DCU at the NTCIR-14 OpenLiveQ-2 Task

Piyush Arora and Gareth J.F.Jones

ADAPT Centre, School of Computing
Dublin City University, Dublin 9, Ireland
{Piyush.Arora,Gareth.Jones}@dcu.ie

Abstract. We describe the DCU-ADAPT team's participation in the NTCIR-14 OpenLiveQ-2 task. In this task for a given query and a set of questions with their answers, we were required to return a ranked list of questions that potentially match and satisfy the user's query effectively. Submitted runs were evaluated using both offline and online measures. Offline evaluation was done using evaluation metrics such as NDCG@10, ERR@10. Online evaluation was conducted in two phases using a pairwise preference multileaving approach. In this task we focus on exploring different LearningToRank (L2R) models, feature selection and data normalisation techniques. Overall, we submitted fourteen systems in the benchmark competition which were evaluated in the offline and first phase of the online evaluation. Five of our best systems (5/14) were selected for the final evaluation in the online evaluation phase. Our best run was ranked 6 out of the 65 submissions for the task. We performed detailed analysis of our system submissions and found that the ranking of different systems in this task varies considerably depending on the evaluation metric chosen. The offline and online metrics used in this task do not match well, indicating that use of only relevance-based measures might not reflect well the manner in which users interact with the information in an online setting.

Team Name. DCU-ADAPT

Subtasks. OpenLiveQ-2 (Japanese)

Keywords: LearningToRank models, Question-Answers Re-Ranking, Modelling users click, Interleaving, Online and Offline testing

1 Introduction

The driving force of human intellect is the ever increasing desire to discover, learn and know more about different topics and find answers to problems with mutual collaboration. Interactive websites for community based question answering (CQA) provide opportunities to search and ask questions ranging from critical topics related to health, education and finance to recreational queries for the purpose of fun and enjoyment etc [1]. Yahoo Chiebukuro (YCH) [1] is a

[1] https://chiebukuro.yahoo.co.jp/

community question-answering service which provides question retrieval system in Japanese language managed by the Yahoo Japan Corporation. The NTCIR-14 OpenLiveQ-2 task aimed to provide an open live test environment using the Yahoo Chiebukuro engine where given a query and a set of questions with their answers, participants had to return a ranked list of questions [5, 7]. The submitted systems were evaluated using both offline and online evaluation metrics (discussed later in Section 2). Final evaluation of the results was based on real user feedback. Involving real users in evaluation helps to consider the diversity of search intents and relevance criteria by utilising real queries and feedback from users who are engaged in real search tasks which makes this task more interesting [5, 7].

"Topical Relevance" has been the main focus of the Information Retrieval (IR) research community, but in general users look at different features such as freshness, concreteness, trustworthiness and conciseness of the information being retrieved while interacting with search engine results. How to model these diverse aspects of relevance, freshness, conciseness etc while modelling the information being retrieved and presented to a user is a complex challenge which forms the focus of this task.

TASK Challenges: Queries are typically short and ambiguous in nature and might not capture the user's intention effectively. For example for Japanese query: Q1009, English translation: "smoking", from such a short query it is hard to infer whether the person is interested in finding questions and information on "dangers of smoking" or "smoking health effects" or "mechanism to quit smoking". Without understanding the user's intent and focus of the query, it becomes challenging to re-rank the questions being retrieved for a given query to satisfy their information need. Thus this task focuses on modelling textual based information and click logs based information to re-rank questions to handle the challenges of queries being ambiguous and having diverse intent.

The remainder of this paper is organised as follows: Section 2 introduces the dataset, tools used and the evaluation strategy, Section 3 describes the approach adopted in our participation in this task, Section 4 gives results and analysis of our submissions to the task, and finally Section 5 concludes.

2 Dataset, Tools and Evaluation

As a part of the dataset for the OpenLiveQ-2 task, the organisers provided the query logs and for each query a corresponding set of questions with a best answer retrieved by the YCH engine. Table 1 presents information regarding the number of queries, questions in the training and the test sets. Since the data is in the Japanese language, so as to facilitate participation from diverse and non-native speaking teams in the development of effective systems, the task organisers provided a list of textual features indicating the scores of relevance models such as BestMatch (BM25) [13], Language Model (LM) [12] etc., for a query and corresponding set of questions. Table 2 presents a list of the complete

features which were provided by the task organisers comprising of textual and click-log based information. We refer interested readers to [5, 7] for more details of these features and the dataset construction for this task.

