World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:10, No:1, 2016

Leveraging Quality Metrics in Voting Model Based Thread Retrieval

Atefeh Heydari, Mohammadali Tavakoli, Zuriati Ismail, Naomie Salim

Open Science Index, Computer and Information Engineering Vol:10, No:1, 2016 publications.waset.org/10003397/pdf

***Abstract*—**Seeking and sharing knowledge on online forums have made them popular in recent years. Although online forums are valuable sources of information, due to variety of sources of messages, retrieving reliable threads with high quality content is an issue. Majority of the existing information retrieval systems ignore the quality of retrieved documents, particularly, in the field of thread retrieval. In this research, we present an approach that employs various quality features in order to investigate the quality of retrieved threads. Different aspects of content quality, including completeness, comprehensiveness, and politeness, are assessed using these features, which lead to finding not only textual, but also conceptual relevant threads for a user query within a forum. To analyse the influence of the features, we used an adopted version of voting model thread search as a retrieval system. We equipped it with each feature solely and also various combinations of features in turn during multiple runs. The results show that incorporating the quality features enhances the effectiveness of the utilised retrieval system significantly.

***Keywords*—**Content quality, Forum search, Thread retrieval, Voting techniques.

I. Introduction

WITH the fast growth of the internet in recent years, online forums are becoming one of the most common forms of social media for discussions over a wide range of general, such as politics, medical issues and finance, and specific topics, including software and hardware discussions, scientific issues etc. The question-answer nature of online forums allows users to illuminate their problems in their own words. Thus, domain experts can easily generate customised solutions for the question. This property of online forums distinguishes them from other online social media. Consequently, more and more people are becoming contributors to online forums in order to alleviate their own problems or challenge other users to generate solutions for open threads. This routine has turned online forums into a source of huge volumes of user-generated data.

Since the profusion of information is growing in recent years, the lake of systems for automatically retrieving valuable information is undeniable. The problem of searching for and extracting relevant information emerged in online forums due to the wealth of valuable information generated in the form of threads. The information obtained from online forums is not only advantageous for individuals as it facilitates them to get customised answers for their questions but is also useful for organisations as it can help them shape their products and services to customers' requirements [1], let communities know their strengths and weaknesses, foster politicians to trade off their situation in society and analyse their policies' pros and cons and enable scientists to extract valuable information from it, such as educational area researchers to assess students' collaboration. Therefore, rapidly accumulating information in online forums and its profitableness for individuals and societies has intensified the significance of thread retrieval, which that can be viewed as an extension of information retrieval with the aim of extracting relevant and needed information from these massive sources of textual data.

Although thread retrieval is facilitating information access for users, the content quality of retrieved information is a challenging area of research. Users of online forums desire to obtain the most relevant threads to their queries. However, the majority of state-of-the-art thread retrieval systems have ignored the importance of the content quality of retrieved documents. In this research, we have equipped a state-of-the-art thread retrieval method with a number of quality features in order to assess the efficiency of the features in improving the quality of retrieved documents.

Atefeh Heydari, Mohammadali Tavakoli, Naomie Salim are with the University Teknologi Malaysia, Johor Bahru, 81300 Malaysia (phone: +989212378119; e-mail: hatefeh@live.utm.my, tmohammadali2@utm.my, naomie@utm.my).

Zuriati Ismail is with the Universiti Teknologi MARA Johor, KM12 Jalan Muar, 85009 Segamat, Johor, Malaysia (e-mail: zuria986@johor.uitm.edu.my) and a PhD candidate in Computer Science, Universiti Teknologi Malaysia, Johor Bahru.

II. Related Work

A. Document Representation

In thread retrieval tasks, thread content could be considered as a virtual document or a collection of text units. These document representation models were proposed by [2] at first and then used in thread retrieval by [3]. The large document model considers the whole thread's text as a single document while the small document model is assigned to treating a thread as a collection of messages. A message could be an initial message written by the thread creator or a reply message, which is created in response to the initial message or other reply messages. When the small document model is used in thread retrieval, although relevance scores are assigned to messages, a ranked list of threads is expected for the result. Thus, aggregating the relevance scores of messages in order to obtain the score of associated threads is a challenging problem in thread retrieval.

