2012 IEEE International Conference on Systems, Man, and Cybernetics
October 14-17, 2012, COEX, Seoul, Korea

Automatic Construction of An Evaluation Dataset from Wisdom of the Crowds for Information Retrieval Applications

Chieh-Jen Wang Hung-Sheng Huang Hsin-Hsi Chen
Department of Computer Science and Information Engineering
National Taiwan University
Taipei 106, Taiwan
{cjwang, hshuang}@nlg.csie.ntu.edu.tw, hhchen@ntu.edu.tw

***Abstract*—A benchmark evaluation dataset which reflects users' search behaviors in the real world is indispensable for evaluating the performance of information retrieval applications. A typical evaluation dataset consists of a document set, a topic set and relevance judgments. Manual preparation of an evaluation dataset needs much human cost, and human-made topics may not fully capture users' real search needs. This paper aims at automatically constructing an evaluation dataset from wisdom of the crowds in search query logs for information retrieval applications. We begin with collecting documents of clicked documents in search query logs, selecting suitable queries in terms of topics, sampling documents from the document collection for each query and estimating the multi-level relevance of document samples based on click count, normalized count and average count functions. The machine-made evaluation dataset is trained and tested by three learning to rank algorithms, including linear regression, SVM^{Rank} and FRank. We compare their performance on a testing collection MQ2007 of LETOR which is a well-known human-made benchmark dataset for learning to rank. The experimental results show that the performance tendency is similar by using machine-made and human-made evaluation datasets. That demonstrates our proposed models can construct an evaluation dataset with similar quality of human-made.**

Keywords- retrieval evaluation*; *search query logs analysis*; *evaluation dataset construction

I. INTRODUCTION

Predicting the ranking sequence of retrieved results list is fundamental in information retrieval. Several kinds of evaluation datasets are available in TREC, CLEF and NTCIR. A typical evaluation dataset includes a document collection, a set of queries, and relevance judgments. The relevance degree of a document may be binary (e.g., relevant vs. irrelevant) or n-ary (e.g., highly relevant, relevant, partially relevant, and irrelevant). Labeling relevance manually for documents needs much human cost, and topic preparation may be ad hoc and cannot fully represent users' real search needs.

Search query logs keep users' search behaviors in the real world. The search logs contain queries, the clicked URLs for each query, the positions of clicked URLs in retrieved results list, and the timestamps of query submissions and URL clicks. Search query logs bring opportunities to construct an evaluation dataset to reflect user search behavior in the real world. Users issuing queries play the similar roles of annotators. A click on a URL for a query by a user can be considered as an annotation of relevance of this query on the URL. In this way, the relevance is labeled by users collaboratively when they interact with search engines for retrieving information. Thus, search query logs can be considered as a type of wisdom of crowds if knowledge is mined. The wisdom of crowds embedded in search query logs is used to create an evaluation dataset for information retrieval applications. However, a clicked URL may not be always relevant to a query due to the performance of retrieval systems and/or the user comprehension. Even knowing a clicked URL is relevant, it is still challenging to predict its relevance degree in the case of multiple level relevance tagging. Besides this issue, the document of a clicked URL in search query logs may not exist on the web because it has been removed or redirected.

Modeling the users' click behaviors has been studied intensively in recent years. Joachims *et al*. [1] adopted an eye-tracking technique to observe the users' search behaviors when browsing web pages. They found that the web pages of higher ranks have higher click probability. The works on click models, e.g., the cascade model [2], the multiple-click model [3], the click chain model [4], and the dynamic bayesian network click model [5], predicted click probability of a web page under different postulations.

In this paper, we will propose three models – say, click count, normalized click count and average click count, to estimate the relevance degree of web pages, and then construct evaluation datasets with multiple-level relevance tagging by different models, and explore them on three learning to rank algorithms including linear regression (LR), SVM^{rank} [10] and FRank [11] to demonstrate their quality. Different metrics measure the performance of these algorithms and show the effectiveness of automatically constructed evaluation datasets.

II. CONSTRUCTION OF AN EVALUATION DATASET

An evaluation dataset T is defined as a quadruple $<D, Q, S, f>$ where D is a document collection, Q is a set of queries, S is a set of $<q, D'>$ tuples where $q \in Q$ and $D' \subset D$, and f is a relevance function of $q \in Q$ and $d \in D$ to measures relevance degree of d to q. In TREC evaluation [6], task organizers formulate a set of candidate queries, search relevant documents

978-1-4673-1714-6/12/$31.00 ©2012 IEEE

in D for each candidate query, and select queries with reasonable number of retrieved documents to form Q. Then they pool together the retrieval results contributed by task participants who employ Q and D on their proposed information retrieval systems. The pooling action *samples* a set of $< q, D' >$ tuples to form S, where D' is the union of all retrieved documents by a query q. Human annotators label each document in the samples a relevance degree, and regard those not in the samples as irrelevant.