Training set	**Size**	**Test set**	**Size**
Number of Queries	1000	Number of Queries	1000
Number of Questions	986125	Number of Questions	985691
Number of click logs	288502	Number of click logs	148388

Table 1. Dataset details

From the machine learning perspective, the OpenLiveQ-2 task can be reduced to learning effective weights to model the different features described in Table 2. The objective is to learn the hypothesis function that represents the data effectively. This hypothesis function should generalise well to re-rank questions to suit the user's query intent and satisfy their information needs effectively.

As outlined in Section 1, this task had offline and online evaluation phases. In the offline evaluation phase, system performance was measured using *NDCG@10* [4], *ERR@10* [2], and *Q-measure* [14, 15]. For the online evaluation phase, a *pairwise preference multileaving (ppm)* approach was used [6, 11]. The proposed methodology in OpenLiveQ-2 focused on two phase online evaluation, in the first phase all the systems were evaluated online to identify the top-k systems, these top-k systems were then compared intensively to ensure that the top systems could be statistically distinguished. For each of the submitted rankings of questions, a multileaving approach was used to form a new set of combined rankings and shown to the users as part of the YCH engine. For a given query each of the questions in the original ranked list that was clicked when presented to a user received a credit, these credit scores were aggregated over the ranked list and are referred to as the cumulative credit (CC). This CC score was used to rank the systems in the online evaluation phase [5].

A baseline system consisting of the original rank of the questions as provided by the YCH engine was provided by the organisers. A natural language processing pipeline for the Japanese language to extract and process the textual based features and an evaluation toolkit for preparing the output of the systems was also provided by the task organisers. The resources provided for this task are openly available at github.[2] After the release of the test set, we were allowed to make at most 1 submission per day. The evaluation scores based on the Q-measure were provided for the test set and displayed on the task leaderboard by the task organisers to gauge and inspect the performance of alternative submitted systems on the test set on an ongoing basis.

[2] https://github.com/mpkato/openliveq

Title	ID	Snippet	ID	Question Body	ID	Best Answer	ID	Click Logs	ID
tf_sum	F1	tf_sum	F18	tf_sum	F35	tf_sum	F52	answer_num	F69
log_tf_sum	F2	log_tf_sum	F19	log_tf_sum	F36	log_tf_sum	F53	log_answer_num	F70
norm_tf_sum	F3	norm_tf_sum	F20	norm_tf_sum	F37	norm_tf_sum	F54	view_num	F71
log_norm_tf_sum	F4	log_norm_tf_sum	F21	log_norm_tf_sum	F38	log_norm_tf_sum	F55	log_view_num	F72
idf_sum	F5	idf_sum	F22	idf_sum	F39	idf_sum	F56	is_open	F73
log_idf_sum	F6	log_idf_sum	F23	log_idf_sum	F40	log_idf_sum	F57	is_vote	F74
icf_sum	F7	icf_sum	F24	icf_sum	F41	icf_sum	F58	is_solved	F75
log_tfidf_sum	F8	log_tfidf_sum	F25	log_tfidf_sum	F42	log_tfidf_sum	F59	rank	F76
tfidf_sum	F9	tfidf_sum	F26	tfidf_sum	F43	tfidf_sum	F60	updated_at	F77
tf_in_idf_sum	F10	tf_in_idf_sum	F27	tf_in_idf_sum	F44	tf_in_idf_sum	F61		
bm25	F11	bm25	F28	bm25	F45	bm25	F62		
log_bm25	F12	log_bm25	F29	log_bm25	F46	log_bm25	F63		
lm_dir	F13	lm_dir	F30	lm_dir	F47	lm_dir	F64		
lm_jm	F14	lm_jm	F31	lm_jm	F48	lm_jm	F65		
lm_abs	F15	lm_abs	F32	lm_abs	F49	lm_abs	F66		
dlen	F16	dlen	F33	dlen	F50	dlen	F67		
log_dlen	F17	log_dlen	F34	log_dlen	F51	log_dlen	F68		

Table 2. All extracted features provided in the dataset

3 System Development: Approaches Used

Information retrieval (IR) focuses on retrieving and ranking of documents for a given user query to satisfy a user's information need effectively. Traditionally the area of "Ranking" has focused on unsupervised models such as BM25 and TF-IDF to measure the extent of topical relevance between a user query and a document. However combining multiple query independent features (page views, page rank) and query dependent features (BM25 and TF-IDF scores) effectively is a complex task. How to combine these multiple features to rank a set of documents has been explored quite extensively under the research area of LearningToRank (L2R) [9, 10]. In L2R models, a ranking function is created using the training data, such that the model can precisely predict the ranked lists in the training data. Given a new query, the ranking function is used to create a ranked list for the documents associated with the query. The focus of L2R technologies is to successfully leverage multiple features for ranking, and to learn automatically the optimal way of combining these features. Submissions to the previous OpenLiveQ task showed positive results using L2R models [8], thus as a part of our investigation we focused on exploring L2R models in this work.

