Microsoft Research Asia with Redmond at the NTCIR-8 Community QA Pilot Task

Young-In Song[1], Jing Liu[2], Tetsuya Sakai[1], Xin-Jing Wang[1], Guwen Feng[3], Yunbo Cao[1], Hisami Suzuki[4], Chin-Yew Lin[1]
[1, 2, 3] Microsoft Research Asia, Beijing, China
[4] Microsoft Research, Redmond, WA, USA
{yosong, tesakai, xjwang, yucao, cyl}@microsoft.com[1], hisamis@microsoft.com[4], jliu@ir.hit.edu.cn[2], linvondepp@gmail.com[3]

ABSTRACT
In this paper, we describe our approaches that we used for the NTCIR-8 Community QA Pilot task and report on its results. Specifically in the pilot task, we mainly focused on discovering effective features for evaluating quality of answers, for example, features on relevance of an answer to a question, authority of an answerer, or informativeness of an answer. Also, we examined two different statistical learning approaches for finding the best quality answer. The official evaluation results of our runs showed that our proposed features and learning approaches are effective in terms of finding the best quality answers.

Keywords
Community QA, Answer quality evaluation, Best answer finding

1. INTRODUCTION
For recent few years, a vast amount of questions and their answers has been accumulated in various kinds of Web sites, for instance, a community QA site (CQA) or a forum. Those question and answer threads (QA threads) become one of valuable knowledge resources for many information seekers by allowing them to search answered questions that satisfying their information needs.

In using online QA threads as a knowledge resource, one interesting challenge is how to assess quality of answers automatically. Because a CQA site or forum generally has no or little editorial control in answer posting process, the quality of answers in a thread varies greatly from an informative, well-written answer to a useless or inappropriate answer. A user may have to distinguish good quality information from answers after finding relevant QA threads to his or her information need. Clearly, this can have a negative effect on user's experience.

Although many CQA sites try to solve this problem by encouraging an asker to select a 'Best Answer' among posted answers to his or her questions, it cannot be a perfect solution: The 'best answer' selected by an asker is not always the real best answer [1,2]. There can be many equivalently good or even better quality answers comparing to the selected best answers. Furthermore, QA treads in a forum do not have any explicit 'best answer' selected by an asker posting a question. Therefore, with the increasing quantity of available QA threads, there is a clear need to enhance the user experience by distinguishing high quality answers from low quality ones [3,16].

In this paper, we describe our efforts to build an automatic system to find out the best quality answer in a QA thread, which is used for NTCIR-8 Community QA Pilot task. Specifically, our main research interests were:

(1) What could be an indicator for high or low quality of answers in a QA thread?

(2) What is an appropriate statistical learning approach for the best quality answer finding?

To find answers, we examined various features reflecting different aspects of answer quality, for example, relevance of an answer to a question, its informativeness, authority of an answerer, or discourse and modality of an answer, with two different statistical learning approaches, SVM rank [4] and the analogical reasoning model [5].

This paper is organized as follows: In the next section, we briefly introduce NTCIR community QA Pilot task. In Section 3, our feature sets and learning approaches are described in detail. Then, evaluation results are reported and analyzed in Section 4 and, finally in Section 5, we conclude our work.

2. NTCIR COMMUNITY QA PILOT TASK
NTCIR-8 Community QA pilot task is motivated to encourage research on developing and evaluating an automatic system to find the best quality answer in an online QA thread. Formally, the pilot task is defined:

- For a given QA thread consisting of one question q and its answers $a_1,...,a_n$ ($n \geq 1$), rank answers according to their quality for q.

For developing of a participant system, approximately 3M of Japanese QA threads from Yahoo! Answers Japan[1] are provided as training data. Each QA thread contains one question and their answers with several meta data on a question or answer, such as a posting time of an answer, an asker and answer identifier, and so on. Also, exactly one answer of a QA threads in training data is labeled as the 'best' answer (BA), denoting the answer which is selected by the asker as the best quality answer. It can be used for training a system.

All participants are asked to submit runs for the testing data consisting of 1,500 QA threads. The participating systems are to

[1] QA threads accumulated for 1 year between 2004 and 2005 in Yahoo! Answers Japan (http://chiebukuro.yahoo.co.jp). For the detail information on the data, refer [1] and [2].

rank all the answers for each QA thread in descending order of answer quality. The submitted runs are evaluated by the pilot task committee with two ways: One is using BAs as a ground truth by regarding BAs as the only good quality answers, and the other is using the ground truth based on graded quality assessment results from 4 assessors (called as Good Answers data; GA). In GA data, each answer in QA threads is manually assigned into one of four grades, L3 (highly relevant), L2 (relevant), L1 (partially relevant) and L0 (not relevant), in absolute terms of answer quality. On average, one QA thread has 1.87 L3 answers, 1.94 L2 answers, 1.12 L1 answers, and 0.03 L0 answers. Note that there are about 2 L3 answers and 2 L2 answers are posted in average to one question. It also supports our observation that there are many good quality answers besides BAs. GA data set is independently constructed from BAs and regarded as a primary ground truth in the evaluation. For the detail information on NTCIR pilot task, we refer the reader to [1] and [2].

