2006

A Framework to Predict the Quality of Answers with NonTextual

University of Massachusetts Amherst

A Framework to Predict the Quality of Answers with Non-Textual Features

Jiwoon Jeon[1], W. Bruce Croft[1], Joon Ho Lee[2] and Soyeon Park[3]
[1]Center for Intelligent Information Retrieval, University of Massachusetts-Amherst, MA, 01003, USA
[jeon,croft]@cs.umass.edu
[2]School of Computing, College of Information Science, Soong-sil University, Seoul, South Korea
joonho@computing.soongsil.ac.kr
[3]Department of Library and Information Science, Duksung Women's University, Seoul, South Korea
sypark@duksung.ac.kr

ABSTRACT

New types of document collections are being developed by various web services. The service providers keep track of non-textual features such as click counts. In this paper, we present a framework to use non-textual features to predict the quality of documents. We also show our quality measure can be successfully incorporated into the language modeling-based retrieval model. We test our approach on a collection of question and answer pairs gathered from a community based question answering service where people ask and answer questions. Experimental results using our quality measure show a significant improvement over our baseline.

Categories and Subject Descriptors

H.3.0 [**Information Search and Retrieval**]: General

General Terms

Algorithms, Measurement, Experimentation

Keywords

Information Retrieval, Language Models, Document Quality, Maximum Entropy

1. INTRODUCTION

Every day new web services become available and these services accumulate new types of documents that have never before existed. Many service providers keep non-textual information related to their document collections such as click-through counts, or user recommendations. Depending on the service, the non-textual features of the documents may be numerous and diverse. For example, blog users often recommend or send interesting blogs to other people. Some blog services store this information for future use. Movie sites saves user reviews with symbolic representations rating the movie (such as **A** or ⋆⋆⋆⋆⋆).

This non-textual information has great potential for improving search quality. In the case of the homepage finding, link information has proved to be very helpful in estimating the authority or the quality of homepages [2, 10]. Usually textual features are used to measure relevance of the document to the query and non-textual features can be utilized to estimate the quality of the document. While smart use of non-textual features is crucial in many web services, there has been little research to develop systematic and formal approaches to process these features.

In this paper, we demonstrate a method for systematically and statistically processing non-textual features to predict the quality of documents collected from a specific web service. For our experiment, we choose a community based question answering service where users ask and answer questions to help each other. Google Answers[1], Ask Yahoo[2], Wondir[3] and MadSciNet[4] are examples of this kind of service. These services usually keep lots of non-textual features such as click counts, recommendation counts, etc. and therefore can be a good testbed for our experiments.

In order to avoid the lag time involved with waiting for a personal response, these services typically search their collection of question and answer (Q&A[5]) pairs to see if the same question has previously been asked. In the retrieval of Q&A pairs, estimating the quality of answers is important because some questions have bad answers. This happens because some users make fun of other users by answering nonsense. Sometimes irrelevant advertisements are given as answers. The followings are examples of bad answers found from community based question answering services.

SIGIR'06, August 6–10, 2006, Seattle, Washington, USA.

[1]http://answers.google.com/
[2]http://ask.yahoo.com/
[3]http://www.wondir.com/
[4]http://www.madsci.org/
[5]In this paper, **Q&A** means **'question and answer'** and is used only as an adjective such as **'Q&A pairs'** and **'Q&A collections'**. **Q&A** must be discerned from **QA** that is often used to refer to automated question answering. Therefore, in this paper, **'Q&A service'** means services such as Google Answers where people answer other people's questions.

Q: What is the minimum positive real number in Matlab?
A: Your IQ.

Q: What is new in Java2.0?
A: Nothing new.

Q: Can I get a router if I have a usb dsl modem?
A: Good question but I do not know.

The answer quality problem becomes important when there are many duplicated questions, or many responses to a single question. The duplicated questions are generated because some users post their questions without carefully searching existing collections. These semantically duplicated questions have answers with varying quality levels, therefore measuring relevance alone is not enough and the quality of answers must be considered together.

We use kernel density estimation [5] and the maximum entropy approach [1] to handle various types of non-textual features and build a stochastic process that can predict the quality of documents associated with the features. We do not use any service or collection specific heuristics, therefore our approach can be used in many other web services. The experimental results show the predictor has the ability to distinguish good answers from bad ones.