B. Voting Model

To alleviate the problem of predicting and ranking people's expertise with respect to the topic of interest, called expert finding, [4] proposed the voting model. In this model, a

Open Science Index, Computer and Information Engineering Vol:10, No:1, 2016 publications.waset.org/10003397/pdf

candidate's profile is considered as a collection of documents. Then a ranked list of documents is provided. Then the relevance scores of documents written by a candidate are fused as an overall score for the candidate in order to create a ranked list of candidates. This model is used to rank aggregates in expert finding [4], blog topic distillation [5], news article ranking [6], and thread retrieval [7]. This voting model was reported to be superior to the approaches used as a baseline in [4], [5].

C. Content Quality in Ranking Threads

Content quality of the retrieved threads is a challenging research area. A few studies have been done on how to retrieve high quality threads. Although argument quality and source credibility are used in [8], [9] as content quality factors, the authors treated threads as concatenations of their messages, which is inferior to small document representation [2], [7], [10]-[12]. Reference [8] applied a Knowledge Adaption Model to estimate various quality features such as thread age, number of messages, posting rate of a thread and number of users. However, they also used large document representation in their experience. Reference [13] used the number of replies, thread length, linkage to the thread and user authority to estimate the quality of a thread. Thread author's activeness is also assessed by [14]. Reference [15] attempted to classify messages in terms of quality. They employed cosine similarity between message content and sub forum content, number of words, number of Part of the Speech (POS) tags, frequency of all capital-letter words and many other quality features in their studies. Some of these quality features were further used [16] to classify user expertise. To assess the quality of messages and classify them, [17] combined their proposed quality features such as number of copy, position of message in thread (first page, last page, first message, last message) and author membership group, with the above-mentioned features. The same thing happened in [18] but with different quality features and techniques. Reference [7] adopted a voting model for thread retrieval and reported an on-going study on employing quality features in thread retrieval in [10]. Then the authors built upon this [11] to leverage quality features, including number of characters, words, vocabulary, and sentences in order to improve thread retrieval using the voting model.

III. Adopt Quality Metrics in Thread Retrieval

This section comprehensively explains the quality metrics used in this research in order to improve the performance of thread retrieval methods. There are two tiers of the quality features in thread retrieval, i.e. thread-related and message-related features. The first tier contains a collection of features that have unique value in all of the messages throughout a thread. However, the second tier consists of a collection of features that vary from one message to another in terms of their values.

A. Thread Related Metrics

Each of the quality features in this group has a unique value throughout a thread. In fact, these features carry some holistic information about the thread that could be stored in each of the associated messages. The dimensions that could be considered as thread level features certain dimensions are as follows:

1) Completeness

The major focus of the decomposed features from this dimension is on the thoroughness of a thread. Due to unsolved questions in a variety of threads, finding a thread with appropriate solutions proposed for the initial question is significant. Thus features in this dimension used to assess the completeness of a thread are:

- *Number of replies* [9], [13], [19]: this feature is the number of reply messages in a thread.
- *Status of the thread*: this feature searches for the words "solved" and "completed" in the thread to know whether the initial problem has been addressed or not.

2) Comprehensiveness

Features falling in this dimension represent how a thread is comprehensive in terms of the attention of users involved in the thread and how the discussion was assessed between users.

- *Number of users* [9], [17]: this feature indicates how many users are involved in resolving the problem. It shows the attraction and importance of the topic for forum members.
- *Ratio of number of users to number of messages* [9]: this feature is computed by the number of users divided by the number of messages in a thread. It shows if a discussion happened between the users of a thread or whether each user just sent his/her solution.

B. Message Related Metrics

In this section, the quality features that are used to assess the quality and practicality of a message, which is the smallest piece of information in a thread, are categorised in different dimensions.

1) Politeness

Politeness is a criterion that assesses the civility of an author. A message created by an author who uses a polite method to explain their opinion and propose solutions for others should have priority in retrieval tasks. Therefore, its effectiveness is assessed in this research. The features that could be part of this category are:

- *Amount of thanks*: this feature is the number of occurrence of "thanks" in the content of a message.
- *Number of swear words* [15]: this feature is the overlap between words in the content of a message and a list of predefined bad words.
- *Number of attacks on other users*: this feature is the amount of bad words used against other users in the thread.

Open Science Index, Computer and Information Engineering Vol:10, No:1, 2016 publications.waset.org/10003397/pdf

IV. Evaluation

A. Experimental Resources

Two datasets containing Travel and Ubuntu forums' data have formed the corpus of this research. The first one is a tour and travel information forum. Historical interests, tourist attractions and valuable places to visit are the most interesting topics in these forums. The second forum covers discussions on the Ubuntu operation system's general usage, problems with software, applications running in the Linux environment, and even development in Linux. A summarised statistic of the corpora is shown in Table I. The queries were selected from real users' search queries performed on the forums.