In this paper, the MSN Search Query Log excerpt (RFP 2006 dataset) [7] is employed to set up such an evaluation dataset. Q and D (D') are selected from queries and clicked URL in the search query logs. Since documents of clicked URLs in the search query logs are recorded in May 2006, not all clicked documents in the search query logs still exist nowadays. Besides, even those documents exist may not be always relevant to queries. We will propose three relevance functions to measure the relevance degree in the sampling data.

A. Environment Reconstruction

The MSN Search Query Log excerpt contains 15 million queries sampled in May 2006, and documents for preparing an evaluation dataset were collected between October and December 2009. We submitted each clicked URL in the search query logs to gather the corresponding documents and received the following three types of *Http Standard Response Codes*.

- 2xx (successful): The documents of the clicked URLs can be downloaded successfully.
- 3xx (redirected): The original URL was redirected.
- Other codes: They include no response, time out, or unsuccessful.

Table I shows the statistics of *Http Standard Response Codes* of the clicked URLs in the search query logs. A total of 4,971,003 clicked URLs are tested. Total 48%, 21%, and 31% of the 4,971,003 clicked URLs belong to the successfully type, the redirected type, and other types, respectively. Only the documents of the clicked URLs of the former two types are considered in the latter experiments.

TABLE I. THE STATISTICS OF CLICKED URLS

Type	Number	Percentages
2xx (successful)	2,371,815	48%
3xx (redirected)	1,040,298	21%
Others	1,558,890	31%
Total	4,971,003	100%

B. Selection of Query and Document Collection

Selecting appropriate queries to form Q is important for construction of an evaluation dataset. Many queries were submitted only a few times in the search query logs. The number of unique clicked URLs for a query is related to the occurrences of this query. Let U_q be the number of unique URLs for a query q, r be the frequency of a query, N_r be the number of queries that occur r times, and Q_r be a set of queries of frequency r. The average number of unique clicked URLs U^r for queries of frequency r is defined as Formula (1).

$$U^r = \frac{1}{N_r} \sum_{q \in Q_r} U_q \qquad (1)$$

Fig. 1 illustrates the relationship between r and U^r. We can observe that U^r increases when r increases, but the increase rate goes smoothly. Fig. 1 also shows that the number of unique clicked URLs of low frequent queries may be not large enough to represent relevance ranking. Thus, random sampling which may include those queries of low frequency in the query set is infeasible.

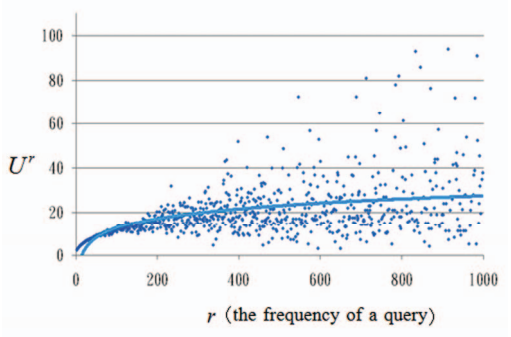

Figure 1. Relationship between query frequency r and average number of unique clicked URLs U^r

Although high frequent queries have larger number of unique clicked URLs, they are fewer than low frequent ones. How to trade off the number of high frequent queries and the number of unique clicked URLs is an important issue. In this paper, we propose a selection criterion: both the frequency of a query and its clicked count of a URL should be larger than a threshold. Table II lists the statistics under different threshold settings. The second column denotes the number of queries exceeding the designated threshold specified in the first column, and the third column denotes total number of documents of clicked URLs for the queries exceeding the threshold.

TABLE II. THE STATISTICS OF QUERIES AND DOCUMENTS UNDER VARIOUS THRESHOLDS

Threshold	Number of queries	Number of documents
3	358,552	1,158,509
5	166,882	699,870
10	70,446	396,636
50	11,442	115,253
100	5,182	66,123

In the latter experiments, we will explore thresholds c to select various document collections D. Besides, we will study thresholds n to sample queries and the corresponding clicked documents for relevance judgments. That will form different sample sets S.