In order to test whether quality prediction can improve the retrieval results, we incorporate our quality measure into the query likelihood retrieval model [18]. Our goal in the retrieval experiments is to retrieve relevant and high quality Q&A pairs for a given query. In other words, the question and the query must describe the same information needs and the quality of answer must be good. Experimental results show significant improvement in retrieval performance can be achieved by introducing the quality measure.

We discuss related work in section 2. Section 3 describes our data collection. Section 4 explains in detail how we calculate the quality of answers. The retrieval experiments and results are presented in section 5. Section 6 concludes this paper.

2. RELATED WORK

Many factors decide the quality of documents (or answers). Strong et al. [20] listed 15 factors and classified those factors into 4 categories: contextual, intrinsic, representational and accessibility. Zhu and Gauch [24] came up with 6 factors to define the quality of web pages. However, so far, there is no standard metric to measure and represent the quality of documents.

There has been extensive research to estimate the quality of web pages. Much of the work is based on link analysis [2, 10]. A few researchers [24, 23] tried to use textual features. Zhou and Croft [23] proposed a document quality model that uses only content based features such as the information-noise ratio and the distance between the document language model and the collection model. However, little research has been done to estimate the quality of answers in a collection of question and answer pairs.

FAQ retrieval research [3, 19, 13, 21, 9] has focused on finding similar questions from FAQ collections. More recently, Jeon et al. [6] proposed a retrieval method based on machine translation to find similar questions from community based question and answering services. However, none of them have considered the quality of answers in the retrieval process.

Quality of Answers, Test Samples

Bad	Medium	Good
208 (12.2%)	393 (23.1%)	1099 (64.7%)

Quality of Answers, Training Samples

Bad	Medium	Good
81 (9.1%)	212 (23.7%)	601 (67.2%)

Table 1: The relationships between questions and answers in Q&A pairs are manually judged. The test samples consist of 1700 Q&A pairs. The training samples have 894 Q&A pairs. Both training and test samples show similar statistics.

The language modeling framework [18] provides a natural way of combining prior knowledge in the form of a prior probability. Prior information such as time, quality and popularity has been successfully integrated using as a prior probability on the document [11, 23, 14]. We also use the prior probability to combine quality and relevance.

Berger et al. [1] proposed the use of the maximum entropy approach for various natural language processing tasks in mid 1990's and after that many researchers have applied this method successfully to a number of other tasks including text classification [16, 17] and image annotation [7].

3. DATA COLLECTION

3.1 Test Collection Building

We collected 6.8 million Q&A pairs from the Naver Q&A service[6]. All questions and answers are written in Korean. We randomly selected 125 queries from the search log of a single day. We used a pooling technique [4] to find relevant Q&A pairs for those queries. We ran 6 different search engines and gathered the top 20 Q&A pairs from each search result. Annotators manually judged the candidates in three levels: Bad, Medium and Good. Annotators read the question part of the Q&A pair. If the question part addressed the same information need as the query, then the Q&A pair was judged as relevant. When the information need of a query was not clear, annotators looked up click-through logs of the query and guessed the intent of the user. In all, we found 1,700 relevant Q&A pairs.

3.2 Manual Judgment of Answer Quality and Relevance

The quality of a Q&A depends on both the question part and the answer part. The followings are examples of bad questions that can be found from community based Q&A services.

What is one plus one?
Who is more handsome than me?
I am sad.

[6]http://www.naver.com/ Naver provides a community based question answering service in South Korea. In this service, users help each other by posting and answering questions. This service is very popular and has accumulated more than 10 million Q&A pairs over last 3 years.

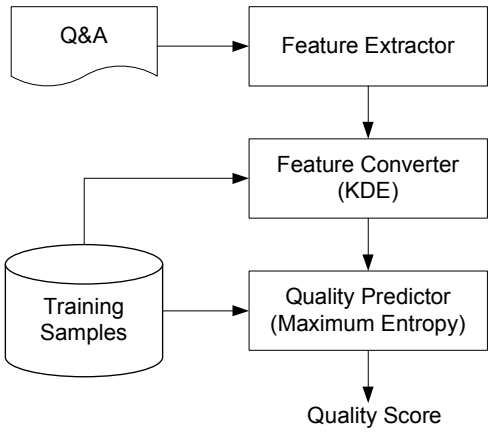

Figure 1: Architecture of the quality predictor.

