NTCIR 17 Conference: Proceedings of the 17th NTCIR Conference on Evaluation of Information Access Technologies, December 12-15, 2023, Tokyo, Japan
DOI: https://doi.org/10.20736/0002001322

BITIR at the NTCIR-17 Session Search Task

Dongshuo Liu
Beijing Institute of Technology
China
3120230888@bit.edu.cn

Yidong Liang
Beijing Institute of Technology
China
3220231233@bit.edu.cn

Zhijing Wu
Beijing Institute of Technology
China
zhijingwu@bit.edu.cn

ABSTRACT

The BITIR team participated in the IR subtask of the NTCIR-17 Session Search(SS) Task. This paper reports our approach to solving the problem and discusses the official results. More specifically, for FOSS and POSS tasks, we submit two times by using the classical retrieval model BM25[5] and graph-based context-aware document ranking model HEXA[6]. Results show that our runs perform well on the test dataset with relevance label, but poorly on the official test dataset provided This may be due to the problem of noise and a small candidate set. For SSEE task, We use two traditional metrics: sDCG and sRBP. The result indicates that sRBP has a higher consistency with golden user satisfaction based on our settings.

KEYWORDS

Session Search, Document Ranking, Evaluation

TEAM NAME

BITIR

SUBTASKS

FOSS (Chinese) POSS (Chinese) SSEE (Chinese)

1 INTRODUCTION

The BITIR team participated in the IR subtask of the NTCIR-17 Data Search 2 Task including FOSS, POSS, and SSEE subtaks[3]. The FOSS aims to re-rank the candidate documents for the last query of a session while the goal of POSS is to re-rank documents for the last $k-n$ queries(query) according to the partially observed contextual information in previous search rounds where $k \geq 2$ and $1 \leq n \leq k-1$. The goal of SSEE is to use user feedback to construct session-level search effectiveness evaluation measures.

For FOSS and POSS tasks, we apply the traditional retrieval model BM25 and graph-based context-aware document ranking model HEXA for re-ranking documents. HEXA is a heterogeneous graph-based context-aware document ranking framework that leverages the current session and other sessions by heterogeneous graphs to capture accurate intent. When it comes to the SSEE task, we implement two traditional metrics: sDCG[2] and sRBP[4].

A detailed description of our approach is in section 2 and we discuss the experimental and official results in section 3

2 METHODS

In this section, we first define the problem and some notations. Then we describe our approaches for FOSS, POSS, and SSEE task respectively.

2.1 Problem Definition

The problem of Session Search aims at exploring better ranking approaches for context-aware search scenarios. We briefly formulate the tasks as follows. The session in the task can be denoted as $S = \{(q_1, \mathcal{D}_1), (q_2, \mathcal{D}_2), ..., (q_M, \mathcal{D}_M)\}$,each query q_i has a list of candidate documents $\mathcal{D}_i = \{d_{i,1}, ..., d_{i,n}\}$ with click binary labels ($y_{i,j} = 1$ if clicked). We also denote the last query q_M in the session as the current query and denote the corresponding $\mathcal{D}_M$ as candidate documents to be ranked.

2.2 FOSS Subtask

The goal of the FOSS task is to score and re-rank candidate documents with full session contexts. In this task, we tried two methods, BM25 and Heterogeneous Graph-based Context-aware Document Ranking(HEXA). We will introduce these two methods as follows.

2.2.1 BM25. BM25[5] is a classical retrieval algorithm that can rank documents based on the term frequency, inverse document frequency, and other adjustable parameters. The calculation formula of BM25 is shown as follows:

$$BM25(d,q) = \sum_{i=1}^{n} IDF(q_i) \frac{f(q_i,d)(k_1+1)}{f(q_i,d) + k_1(1 - b + b\frac{|d|}{avgdl})}, \quad (1)$$

$$IDF(q_i) = ln(\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} + 1), \quad (2)$$

where $n(q_i)$ is the number of documents that contain words q_i in document set, $f(q_i, d)$ is the term frequency of the q_i in document d, N is the total number of documents in document set, *avgdl* is the average length of documents in document set, k_1 and b are adjustable parameter.

In order to improve computing efficiency and reduce the noise in the document, we first segment the sentence and remove stop words in the document, then we pre-count the index and number of occurrences of all words in the body of the document.