C. Relevance Functions

Relevance is divided into v levels, 0, 1, …, v-1, in multi-level relevance judgments. The higher the level of a clicked URL is, the larger the relevance degree it has. The clicked URLs on the same level have the same relevance degree. The following proposes three relevance functions f to map a click count to a relevance degree.

1) Click Count (CC) function

A click count of a URL u for a query q, $c(u, q)$, denotes how many times users click u in q. Formula (2) defines

relevance degree *CC* of *u* in *q*. Formula (3) projects *CC* into relevance degree *PCC* of a URL *u* in a query *q*, where *floor* function returns the largest integer value less than or equal to $CC(u, q)$ divided by *dif* > 0, and *dif* is a parameter to reflect the difference degree of clicked URLs. For a larger *dif*, the difference of $c(u,q)$ and $c(u',q)$ should be much larger in order to distinguish the relevance degree of *u* from degree of *u*'. In the extreme case, *dif*=1, the absolute click count itself reflects the relevance degree.

$$CC(u,q) = c(u,q) \tag{2}$$

$$PCC(u,q) = floor(\frac{CC(u,q)}{dif}) \tag{3}$$

2) Normalized Count (NC) function

Assume there are *m* unique clicked URLs, $u_1, u_2, \ldots, u_m$, for query *q*. Formula 4 defines a normalized count *NC* for each URL u_j. Here the absolute click count is mapped into a value between 0 and 1. Formula (5) projects *NC* into *PNC* of a URL *u* in a query *q*, where *v* is the number of relevance levels.

$$NC(u_j,q) = \frac{c(u_j,q)}{\sum_{i=1}^{m} c(u_i,q)} \tag{4}$$

$$PNC(u,q) = floor(NC(u,q) \times v) \tag{5}$$

3) Average Click (AC) function

Formula (6) defines the average count, $AC(u, q)$, of a URL *u* for a query *q*, where M_q is total occurrences of *q*. Formula (7) projects *AC* into a relevance degree *PAC* of a URL *u* in a query *q*, where *v* is the number of relevance levels and *ceiling* function returns the smallest integer greater than or equal to $AC(u, q)$ multiplying *v*.

$$AC(u,q) = \frac{c(u,q)}{M_q} \tag{6}$$

$$PAC(u,q) = ceiling(AC(u,q) \times v) - 1 \tag{7}$$

III. EXPERIMENT DESIGN

The three functions create different evaluation datasets. To evaluate directly which function is better is challenging. Examining each evaluation dataset by human not only cost much, but also has agreement problems in relevance judgments. We adopt an indirect strategy: comparing the performance of each evaluation dataset on different learning to rank algorithms. Moreover, we also compare the results with a human-made dataset, LETOR, and discuss the feasibility of the machine-made datasets. In this section, we introduce LETOR firstly, then three learning to rank algorithms, and end with the evaluation procedure.

A. LETOR

LETOR (LEarning TO Rank) [8] developed by Microsoft Research Asia is a well-known dataset widely used to evaluate learning to rank algorithms. LETOR 4.0 released in July of 2009 consists of two evaluation datasets MQ2007 and MQ2008. Table III lists their statistics. As MQ2008 is relatively smaller, we adopt MQ2007 in this study.

There are four fields shown as follows for each entry in LETOR.

TABLE III. STATISTICS OF LETOR 4.0

Data set	Number of queries	Number of documents
MQ2007	1,692	69,622
MQ2008	784	15,211

- Relevance degree

Relevance degree measures the relevance between a query and a document. There are 3 relevance levels, i.e., 0-irrelevant, 1-partially relevant, and 2-relevant. The degree is annotated by human. The relevance distribution of documents is: 74% irrelevant, 20% partially relevant and 6% relevant.

- Query id

MQ2007 is selected from TREC 2007 Million Query track. From query id, we can find queries and the detailed information needs, and descriptions.

- Document id

Documents are sampled from GOV2 (collected in .gov domain in 2004). From each document id, we can find the URL of documents and their contents in GOV2.

- Features

A total of 46 features are extracted from the LETOR. They cover text and hyperlink features. Text features are selected from Body, Anchor, Title, URL and Whole document. Statistics like TF, IDF, TF-IDF, DL, *etc*. and some higher level features like BM25, language models with various smoothing techniques, *etc*. are provided. Hyperlink features include Inlink, Outlink, PageRank, and number of child pages.

B. Learning to Rank Algorithms

The studies of learning to rank are very intensive recently and many algorithms are evaluated with LETOR. We adopt three learning to rank algorithms such as linear regression (LR) [9], SVM^{rank} [10] and FRank [11] in this study. The first is a point-wise method, and the last two are pair-wise methods.