Users can not get any useful information by reading answers for these bad questions. We found that bad questions always lead to bad quality answers. Answers for these bad questions usually blame the questioner with short insulting words. Therefore, we decide to estimate only the quality of answers and consider it as the quality of the Q&A.

In general, good answers tend to be relevant, informative, objective, sincere and readable. We may separately measure these individual factors and combine scores to calculate overall the quality of the answer. But this approach requires development of multiple estimators for each factor and the combination is not intuitive. Therefore, we propose to use a holistic view to decide the quality of an answer. Our annotators read answers, consider all of the above factors and specify the quality of answers in just three levels: Bad, Medium and Good. This holistic approach shifts the burden of combining individual quality metrics to human annotators.

In subsection 3.1, we explained how we found 1700 relevant Q&A pairs to the 125 queries. For the 1,700 Q&A pairs, we manually judged the quality of answers. In this step, the query was ignored and only the relationships between questions and answers in Q&A pairs are considered. The results of the quality judgment are in Table 1. Around 1/3 of the answers have some sort of quality problems. Approximately 1/10 of the answers are bad. Therefore, we need to properly handle these bad documents (Q&A pairs).

To build a machine learning based quality predictor, we need training samples. We randomly selected 894 new Q&A pairs from the Naver collection and manually judged the quality of the answers in the same way. Table 1 shows the test and the training samples have similar statistics.

4. ESTIMATING ANSWER QUALITY

In this section, we explain how to predict the quality of answers. Figure 1 shows the architecture of our quality prediction system. The input of the system is a Q&A pair and the output is the probability that the Q&A pair has a good answer. The following subsections discuss each component in detail.

4.1 Feature Extraction

First we need to extract feature vectors from a Q&A pair. We extract 13 non-textual features. Table 2 shows the list of the features. In the Naver Q&A service, multiple answers are possible for a single question and the questioner selects the best answer. Unless otherwise mentioned, we extract features only from the best answer. The following is a detailed explanation of each individual feature.

Answerer's Acceptance Ratio The ratio of best answers to all the answers that the answerer answered previously.

Answer Length The length of the answer. Depending on points of view, this feature can be thought of as a textual feature. However, we add this feature because it can be easily extracted without a serious analysis of the content of the text and is known to be helpful in measuring the quality of online writings [12].

Questioner's Self Evaluation The questioner gives from one to five stars($\star$) to the answer when they select the answer.

Answerer's Activity Level If a user asks and answers many times in the service, the user gets a high activity score.

Answerer's Category Specialty If a user answers many questions in a category, the user gets a high category specialty score for that category.

Print Counts The number of times that users print the answer.

Copy Counts The number of times that users copy the answer to their blog.

Users' Recommendation The number of times the Q&A pair is recommended by other users.

Editor's Recommendation Sometimes editors of the service upload interesting Q&A pairs on the front page of the service.

Sponsor's Answer For some categories, there are approved answerers who are nominated as a 'sponsor' of the category.

Click Counts The number of times the Q&A pair is clicked by other users.

Number of Answers The number of answers for the given question.

Users' Dis-Recommendation The number of time the Q&A pair is dis-recommended by other users.

Although some features are specific to the Naver service, other features such as answer length, the number of answers and click counts are common in many Q&A services. Some features such as recommendation counts and evaluation scores using stars can be found in many other web services. As can be seen from table 2, various numerical types are used to represent diverse features.

Features	Type	Corr
Answerer's Acceptance Ratio	Percentile	0.1837
Answer Length	Integer	0.1733
Questioner's Self Evaluation	1,...5	0.1675
Answerer's Activity Level	Integer	0.1430
Answerer's Category Specialty	Integer	0.1037
Print Counts	Integer	0.0528
Copy Counts	Integer	0.0469
Users' Recommendation	Integer	0.0351
Editor's Recommendation	Binary	0.0285
Sponsor's Answer	Binary	0.0232
Click Counts	Integer	-0.0085
Number of Answers	Integer	-0.0297
User's Dis-Recommendation	Integer	-0.0596

Table 2: List of features. The second column shows numerical types of the features. The last column shows the correlation coefficients between the feature values and the manually judged quality scores. Higher correlation means the feature is a better indicator to predict the quality of answers. Minus values means there are negative correlations.