3. LEARNING BEST ANSWERS

The best quality answer finding task can be viewed as a statistical learning problem on a preference to the best quality answer. In our approach, each answer in a QA thread is represented as a feature vector, which is a set of evidences potentially denoting answer quality, and a statistical model is learned from QA threads by regarding BA labels as a ground truth. In the testing phase, the output score of the model is used for ranking answers.

In the following subsections, we will describe our feature sets and learning models in detail.

3.1 Features

We experimented with various features that are potentially useful for discriminating good quality answers from others. The four aspects are mainly considered in the selection of the features; *relevance to a question, authority and expertise of an answerer, informativeness of an answer, and discourse and modality*.

3.1.1 Relevance features

It is obvious that quality of an answer should be defined in the context of a question which the answer targets to resolve. If an answer is not relevant to a question, it is worthless as an answer in regardless of quality of information that it contains. From this viewpoint, we define two features on relevance of an answer to a question in QA thread.

LM based Relevance Score (LMRS):

This is a real value feature indicating a relevance score of an answer for a question in QA thread, which is estimated by using unigram query likelihood model for information retrieval [6]. Jelinek-Mercer smoothing method is used to interpolate a document language model (in our case, answer model) with a collection language model.

Graph-based Relevance Score (GRS):

Cong et al. proposed an alternative way to measure relevance of an answer by considering relation between answers [7]. The main hypothesis of the approach is that if an answer is related (e.g., similar to) an authoritative (relevant) answer with high score, the answer is also likely to be a relevant answer for a given question. Based on the hypothesis, they proposed a random walking algorithm to compute relevance scores of answers using an answer graph where a node is an answer and an edge is determined based on similarity between answers. We employed their method to calculate a relevance score for the best quality answer finding. For the detail information on their graph-based propagation approach, we refer the reader to [7].

3.1.2 Authority and expertise feature

One of our interesting observations on online QA community is that there is a tendency that a very small number of users provide a significantly large portion of BAs. For example, in the training data for NTCIR pilot task, only 10% of answerers composed about 90% of best answers. It implies that there are a group of authoritative users having expert knowledge, who also participate very passionately to the community. In another words, there is a high probability that answers provided by such an authoritative user are good quality answers if a question matches to his or her expertise domain. Based on this intuition, we defined several features on authority and expertise of an answerer.

Normalized number of best answers (NBA):

One of the simplest ways to measure authority of an answerer would be counting the BAs that the answerer composed previously. If one produced many BAs, it would be a good indicator that he or she is a high authoritative answerer. Based on this intuition, we introduce a normalized number of best answers (NBA) as one of our features denoting authority of an answerer:

$$NBA(u_i; T_j) = \frac{C(BA; u_i)}{\max_{u \in T_j} C(BA; u)} \quad (1)$$

where u_i is an answerer participating a QA thread T_j, and $C(BA; u_i)$ denotes the number of best answers provided by the answerer u_i.

Precision score of Answerer (PS):

Our user precision score *PS* is basically defined as a success rate that answers posted by an answerer *u* are selected as BAs by askers:

$$PS(u_i) = \frac{C(BA; u_i)}{C(A; u_i)} \quad (2)$$

where $C(A; u_i)$ denotes the number of answers provided by the answerer u_i. Intuitively, if the most of answers posted by an answerer was selected as BAs, he or she is very likely to be an authoritative answerer.

Unfortunately, the equation (1) has a problem when $C(A; u_i)$ is small. For example, suppose two answerers: one posting 1,000 answers with 800 BAs, and the other posting only 1 answer which is selected as a BA. In this case, the precision score of latter answerer will be higher than the first answerer, but it is doubtful that the latter one is truly a better authoritative answerer because he or she may not be an active user contributing to the QA community.

Thus, we modify the equation (1) with consideration on the degree of participation of an answerer to a QA community:

$$PS_s(u_i) = \frac{C(A; u_i)}{C(A; u_i) + \alpha} \cdot PS(u_i) + \frac{\alpha}{C(A; u_i) + \alpha} \cdot PS_{avg} \quad (3)$$

where

$$PS_{avg} = \frac{1}{|u|} \cdot \sum_{\forall u} PS(u), \quad \alpha = \frac{1}{|u|} \cdot \sum_{\forall u} C(A; u)$$

Here, $|u|$ indicates the number of answerers in the training data. In Equation (3), an output value will be close to the output value of the equation (2) when $C(A;u_i)$ is much larger than the average number of answers for a user, α, in the training data. In the opposite case, it will be close to the average of PS over answerers, PS_{avg}, in the data.