For linear regression, we employ least squares to compute the best linear combination. For the second method, the toolkit SVM^{rank} [1] is used, where the kernel function is set to linear function and the termination criterion is set to 0.001. We will explore the trade off between training error and margin, and select the best value (0.0001~10000) at validation stage. For the third method, we will explore different margin cost (1~10) and the number of weak learners (1~100) at validation stage.

C. Experiment Procedure

In this study, we have machine-made datasets named *CC*, *NC* and *AC* based on the adopted relevance functions, and one human-made dataset (LETOR). LETOR dataset is divided into 5 folds. We train the three learning to rank algorithms with *CC*, *NC* and *AC* datasets, and then validate and test them with 4/5 and 1/5 of LETOR dataset, respectively. For comparisons, we also train the three learning to rank algorithms with 3/5 of LETOR data set, and validate and test them with the remaining 2/5 data sets. Table IV and Table V show the experimental designs by using machine-made and human-made with LETOR, respectively. Note that the testing is used the same fold of data.

[1] http://www.cs.cornell.edu/People/tj/svm_light/svm_rank.html

TABLE IV. TRAINING WITH MACHINE-MADE DATASET, AND VALIDATING AND TESTING WITH LETOR

LETOR	1	2	3	4	5
Fold 1	Validation	Validation	Validation	Validation	**Testing**
Fold 2	**Testing**	Validation	Validation	Validation	Validation
Fold 3	Validation	**Testing**	Validation	Validation	Validation
Fold 4	Validation	Validation	**Testing**	Validation	Validation
Fold 5	Validation	Validation	Validation	**Testing**	Validation

TABLE V. TRAINING, VALIDATING AND TESTING WITH LETOR

LETOR	1	2	3	4	5
Fold 1	Training	Training	Training	Validation	**Testing**
Fold 2	**Testing**	Training	Training	Training	Validation
Fold 3	Validation	**Testing**	Training	Training	Training
Fold 4	Training	Validation	**Testing**	Training	Training
Fold 5	Training	Training	Validation	**Testing**	Training

We select the same features for all the experimental setups. Since we could not capture the web structure in 2006 during environment reconstruction which is specified in Section II.A, 4 hyperlink and 8 anchor features in LETOR are removed for fair comparison. A total of 34 features remain for experiments.

We adopt the same evaluation metrics as LETOR4.0, including *Precision@m*, Mean Average Precision (*MAP*), *Mean nDCG* and *nDCG@m*, where *m*=1, 3, 10. Besides the eight metrics, we also consider *winner numbers* when comparing the performance of learning to rank algorithms on different datasets.

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

The evaluations are divided two main parts. We will study the effects of some parameters when selecting suitable queries and documents in the first part. Then, the performances of different learning to rank algorithms are compared to verify the machine-made evaluation datasets.

A document collection *D* is a union of MQ2007 and the documents selected from the MSN Search Query Log excerpt under the following criterion: the selected documents must be clicked at least *c* times in total and those queries retrieving the documents must be submitted at least *c* times. A query is selected when it appears at least *n* times in the search query logs, and at least *n* URLs are clicked under this query. Besides, two parameters, *dif* and *v*, have to be set for computing relevance degrees, as mentioned in Section II.C.

A. The First Part of Experiments

1) Feature Re-Extraction

Because of environment reconstruction specified in Section II.A, 4 hyperlink and 8 anchor features in LETOR are removed, i.e., only 34 features are used in the latter experiments. Table VI shows the performance using MQ2007 dataset before/after feature re-extraction. The performance dropped very much shows hyperlink and anchor features are important for ranking.

2) Document Collection D

Parameter *c* relates to the selection of document set *D*. It specifies the minimum numbers of times a URL being clicked and a query being issued. We use MQ2007 to explore the value and set *c* to 10, 5, and 3 for selecting documents in the search query logs. Table VII lists the performance on different document sets. Although *c*=10 is 3.33 times stricter than *c*=3, the performance of using smaller and larger data sets is similar. In the latter experiments, we set *c* to 10 to select document collection *D* from the search query logs.

