answer being assessed (that is, categories in set $\mathcal{T}$, defined in Section 3.2.1). With the same goal, we created a sample of 100 articles, classified as Feature Articles according to the Wikipedia quality taxonomy[7], and its discussion pages in order to compare the answer language model to the language models of the Wikipedia articles and of the discussion pages, creating the features *ty-klwi* and *ty-klwid*.

Features *ty-slp*, *ty-lpr*, *ty-spr*, *ty-avc*, *ty-qc*, *ty-pc*, *ty-pvc*, *ty-cjr*, *ty-nr*, *ty-rpr*, *ty-ber*, *ty-sp*, *ty-sa*, *ty-scc*, *ty-sscc*, *ty-sipc*, and *ty-spr* were also previously used to assess content quality in collaborative encyclopedias [10]. We use them here for the first time in the Q&A domain. To calculate these features, we used the same terms used by the authors in [10]. Finally, for features *ty-spr* and *ty-lpr* we used threshold values of the Style and Diction software program.

3.2.7 Readability Features

Readability features were first used in [24] to predict quality of content in digital libraries. They aim at estimating the age or (USA) grade level necessary to comprehend a text. The intuition behind them is that good articles should be well written, understandable, and free of unnecessary complexity. The features we used are described in Table 5. Due to space constraints, we refer the interested reader to the more detailed descriptions available in the original papers, show in the table.

3.2.8 Relevance Features

Relevance features, first used in this domain in [31], try to identify the similarities between the answer and the question, in order to measure how relevant the first is to the second. These features are particularly useful to identify answers not related to the query. The features are shown in Table 6. Note that since the question has two sections, title and body, we use two features for each metric: the first one matches the title of a question to the body of the answer whereas the second one matches the body of the question to the body of the answer.

As in [31], computation of these features required different preprocessing tasks to be performed. The preprocessing tasks were stop-word removal and stemming. Content was represented using bags of terms, where terms could be words, part-of-speech (POS) tags, bi-grams, syntactic dependencies[8] and generalizations. A generalization corresponds to the transformation of each term into its corresponding category in WordNet Supersense, that is, a set of 46 categories which can be assigned to nouns and verbs (eg, "dog" is generalized to "animal", a person name is generalized to "person", a verb such as "swash" is generalized to "verb-motion"). A tagger[9] was used to extract POS tags and generalize words.

More specifically, features *tm-bm25*, *tm-bm25b*, *tm-swm*, and *tm-swmb* used stemming and content representations based on words, bi grams, dependencies, generalized bi-grams and generalized dependencies. Features *tm-span*, *tm-spanb*, *tm-swo*, and *tm-swob* used stemming and a content representation based on words. Finally, features *tm-nwad*, *tm-nwn*, *tm-nwv*, *tm-nwadb*, *tm-nwnb*, and *tm-nwvb* used a content representation based on POS tags.

4. EXPERIMENTS

Using the features described in Section 3.2, we performed a set of experiments using a Q&A test collection extracted from Stack Overflow. We now describe our experimental design, dataset, and results.

4.1 Dataset

Our dataset consists of a sample of Stack Overflow, a Q&A Forum for programmers. We chose this collection because it is freely available for download[10] and is the largest forum hosted by Stack Exchange. As any Stack Exchange forum, Stack Overflow focuses on specific topics and questions not related to these topics are removed or marked as closed. Thus, it probably has fewer spam and distractions when compared to general Q&A Forums such as Yahoo Answers.

The sample we used consists on 10,000 questions randomly extracted from the dataset. Note we extracted only questions that have, at least, 4 answers since that is the most interesting case for ranking. Questions with less than 4 answers are answers easily assessed by the users and an automatic rating is much less useful. From the resulting set, we also removed questions with no rated answers, since they could not be evaluated in a machine learning approach. These two procedures resulted in removing less than 3% of the answers in our dataset. In the end, our sample consisted of 9,721 questions with 53,263 answers[11]. To create the user graph (cf. Section 3.2.1), we considered all the Stack Overflow users and their questions and answers.