Likelihood to be Winner (LW):

By taking relations between users in a QA community into the consideration, we may estimate authority of answerers better. Based on this idea, Zhang [8] and Jurczyk [9,10] have proposed PageRank and HITS based approaches to compute authority of an answerer in an online QA community. They assumed that if a user answers a question asked by a high authoritative person, the user (answerer) should have a high authority score. To evaluate authority of a user, they firstly define a graph based on the relation between askers and answerers, and then perform random walking based on the graph to estimate authority scores of users. However, their assumption may not be a realistic assumption in CQA data because there is generally an only small amount of overlaps between answerers and askers in a CQA community [11]. An asker seldom answered a question, and also an answerer seldom asked a question. This implies that authority of a user cannot be propagated well to other users by using asker-answerer relations.

To alleviate the problem, we proposed a new approach to compute user authority, which is also based on a random walking algorithm but uses a different graph representation. In our approach, each QA thread is viewed as a competition, in which the winner is the answerer posting BA. Users posting non BA are regarded as losers. From this viewpoint, a directed graph can be constructed from QA threads by regarding answerers as nodes and connecting answerers with a directed edge from a loser to a winner in a competition. Based on the graph, the authority score of each answerer at the t-th iteration, $P_t(u_i)$, can be computed by the following equation:

$$P_t(u_i) = \lambda \cdot P_0(u_i) + (1-\lambda)\cdot\left(\sum_{\forall u_j} T(u_j \rightarrow u_i) P_{t-1}(u_j)\right) \quad (4)$$

where

$$P_0(u_i) = \frac{C(BA;u_i)}{\max_{\forall u} C(BA;u)}$$

$$T(u_j \rightarrow u_i) = \begin{cases} if\ u_j = u_i, & \frac{\#\ of\ questions\ u_j\ wins}{\#\ of\ questions\ that\ u_j\ participate} \\ else, & \frac{\#\ of\ questions\ u_i\ wins\ u_j}{\#\ of\ questions\ that\ u_j\ participate} \end{cases}$$

Equation (4) can be computed using power iteration method.

User Expertise Score (UE):

If a question is well matched to an answerer's knowledge, there will be a high probability that his or her answer can be a good quality answer. Relevance of answerer's expertise knowledge to a question can be a good indicator to predict answer quality.

By assuming that an answer reflects the knowledge or skills of its asker (so to speak, "you're what you wrote"), the expertise of an answerer can be inferred by his or her answers. Given a question q and an answer a posted by a user u_i, the relevance score of the expertise of user u_i on a given question q can be estimated:

$$P_{UE}(q|a,u_i) \propto \lambda_1 P(q|a) + \lambda_2 P(q|C_{u_i}) + \lambda_3 P(q|C_c) \quad (5)$$

where

$$P(q|a) = \frac{C(w;a)}{|a|}, P(q|C_{u_i}) = \frac{C(w;C_{u_i})}{|C_{u_i}|}, P(q|C_c) = \frac{C(w,C_c)}{|C_c|},$$

$$\lambda_1 + \lambda_2 + \lambda_3 = 1,$$

C_{u_i} means the collection of all answers posted by the user u_i, and C_c denotes the collection of all documents in category c. Similar methods have been adopted to build the profiles of users and rank users in enterprise search [12].

3.1.3 Informativeness features

Informativeness features are designed to measure how informative an answer is for a given question. This is obviously one of important aspect in answer quality, so we define the following features to measure informativeness degree of an answer.

Square of normalized length of answer (NLA):

A length of answer is a simple but effective indicator for how informative an answer is. Naturally, a length of answer has a tendency to be longer when it contains richer information. Also, in many previous studies on answer quality [13, 17], it has been reported as one of the most effective features to find a good quality answer. Thus, we use a square of length of an answer, normalized by the maximum answer length in a QA thread as one of the features to measure informativeness:

$$NLA(a_i, T_j) = \left(\frac{|a_i|}{\max_{a_k \in T_j}(|a_k|)}\right)^2 \quad (6)$$

where a_i is an answer in a QA thread T_j, and $|a|$ is the number of content words (noun, verb, adjective, adverb) in a. NLA is defined as a real value feature.

Existences of URL address (URL):

Because many of questions posted to online QA community ask about information on the Web, URL address appeared in an answer can be a good indicator that the answer provides useful information. Thus, we define a binary feature indicating whether or not an answer contains at least one URL address.

Lexical centrality of an answer in a thread (LEX):

In terms of informativeness, the best quality answer in a QA thread would be the best summary of the thread, minimizing information loss. From this viewpoint, measuring informativeness of an answer can be similarly defined to a task finding the best summary sentence (in our case, an answer) in a document (a QA thread). Based on this intuition, we examine one of the graph-based summarization approaches, 'LexRank', evaluating salience of a sentence in a document by measuring its lexical centrality in the document [14, 17].