In this work, we used the Lemur RankLib toolkit [3]. This toolkit provides an implementation of a range of L2R algorithms which have been shown to be successful in earlier work. As a part of our investigation for this task we explored four main aspects which are discussed below:

– **Learning to Rank algorithms:** We explored various L2R algorithms including Coordinate Ascent and MART on the training set since they have been shown to perform quite well for this task [8]. The Coordinate Ascent algorithm iteratively optimises the weights of the hypothesis function by

performing a series of one dimensional searches. It repeatedly cycles through each parameter, holding all other parameters fixed, and optimises over the free parameter. Whereas, the MART algorithm produces a prediction model in the form of an ensemble of weak prediction models, which are decision trees. Thus instead of learning a single linearly combined function as in Coordinate Ascent, MART combines multiple decision tress (prediction models) to represent the training data. Our goal is to determine which of these algorithms should be used for further experiments to develop an effective solution to address the OpenLiveQ-2 task.

- **Feature Selection & Combination**: The main focus of this task was to effectively combine textual based features measuring the similarity of queries with a set of questions and click based information captured through user logs. We investigated feature selection extensively to determine a good set of features to re-rank the questions effectively for a given set of test queries. A complete set of features is shown in Table 2. To select features and combine them effectively, we broadly categorised the set of 77 features into 5 main categories, as shown in Table 3. As shown in Table 3, we have diverse feature sets capturing relevance of: i) user query to question title (*Title set*), ii) user query to question body (*Body set*), iii) user query to question snippets (*Snippet set*), iv) user query to the best answer (*Answer set*), and v) click logs based information (*Click set*). We explored alternative combinations of these diverse features set.
- **Parameter selection:** We studied varying L2R model parameters to learn effective hypothesis functions from the dataset.
- **Scores Normalisation:** The scale of the features (77 features) varies considerably as some features are on logarithmic scales (log-based values), thus we explored three feature normalisation techniques: average scores (feature scores were normalised by the sum of the value of the feature set in the data set), zscores (data normalisation by factoring in to account for the mean and standard deviation of a feature distribution) and scaling feature scores between [0-1] (by dividing each feature value with the maximum value of the feature set).

Type of Features	**Feature's ID Range**
Title Based Textual Features (Title set)	[F1-F17]
Snippet Based Textual Features (Snippet set)	[F18-F34]
Question Body Based Textual Features (Body set)	[F35-F51]
Body Answer Based Textual Features (Answer set)	[F52-F68]
Click Log Features (Click set)	[F69-F77]

Table 3. Feature set

Run Submissions: As described above, we used the RankLib toolkit to experiment with different algorithms and perform parametric optimisation. Models were trained on the training dataset comprising of about 1M questions (data points) and among which about 300k questions (data points) had information about user interactions. The models were optimised based on ERR@10 metric. We submitted 14 systems as a part of this investigation. Table 4 provides a basic overview and range of features that were linearly combined in each of the systems that we submitted for the task. Next, we briefly outline these 14 systems which we officially submitted for evaluation.

Systems	System-ID	Features combined
System-1	99	Baseline Model
System-2	106	All 77 Features (F1:F77)
System-3	110	All 77 Features (F1:F77), model trained using Mart algorithm
System-4	112	Features F1:F17 and F35:F77
System-5	118	Features F69:F77 only click logs based features
System-6	123	Features F9, F11, F13, F14, F43, F45, F47, F48, F60, F62, F64, F65, F71, F72, F75, F76
System-7	126	Features in System-6 + F77
System-8	128	Features F11, F13, F14, F45, F47, F48, F62, F64, F65, F71, F75, F76
System-9	130	Features in System-6 + F26, F28, F30, F31
System-10	133	Features in System-6 (Varied iteration of training data to 50)
System-11	143	Features F9, F11, F13, F43, F45, F47, F60, F62, F64, F71, F72
System-12	147	Features F9, F11, F13, F26, F28, F30, F71, F72
System-13	150	Features F9, F11, F26, F28, F60, F62, F72
System-14	152	Features in System-6 (Features normalisation performed)

Table 4. System Submissions, all the models, unless mentioned, were trained using a Coordinate Ascent algorithm, with default parameters: tolerance=0.001, iterations = 25, random restarts=5

All the L2R models were trained using a Coordinate Ascent algorithm, unless mentioned otherwise.