TABLE VI. EFFECTS OF FEATURES ON THE PERFORMANCE OF USING MQ2007

Before	LR	SVM^{Rank}	FRank
MAP	0.4497	0.4635	0.4593
Mean nDCG	0.4845	0.4954	0.4945
After	LR	SVM^{Rank}	FRank
MAP	0.3821	0.3830	0.3827
Mean nDCG	0.4051	0.4047	0.4064

TABLE VII. EFFECTS OF PARAMETER *C* ON THE PERFORMANCE OF USING MQ2007

	MAP			*Mean nDCG*		
	c=10	*c*=5	*c*=3	*c*=10	*c*=5	*c*=3
LR	0.3821	**0.3823**	0.3822	0.4051	0.4054	0.4055
SVM^{Rank}	**0.3830**	0.3823	0.3822	0.4047	0.4051	0.4055
FRank	**0.3827**	0.3822	0.3818	0.4064	0.4061	0.4050

3) Query and Document Samples S

Parameter *n* relates to the query selection. It specifies the minimum occurrences of a query and the clicked URLs. Table VIII shows the performance of using different *n* to select samples *S*. In the experiments, we fix *c* to 10 to select a document collection *D*, and explore relevance functions defined in Formulas 2, 4, and 6. Note that the functions without projection have the same effect on pairwise ranking algorithms, i.e., SVM^{Rank}, and FRank, so that the performance is shown together. For linear regression, the best setting is 100. As mentioned in Table II, the selected number of documents is smaller if a threshold is larger. Smaller *n* will result in more samples, but the quality of the samples is not guaranteed. SVM^{Rank} and FRank perform better than linear regression (LR).

4) Relevance Functions f

Formulas (2)-(7) define 3 sets of relevance functions, (*CC*, *PCC*), (*NC*, *PNC*), and (*AC*, *PAC*). Normalized Count (*NC*) and Average Count (*AC*) are derived from Click Count (*CC*). *PCC*, *PNC*, and *PAC* are projected from *CC*, *NC*, and *AC*, respectively. Various datasets are generated under different relevance functions. Tables IX, X, and XI show the performance of using these sets by linear regression, SVM^{Rank}, and FRank, respectively. Here both *c* and *n* are set to 10. Table IX shows that *AC* performs the best, and *PAC* is better than *PNC* and *PCC*. It means the dataset generated by *AC* function is more suitable to linear regression. Tables X and XI show SVM^{Rank} and FRank employing datasets generated by relevance functions without projections, i.e., Formulas 2, 4, and 6, achieve good performance.

5) Samples S and Relevance Function f.

Fig. 2-4 show linear regression on *CC*, *NC* and *AC* datasets. Linear regression does not favor absolute click count (*CC*). It performs better on the two relative click counts (*NC* and *AC*). Stricter sampling results in better performance. Fig. 5-7 show SVM^{Rank} on *CC*, *NC* and *AC* data sets. When adopting looser sampling, projection with *dif*=2 or *dif*=3 get better performance.

Comparing Fig. 6-7, projection with more levels is good for stricter sampling. The methods without projection (i.e., *original* in the figures) are better than those with projection in almost all the cases.

B. The Second Part of Experiments

Table XII shows the performance of the three learning to rank algorithms on the original MQ2007 evaluation dataset: SVM^{Rank} > FRank > LR. Table XIII shows the performance after features are re-extracted (refer to Sections III.A and IV.A): SVM^{Rank} > FRank > LR in precision metrics, and FRank > SVM^{Rank} > LR in Mean *nDCG* metrics.

TABLE VIII. EFFECTS OF PARAMETER *N* ON THE PERFORMANCE OF USING MACHINE-MADE DATA SETS

	MAP			*Mean nDCG*		
LR	*n*=10	*n*=50	*n*=100	*n*=10	*n*=50	*n*=100
CC (Formula 2)	**0.2881**	0.2849	0.2842	0.2783	0.2736	0.2709
NC (Formula 4)	0.3034	0.3106	**0.3158**	0.3016	0.3117	0.3180
AC (Formula 6)	0.3066	0.3128	**0.3176**	0.3070	0.3158	0.3205
	MAP			*Mean nDCG*		
SVM^{Rank}	*n*=10	*n*=50	*n*=100	*n*=10	*n*=50	*n*=100
CC, *NC*, *AC*	0.3413	**0.3457**	0.3441	0.3471	0.3582	0.3500
	MAP			*Mean nDCG*		
FRank	*n*=10	*n*=50	*n*=100	*n*=10	*n*=50	*n*=100
CC, *NC*, *AC*	0.3392	0.3409	**0.3417**	0.3491	0.3467	0.3476

TABLE IX. EFFECTS OF RELEVANCE FUNCTION *F* ON THE PERFORMANCE (LINEAR REGRESSION)