TABLE I
STATISTICS OF THE DATASET

Dataset	Ubuntu	Travel
No of threads	113277	83072
No of users	103280	39454
No of messages	676777	590021
No of queries	25	25
No of evaluated threads	4512	4478

B. Experimental Setup

In this research, we have used an inverted index data structure with a unique segmented architecture to store input documents. To perform this task for each one of the datasets, we used Lucene library tools. Before indexing, for pre-processing, we used an English Analyser. An analyser represents the rules for extracting index terms from text. It breaks text fields into index-able tokens. English Analyser uses a default stop words list from the Lucene library to remove stop words and the Porter Stemmer for stemming purposes.

In this research, our purpose is to leverage quality features in voting model-based thread retrieval. Forthis aspect, retrieving a list of initial ranked messages, regardless of their corresponding thread, according to their relevance scores is the first step. We implemented the language model proposed [20] with Bayesian smoothing using Dirichlet priors [21] as represented in (1):

$$P(Q|M) = \prod_{i=1}^{|Q|} \left(\frac{tf(q_i;M) + \mu \frac{tf(q_i;C)}{|C|}}{|M| + \mu} \right) \quad (1)$$

where Q is the query set, M is a message, C is the whole corpus, q_i is i^{th} query of Q, $tf(q_i;M)$ is frequency of q_i in M, and μ is a free parameter.

Then we multiplied the result of (1), *Rel(Q,M)*, by the sigmoid transformation [22] of utilised quality feature f as shown in (2):

$$RelScore\ (Q,M,f) = Rel(Q,M) \times sigm(f) \quad (2)$$

where computation of *sigm(f)* is shown in (3).

$$sigm(f) = w\ \frac{f^a}{k^a + f^a} \quad (3)$$

We experimented with different values for w, a, and k, as suggested by [22] and we found that, in our datasets, the best values for these variables are (w = 1.0, a = 0.6 and k = 1.0). So far, for each message, we have a relevance score obtained from (1) and one score for each utilised quality feature estimated using (2). For instance, if there are five quality features in use, we will have six scores assigned to each message, one as query relevance score obtained from used retrieval system and five for quality features. Turning message-level scores into thread-level scores is necessary in this part of the study because the relevancy judgments are at thread-level in our datasets. To alleviate this problem, pertaining to the relevance score of a thread, we used the BordaFuse [23] method from the voting model to fuse the message-level features of all messages of the thread in order to estimate the relevance score of that particular thread. We used the BordaFuse technique due to its better performance and results compared with other voting models. We considered 300 as a possible value of the initial ranked list's size. In fact, we utilised a range of cutoffs, including 300, 500, 1000, 1500, 2000. However, 300 produced the best results. To aggregate quality features scores we used BordaFuse and four more different aggregation strategies proposed by [24] to fuse the message-level scores of each thread and obtain a thread-level score. When a combination of all voting techniques is required, the quality features would be aggregated using all of the methods, then a score for each one would be sent to the Coordinate Ascent. A list of utilised voting techniques in this study is given in Table II (T = thread, n = number of message(s) in the thread, Q = Query, R_Q= size of the ranked list, $Rank(Q, M_i)$= Rank of i^{th} message, and $Score(Q, M_i)$ = Score of ith message)

TABLE II
LIST OF EMPLOYED VOTING TECHNIQUES

Method	Formula	Source
BordaFuse	BordaFuse Score (T,Q) = $\sum_{i=1}^{n}(\lvert R_Q \rvert - Rank(Q, M_i))$	[23]
CombSUM	CombSUM Score (T,Q) = $\sum_{i=1}^{n} Score(Q, M_i)$	[24]
CombMAX	CombMAX Score (T,Q) = $max_{i \in n} Score(Q, M_i)$	[24]
CombMED	CombMED Score (T,Q) = $med_{i \in n} Score(Q, M_i)$	[24]
CombMIN	CombMIN Score (T,Q) = $min_{i \in n} Score(Q, M_i)$	[24]
All	Combination of all above voting techniques	[24]

With the aggregated thread-level scores in hand, we trained a list-wise learning to rank algorithm, Coordinate Ascent [25], in order to learn the ranking functions of documents using combined features. We used five-fold cross validation to optimise Normalised Discounted Cumulative Gain (NDCG). Text REtrieval Conference (TREC) was the evaluation tool used to evaluate the effectiveness of the method.