As ground truth for our machine learning approach, we use a function of the difference between upvotes and downvotes received by the answer. We refer to this function as the answer rating. r is given by Equation 1:

$$r_a = r'_a + r'_{min} \quad (1)$$

where $r'_a = u_a - d_a$ is the difference between the number of upvotes u_a and downvotes d_a received by answer a, and r'_{min} is the minimum difference between upvotes and downvotes observed in the collection, used to avoid negative values in Equation 2.

Note that the answer rating distribution follows a power law as we can see in Figure 2. Ratings vary from -16 to 505, with values from -20 to 140 corresponding to 99% of the instances in our

[7] http://en.wikipedia.org/wiki/Wikipedia:ASSESS

[8] Dependencies were detected by the tool described in [4] and available in http://sourceforge.net/projects/desr.

[9] We used a tagger based on Wordnet, described in [8] and available in http://sourceforge.net/projects/supersensetag.

[10] http://data.stackexchange.com/

[11] Data for the used sample can be downloaded at: http://www.lbd.dcc.ufmg.br/lbd/collections/ranking-q-a-forums

Table 6: Text Relevance features.

Feature	Description	Ref.
tm-bm25, *tm-bm25b*	BM25 ranking function, based on a probabilistic retrieval framework [26]	[31]
tm-ssm, *tm-ssmb*	Number of sentences shared by question and answer	[31]
tm-swm, *tm-swmb*	Number of words shared by question and answer	[31]
tm-swo, *tm-swob*	Number of words which appear in answer and question in the same order	[31]
tm-span, *tm-spanb*	Largest distance between two words that appear in answer and questions	[31]
tm-nwad, *tm-nwn*, *tm-nwv*, *tm-nwadb*, *tm-nwnb*, *tm-nwvb*	Number of new adjectives, nouns and verbs in the answer which did not appear on the question	[31]

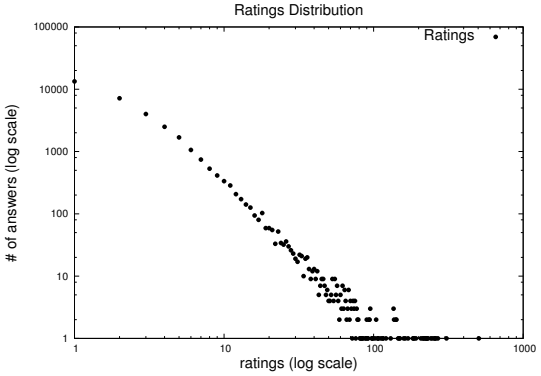

Figure 2: Ratings distribution for Stack Overflow sample.

sample. Such a skewed distribution is due to the popularity of the answers, with a few of them attracting large audiences.

4.2 Evaluation Methodology

The experiments we conduct have two main goals. First, to perform a comparative analysis between different machine learning methods in the task of ranking answers. Second, to analyze the impact of each group of features in this task.

We compare the methods using the *Normalized Discounted Cumulative Gain at top k* (NDCG@k, for short). This is a ranking evaluation metric first proposed in [15]. It allows us to measure how close the predicted answer ranking is to the ground truth ranking. More formally, NDCG@k is defined as:

$$NDCG@k = \frac{1}{N}\sum_{i=1}^{k}(\frac{2^{r_i}}{\log_2(i+1)}) \quad (2)$$

where r_i is the true rating assessment for the answer at position i in the ranking, and N is a normalization factor. The factor N is equal to the *discounted cumulative gain* (the sum part in Equation (2)) of the *ideal ranking*, i.e. the ranking where, given a pair of answers (a_i, a_j), a_i is better ranked than a_j if r'_i is greater than r'_j (cf. Equation 1).

To perform the comparative experiments, we used a five-fold cross-validation method [21] with a validation set. Thus, each dataset was randomly split into five parts, such that, in each run, one part was used as test set, one part was used as validation set, and the remaining three parts were use as training set. The split on training, validation, and test sets was the same in all experiments. The final results of each experiment represent the average of the five runs. Note that the folds were split such that answers provided to the same question belong to the same fold.