MAP					*Mean nDCG*				
PCC(*dif*=2)	*PCC*(*dif*=3)	*PCC*(*dif*=4)	*PCC*(*dif*=5)	*CC*	*PCC*(*dif*=2)	*PCC*(*dif*=3)	*PCC*(*dif*=4)	*PCC*(*dif*=5)	*CC*
0.2881	0.2880	0.2879	0.2878	**0.2881**	0.2783	0.2783	0.2783	0.2783	0.2783
PNC(*v*=40)	*PNC*(*v*=20)	*PNC*(*v*=10)	*PNC*(*v*=3)	*NC*	*PNC*(*v*=40)	*PNC*(*v*=20)	*PNC*(*v*=10)	*PNC*(*v*=3)	*NC*
0.3031	0.3026	0.3021	0.2983	**0.3034**	0.3012	0.3004	0.2990	0.2946	0.3016
PAC(*v*=40)	*PAC*(*v*=20)	*PAC*(*v*=10)	*PAC*(*v*=3)	*AC*	*PAC*(*v*=40)	*PAC*(*v*=20)	*PAC*(*v*=10)	*PAC*(*v*=3)	*AC*
0.3065	0.3062	0.3049	0.2998	**0.3066**	0.3068	0.3062	0.3039	0.2962	0.3070

TABLE X. EFFECTS OF RELEVANCE FUNCTION ON *F* ON THE PERFORMANCE (SVM^{RANK})

MAP					*Mean nDCG*				
PCC(*dif*=2)	*PCC*(*dif*=3)	*PCC*(*dif*=4)	*PCC*(*dif*=5)	*CC*	*PCC*(*dif*=2)	*PCC*(*dif*=3)	*PCC*(*dif*=4)	*PCC*(*dif*=5)	*CC*
0.3415	**0.3417**	0.3345	0.3354	0.3413	0.3472	0.3476	0.3388	0.3452	0.3471
PNC(*v*=40)	*PNC*(*v*=20)	*PNC*(*v*=10)	*PNC*(*v*=3)	*NC*	*PNC*(*v*=40)	*PNC*(*v*=20)	*PNC*(*v*=10)	*PNC*(*v*=3)	*NC*
0.3395	0.3378	0.3329	0.3195	**0.3413**	0.3480	0.3481	0.3421	0.3244	0.3471
PAC(*v*=40)	*PAC*(*v*=20)	*PAC*(*v*=10)	*PAC*(*v*=3)	*AC*	*PAC*(*v*=40)	*PAC*(*v*=20)	*PAC*(*v*=10)	*PAC*(*v*=3)	*AC*
0.3409	0.3338	0.3293	0.3195	**0.3413**	0.3479	0.3467	0.3371	0.3246	0.3471

TABLE XI. EFFECTS OF RELEVANCE FUNCTION *F* ON THE PERFORMANCE (FRANK)

MAP					*Mean nDCG*				
PCC(*dif*=2)	*PCC*(*dif*=3)	*PCC*(*dif*=4)	*PCC*(*dif*=5)	*CC*	*PCC*(*dif*=2)	*PCC*(*dif*=3)	*PCC*(*dif*=4)	*PCC*(*dif*=5)	*CC*
0.3388	0.3359	0.3323	0.3314	**0.3392**	0.3457	0.3445	0.3399	0.3352	0.3491
PNC(*v*=40)	*PNC*(*v*=20)	*PNC*(*v*=10)	*PNC*(*v*=3)	*NC*	*PNC*(*v*=40)	*PNC*(*v*=20)	*PNC*(*v*=10)	*PNC*(*v*=3)	*NC*
0.3351	0.3331	0.3312	0.3302	**0.3392**	0.3417	0.3405	0.3392	0.3311	0.3491
PAC(*v*=40)	*PAC*(*v*=20)	*PAC*(*v*=10)	*PAC*(*v*=3)	*AC*	*PAC*(*v*=40)	*PAC*(*v*=20)	*PAC*(*v*=10)	*PAC*(*v*=3)	*AC*
0.3354	0.3347	0.3328	0.3301	**0.3392**	0.3459	0.3401	0.3402	0.3327	0.3491