TABLE III
THE IMPACT OF THE COMPLETENESS METRICS

		TRAVEL				UBUNTU			
Feature Name	Voting Model	NDCG @100	NDCG @10	MAP @10	P@10	NDCG @100	NDCG @10	MAP @10	P@10
NoFeature		0.3113	0.293	0.042	0.408	0.3075	0.237	0.045	0.312
NoRply	bc	0.3081	0.29	0.042	**0.412**	0.3069	0.23	0.044	0.3
	max	0.3082	0.289	0.041	0.404	0.3075	0.237	0.045	0.312
	med	0.3082	0.289	0.041	0.404	0.3075	0.237	0.045	0.312
	min	0.3082	0.289	0.041	0.404	0.3075	0.237	0.045	0.312
	sum	**0.3156**	0.291	0.042	0.404	0.3072	0.236	0.045	0.312
	All	$\mathbf{0.3207}^{\Delta}$	0.29	**0.043**	0.4	0.3052	0.231	0.044	0.308
StOfTrd	bc	0.2809	0.28	0.042	0.392	0.3058	0.233	0.045	0.304
	max	0.3113	0.293	0.042	0.408	0.3075	0.237	0.045	0.312
	med	0.3113	0.293	0.042	0.408	0.3075	0.237	0.045	0.312
	min	0.3113	0.293	0.042	0.408	0.3075	0.237	0.045	0.312
	sum	0.3113	0.293	0.042	0.408	0.3075	0.237	0.045	0.312
	All	$\mathbf{0.3214}^{\Delta}$	0.292	**0.044**	0.404	0.3059	0.233	0.044	0.304
NoRply- StOfTrd	bc	**0.3198**	**0.295**	**0.045**	**0.416**	0.3035	0.224	0.043	0.288
	max	0.3091	0.292	0.042	0.408	0.3071	0.237	0.045	0.312
	med	0.3091	0.292	0.042	0.408	**0.3076**	0.236	0.045	0.308
	min	0.3091	0.292	0.042	0.408	**0.3076**	0.236	0.045	0.308
	sum	**0.3144**	0.291	0.042	0.404	0.3062	0.233	0.044	0.304
	All	**0.3199**	0.289	**0.043**	0.396	0.3048	0.218	0.042	0.28

C. Quality-Based Ranking Performance

This section discusses the results of testing the effects of quality features and their combinations across each dimension in order to significantly improve thread search effectiveness. All significance tests in this research are conducted using t-test at $p < 0:05$. Detailed results of running the retrieval method equipped with aforementioned quality features on each of the datasets are discussed below:

1) Impact of Completeness Quality Metrics on Retrieval Method

Effectiveness of the retrieval method on Travel and Ubuntu datasets with quality metrics in turn is assessed by running the method recurrently with each one of the quality features and their different combinations and evaluating the results (Table III).

In the results obtained from running the method on Ubuntu corpora, although it is obvious that there are some improvements in running the method with quality features of completeness dimension in turn, the improvements are not significant. It might be due to technical aspects of the discussions in the Ubuntu forum. . Oppositely, significant improvements could be observed in the Travel dataset. In this dataset, including *Number of reply posts* and *Status of the thread* in quality features revealed better results. However, combining these two did not perform as well as using them individually. Moreover, pertaining to aggregation strategies, the combination of all voting techniques was superior compared with other techniques. The second best voting technique was BordaFuse in terms of its completeness dimension. A detailed statistic of the performance of voting techniques is illustrated in Table IV.

TABLE IV
PERFORMANCE OF VOTING METHODS IN AGGREGATING MESSAGE-LEVEL SCORES OF COMPLETENESS QUALITY FEATURES

Voting Model	Better than base line	Significant Improvement
Bc	5	0
Max	1	0
Med	1	0
Min	1	0
Sum	2	0
All	6	2

All in all, it is noticeable that the most effective feature in this dimension is *Number of reply posts* in opposition to our assumption that a "closed" and "solved" thread (*Status of thread*) would reveal more accurate results with higher quality.

2) Impact of Comprehensiveness Quality Metrics on Retrieval Method

Comprehensiveness features that are from the thread-related features category are also assessed individually and in combination form. Table V presents the results of running the method with comprehensiveness quality features in turn. Using *Number of thread users* solely as a quality feature did not improve the results of the retrieval system significantly. On the other hand, combining it with *Ratio of users to posts* reveals better results with significant improvements in NDCG@10 and MAP@10. However, these improvements are for including *Ratio of users to posts* because using this individually reveals very good results.