For all comparisons reported in this work, we used the signed-rank test of Wilcoxon [33] to determine if the differences in effectiveness were statistically significant. This is a nonparametric paired test that does not assume any particular distribution on the tested values. In all cases, we only draw conclusions from results that were considered statistically significant with a 95% confidence level.

In order to evaluate the impact of the chosen features, we used the information gain measure (infogain, for short) [21]. Infogain is a statistical measure of how much a given feature contributes to discriminate the class to which any given article belongs. It is normally used for feature selection but, since it provides a ranking of features based on their discriminative power, here we used to study the analyzed features. Infogain was computed for all features and the results are reported in Section 4.3.3.

4.3 Results

We now describe the experiments used to evaluate our proposed method. We first compare our method to others previously proposed in the literature and then we provide a comprehensive evaluation of the features.

4.3.1 Comparison to Previous Work

In this section we compare our method to previous work published in the literature. Since our work addresses the problem of ranking answers, our baselines are slightly modified versions of the methods proposed by Suryanto et al [32], Shah et al [28], and Burel et al [1]. In the first case, we provide the answer ratings as estimates for answer goodness and take the similarities returned by the method as rank values. In the second and third cases, we use the probability to be the best answer as rank value. We refer to our method as RF, to the method proposed by Suryanto et al as EXQD, to the method proposed by Shah et al as SHAH, and to the method proposed by Burel et al as AdTree. Note that RF uses all the features previously described.

In order to test the effectiveness of the new features proposed, we implemented a version of RF where Q&A pairs are represented only with features used by the baselines (the ones not marked as 'New' in Section 3.2). We refer to this method as RF-BaseFeatures. We also note that since SHAH uses a regression strategy, it is simple to adapt it to learn the answer rating instead the best answer. Thus, we tested several variants of the answer rating function as its target attribute. The one that achieved the best result was $\log r_a$, where r_a as given by Equation 1. We refer to this variant as SHAH-LR. Since the original implementations of these algorithms are not publicly available, we implemented them ourselves.

Figure 3 shows the NDCG@k figures for AdTree, SHAH-LR, SHAH, EXQD, RF-BaseFeatures, and RF using the best query-answer representations proposed for each method in our test dataset. All the differences pointed out between RF and the other methods were statistically significant, at all NDCG@k points, according to the Wilcoxon test.

We observe that RF and RF-BaseFeatures outperformed all the remaining methods for all values of k in our dataset. The larger gains were obtained for the top-ranked answers. The gain obtained by RF over RF-BaseFeatures indicates that our new features were able to improve the answer ranking. As for the baselines, when we compare the performance of RF and EXDQ, which was the second

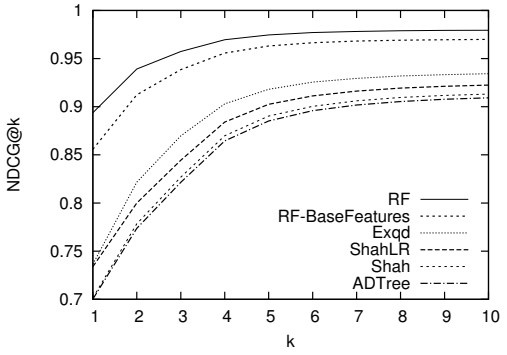

Figure 3: NDCG@k obtained for methods RF, RF-BaseFeatures, EXQD, SHAH, SHAH-LR, and AD-Tree.

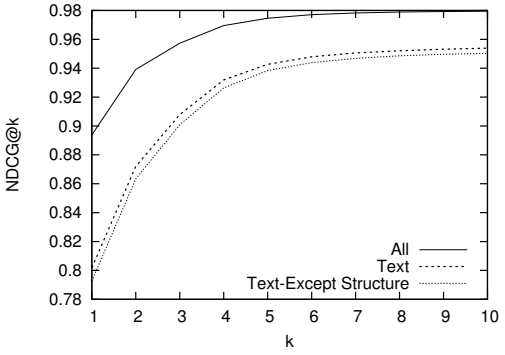

Figure 5: Random Forests effectiveness for textual features ranked according to their NDCG@k.

best method, we note gains ranging from 6% (NDCG@10) to 21% (NDCG@1). The smaller gains for the largest k values were expected since many questions have less than $k = 10$ answers then, for larger values of k, the highest rated answers are likely to be taken into account when calculating the NDCG. From the SHAH versions, the best performer was SHAH-LR. We also observe that AdTree and SHAH achieved the worst performance. Such result was expected since AdTree and SHAH were trained to find the best answers selected by the asker not those with the best ratings, which in our view may be even more informative, since the asker may have chosen the "best" one before the discussions have matured.