The basic idea of LexRank method is that if a topic in a sentence is valuable as a summary, there can be many lexically similar sentences in a document. Among those sentences, the most central

sentence can be regarded as the most representative one in a document. LexRank estimates the centrality of a sentence in a manner similar to the PageRank, based on a graph where each node represents a sentence and two nodes are edged such that similarity between them exceeds a certain threshold value. By replacing a sentence and a document into an answer and a QA thread, we can apply this method to measure informativeness of an answer.

In the LexRank method, a prior weight should be assigned for each node, which denotes the best guess on salience of a node. We examine two different priors: One is NLA feature value of an answer, and the other is PS feature value of an answer. In our system, the output values of LexRank with two different priors, NLA and PS, are used as two separated real value features, LEX+NLA and LEX+PS respectively.

3.1.4 Discourse and modality features

Features from discourse structure of QA thread (e.g., a position of an answer in a thread) or modality of an answer (e.g., kindness of an answer) can be also effective for the best quality answer finding. We also investigated the features from the aspect.

Position of answer (PA):

In [15], Nam, et al reported one interesting phenomenon on an answerer's behavior in a CQA community. They interviewed a number of top contributors in one famous Korean CQA community, Naver's Knowledge-In[2] and found that they have a tendency to answer questions only in the case that it is necessary, mainly for the purpose of saving their time: If there is already a sufficiently good answer posted to a question, they will skip the question and move to another question. If not, they will post a new answer to this question. If this behavior is general in all answerers in a QA community, a lastly posted answer in a thread would be likely to be the best quality answer for the question because it makes all other users skip the question.

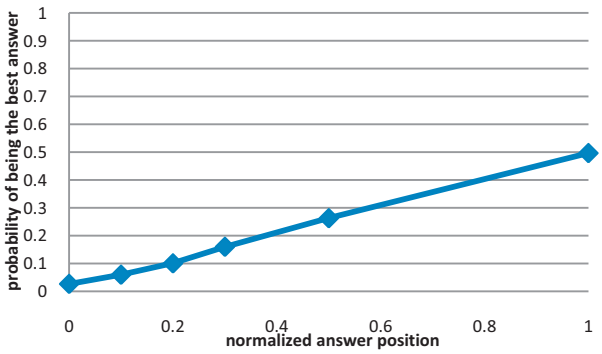

Figure 1: The change of the probability of an answer being BA and its position in a thread.

In our observation, it looks a true and general behavior in CQA. Figure 1 shows the correlation between that probability of an answer being BA and its position in a thread. In the figure, x-axis denotes that the normalized answer position which is computed by:

$$PA(a_i) = \frac{1}{|T_j| - Pos(a_i)} \quad (7)$$

where a_i is an answer in a QA thread T_j, $|T_j|$ is the number of answers in the thread T_j, and $Pos(a_i)$ is the rank of the answer when all answers in T_j are sorted by its posting time in ascending order. As shown in the figure, the probability of the lastly posted answer being BA is much higher than the probabilities of the other positions.

Based on this observation, we use a value of Equation (7) as one of the features.

Negative Words (NW):

We extract words appearing more frequently in non BAs than BAs and manually select three types of cue words for a low quality answer; (1) casual words, (2) rude words and (3) sexual words. These cue words are named as negative words. Based on the negative word set, we defined one binary feature (NW): a value of NW feature is 1 if any negative word appears in an answer; otherwise, feature value is 0.

Agreement relationship between question and answer (AR):

In our observation on online QA threads, one of the frequently asked question types is a question begging an agreement from others, for example, "*Am I normal?*" Also, we found that for such a kind of question, BA often contains an agreement expression, e.g., "*Yes, you are fine*". Based on this observation, we manually built small amount of question lexical patterns indicating questions begging an agreement and answer lexical patterns indicating agreement expressions. We defined a binary feature, AR, indicating both a question and an answer contains one of patterns.

3.2 Model

To learn a preference on the best quality answers, we examined two different statistical approaches; SVM Rank [4] and Analytical model [5]. SVM Rank is selected to investigate effectiveness of a pair wise learning approach in the best quality answer finding task, and the analytical model is chosen to consider a dependency between questions. In the following subsections, we will describe each model in detail.

3.2.1 SVM Rank

Because all answers in the training data are classified into two binary labels, BA and non BA, we can apply any of statistical classification models to find the best quality answer by regarding all BAs as good quality answers (positive) and all non-BAs as non-good quality answers (negative). However, this might be a too crude assumption; as aforementioned, there can be many good quality answers among non BAs. Also, the quality of one non-BA in a QA thread can be superior to the quality of BAs in other threads. These facts can make a statistical model suffer from many false negatives. Moreover, because the amount of non BAs overwhelms the amount of BAs in online QA threads, it could make a bias to negative labels in model training.