Open Science Index, Computer and Information Engineering Vol:10, No:1, 2016 publications.waset.org/10003397/pdf

TABLE V
PERFORMANCE OF COMPREHENSIVENESS QUALITY FEATURES ON UBUNTU AND TRAVEL

		TRAVEL				UBUNTU			
Feature Name	Voting Model	NDCG @100	NDCG @10	MAP @10	P@10	NDCG @100	NDCG @10	MAP @10	P@10
NoFeature		0.3113	0.2932	0.042	0.408	0.3075	0.2372	0.045	0.312
NoTrdUsr	bc	0.3098	0.2895	0.0417	0.4040	0.3066	0.2336	0.045	0.308
	max	0.3098	0.2917	**0.0421**	0.4080	**0.308**	**0.2384**	0.045	0.312
	med	0.3098	0.2917	**0.0421**	0.4080	**0.3083**	0.2367	0.045	0.308
	min	0.3098	0.2917	**0.0421**	0.4080	**0.308**	**0.2384**	0.045	0.312
	sum	**0.3163**	0.2878	0.0420	0.3960	0.3063	0.2317	0.044	0.304
	All	**0.3161**	0.2906	**0.0431**	0.4040	0.3042	0.2298	0.042	0.3
RtoUsrToPst	bc	**0.3171**	**0.2964**	**0.0446**	**0.4200**	**0.3093**	$\mathbf{0.254}^{\Delta}$	$\mathbf{0.05}^{\Delta}$	0.308
	max	0.3084	**0.3036**	**0.0443**	**0.4160**	**0.3078**	**0.2449**	**0.047**	**0.316**
	med	0.3106	**0.3015**	**0.0441**	**0.4160**	0.3064	**0.2421**	**0.047**	0.312
	min	0.3027	**0.2991**	**0.0435**	0.4080	0.3029	**0.2404**	**0.046**	0.308
	sum	**0.3217**	**0.2955**	**0.0450**	**0.4160**	0.3009	0.2254	0.04	0.296
	All	$\mathbf{0.3235}^{\Delta}$	0.2929	**0.0455**	**0.4120**	**0.3146**	**0.2675**	$\mathbf{0.054}^{\Delta}$	**0.32**
NoTrdUsr-RtoUsrToPst	bc	0.3057	0.2763	0.0420	0.3880	**0.3117**	$\mathbf{0.2617}^{\Delta}$	$\mathbf{0.0558}^{\Delta}$	**0.3440**
	max	0.3042	**0.3011**	**0.0436**	0.4080	0.3007	**0.2401**	**0.047**	0.296
	med	0.3042	**0.3024**	**0.0451**	**0.4120**	0.306	**0.2398**	**0.046**	0.308
	min	0.3045	**0.3011**	**0.0436**	0.4080	0.3030	**0.2378**	**0.0459**	0.3000
	sum	**0.3230**	**0.3064**	**0.0463**	**0.4360**	0.3021	0.2271	0.043	0.296
	All	0.3095	0.2835	**0.0432**	0.4000	**0.3123**	$\mathbf{0.2671}^{\Delta}$	$\mathbf{0.057}^{\Delta}$	**0.328**

TABLE VI
PERFORMANCE OF VOTING METHODS IN AGGREGATING MESSAGE-LEVEL SCORES OF COMPREHENSIVENESS QUALITY FEATURES

Voting Model	Improvements	Significant Improvement
Bc	11	4
Max	14	0
Med	12	0
Min	11	0
Sum	9	0
All	14	4

Ratio of users to posts was a superior quality feature compared to others with significant improvements in NDCG@10 and MAP@10. Among voting methods, the combination of all techniques and then BordaFuse performed better than the others (Table VI). Thus, using *Ratio of users to posts* as a quality feature and aggregating message level scores using a combination of all voting techniques caused the most improvement in the results of our experiment.

3) Impact of Politeness Quality Metrics on Retrieval Method

A detailed analysis of the results obtained from using quality features of the politeness dimension is represented in Table VII. Politeness, unlike completeness and comprehensiveness, is a collection of message-level quality features that vary among different messages. Results show that similar to completeness, quality features in the politeness dimension did not cause any significant improvement in the Ubuntu corpora. Among quality features, running the experiment using a combination of *Number of thanks words* and *Number of attacks on other users* in turn was superior to others. Pertaining to voting models again BordaFuse performance and then a combination of all techniques were ranked above other techniques (Table VIII).