4.3.2 Analysis of the Groups of Features

To analyze the impact of each group of features, we divided our feature set into 8 groups: Structure, Length, Style, Relevance, Review History, User, User Graph, and Readability. We then conducted two series of experiments. First, we represented our question-answer pairs using only the features of each group in isolation, in order to determine the individual impact of the group. Following, we represented the Q&A pairs using all the features, leaving out one group at a time. This way, we can verify how each group is able to contribute to the results, independently from the other groups.

Figure 4 presents the results obtained for each of the feature groups. These groups are evaluated when used in isolation (Figure 4.3.2 (a)) and when excluded from the full feature set (Figure 4.3.2 (b)). For each group, we present the NDCG@k for k from 1 to 10. Note that in Figure 4.3.2 (b), feature groups whose exclusion did not result in a statistically significant loss are not shown.

As we can see in Figure 4.3.2(a), where groups are taken in isolation, no group is (statistically) significantly better than the combination of all features. Interestingly, the User features are the most relevant group. The User features are also the most important when we analyse their impact when removed (Figure 4.3.2 (b)), which highlights the importance of the profile and the history of the user to assess the quality of an answer. The second best set of features is the Review group. These features are useful to measure the engagement of the users in an answer (commenting, editing, etc.) and this engagement is probably proportional to the answer rating.

While User and Review features are the most important, text features are also useful and, more importantly, much less demanding to obtain, in terms of the preprocessing required. In addition, these features are always available from answers in any Q&A Forum, making the method more easily applicable in different forums. Thus, they are worth studying by themselves. In Figure 5, we compare the results of using all features, only textual features, and textual features excluding the structure group. Other textual feature groups are not shown since their removal did not show statistically significant differences. We can see, both in Figure 4.3.2(a) and Figure 5, that Structure features have the best performance when compared to the other textual features. Thus, we can conclude that, as previously found for collaborative encyclopedias [10], Structure is the best group of textual features.

Besides User, Review, and Structure, the remaining groups of features perform significantly worse and have no impact when removed. Relevance features have no impact probably due to the fact that most of the answers in Stack Overflow are relevant to their questions. Readability features are probably not suitable for this scenario, characterized by short text and many code snippets. Length features are probably redundant since many other features (eg, number of prepositions) are correlated to length. Finally, User Graph features presented a bad performance probably because they failed to capture user expertise in our sample dataset. This is interesting, for instance, to reduce the feature space, allowing to find even better ranking functions with less computational effort.

4.3.3 Feature Analysis using Infogain

To complement our previous study, we also ranked all the features according to the information gain metric. The results are summarized in Table 7, which shows the distribution of the features in groups of ten, starting from the best ranked.

Table 7: Number of features at top positions, ranked using infogain.

Group	# of features at top...					
	1-10	11-20	21-30	31-40	41-50	>50
User	7	8	2	3	3	36
Review	3	2	3	0	1	9
Structure	0	0	1	0	1	23
Length	0	0	2	1	0	0
Style	0	0	2	1	4	23
Relevance	0	0	0	5	1	34
Readability	0	0	0	0	0	7
User Graph	0	0	0	0	0	4

This table confirms the good performance of user and review features. Textual features appear only among the top-30 (length, style and structure features). Relevance features appear among the top-40. Readability and user graph features do not appear before the top-50. These two groups have already presented a bad performance in a previous work using the Yahoo! Q&A Forum [2] (User graph features) and collaborative encyclopedias [10] (Readability features).