- **System-1**: Our first submission was a baseline system provided by the task organisers to check the consistency of the submission format. As described earlier, the baseline system consisted of the rank of questions as provided by the YCH engine.

- **System-2**: This submission combined all the 77 features, comprising of Title, Body, Answer, Snippet and Click set.

- **System-3**: This submission combined all the 77 features, the L2R model was trained using the Mart algorithm.

- **System-4**: This submission combined the best feature set comprising of Title, Body, Answer and Click set.

- **System-5**: This submission combined only the click-logs based features (Click set).

- **System-6**: During initial analysis of different L2R models and feature set (Title, Body, Answer, Snippet, Click) we found that some features are repetitive in nature, thus we selected 4 main features indicating: i) TF-IDF, ii) BM25, iii) LM with Dirichlet smoothing and iv) LM with Jelinek Mercer smoothing across Title, Body, Answer features set for combination. As this combination of features showed quite positive results most of the following approaches were based on the incremental changes to System-6.

- **System-7**: This system included the time based feature to the features used in System 6.

- **System-8**: To ensure we are capturing diverse relevance-based information, we explored removing the Tf-IDF feature from the features used in System-6. Thus we had three features i) BM25, ii) LM with Dirichlet smoothing and iii) LM with Jelinek Mercer smoothing across all feature sets for combination.

- **System-9**: We added the four features: (i) TF-IDF, ii) BM25, iii) LM with Dirichlet smoothing and iv) LM with Jelinek Mercer smoothing for the Snippet feature set to the features used in System 6.

- **System-10**: Keeping the features exactly same as in System-6, we varied the number of iterations to 50 while training the Coordinate Ascent algorithm.

- **System-11**: Similar to System-8 instead of removing the TF-IDF feature, we removed LM with Jelinek Mercer feature from the features used in System-6. Thus we had three features: i) TF-IDF, ii) BM25 and iii) LM with Dirichlet smoothing across Title, Body, and Answer feature set for combination. For each feature, we calculated the average score and used average scores based normalised data for training the L2R model.

- **System-12**: In this approach we used the Title and Snippet feature set based information. We consider three features from each category similar to system-11 consisting of: i) TF-IDF, ii) BM25 and iii) LM with Dirichlet smoothing. For each feature we calculated the zscore and used the zscore based normalised data for training the L2R model.

- **System-13**: In this approach we used just the Title, Snippet, and Answer feature set based information. We consider two features from each category consisting of: i) TF-IDF, and ii) BM25 scores. For each feature we normalised the feature values in the range of [0-1] and use normalised data for training

the L2R model.

– **System-14**: Keeping the features exactly same as in System-6 we performed feature scaling and normalisation before training the model. For each feature we normalised the feature values in the range of [0-1] and used normalised data for training the L2R model.

Systems	System-ID	NDCG@10	ERR@10	Q-Measure
Best Scores	131	**0.3327**	**0.2089**	**0.5015**
Average Scores	NA	0.2039	0.1281	0.4361
System-1	99	0.0741	0.0442	0.3820
System-2	106	0.2374	0.1710	0.4537
System-3	110	0.2388	0.1384	0.4441
System-4	112	0.1881	0.1369	0.3705
System-5	118	0.1170	0.1059	0.3395
System-6	123	0.3260	0.2018	**0.4954**
System-7	126	0.2041	0.1379	0.4385
System-8	128	0.2849	0.1908	0.4590
System-9	130	**0.3308**	**0.2031**	0.4640
System-10	133	0.2271	0.1483	0.4495
System-11	143	0.3016	0.1888	0.4449
System-12	147	0.2866	0.1785	0.4663
System-13	150	0.2945	0.1812	0.4640
System-14	152	0.2581	0.1541	0.4905

Table 5. Offline evaluation scores, best scores are in boldface.