V. CONCLUSION

In this research, we assessed the significance of quality metrics in voting model thread retrieval. Two different corpuses, Ubuntu and Travel, were used to investigate the relevance score of retrieval methods with and without quality features. The quality metrics were obtained from previous researches but had never been used to promote a voting model thread retrieval technique. The results show that some of the quality features of completeness, comprehensiveness, and politeness improved the retrieval method and could be used in further research to investigate their role in different retrieval methods.

Our experimentation has demonstrated that quality optimizations are a reliable way to improve thread retrieval performance. With this idea in mind, we believe that thread retrieval throughput can be improved even more by exploring and assessing new dimensions such as: readability and popularity of a thread, ease of understanding of a thread and expertise of authors in a thread. In this sense, novel retrieval systems can take advantage of this information to successfully select quality metrics having high influence on accuracy of the system for concurrent execution. Therefore, attempting to propose effective quality metrics seems to be a reliable approach to improve retrieval performance, and should be exploited in future works. These proposals constitute the main guidelines for future developments and research activities.

Open Science Index, Computer and Information Engineering Vol:10, No:1, 2016 publications.waset.org/10003397/pdf

Open Science Index, Computer and Information Engineering Vol:10, No:1, 2016 publications.waset.org/10003397/pdf

TABLE VII
PERFORMANCE OF POLITENESS QUALITY FEATURES ON UBUNTU AND TRAVEL

		TRAVEL				UBUNTU			
Feature Name	Voting Model	NDCG @100	NDCG @10	MAP @10	P @10	NDCG @100	NDCG@10	MAP@10	P@10
NoFeature		0.3113	0.2930	0.0420	0.4080	0.3075	0.2370	0.0450	0.3120
NoAtk	bc	**0.3156**	**0.2939**	**0.0422**	0.4040	0.3067	0.2345	0.0446	0.3080
	max	**0.3116**	**0.2931**	**0.0424**	0.4080	0.3073	0.2370	0.0448	0.3120
	med	**0.3117**	**0.2932**	**0.0424**	0.4080	0.3075	**0.2372**	0.0449	0.3120
	min	**0.3117**	**0.2932**	**0.0424**	0.4080	0.3075	**0.2372**	0.0449	0.3120
	sum	**0.3116**	**0.2931**	**0.0424**	0.4080	0.3073	0.2370	0.0448	0.3120
	All	**0.3219**	**0.2957**	**0.0439**	**0.4120**	0.3052	0.2314	0.0439	0.3040
NoSwr	bc	**0.3216**$^{\Delta}$	**0.2941**	**0.0439**	0.4080	0.3059	0.2334	0.0446	0.3040
	max	0.3088	0.2844	0.0410	0.3960	0.3075	**0.2372**	0.0449	0.3120
	med	0.3088	0.2844	0.0410	0.3960	0.3075	**0.2372**	0.0449	0.3120
	min	0.3089	0.2844	0.0410	0.3960	0.3075	**0.2372**	0.0449	0.3120
	sum	0.3088	0.2844	0.0410	0.3960	**0.3076**	**0.2373**	0.0449	0.3120
	All	**0.3175**	0.2854	**0.0424**	0.3960	0.3048	0.2334	0.0449	0.3120
NoSwr-NoAtk	bc	**0.3209**$^{\Delta}$	**0.2973**	**0.0452**	**0.4160**	0.3059	0.2333	0.0445	0.3040
	max	0.3089	0.2843	0.0410	0.3960	0.3073	0.2370	0.0448	0.3120
	med	0.3061	0.2844	0.0410	0.3960	0.3075	**0.2372**	0.0449	0.3120
	min	0.3090	0.2844	0.0410	0.3960	0.3069	**0.2372**	0.0449	0.3120
	sum	0.3089	0.2843	0.0410	0.3960	0.3073	**0.2371**	0.0448	0.3120
	All	**0.3148**	0.2842	0.0419	0.3960	0.3075	**0.2463**	**0.0485**	**0.3240**
NoTnk	bc	**0.3148**	**0.2953**	**0.0430**	**0.4160**	0.3072	0.2324	0.0431	0.3040
	max	**0.3114**	**0.2934**	**0.0425**	0.4080	0.3072	0.2362	0.0449	0.3080
	med	0.3111	0.2926	**0.0424**	0.4080	0.3069	0.2359	0.0448	0.3080
	min	0.3113	**0.2932**	**0.0424**	0.4080	0.3069	0.2359	0.0448	0.3080
	sum	0.3113	**0.2932**	**0.0424**	0.4080	0.3069	0.2368	0.0446	0.3120
	All	**0.3162**	0.2924	**0.0427**	0.4080	0.3060	0.2286	0.0426	0.2960
NoTnk-NoAtk	bc	**0.3215**$^{\Delta}$	**0.2941**	**0.0437**	**0.4120**	0.3059	0.2314	0.0430	0.3040
	max	**0.3116**	0.2926	**0.0424**	0.4080	**0.3078**	0.2359	0.0447	0.3080
	med	0.3110	0.2926	**0.0424**	0.4080	**0.3076**	**0.2372**	0.0449	0.3120
	min	**0.3114**	0.2924	**0.0423**	0.4080	0.3070	0.2359	0.0448	0.3080
	sum	0.3111	**0.2931**	**0.0424**	0.4080	0.3069	0.2368	0.0446	0.3120
	All	**0.3240**$^{\Delta}$	**0.2942**	**0.0440**	0.4080	0.2994	0.2135	0.0407	0.2800
NoTnk-NoSwr	bc	**0.3162**	0.2911	**0.0435**	**0.4120**	0.3061	0.2289	0.0437	0.2960
	max	0.3078	0.2877	0.0415	0.4000	**0.3079**	0.2359	0.0447	0.3080
	med	0.3049	0.2845	0.0411	0.3960	**0.3083**	0.2341	0.0444	0.3040
	min	0.3085	0.2843	0.0410	0.3960	0.3064	0.2359	0.0448	0.3080
	sum	0.3079	0.2846	0.0411	0.3960	0.3036	0.2334	0.0444	0.3080
	All	**0.3183**	0.2865	**0.0427**	0.4000	0.2887	0.2001	0.0393	0.2800
NoTnk-NoSwr-NoAtk	bc	**0.3210**$^{\Delta}$	0.2929	**0.0439**	0.4080	0.3063	0.2281	0.0433	0.2960
	max	0.3081	0.2882	0.0416	0.4000	0.3067	0.2336	0.0443	0.3080
	med	0.3050	0.2846	0.0411	0.3960	0.3065	0.2347	0.0445	0.3080
	min	0.3088	0.2844	0.0410	0.3960	0.3072	0.2359	0.0448	0.3080
	sum	0.3087	0.2844	0.0410	0.3960	0.3077	0.2347	0.0447	0.3080
	All	**0.3174**	0.2841	**0.0425**	0.3960	0.3028	0.2289	0.0446	0.3000