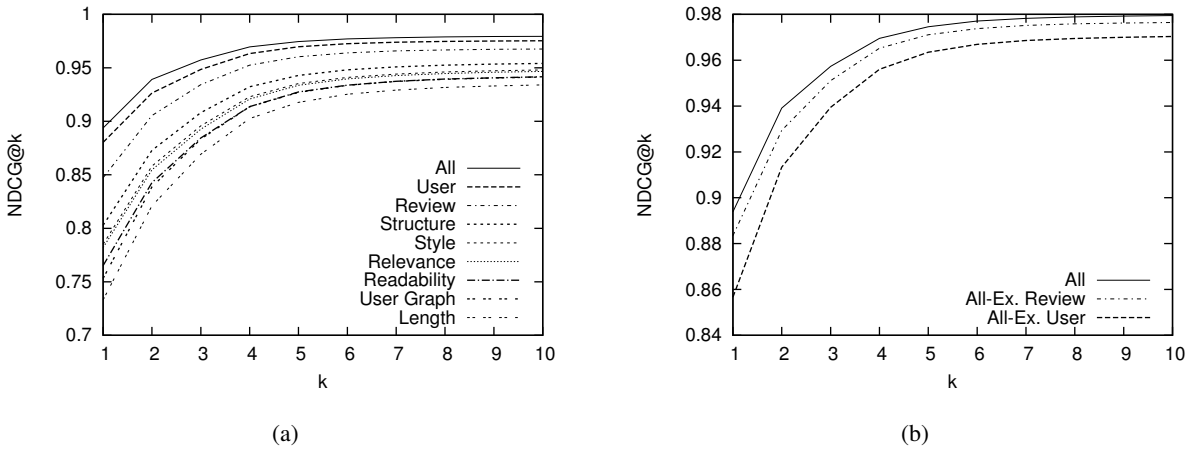

Figure 4: Random Forests effectiveness by using feature groups taken in isolation (a) or excluded from the complete set (b). Features Groups are ranked according to their NDCG@k.

Table 8: Features at top 5 positions per group and its general rank position, ranked using infogain.

User	Review	Structure	Length	Style	Relevance	Readability	User Graph
u-rka(1)	*r-au*(2)	*ts-pc*(27)	*tl-ccount*(21)	*ty-poc*(28)	*tm-nwn*(32)	*tr-sg*(74)	*ug-au*(54)
u-bdg(4)	*r-acc*(3)	*ts-cod*(49)	*tl-scount*(24)	*ty-spr*(30)	*tm-nwnb*(34)	*tr-cl*(79)	*ug-hu*(57)
u-rta(5)	*r-count*(7)	*ts-sdc*(53)	*tl-wcount*(35)	*ty-pvc*(38)	*tm-nwad*(36)	*tr-fki*(80)	*ug-pr*(73)
u-dayc(6)	*r-aepu*(17)	*ts-xlc*(62)		*ty-rpr*(41)	*tm-nwadb*(37)	*tr-ari*(81)	*ug-er*(77)
u-edt(8)	*r-aage*(20)	*ts-ilc*(65)		*ty-cpc*(47)	*tm-nwv*(39)	*tr-lix*(82)	

To better assess the best individual features, we show for each group the top-5 best features and its general rank in Table 8. As we can see, the best feature in User group is the rank position of the user according to its question answering rating (*u-rka*). This feature is evidently effective at capturing the relative expertise of the user.

The best features in the Review group are those related to comments (*r-au* and *r-acc*). These help to quantify how much attention other users have given to an answer. Other important features are the number of edits and editing users (*r-count*, *r-aepu* and *r-aepu*), also indicators of engagement, and answer age (*r-aage*). This last feature is useful because old answers have more opportunities to be voted.

The best feature of the structure group was the paragraph count (*ts-pc*), which provides hints about the answer structural organization. Other good features were related to code, links, and quotes, which indicate informative content with examples and references.

Although length features did not performed well in per-group analysis, they appear among the top 30. As in the encyclopedia domain, this probably happens because, in spite of this feature being related to quality, it carries information that is also indirectly provided by other features.

Regarding style features, the best are punctuation count and the number of short phrases (*ty-poc*, *ty-spr* respectively), which suggests that proper punctuation and well formed phrases are useful to predict quality in Q&A.