4.2 Feature Analysis

We calculate the correlation coefficient (or Pearson's correlation) between individual features and the manual quality judgment scores (good answers have higher scores: Bad=0, Medium=1, Good=2). The third column in table 2 shows the coefficient values.

Surprisingly, "Questioner's Self Evaluation" is not the feature that has the strongest correlation with the quality of the answer. This means the questioner's self evaluation is subjective and often does not agree with other users opinion about the answer. Many people simply appreciate getting answers from other people regardless of the quality of the answers, and give high scores for most of the answers. This user behavior may be related to the culture of Korean users. Performing similar analysis with other user groups, for example with North American users, may give an interesting comparison.

"Sponsor's Answer" and "Editor's Recommendation" are good features because they always guarantee the quality of answers but only small number of Q&A pairs are recommended by editors or written by sponsors. Therefore, these features have little impact on overall performance and the coefficient values are relatively small.

With the exception of the answer length, most of the important features are related to the expertise or the quality of the answerer. This result implies that knowing about the answerer is very important in estimating the quality of answers. We may get better estimations using these non-textual features than analyzing contents of answers using textual features because accurately understanding the contents of the text is very hard with the current technology.

4.3 Feature Conversion using KDE

Maximum entropy models require monotonic features that always represent stronger evidence with bigger values. For example, the number of recommendations is a monotonic feature since more recommendations means better quality. However, the length of an answer is not a monotonic feature because longer answers do not always mean better answers.

Features	Corr (Original)	Corr (KDE)
Answer Length	0.1733	0.4285
Answerer's Activity Level	0.1430	0.1982
Answerer's Category Specialty	0.1037	0.2103

Table 3: Feature conversion results. The second column represents the correlation between the raw feature value and the quality scores. The third column shows the correlation coefficients after converting features using kernel density estimation. Much stronger correlations are observed after the conversion.

Most of the previous work [16, 17] on text classification using the maximum entropy approach used only monotonic features such as frequency of words or n-grams. Therefore little attention was given to solve the problem of non-monotonic features. However, we have non-monotonic features and need to convert these features into monotonic features.

We propose using kernel density estimation (KDE) [5]. KDE is a nonparametric density estimation technique that overcomes the shortcomings of histograms. In KDE, neighboring data points are averaged to estimate the probability density of a given point. We use the Gaussian kernel to give more influence to closer data points. The probability of having a good answer given only the answer length, $P(good|AL)$, can be calculated from the density distributions.

$$P(good|AL) = \frac{P(good)F(good|AL)}{P(good)F(good|AL) + P(bad)F(bad|AL)} \quad (1)$$

where AL denotes the answer length and $F()$ is the density function estimated using KDE. $P(good)$ is the prior probability of having a good quality answer estimated from the training data using the maximum likelihood estimator. $P(bad)$ is measured in the same way.

Figure 2 shows density distributions of good quality answers and bad quality answers according to the answer length. Good answers are usually longer than bad answers but very long and bad quality answers also exist. The graph shows $P(good|AL)$ calculated from the density distributions. The probability initially increases as the answer length becomes longer but eventually starts decreasing. The probability that an answer is high quality is high for average-length answers, but low for very long answers. This accurately reflects what we see in practice in the Naver data.

We use $P(good|AL)$ as our feature value instead of using the answer length directly. This converted feature is monotonic since a bigger value always means stronger evidence. The 894 training samples are used to train the kernel density estimation module. Table 3 shows the power of this conversion. We calculate the correlation coefficients again after converting a few non-monotonic features. In the case of the answer length, the strength of the correlation is dramatically improved and it becomes the most significant feature.

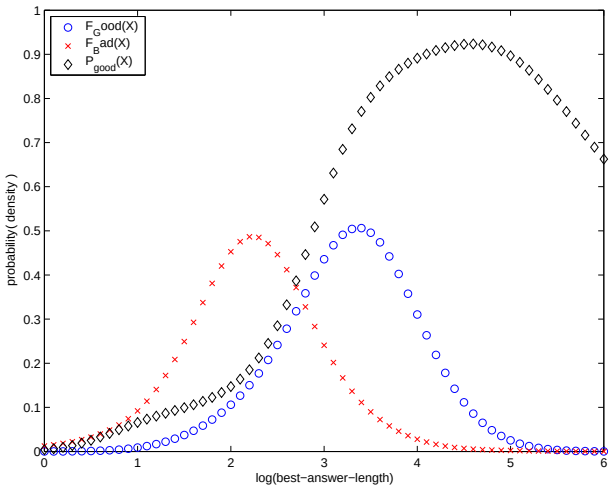

Figure 2: Density distributions of good answers and bad answers measured using KDE. The x axis is log(answer length) and the y axis is the density or the probability. The graph also shows the probability of having a good answer given the answer length.