TABLE VIII
PERFORMANCE OF VOTING METHODS IN AGGREGATING MESSAGE-LEVEL SCORES OF POLITENESS QUALITY FEATURES

Voting Model	Improvements	Significant Improvement
Bc	23	4
Max	10	0
Med	10	0
Min	10	0
Sum	10	0
All	19	1

ACKNOWLEDGMENT

'This work is supported by the Ministry of Education Malaysia and Soft Computing Research Group (SCRG) of Universiti Teknologi Malaysia (UTM). This Work is also supported in part by grant from Vote 4F373.

REFERENCES

[1] Heydari, Atefeh, Mohammad ali Tavakoli, Naomie Salim, and Zahra Heydari. "Detection of review spam: A survey." Expert Systems with Applications 42, no. 7 (2015): 3634-3642.

[2] Elsas, J. L., Arguello, J., Callan, J. and Carbonell, J. G. (2008). Retrieval and feedback models for blog feed search. In: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, ACM, New York, NY, USA, SIGIR '08, pp. 347-354, DOI 10.1145/1390334.1390394, URL http://doi.acm.org/10.1145/1390334.1390394

[3] Elsas, J.L. and Carbonell, J. G. (2009) It pays to be picky: an evaluation of thread retrieval in online forums. In: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, ACM, New York, NY, USA, SIGIR '09, pp. 714-715, DOI 10.1145/1571941.1572092, URL http://doi.acm.org/10.1145/1571941.1572092.

[4] Macdonald, C. and Ounis, I. (2008a). Voting techniques for expert search. Knowl Inf Syst., 16(3), pp. 259-280. DOI 10.1007/s10115-007-0105-3, URL http://dx.doi.org/10.1007/s10115-007-0105-3

[5] Macdonald, C. and Ounis, I. (2008b). Key blog distillation: ranking aggregates. In: Proceedings of the 17th ACM conference on Information and knowledge management, ACM, New York, NY, USA, CIKM '08,

pp. 1043-1052, DOI 10.1145/1458082.1458221, URL http://doi.acm.org/10.1145/1458082.1458221

[6] McCreadie,R. M. C.,Macdonald, C. and Ounis, I. (2010). News article ranking: leveraging the wisdom of bloggers. In Adaptivity, Personalization and Fusion of Heterogeneous Information, RIAO '10, pp. 40-48, Paris, France. Le Centre de Hautes Etudes Internationals D'Informatique Documentaire.http://dl.acm.org/citation.cfm?id=1937055.1937064.