It is also interesting to note that, in the relevance group, the best features were about the new adjectives, nouns, and verbs in the answer. This shows that the appearance of new information in the answer is indicative of its quality.

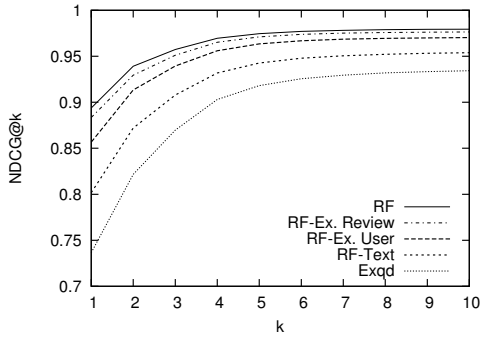

Figure 6: NDCG@k for the case of new answers and new users.

4.3.4 Rating New Answers and New Users

To finalize our evaluation, we should note that, not all the feature groups presented so far are always available to rank the answers. In particular, relatively new answers may not have any relevant Review features. Likewise, relatively new users may not have any relevant User features. For this reason, we performed experiments taking these possibilities into account.

Figure 6 shows the NDCG@k values for three cases: our approach excluding Review features, representing the case where the answer is new; our approach excluding User features, representing the case where the user is new; and our approach using only text features, representing the case where both the answer and the user are new. For comparison, we include our approach using all features (RF) and the best performing baseline (EXQD).

We can see that, even without User or Review features, we can still obtain very high NDCG@k values, close to those obtained with

all features. Using only text features, on the other hand, results visibly decrease. However, they are still well above the EXQD baseline, thus showing the strength of our method, even when using much less information.

5. CONCLUSIONS

In this work we proposed an L2R approach for ranking answers in Q&A Forums. In particular, we adopted an approach based on Random Forests and represented the Q&A pairs using eight groups of features. In total, we evaluated 186 features and, to the best of our knowledge, 89 of them were not used in the Q&A domain before. These features capture several aspects of a Q&A pair which we classified in the following groups: review, user, user graph, structure, style, length, readability, and relevance. Our L2R method was trained to learn the answer rating, based on the feedback users give to answers in Q&A Forums. By using a dataset from the StackOverflow Q&A Forum, we evaluated the sets of features and compared our method to 3 other ones previously published in literature.

We found that, unlike what was previously observed for collaborative encyclopedias, review and user features are the most important in the Q&A domain. Further, text features, which are very simple to compute, are useful for assessing quality of new answers (which did not had enough time to acquire many reviews). We have shown that the set of new features proposed was able to yield even better quality rankings. We also have shown that our method was able to outperform the best baseline with statistically significant gains of up to 21% in NDCG@k.

As future work, we intend to study a multi-view combination method for this problem, based on a meta-learning stacking strategy. We also intend to study more reliable strategies to determine the expertise of the users and their importance for the forums.

Acknowledgments

This research is partially funded by InWeb - The Brazilian National Institute of Science and Technology for the Web (MCT/CNPq/ FAPEMIG grant number 573871/2008-6), FCT (Portugal) under project SMARTIS - PTDC/EIA-EIA/115346/2009, and by the authors's individual research grants from FAPEMIG, CNPq, CAPES, and Google.

6. REFERENCES

[1] Automatic identification of best answers in online enquiry communities. In G. Burel, Y. He, and H. Alani, editors, *9th Extended Semantic Web Conference*, volume 7295 of *Lecture Notes in Computer Science*, Crete, 2012. Springer Berlin Heidelberg.

[2] E. Agichtein, C. Castillo, D. Donato, A. Gionis, and G. Mishne. Finding high-quality content in social media. In *WSDM '08*, pages 183–194, Palo Alto, California, USA, 2008. ACM.

[3] 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 *KDD'12*, 2012.

[4] G. Attardi, F. dell'Orletta, M. Simi, A. Chanev, and M. Ciaramita. Multilingual dependency parsing and domain adaptation using desr. In *EMNLP-CoNLL*, pages 1112–1118, 2007.

[5] C. Björnsson. *Lesbarkeit durch Lix*. Stockholm: Pedagogiskt Centrum, 1968.