Comparing to using the statistical classification approach, using a pairwise learning approach has a clear advantage in best quality answer finding. As different from a classification approach, it aims to learn a preference on a pair. Thus, its assumption on the training data is different from the one under the setting of

[2] http://kin.naver.com/

classification: a BA answer is better than a Non BA answer in a QA thread. The assumption in pairwise learning can be also wrong because there can be an objectively better quality answers than a BA in a thread, but it may be a safer assumption than the assumption of classification. In pairwise learning, it is not necessary to assume that Non BA answer is not a good quality answer. A false negative in pairwise learning only happens when there are an objectively better quality answers than the BA in a QA thread, so an amount of false negatives would be smaller than the classification.

For this reason, we select SVM rank as our baseline model for the best quality answer finding, which is one of the popularly used pairwise learning approaches for ranking.

3.2.2 Analogical Reasoning Model

Different from the SVM Rank model, the analogical reasoning model proposed in [5] not only considers the relationship between a question and an answer, but also regards questions as relational data and attempts to leverage the knowledge embedded in available questions and their answers.

The task of detecting BA is challenging because there is not only lexical gap but also semantic gap between questions and answers. The lexical gap is result not only from textual mismatch between questions and answers, but also from existing spam. The semantic gap is even severe, e.g. the best answer of the question "how to pronounce the Congolese city Kinshasa" is "'kin' as in your family, 'sha' as in sharp and 'sa' as in sergeant". Though the BA is semantically relevant to the question, one cannot find any textual clues from the Q and A.

Fortunately, one can find more clues from previous QA threads to bridge the gaps. The basic idea is, though a key term which appears in a question (or its BA) does not appear in its BA (or the question instead), possibly it has appeared in some related questions that asked by some other users before. Therefore, by identifying the previous related questions and leveraging the characteristics of their best answers, we are able to infer that which answer should be the BA to the new question.

The analogical reasoning model contains two stages: one is the training stage which learns a Bayesian logistic regression model from the positive Q-A pairs and negative ones observed from the training data. The other is the testing stage in which given a new question, it retrieves a set of relevant positive Q-A pairs, and evaluate the analogy of a new Q-A candidate to the retrieved knowledge. The best answer is the one that achieves the highest analogy score.

4. EXPERIMENTS

In this section, we report the evaluation results of our approaches in NTCIR pilot task.

4.1 Experiment Setting

We trained our models with about 3M of Yahoo! Answers Japan QA threads provided by NTCIR committee. Binary relevance is assumed for training by regarding BAs as relevant answers (good quality answers).

Also, we learn separate models for categories of QA threads and applied one of them according to the category of an input QA thread in testing phase. Our intuition is that characteristics of good quality answers in one category could be different from other categories. For example, good quality answers of 'LOVE' category can be different from ones of 'NEWS' category in many aspects.

Yahoo! Answers Japan uses a hierarchical taxonomy for categorizing a QA thread. It has 14 root level categories, and each QA thread in Yahoo! Answers Japan belongs to exactly one root category. We utilized the root level categories for training our models.

To process Japanese text in QA threads, we used NLPWin as a parser, which is developed by Microsoft Research [18, 19].

4.2 Run Configuration

To select features for official runs, we performed preliminary experiments by creating our own training data and testing data. For that, we separated QA threads posted to the last one month from the original training data, and regarded them as testing data. The rest of original training data is used as training data for the preliminary experiments. In our preliminary experiments, BAs are only regarded as relevant answers. We mainly relied on the precision at top 1 rank (BA-Hit@1) measure in selecting effective features for best quality answer finding. Note that the training and testing data in our preliminary experiments is different from the ones used for our final runs submitted for official evaluation. We train our models again by using whole training data to produce our official runs.

Table 1 shows the features used for our 5 runs (from Run 1 to Run 5), which are resulted from the preliminary experiment results. For Run 1, 2, 3, and 4, SVM Rank is used as a learning model. For Run 5, the analogical reasoning model is used.

Table 1: Feature configurations of our runs

	Feature	Run 1	Run 2	Run 3	Run 4	Run 5
Relevance	LMRS[3]					
	GRS	√	√			√
Authority and Expertise	NBA			√		
	PS	√	√	√		√
	LW			√	√	
	UE		√	√	√	
Informativeness	NLA	√	√			√
	URL	√	√	√	√	√
	LEX+NLA				√	
	LEX+PS			√		
Discourse and Modality	PA	√	√	√	√	√
	NW		√	√	√	
	AR		√	√	√	

[3] LMRS is not used for any run because it consistently decreases performances in our preliminary experiments.