4.4 Maximum Entropy for Answer Quality Estimation

We use the maximum entropy approach to build our quality predictor for the following reasons. First, the approach generates purely statistical models and the output of the models is a probability. The probability can be easily integrated into other statistical models. Our experimental results show the output can be seamlessly combined with statistical language models. Second the model can handle a large number of features and it is easy to add or drop features. The models are also robust to noisy features.

We assume that there is a random process that observes a Q&A pair and generates a label y, an element of a finite set $Y = \{good, bad\}$. Our goal is making a stochastic model that is close to the random process. We construct a training dataset by observing the behavior of the random process. The training dataset is $(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)$. x_i is a question and answer pair and y_i is a label that represents the quality of the answer. We make 894 training samples from the training data.

4.4.1 Predicate Functions and Constraints

We can extract many statistics from the training samples and the output of our stochastic model should match these statistics as much as possible. In the maximum entropy approach, any statistic is represented by the expected value of a feature function. To avoid confusion with the document features, we refer to the feature functions as predicates. We use 13 predicates. Each predicate corresponds to each document feature that we explained in the previous section.

$$f_i(x,y) = \begin{cases} kde(x_{fi}) & \text{if } i^{th}\text{feature is non-monotonic} \\ x_{fi} & \text{otherwise} \end{cases} \quad (2)$$

where $f_i(x,y)$ is the i^{th} predicate and x_{fi} is the raw value of the i^{th} feature in Q&A pair x.

The expected value of a predicate with respect to the training data is defined as follows,

$$\tilde{p}(f_i) = \sum_{x,y} \tilde{p}(x,y) f_i(x,y) \quad (3)$$

where $\tilde{p}(x,y)$ is a empirical probability distribution that can be easily calculated from the training data. The expected value of the predicate with respect to the output of the stochastic model should be the same with the expected value measured from the training data.

$$\sum_{x,y} \tilde{p}(x,y) f_i(x,y) = \sum_{x,y} \tilde{p}(x) p(y|x) f_i(x,y) \quad (4)$$

where $p(y|x)$ is the stochastic model that we want to construct. We call the equation (4) a constraint. We have to choose a model that satisfy these constraints for all predicates.

4.4.2 Finding Optimal Models

In many cases, there are infinite number of models that satisfy the constraints explained in the previous subsection. In the maximum entropy approach, we choose the model that has maximum conditional entropy

$$H(p) = -\sum_{x,y} \tilde{p}(x) p(y|x) \log p(y|x) \quad (5)$$

There are a few algorithms that find an optimal model which satisfy the constraints and maximize the entropy. Generalized Iterative Scaling and Improved Iterative Scaling have been widely used. We use Limited Memory Variable Metric method which is very effective for Maximum Entropy parameter estimation [15]. We use Zhang Le's maximum entropy toolkit[7] for the experiment.

The model is represented by a set of parameters λ. Each predicate has a corresponding parameter and the following is the final equation to get the probability of having a good answer or bad answer.

$$p(y|x) = \frac{1}{Z(x)} \exp\left[\sum_{i=1}^{13} \lambda_i f_i(x,y)\right] \quad (6)$$

where $Z(x)$ is a normalization factor.

4.4.3 Performance of the Predictor

We build the predictor using the 894 training samples and test using the 1700 test samples. The output of the predictor is the probability that the answer of the given Q&A pair is good. The average output for good Q&A pairs is 0.9227 and the average output for bad Q&A pairs is 0.6558. In both cases, the averages are higher than 0.5 because the prior probability of having a good answer is high. As long as this difference is consistent, it is possible to build an accurate classifier using this probability estimate.