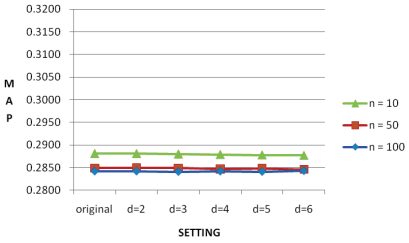

Figure 2. Linear regression with *CC* dataset

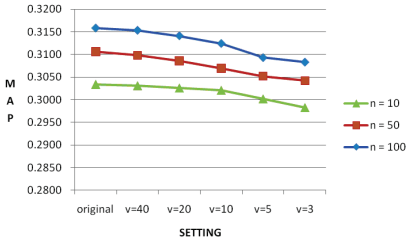

Figure 3. Linear regression with *NC* dataset

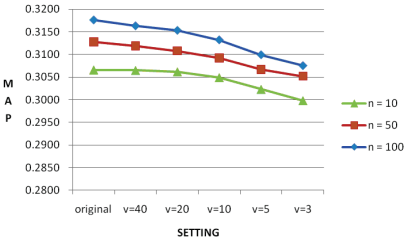

Figure 4. Linear regression with *AC* dataset

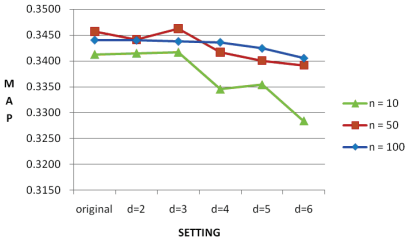

Figure 5. SVM^{Rank} with *CC* dataset

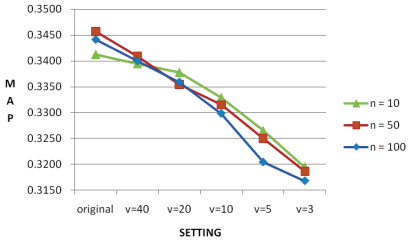

Figure 6. SVM^{Rank} with *NC* dataset

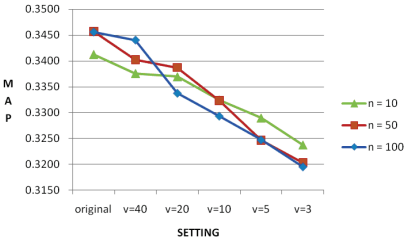

Figure 7. SVM^{Rank} with *AC* dataset

Table XIV shows the performance of the three learning to rank algorithms under the best setting of machine-made evaluation datasets: linear regression (*c*=3, *n*=100, *f*=*AC*), SVM^{Rank} (*c*=5, *n*=100, *f*=*AC*), and FRank (*c*=3, *n*=100, *f*=*AC*). SVM^{Rank} performs the best on 3 precision and 2 *nDCG* metrics, while FRank achieves the best performance in 1 precision and 2 *nDCG* metrics. Linear regression is the worst in the evaluation dataset. This is because it is less probable to have the same absolute click counts for two different documents at the same relevance degree.

TABLE XII. THREE ALGORITHMS ON THE ORIGINAL MQ2007

	Precision@1	*Precision@3*	*Precision@10*	*MAP*
LR	0.4356	0.4204	0.3723	0.4497
SVM^{Rank}	**0.4723**	**0.4303**	**0.3830**	**0.4635**
FRank	0.4515	0.4271	0.3700	0.4593
	nDCG@1	*nDCG@3*	*nDCG@10*	*Mean nDCG*
LR	0.3750	0.3935	0.4288	0.4845
SVM^{Rank}	**0.4085**	**0.4035**	**0.4414**	**0.4954**
FRank	0.4012	0.4023	0.4395	0.4945

TABLE XIII. THREE ALGORITHMS ON THE REVISED MQ2007

	Precision@1	*Precision@3*	*Precision@10*	*MAP*
LR	**0.3280**	0.3275	0.3274	0.3821
SVM^{Rank}	0.3168	**0.3285**	**0.3312**	**0.3830**
FRank	0.3210	0.3281	0.3302	0.3827
	nDCG@1	*nDCG@3*	*nDCG@10*	*Mean nDCG*
LR	**0.2820**	0.2937	0.3402	0.4051
SVM^{Rank}	0.2695	0.2890	0.3399	0.4047
FRank	0.2800	**0.3062**	**0.3416**	**0.4064**

TABLE XIV. THREE LEARNING TO RANK ALGORITHMS ON THE MACHINE-MADE DATASET

	Precision@1	*Precision@3*	*Precision@10*	*MAP*
LR	0.1472	0.2210	0.2590	0.3170
SVM^{Rank}	**0.2500**	**0.2790**	0.2943	**0.3487**
FRank	0.2490	0.2745	**0.3011**	0.3462
	nDCG@1	*nDCG@3*	*nDCG@10*	*Mean nDCG*
LR	0.1152	0.1576	0.2475	0.3191
SVM^{Rank}	0.2047	**0.2383**	**0.2913**	0.3606
FRank	**0.2133**	0.2210	0.2908	**0.3748**