[6] L. Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, Oct. 2001.

[7] S. L. Cessie and J. V. Houwelingen. Ridge estimators in logistic regression. *Journal of the Royal Statistical Society. Series C*, 41(1):191–201, 1992.

[8] M. Ciaramita and Y. Altun. Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger. In *EMNLP '06*, EMNLP '06, pages 594–602, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[9] M. Coleman and T. L. Liau. A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, 60(2):283–284, 1975.

[10] D. H. Dalip, M. A. Gonçalves, M. Cristo, and P. Calado. Automatic assessment of document quality in web collaborative digital libraries. *ACM Journal of Data and Information Quality*, 2(13), 2011.

[11] R. Flesch. A new readability yardstick. *Journal of Applied Psychology*, pages 221–235, 1948.

[12] Y. Freund and L. Mason. The alternating decision tree learning algorithm. In *ICML'99*, 1999.

[13] J. H. Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, Feb. 2002.

[14] R. Gunning. *The Technique of Clear Writing*. McGraw-Hill International Book Co, 1952.

[15] K. Järvelin and J. Kekäläinen. IR evaluation methods for retrieving highly relevant documents. In *SIGIR '00*, pages 41–48, Athens, Greece, 2000.

[16] J. Jeon, W. B. Croft, J. H. Lee, and S. Park. A framework to predict the quality of answers with non-textual features. In *SIGIR '06*, pages 228–235, Seattle, Washington, USA, 2006. ACM.

[17] J. M. Kleinberg. Hubs, authorities, and communities. *ACM Comput. Surv.*, 31(4es), Dec. 1999.

[18] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[19] B. Li, T. Jin, M. R. Lyu, I. King, B. Mak, T. Chinese, and H. Kong. Analyzing and Predicting Question Quality in Community Question Answering Services Categories and Subject Descriptors. pages 775–782, 2012.

[20] G. H. McLaughlin. Smog grading: A new readability formula. *Journal of Reading*, pages 639–646, 1969.

[21] T. M. Mitchell. *Machine Learning*. McGraw-Hill Higher Education, 1997.

[22] A. Mohan, Z. Chen, and K. Weinberger. Web-search ranking with initialized gradient boosted regression trees. *JMLR Workshop and Conference Proceedings: Proceedings of the Yahoo! Learning to Rank Challenge*, 14:77–89, June 2011.

[23] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical report, Stanford Digital Library Technologies Project, 1998.

[24] L. Rassbach, T. Pincock, and B. Mingus. Exploring the feasibility of automatically rating online article quality. `http://upload.wikimedia.org/wikipedia/wikimania2007/d/d3/RassbachPincockMingus07.pdf`, 2007.

[25] S. Ressler. *Perspectives on electronic publishing: standards, solutions, and more*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[26] S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR '94*, SIGIR '94, pages 232–241, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[27] T. Sakai, D. Ishikawa, N. Kando, Y. Seki, K. Kuriyama, and C.-Y. Lin. Using graded-relevance metrics for evaluating community QA answer selection. In *WSDM '11*, page 187, New York, New York, USA, Feb. 2011. ACM Press.

[28] C. Shah and J. Pomerantz. Evaluating and predicting answer quality in community QA. In *SIGIR'10*, number March 2008, 2010.

[29] E. A. Smith and R. J. Senter. Automated readability index. *Aerospace Medical Division*, 1967.

[30] B. Stvilia, M. B. Twidale, L. C. Smith, and L. Gasser. Assessing information quality of a community-based encyclopedia. In *ICIQ'05*, pages 442–454, 2005.

[31] M. Surdeanu, M. Ciaramita, and H. Zaragoza. Learning to Rank Answers on Large Online QA Collections. *ACL '08*, 2008.

[32] M. A. Suryanto and R. H. L. Chiang. Quality-Aware Collaborative Question Answering : Methods and Evaluation. In *WSDM '09*, pages 142–151, 2009.

[33] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics*, pages 80–83, 1945.

[34] J. Zhang, M. S. Ackerman, and L. Adamic. Expertise networks in online communities. In *WWW '07*, New York, New York, USA, 2007. ACM Press.