We rank 208 bad examples and 1099 good examples in the test collection together by the descending order of the output values. Figure 3 shows the quality of the ranking using the recall-precision graph. The predictor is significantly better than random ranking. In the top 100, all Q&A pairs are good. The top 250 pairs contain 2 bad pairs and the top 500 pairs contain 9 bad pairs. The results show that the predictor has the ability to discriminate good answers from

[7] http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html

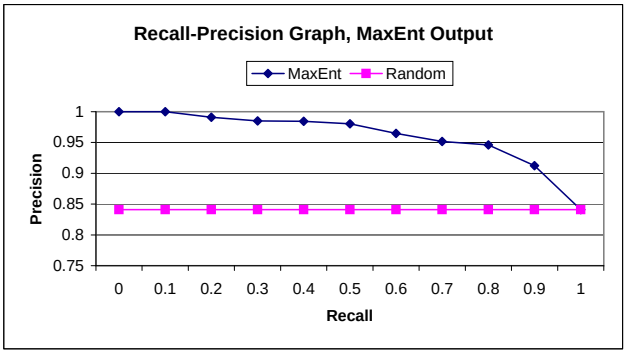

Figure 3: Performance of the quality predictor. 11pt recall-precision graph. Note that the y-axis scale starts from 0.75. 'Random' is the result of random ranking that positions Q&A pairs randomly.

bad answers. In future work, by increasing the size of the training samples, we may get better performance. In the next section, we investigate the effectiveness of the predictor in the context of retrieval.

5. RETRIEVAL EXPERIMENTS

We test whether the quality measure can improve retrieval performance. As a baseline experiment, we retrieve Q&A pairs using the query likelihood retrieval model[18]. The 125 queries are used and the question part of the Q&A pair is searched to find relevant Q&A pairs to the query, because the question part is known to be much more useful than the answer part in finding relevant Q&A pairs [8, 6]. This baseline system may return relevant Q&A pairs, but there is no guarantee about the quality of the answers. We incorporate the quality measure into the baseline system and compare retrieval performance.

5.1 Retrieval Framework

In the query likelihood retrieval model, the similarity between a query and a document is given by the probability of the generating the query from the document language model.

$$sim(Q, D) = P(D|Q) = P(D)P(Q|D)/P(Q) \qquad (7)$$

$P(Q)$ is independent of documents and does not affect the ranking. For the document model, usually, i.i.d sampling and unigram language models are used.

$$P(Q|D) = P(D) \prod_{w \in Q} P(w|D) \qquad (8)$$

$P(D)$ is the prior probability of document D. Query independent prior information such as time, quality and popularity have been successfully integrated into the model using the prior probability [11, 23, 14]. Since our estimation of the quality is given by a probability and query independent, the output of the quality predictor can be plugged into the retrieval model using the prior probability without any modification such as normalization. Therefore, in our approach, $P(D) = p(y|x = D)$ and $p(y|x)$ is given as in equation (6).

To avoid zero probabilities and estimate more accurate document language models, documents are smoothed using a background collection,

$$P(w|D) = (1 - \lambda)P_{ml}(w|D) + \lambda P_{ml}(w|C) \qquad (9)$$

$P_{ml}(w|C)$ is the probability that the term w is generated from the collection C. $P_{ml}(w|C)$ is estimated using the maximum likelihood estimator. λ is the smoothing parameter. We use Dirichlet smoothing [22]. The optimal parameter value is found by exhaustive search of the parameter space. We use the implementation of the query likelihood retrieval model in the Lemur toolkit[8].

5.2 Evaluation Method

In order to automatically evaluate retrieval performance, usually a relevance judgment file is made. This file contains lists of relevant documents to queries and an evaluation system looks up this file to automatically assess the performance of search engines. We made three different relevance judgment files. The first one (Rel_1) considers only the relevance between the query and the question, if the question part of a Q&A pair addresses the same information need as the query, the Q&A pair is considered to be relevant to the query. The second file (Rel_2) considers both the relevance and the quality of Q&A pairs. If the quality of the the answer is judged as 'bad', then the Q&A pair is removed from the relevance judgment file even if the question part is judged as relevant to the query. The last judgment file (Rel_3) requires a stronger requirement of quality. If the quality of the answer is judged 'bad' or 'medium', then the Q&A pair is removed from the file and only relevant and good quality Q&A pairs remain in the file.