4 Results and Analysis

Offline Evaluation: Table 5 presents the results of our submitted systems based on the offline evaluation measures. For the offline evaluation, the number of judged test questions was 43,205, i.e. 4.38% of all the test questions in OpenLiveQ-2 [7]. As the relevance data was incomplete, the organisers filtered out questions without relevance judgements from ranked lists of submitted runs. As a part of the official metrics, the organisers reported and compared the ranks and scores of the systems across all three measures NDCG (normalized discounted cumulative gain), ERR (expected reciprocal rank), and Q-measure. As shown in Table 5, we can see some quite distinct variations across the three scores (NDCG@10, ERR@10 and Q-scores) for the system submissions, indicating that these three evaluation metrics do not show consistent trends. For example *System-14* shows Q-scores similar to the best scores of System-6, however the NDCG@10 and ERR@10 scores are quite low compared to the highest scores of System-9.

Online Phase Evaluation: As discussed in Section 2, the online evaluation was conducted in two phases. In the first phase all 61 distinct system submissions were compared in an online setting using a pairwise preference multileaving approach to select the top 30 submissions which were then compared extensively. Table 6 presents results of both the online evaluation phases. In the first phase of online evaluation only two of our submissions (System: 128 and 133) scored below the average score, the remaining 11 systems performed better than the average score, and 5 of our 13 systems were selected to be compared in the final phase of online evaluation. In the final phase of online evaluation only 1 of 5 systems scored below the average score. Our best run System-118, was ranked "6" among the top 30 systems. We had 3 systems in top 10 final systems which were ranked at positions 6, 7 and 10.

Systems	System-ID	Cumulative Credit	Rank	Cumulative Credit	Rank
Best Scores	NA	**2633.20**	1	**1867.44**	1
Average Scores	NA	-4.92		-13.94	NA
System-1	99	-1420.91	61	NA	NA
System-2	106	**1843.81**	7	**1002.85**	7
System-3	110	190.39	40	NA	NA
System-4	112	**1721.43**	8	**428.37**	10
System-5	118	**2006.33**	4	**1129.58**	6
System-6	123	70.80	43	NA	NA
System-7	126	**1326.38**	14	**241.36**	12
System-8	128	-83.10	46	NA	NA
System-9	130	282.39	38	NA	NA
System-10	133	-40.83	44	NA	NA
System-11	143	171.30	41	NA	NA
System-12	147	**452.90**	29	**-418.21**	23
System-13	150	276.77	39	NA	NA
System-14	152	369.79	35	NA	NA

Table 6. Online evaluation scores for phase 1 and final phase. The best scores and our top 5 systems which were included in the final phase are in bold face.

Most of our submissions were heavily tuned to focus on the relevance-based features such as BM25 and LM scores, measuring the similarity of queries with a set of questions to be re-ranked as shown in Table 4. However, we found that our best systems in the online phase System-5 and System-2 ranked "6" and "7", focused on modelling users click logs (Click set) along with relevance based features effectively.

During the analysis of the performance of our alternatives systems we found that the ranking of systems varies considerably depending on the evaluation metric being considered for measuring systems performance. Table 7 presents the rank of our different systems depending on the evaluation metric chosen. In the previous edition of the task, the organisers found that Q-measure correlates

more with the online evaluation [5], but seeing the ranking of systems as shown in Table 7, it seems that the ranking of systems based on online and offline evaluation metrics does not go hand in hand. For example based on the Q-scores our System-6 was ranked "6", but was ranked "43" using the online phase-1 evaluation.

L2R models are trained using an evaluation metric such as ERR@10, NDCG@10 etc. The trained model is then used to predict the ranking of questions for the test set, which are then evaluated in offline and online settings. If the metric on which the model is trained (e.g. ERR@10) varies considerably more than the metric on which the model is evaluated (e.g. cumulative credit), then the model is found not to perform well due to differences in the nature of the evaluation criteria. Thus there is a need for further investigations to find online and offline evaluation metrics that correlate well to built effective models to address the task of ranking questions.

Systems	System - ID	NDCG@10	ERR@10	Q-Measure	CC: Phase 1	CC: Final Phase
System-1	99	59	59	56	61	NA
System-2	106	32	24	26	7	7
System-3	110	31	33	29	40	NA
System-4	112	36	35	64	8	10
System-5	118	45	38	65	**4**	**6**
System-6	123	5	5	**6**	43	NA
System-7	126	34	34	32	14	12
System-8	128	22	20	24	46	NA
System-9	130	**2**	**3**	21	38	NA
System-10	133	33	32	27	44	NA
System-11	143	19	21	28	41	NA
System-12	147	21	23	20	29	23
System-13	150	20	22	22	39	NA
System-14	152	25	31	17	35	NA

Table 7. Rank comparison based on different evaluation metrics, best systems using each of the evaluation metric is in boldface. CC stands for Cumulative Credit scores.