[7] Albaham, A. T. and Salim, N. (2012a). Adapting voting techniques for online forum thread retrieval. Advanced Machine Learning Technologies and Applications, volume 322 of Communications in Computer and Information Science, pages 439-448. Springer Berlin Heidelberg. ISBN 978-3-642-35325-3.

[8] Wang, G. A., Jiao, J. and Fan, W.(2009). Searching for Authoritative Documents in Knowledge-Base Communities. ICIS 2009 Proceedings. Paper 109.http://aisel.aisnet.org/icis2009/109

[9] Fan, W. (2009). Effective search in online knowledge communities: A genetic algorithm approach (Doctoral dissertation, Virginia Polytechnic Institute and State University).

[10] Albaham, A. T. and Salim, N. (2012b). Quality-biased retrieval in online forums. Journal of Theoretical and Applied Information Technology, 38(1), pp. 55-62.

[11] Albaham, A. T. and Salim, N. (2013, December). Quality biased thread retrieval using the voting model. In Proceedings of the 18th Australasian Document Computing Symposium (pp. 97-100). ACM.

[12] Albaham, A. T., Salim, N. and Adekunle, O. I. (2014, January). Leveraging Post Level Quality Indicators in Online Forum Thread Retrieval. In Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013) (pp. 417-425). Springer Singapore.

[13] Bhatia, S. and Mitra, P. (2010). Adopting inference networks for online thread retrieval. In Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, pp. 1300-1305, Atlanta, Georgia, USA.

[14] Zuriati Ismail, Atefeh Heydari, Mohammadali Tavakoli, Naomie Salim. "Incorporating Author's Activeness in Online Discussion in Thread Retrieval Model" ARPN Journal of Engineering and Applied Sciences 10 (2), 473-479

[15] Weimer, M. and Gurevych, I. (2007). Predicting the perceived quality of web forum posts. In Proceedings of the Conference on Recent Advances in Natural Language Processing (RANLP), pp. 643-648.

[16] Lui, M. and Baldwin, T. (2010). Classifying user forum participants: Separating the gurus from the hacks, and other tales of the internet. In Proceedings of the Australasian Language Technology Association Workshop 2010, pp. 49-57, Melbourne, Australia, December 2010.

[17] Eng, K. and Chai, K. (2011). A Machine Learning-based Approach for Automated Quality Assessment of User Generated Content in Web Forums. PhD thesis, Digital Ecosystems and Business Intelligence Institute, Curtin University.

[18] Burel, G., He, Y. and Alani, H. (2012). Automatic identification of best answers in online enquiry communities. In 9th Extended Semantic Web Conference, May 2012.

[19] Fan, W., Wang, G. and Liu, X. (2011). A knowledge adaption model based framework for finding helpful user generated content in online communities: In Thirty Second International Conference on Information Systems. AIS Electronic Library (AISeL).

[20] Ponte, J. M. and Croft, W.B. (1998). A language modeling approach to information retrieval. In: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, ACM, NewYork, NY, USA, SIGIR '98, pp. 275-281, DOI 10.1145/290941.291008.

[21] Zhai, C. and Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. ACM Trans Inf Syst, 22(2), pp. 179-214, DOI 10.1145/984321.984322.

[22] Craswell, N., Robertson, S., Zaragoza, H., and Taylor, M. (2005, August). Relevance weighting for query independent evidence. In Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval (pp. 416-423). ACM.

[23] Aslam, J. A. and Montague, M. (2001). Models for metasearch. In: Oft, W. B., Harper, D., Kraft, D. et al. (eds.) Proceedings of ACM SIGIR 2001. ACM Press, New Orleans, pp. 276–284. doi: 10.1145/383952.384007

[24] Fox, E.A. and Shaw, J. A. (1994). Combination of multiple searches. In: Proceedings of TREC-2. NIST, Gaithersburg.

[25] Metzler, D. and Croft, W. B. (2007). Linear feature-based models for information retrieval. Inf. Retr., 10(3), pp. 257-274, June. ISSN 1386-4564.

Open Science Index, Computer and Information Engineering Vol:10, No:1, 2016 publications.waset.org/10003397/pdf