Rel_2 is a subset of Rel_1 and Rel_3 is a subset of Rel_2. From table 1, Rel_1 contains 1700 Q&A pairs, Rel_2 has 1492 pairs and Rel_3 includes 1099 pairs. Most of the previous experiments in FAQ retrieval, only the relevance of the question is considered and they used relevance judgment file like Rel_1. We believe the performance measured using Rel_2 or Rel_3 is closer to real user satisfaction, since they take into account both relevance and quality.

5.3 Experimental Results

We measure retrieval performance using various standard evaluation metrics such as the mean average precision, R-precision and 11pt recall-precision graphs. Table 4 and Figure 4 show the retrieval results.

Table 4 shows that the retrieval performance is significantly improved regardless of the evaluation metric after adding the quality measure. Surprisingly, the retrieval performance is significantly improved even when we use the relevance judgment file that does not consider quality. This implies bad quality Q&A pairs tend not to be relevant to any query and incorporating the quality measure pulls down these useless Q&A pairs to lower ranks and improves the retrieval results overall.

Because Rel_2 has smaller number of relevant Q&A pairs and Rel_3 contains even smaller number of the pairs, the retrieval performance is lower. However, the performance drop becomes much less dramatic when we integrate the quality measure. The retrieval performance evaluated by Rel_2 is better than the performance evaluated by Rel_1, if we incorporate the quality measure.

[8]http://www.lemurproject.org/

Mean Average Precisions

	Rel_1	Rel_2	Rel_3
Without Quality	0.294	0.267	0.222
With Quality	0.322	0.316	0.290
P-value	0.007	1.97E-06	2.96E-11

R-Precisions at Rank 10

	Rel_1	Rel_2	Rel_3
Without Quality	0.366	0.313	0.236
With Quality	0.427	0.404	0.338
P-value	3.59E-05	5.81E-09	1.18E-12

Table 4: Comparison of retrieval performance. The upper table shows mean average precisions and the lower table shows R-precisions at rank 10. The P-value is calculated using the sign test. Smaller value means more significant difference.

We do a sign test to check whether the improvements are statistically significant. The third rows in Table 4 show the P-values of the test. The results show all the improvements are significant at the 99% confidence level. The significance of the improvement is higher when we use stricter requirements for the correct Q&A pairs.

Figure 4 shows 11pt recall-precision graphs. In all recall levels, we get improved precisions by adding the quality measure. The improvement becomes bigger when we use Rel_3 than Rel_1.

6. CONCLUSION AND FUTURE WORK

In this paper, we showed how we could systematically and statistically process non-textual features that are commonly recorded by web services, to improve search quality. We did not use any service or collection specific heuristics. We used statistical methods in every step of the development. Therefore, we believe our approach can be applied to other web services.

We applied our method to improve the quality of the retrieval service that is attached to a community-based question answering web site. We predicted the quality of answers accurately using the maximum entropy approach and kernel density estimation. The predicted quality scores were successfully incorporated into the language modeling-based retrieval model. We achieved significant improvement in retrieval performance.

We plan to improve the feature selection mechanism and develop a framework that can handle both textual and non-textual feature together and apply it to other web services.

7. ACKNOWLEDGEMENTS

This work was supported by NHN Corp. and the Center for Intelligent Information Retrieval. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

8. REFERENCES

[1] A. Berger, S. D. Pietra, and V. D. Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

[2] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[3] R. D. Burke, K. J. Hammond, V. A. Kulyukin, S. L. Lytinen, N. Tomuro, and S. Schoenberg. Question answering from frequently asked question files: Experiences with the faq finder system. *AI Magazine*, 18(2):57–66, 1997.

[4] D. Harman. Overview of the first text retrieval conference (trec-1). In *Proceedings of the First TREC Conference*, pages 1–20, 1992.

[5] J. Hwang, S. Lay, and A. Lippman. Nonparametric multivariate density estimation: A comparative study. *IEEE Transactions of Signal Processing*, 42(10):2795–2810, 1994.

[6] J. Jeon, W. B. Croft, and J. H. Lee. Finding similar questions in large question and answer archives. In *Proceedings of the ACM Fourteenth Conference on Information and Knowledge Management*, pages 76–83, 2005.

[7] J. Jeon and R. Manmatha. Using maximum entropy for automatic image annotation. *Image and Video Retrieval Third International Conference, CIVR 2004, Proceedings Series: Lecture Notes in Computer Science*, 3115:24–32, 2004.