TABLE XV. WINNER NUMBERS OF THE THREE LEARNING TO RANK ALGORITHMS

	Precision@1	*Precision@3*	*Precision@10*	*MAP*
LR	0	0	0	0
SVM^{Rank}	226	221	245	253
FRank	179	184	160	152
	nDCG@1	*nDCG@3*	*nDCG@10*	*Mean nDCG*
LR	0	0	0	0
SVM^{Rank}	196	194	179	184
FRank	209	211	226	221

Besides the above 8 metrics, we also consider winner numbers. Winner numbers count how many learning to rank algorithms performs lower than a target algorithm in each evaluation metric. In this study, we have three factors: *c* (=3, 5, or 10), *n* (=10, 50, or 100), and *f* (=*CC*, *PCC* with *dif* (2, 3, 4, or 5), *NC*, *PNC* with *v* (40, 20, 10, or 3), *AC*, *PAC* with *v* (40, 20, 10, or 3)). Therefore, there are 3×3×15=135 machine-made evaluation datasets. Table XV shows the winner numbers of the three learning to rank algorithms. SVM^{Rank} and FRank perform well in precision and *nDCG* metrics, respectively. That is roughly consistent with Table XIII.

V. CONCLUSION

In this paper, we utilize the wisdom of crowds mining from the MSN Search Query Log excerpt to construct evaluation datasets for information retrieval application. How to select document sets, how to sample queries and documents for relevance judgment and how relevance functions work are discussed. We adopt an indirection approach to verify the machine-made evaluation datasets. The experiments show our proposed models can create evaluation datasets of certain quality. As the domain of machine-made collections (i.e., from the Web) is much wider than that of MQ2007 (i.e., from .GOV), and hyperlink and anchor features are not available, the performance of learning to rank algorithms on the evaluation datasets is lower than those on MQ2007. In spite of the performance drop, the ranks of linear regression, SVM^{Rank}, and FRank in 8 metrics and winner numbers show the similar tendency on both machine-made and human-made evaluation datasets. That demonstrates that evaluation datasets constructed by our proposed models have certain of quality.

VI. ACKNOWLEDGEMENTS

This work was partially supported by National Science Council, Taiwan under grants NSC98-2221-E-002-175-MY3 and NSC99-2221-E-002-167-MY3. We are also grateful to Microsoft Research Asia for the support of MSN Search Query Log excerpt.

VII. REFERENCES

[1] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay, "Accurately interpreting clickthrough data as implicit feedback," in *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, 2005, pp. 154 – 161.

[2] N. Craswell, O. Zoeter, M. Taylor, and B. Ramsey, "An experimental comparison of click position-bias models," in *Proceedings of the international conference on Web search and web data mining*, USA, 2008, pp. 87 – 94.

[3] F. Guo, C. Liu, and Y. M. Wang, "Efficient multiple-click models in web search," in *Proceedings of the 2nd ACM International Conference on Web Search and Data Mining*, 2009, pp. 124 – 131.

[4] F. Guo, C. Liu, A. Kannan, T. Minka, M. Taylor, Y.-M. Wang, and C. Faloutsos, "Click chain model in web search," in *Proceedings of the 18th international conference on World wide web*, 2009, pp. 11 – 20.

[5] O. Chapelle and Y. Zhang, "A dynamic bayesian network click model for web search ranking," in *Proceedings of the 18th international conference on World wide web*, 2009, pp. 1 – 10.

[6] J. L. Vicedo and J. Gómez, "TREC: Experiment and evaluation in information retrieval," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 6, pp. 910 – 911, 2007.

[7] N. Craswell, R. Jones, G. Dupret, and E. Viegas, "Proceedings of the 2009 workshop on Web Search Click Data", 2009, pp. 95.

[8] T. Qin, T. Y. Liu, J. Xu, and H. Li, "LETOR: A benchmark collection for research on learning to rank for information retrieval," *Information Retrieval Journal*, vol. 13, no. 4, pp. 346 – 374, 2010.

[9] D. Yeh, P. C. Sun, and J. W. Lee, "A Linear Regression Model for Assessing the Ranking of Web Sites Based on Number of Visits," in *Web Engineering*, vol. 3140, 2004, pp. 763 – 763.

[10] T. Joachims, "Training linear SVMs in linear time," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, USA, 2006, pp. 217 – 226.

[11] M. F. Tsai, T. Y. Liu, T. Qin, H. H. Chen, and W. Y. Ma, "FRank: a ranking method with fidelity loss," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 2007, pp. 383 – 390.