5 Conclusions

We submitted 14 alternative runs (including the baseline) for the OpenLiveQ-2 task. As a part of our investigation we found that the Coordinate Ascent algorithm seems to perform relatively better than the Mart algorithm. Our best system (System-9) based on NDCG@10 and ERR@10 was ranked "2" and "3" respectively. Based on Q-scores our best system (System-6) was ranked "6". However we found that the ranking of systems based on the online evaluation metric contrasted to the offline evaluation metrics. Based on the cumulative

credit our best system (System-5) was ranked "4" and "6" for online phase-1 and final phase evaluation. Contrary results regarding the rank of offline and online evaluation measures indicates that there is a need for more research and focus to understand the main factors behind such behaviour. We found that our best systems in the online phase focused on modelling users click logs, thus in future we would like to explore more on the effective techniques of modelling user logs and click distributions for ranking questions.

Acknowledgement

This research is supported by Science Foundation Ireland (SFI) as a part of the ADAPT Centre at Dublin City University (Grant No: 12/CE/I2267). We would like to thank Yasufumi Moriya from the ADAPT centre for his assistance in this task and the NTCIR-14 OpenLiveQ-2 task organisers for organising this interesting task.

References

1. Arora, P., Ganguly, D., Jones, G.J.: The Good, the Bad and their Kins: Identifying Questions with Negative Scores in Stackoverflow. In: 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. pp. 1232–1239. ASONAM (2015)
2. Chapelle, O., Metlzer, D., Zhang, Y., Grinspan, P.: Expected Reciprocal Rank for Graded Relevance. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management. pp. 621–630. CIKM (2009)
3. Dang, V.: The Lemur Project-Wiki-Ranklib (2013), http://sourceforge.net/p/lemur/wiki/RankLib
4. Järvelin, K., Kekäläinen, J.: Cumulated Gain-based Evaluation of IR Techniques. In: ACM Transactions on Information Systems 20(4), 422–446. TOIS (2002)
5. Kato, M.P., Liu, Y.: Overview of NTCIR-13. In: Proceedings of the 13th NTCIR Conference on Evaluation of Information Access Technologies (2017)
6. Kato, M.P., Manabe, T., Fujita, S., Nishida, A., Yamamoto, T.: Challenges of Multileaved Comparison in Practice: Lessons from NTCIR-13 OpenLiveQ Task. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management. pp. 1515–1518. CIKM (2018)
7. Kato, M.P., Nishida, A., Manabe, T., Fujita, S., Yamamoto, T.: Overview of the NTCIR-14 OpenLiveQ-2 Task. In: Proceedings of the 14th NTCIR Conference on Evaluation of Information Access Technologies (2019)
8. Manabe, T., Nishida, A., Fujita, S.: YJRS at the NTCIR-13 OpenLiveQ Task. In: Proceedings of the 13th NTCIR Conference on Evaluation of Information Access Technologies (2017)
9. Qin, T., Liu, T.Y., Xu, J., Li, H.: Letor: A Benchmark Collection for Research on Learning to Rank for Information Retrieval. In: Journal of Information Retrieval 13(4), 346–374 (2010)
10. Metzler, D., Croft, W.B.: Linear Feature-based Models for Information Retrieval. In: Journal of Information Retrieval 10(3), 257–274 (2007)

11. Oosterhuis, H., de Rijke, M.: Sensitive and Scalable Online Evaluation with Theoretical Guarantees. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. pp. 77–86. CIKM (2017)
12. Ponte, J.M., Croft, W.B.: A Language Modeling Approach to Information Retrieval. In: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 275–281. SIGIR (1998)
13. Robertson, S., Walker, S., Jones, S., Hancock-Beaulieu, M., Gatford, M.: Okapi at TREC-3. In: NIST special publication (500225), 109–123 (1995)
14. Sakai, T.: Evaluating Evaluation Metrics Based on the Bootstrap. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 525–532. SIGIR (2006)
15. Sakai, T.: On the Reliability of Information Retrieval Metrics Based on Graded Relevance. In: Journal of Information Processing & Management 43(2), 531–548 (2007)