[8] V. Jijkoun and M. de Rijke. Retrieving answers from frequently asked questions pages on the web. In *Proceedings of the ACM Fourteenth Conference on Information and Knowledge Management*, pages 76–83, 2005.

[9] H. Kim and J. Seo. High-performance faq retrieval using an automatic clustering method of query logs. *Information Processing and Management*, 42(3):650–661, 2006.

[10] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[11] W. Kraaij, T. Westerveld, and D. Hiemstra. The importance of prior probabilities for entry page search. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 27–34, 2002.

[12] L. S. Larkey. Automatic essay grading using text categorization techniques. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 90–95, 1998.

[13] M. Lenz, A. Hubner, and M. Kunze. Question answering with textual cbr. In *Proceedings of the Third International Conference on Flexible Query Answering Systems*, pages 236–247, 1998.

[14] X. Li and W. B. Croft. Time-based language models. In *Proceedings of the Twelfth ACM International Conference on Information and knowledge management*, pages 469–475, 2003.

[15] R. Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of Conference on Computational Natural Language Learning*, pages 49–55, 2002.

[16] K. Nigam, J. Lafferty, and A. McCallum. Using maximum entropy for text classification. In *Proceedings of IJCAI-99 Workshop on Machine Learning for Information Filtering*, pages 61–67, 1999.

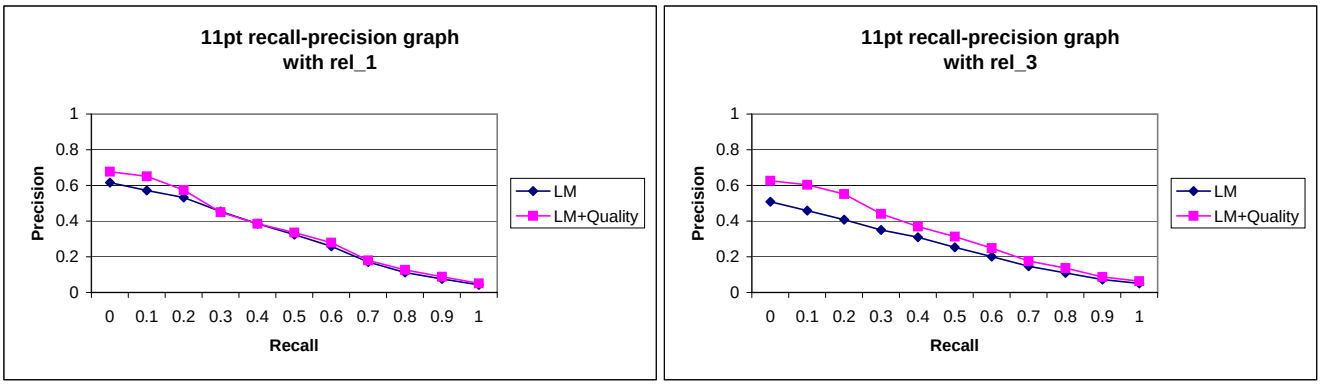

Figure 4: 11pt recall precision graphs. LM is the result of using the query likelihood retrieval model. LM+Quality is the result after incorporating the quality measure into the same retrieval model.

[17] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2002.

[18] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 275–281, 1998.

[19] E. Sneiders. Automated faq answering: Continued experience with shallow language understanding. In *Proceedings for the 1999 AAAI Fall Symposium on Question Answering Systems*, 1999.

[20] D. M. Strong, Y. W. Lee, and R. Y. Wang. Data quality in context. *Communications of the ACM*, 40(5):103–110, 1997.

[21] C.-H. Wu, J.-F. Yeh, and M.-J. Chen. Domain-specific faq retrieval using independent aspects. *ACM Transactions on Asian Language Information Processing*, 4(1):1–17, 2005.

[22] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 334–342, 2001.

[23] Y. Zhou and W. B. Croft. Document quality models for web ad hoc retrieval. In *Proceedings of the ACM Fourteenth Conference on Information and Knowledge Management*, pages 331–332, 2005.

[24] X. Zhu and S. Gauch. Incorporating quality metrics in centralized/distributed information retrieval on the world wide web. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 288–295, 2000.