Table 2: Mean performances of our five runs compared with three baseline runs provided by NTCIR. Runs are sorted by GA-nG@1. In the table, Baseline-1, 2 and 3 denotes the performances of the baseline runs, Δ_{B2} indicates a higher performance than the baseline-2, and Δ_{R1} indicates a higher performance than Run 1 (our baseline). The highest performances are denoted as bold ones.

	BA-Hit@1	GA-Hit@1	GA-nG@1	GA-nDCG	GA-Q
Run 2	0.4980 (Δ_{B2}, Δ_{R1})	0.9967 (Δ_{B2})	**0.9211 (Δ_{B2}, Δ_{R1})**	**0.9747 (Δ_{B2}, Δ_{R1})**	**0.9690 (Δ_{B2}, Δ_{R1})**
Run 1	0.4980 (Δ_{B2})	0.9967 (Δ_{B2})	0.9203 (Δ_{B2})	0.9741 (Δ_{B2})	0.9682 (Δ_{B2})
Run 4	0.4847	0.9973 (Δ_{B2}, Δ_{R1})	0.9202 (Δ_{B2})	0.9745 (Δ_{B2}, Δ_{R1})	0.9688 (Δ_{B2}, Δ_{R1})
Baseline-2 (Length only)	0.4847	0.9953	0.9170	0.9735	0.9680
Run 3	0.4813	0.9960 (Δ_{B2})	0.8956	0.9679	0.9609
Run 5	**0.7773 (Δ_{B2}, Δ_{R1})**	**0.9987 (Δ_{B2}, Δ_{R1})**	0.8863	0.9604	0.9499
Baseline-3 (Posting Time)	0.3820	0.9940	0.8213	0.9460	0.9359
Baseline-1 (Random)	0.2713	0.9920	0.7751	0.9311	0.9169

Our five runs[4] are organized for the following purposes: Run 1 represents our baseline system. We intended to build a system with minimum number of features, which can show consistent and reasonable performances on best quality answer finding. For the purpose, we select the one feature in each feature set, which was most effective one of each set in the preliminary results. One exception is URL feature in the informativeness feature set. It is additionally included to Run 1 because we think that it reflects a somewhat different aspect of informativeness of an answer from NLA feature, which is also one of informativness features.

Run 2 represents our most effective system among our runs. We used all features showing positive effects in BA-Hit@1 in our preliminary experiments. Run 3 and Run 4 are purposed to examine features which was believed as advanced or novel ones. In the selection of features for Run 3 and 4, we did not very consider their effectiveness in the preliminary experiments. In Run 3, authority and expertise features are extensively examined, and In Run 4, some novel features that we newly proposed, for example, LW or LEX+NLA are additionally included. For Run 3 and Run 4, we intentionally exclude some known effective features (e.g., NLA) because they can be somewhat overlapped with newly added features (e.g., LEX+NLA). Run 5 is motivated to examine the analogical reasoning model. The features used for Run 5 are same to Run 1.

Regarding to Run 5, we would like to note that the training data and testing data used for official runs in NTCIR pilot task are not separated. All QA threads in testing data are in the training data. Under this setting, the analytical model will take unrealistic advantages: it will always have a chance to optimize parameters with correct BA answers. This cannot happen in the real world, so our results from Run 5 may not be very meaningful [1].

4.3 Results

Table 2 shows the experimental results of our five runs in the NTCIR pilot task. They are compared to the performances of another three baseline runs that NTCIR provided: (1) the results ranking answers in a thread randomly (Baseline-1), (2) ranking answers by answer length in descending order (Baseline-2), and (3) ranking answers by their posting time in ascending order (Baseline-3). In the table, BA-Hit@1 indicates a performance measured by *hit at rank 1* with BA as a ground truth, GA-Hit@1 means *hit at rank 1* with GA ground truth by regarding all L3, L2, L1 answers as good quality ones, GA-nG@1 is a normalized gain score at 1 with GA, GA-nDCG is NDCG score with GA, and GA-Q indicates Q-measure score using GA. For the detail information on the evaluation metrics, we refer the reader to [1].

As shown in the table, Run 1 and Run 2 consistently performs better than all NTCIR baseline systems for all evaluation metrics, and Run 2 shows the best performances based on all GA-based evaluation metrics except Run 5 at GA-Hit@1. Specifically, when preferring high relevance grades in GAs in an evaluation metric (e.g., GA-nG@1 based on graded relevance vs. GA-Hit@1 based on binary relevance), the differences between performances of Run 2 and the baseline runs show a tendency to become bigger. These results indicate that our approach can predict a good quality answers reasonably well by utilizing different aspects of answer quality.

One interesting fact regarding to Run 5 is that it shows a significantly better performance than others at BA-Hit@1 by taking an advantage considering correct BA for a given QA thread, but in GA-based evaluation results, its performances are only slightly better (at GA-Hit@1) or even worse than others (at GA-nG@1, GA-NDCG, and GA-Q). It can imply that highly optimizing a ranking model based on BAs has a risk to make a serious bias problem in training; basically, BAs are subjective and incomplete ground truths for good quality answers, and a characteristic of a BA in one QA thread can be quite different from other good quality answers in the thread. In this case, if we select parameters optimized for a BA in the thread, a model would fail to rank other good quality answers at high ranks although they can be better in terms of objective answer quality. Similar tendencies are also observed in our other runs. For example, the performance improvements of Run 2 over Baseline-2 in GA-based evaluation metrics are much smaller than in BA-Hit@1. It might indicate a necessity to develop a methodology for distinguishing objective (global) preferences on good quality answers from subjective (local) preferences.

Although our Run 1 and Run 2 show better performances than the simple baseline run using answer lengths (Baseline-2), the amount of improvement is only marginal in GA-based evaluation.

[4] Our final runs are denoted as from MSRA+MSR1 to MSRA+MSR5 in [2].

Interestingly, the NTCIR evaluation results show that such a simple length-based ranking method can achieve a very high performance (more than 0.9) in GA based evaluation metrics. For example, its GA-Hit@1 is more than 0.99. This indicates that a length feature is very strong indicator for answer quality, and it also implies that informativeness is dominant criteria for human to evaluate quality of answers. The same or similar experimental results also have been reported at the previous works on answer quality [13, 17].

Run 3 and Run 4, whose features are organized according to a novelty of features rather than a performance, show relatively lower performances than Run 2. Specifically, the performances of Run 3, which is mainly based on authority and expertise features, are generally lower than the baseline run using a length feature in many evaluation metrics. We suspect the following reasons; at first, for both Run 3 and 4, an answer length is not considered by removing NLA feature, which is the most effective feature for the best answer finding. Also, the authority features are not on an answer itself; they are mainly purposed to evaluate the answerer. However, there is a possibly that good authoritative answerer post a bad quality answer in some cases. Thus, without a help of other good indicators (e.g, NLA feature for informativeness), it cannot achieve a high precision in good answer quality finding.

Table 3: Performance comparison between Run 2 and BA. L3-Hit@1 means *hit at rank 1 with L3 answer* only[5].

	GA-nG@1	L3-Hit@1
Run 2	**0.9211**	**0.8054**
BA as top 1 rank	0.8900	0.7315

One another interesting fact in our analysis is that there is a high probability that top 1 rank in a QA thread from our runs is evaluated as a higher quality answer than BA in the same thread. Table 3 shows the comparison between the evaluation results of Run 2 and the results when BAs are regarded as the top rank in a system output (BA as top 1 rank). As shown in the table, Run 2 shows significantly better performances than BA as top 1 rank. This result shows the fact that the best answer selected by an asker is often not really the best quality answer objectively, and the automatic method to evaluate answer quality can be used for the problem of BAs.

4.3.1 Feature Analysis

To investigate the effectiveness of each feature, we performed '*leave one out*' experiments based on Run 2. The results are shown in the Table 4.

As shown in the table, the most effective features in the good quality answer finding are informative features, particularly, NLA feature utilizing an answer length. Also, GRS (relevance), URL, and NW (modality) feature contribute to the performance improvement, but there is no consistent benefit from the rest of the features used in Run 2 although they were effective ones in the preliminary experiment results. For example, PA, the feature utilizing posting positions of answers, was consistently effective in the preliminary experiments based on BAs, but it fails to improve a performance of BA-Hit@1 in the testing data. More investigation on those features is necessary, and it will be one of our future works. Table 4 also shows that in general, an effective feature in terms of BA-Hit@1 is also effective in GA-based evaluation.

Table 4: Performance changes when removing one feature from Run 2. Bold ones indicate positive features in each evaluation metrics.

	BA-Hit@1	GA-nG@1
Run 2	-	-
Run 2 – GRS	**-0.0020**	**-0.0014**
Run 2 – PS	**-0.0100**	+0.0023
Run 2 – UE	+0.0020	+0.0006
Run 2 – NLA	**-0.0413**	**-0.0527**
Run 2 – URL	**-0.0027**	**-0.0032**
Run 2 – PA	+0.0053	+0.0038
Run 2 – NW	**-0.0040**	**-0.0007**
Run 2 – AR	+0.0007	+0.0009

Table 5: Performance changes when adding a feature to Run 2

	BA-Hit@1	GA-nG@1
Run 2	-	-
Run 2 + LMRS	**+0.0027**	-0.0016
Run 2 + NBA	-0.0013	-0.0012
Run 2 + LW	-0.0027	-0.0010
Run 2 + LEX(NLA)	**+0.0007**	-0.0022
Run 2 + LEX(PS)	**+0.0033**	-0.0006

Also, Table 5 shows the performance changes when we add more features to Run 2. Note that those features examined in the table were not effective in our preliminary experiments. In terms of BA-Hit@1, there are three features improving performances when they are added, but all of them fail to improve a GA-nDCG@1 score. Similarly to our previous leave one out experiments, there is inconsistency in the official results and the preliminary experiment results. It may indicate that there is still a room to investigate on the feature selection.

5. Conclusion

In this paper, we described our approaches to find the best quality answer from online QA threads, which were used for the NTCIR-8 Community QA Pilot task. We investigated multiple different features from four aspects: relevance of an answer, authority and expertise of its answerer, informativeness, and its characteristics on discourse or modality. We also examined the effectiveness of two different statistical learning approaches in utilizing features.

Our systems are trained by utilizing the best answers selected by askers of questions, and evaluated with another ground truth data, which was built by multiple assessors independently from the best answers that askers select. NTCIR evaluation results showed that our approaches can be effective in finding the best quality answers.

[5] L3-Hit@1 scores are computed for QA threads having at least one L3 answer in the testing data. The threads which do not have any L3 answer are excluded in the evaluation.

For a thorough analysis, more experiments are required on the effect of our features. Also, it would be also necessary to conduct an additional experiment for comparing a classification approach and pairwise approach[6]. They will be our future works.

6. ACKNOWLEDGMENTS

This work was done when Jing Liu and Guwen Feng worked in Microsoft Research Asia.

7. REFERENCES

[1] Sakai, T., Ishikawa, D. and Kando, N. Overview of the NTCIR8 Community QA Pilot Task (Part II): System Evaluation. NTCIR-8 Proceedings, 2010.

[2] Ishikawa, D., Sakai, T. and Kando, N. Overview of the NTCIR-8 Community QA Pilot Task (Part I). The Test Collection and the Task. NTCIR-8 Proceedings, 2010.

[3] Agichtein, E., Castillo, C., Donato, D., Gionis, A. and Mishne, G. Finding high-quality content in social media. Proceedings of the international conference on Web search and web data mining, pages 183-194, 2008.

[4] Joachims, T. Training Linear SVMs in Linear Time. Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD), 2006.

[5] Wang, X.J., Tu, X., Feng, D. and Zhang, L. Ranking community answers by modeling question-answer relationships via analogical reasoning. Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, pages 179-186, 2009.

[6] Ponte, J.M. and Croft, W.B. A language modeling approach to information retrieval. Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, pages 275-281, 1998.

[7] Cong, G., Wang, L., Lin, C.Y., Song, Y.I. and Sun, Y. Finding question-answer pairs from online forums. Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, pages 467-474, 2008.

[8] Zhang, J., Ackerman, M.S. and Adamic, L. Expertise networks in online communities: structure and algorithms. Proceedings of the 16th international conference on World Wide Web, pages 221-230, 2007.

[9] Jurczyk, P. and Agichtein, E. Discovering authorities in question answer communities by using link analysis. Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, pages 919-922, 2007.

[10] Jurczyk, P. and Agichtein, E. Hits on question answer portals: exploration of link analysis for author ranking. Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, 2007.

[11] Adamic, L.A., Zhang, J., Bakshy, E. and Ackerman, M.S. Knowledge sharing and yahoo answers: everyone knows something. Proceeding of the 17th international conference on World Wide Web, pages 665-674, 2008.

[12] Balog, K., Azzopardi, L. and de Rijke, M. Formal models for expert finding in enterprise corpora. Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pages 43-50, 2006.

[13] Jeon, J., Croft, W.B., Lee, J.H. and Park, S. A framework to predict the quality of answers with non-textual features. Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pages 227-235, 2006.

[14] Erkan, G. and Radev, D.R. Lexrank: Graph-based lexical centrality as salience in text summarization. Journal of Artificial Intelligence Research, volume 22, number 1, pages 457-479, 2004.

[15] Nam, K.K., Ackerman, M.S. and Adamic, L.A. Questions in, knowledge in?: a study of naver's question answering community. Proceedings of the 27th international conference on Human factors in computing systems, pages 779-788, 2009.

[16] O'Mahony, P., Smyth, B., Using Readability Tests to Predict Helpful Product Reviews. Proceedings of the 9th RIAO Conference (RIAO 2010), 2010.

[17] Lihn, H., Lee, J.T., Song, Y.I., Rim, H.C., A Model for Evaluating the Quality of User-Created Documents. Proceedings of AIRS 2008, 2008.

[18] Hisami Suzuki. 2004. Phrase-Based Dependency Evaluation of a Japanese Parser. In the Proceedings of European Language Resources Association, May 2004

[19] http://research.microsoft.com/en-us/projects/japanesenlp/

[6] We have conducted the experiments for the purpose of comparing those two different approaches, but unfortunately, the experiments were not finished at the point that this paper was written.