2.2.2 HEXA. HEXA is a heterogeneous graph-based context-aware document ranking framework that leverages the current session and other sessions by heterogeneous graphs to capture accurate intent. It mainly contains three stages:(1) graph modeling, (2) ranking stage, (3) optimizing stage

- **Graph Modeling**. Formally, the definition of heterogeneous graph in HEXA is $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{T}, \mathcal{R}\}$, where $\mathcal{V}$ is the set of nodes in the graph, $\mathcal{E}$ is the set of edges in the graph, $\mathcal{T}$ denote the set of node types of the graph including query and document, $\mathcal{R}$ denote the set of edge types coming from query-query, query-document, document-document. As shown in Figure 1, based on click behavior and transition relation between queries and documents, HEXA design eight types of edges. According to this graph schema, HEXA builds graphs from two aspects, *i.e.*, session graph, and query graph. The

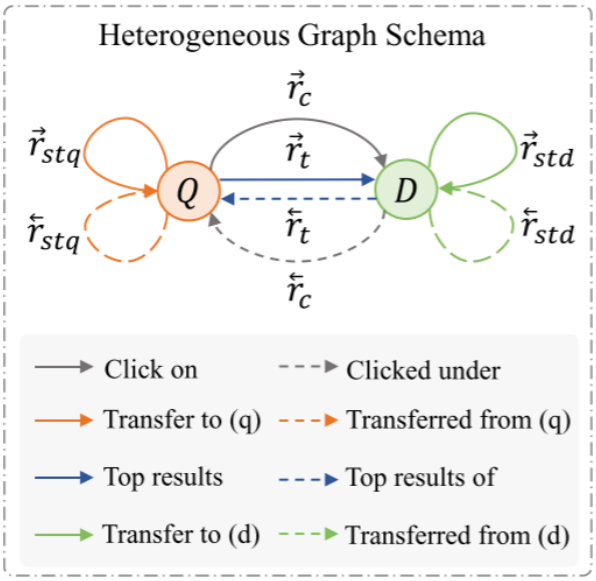

Figure 1: Graph schema. This figure is from the original paper[6]

session graph provides the local search intent with queries and documents in the session, while the query graph is built on relevant queries and documents in the search log to expand the current and provide the global query intent.
Because some of the queries and documents in the test dataset are not seen in the Tiangong-st used for building graphs, we use lazy string matching to link the unseen queries and documents to the graphs.

- **Ranking Stage**. HEXA firstly applies heterogeneous graph neural networks (HGNNs) to graphs for learning node representations used for obtaining session intent, $\mathbf{I}_s$ and query intent $\mathbf{I}_q$:

$$\mathbf{I}_s = \sum_{i=1}^{c} \alpha_i \mathbf{d}_i; \quad \mathbf{d}_i = \mathrm{HGT}(\mathrm{G_s})[\mathrm{d_i}], \tag{3}$$

$$\mathbf{I}_q = \mathrm{HGT}(\mathrm{G_q})[\mathrm{q}], \tag{4}$$

with the obtained intents, HEXA calculates ranking scores as follows:

$$s^g(d) = \mathbf{I}_q \mathbf{d}, \quad s^l(d) = \mathbf{I}_s \mathbf{d}. \tag{5}$$

where $\mathbf{d}$ is the representation of documents by the BERT. Because of great performance in encoding, HEXA applies BERT again for encoding sequence concatenated from session $\mathcal{S}$ and the output of the special token [CLS] is used for computing the ranking score:

$$s^q(d) = \mathbf{MLP}(\mathbf{BERT}(X)_{[CLS]}), \tag{6}$$

Combining these three scores, HEXA lastly outputs the ranking score through MLP layer:

$$s(d) = \mathbf{MLP}[s^g(d); s^l(d); s^q(d)], \tag{7}$$

- **Optimizing**. HEXA applies a point-wise loss to optimize the model. The loss function is introduced as follows:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} y_i \log \mathrm{z_i} + (1 - \mathrm{y_i}) \log(1 - \mathrm{z_i}), \tag{8}$$

where N is the number of training data, $z_i = \mathrm{sigmoid}(\mathrm{s}(\mathrm{d_i}))$

Because of the large quantity of the document dataset, we first use BM25 to filter candidate documents before employing HEXA.

2.3 POSS Subtask

The aim of the POSS task is to re-rank documents with the sessions truncated before the last query. In this subtask, we also apply BM25 and HEXA for re-ranking, regarding the POSS task as several FOSS tasks. With this pipelined mode, we first re-rank the documents of the first query in the session and pass the results to the next query until the documents of the last query are re-ranked.

2.4 SSEE SUBTASK

For the SSEE subtask, we implement two traditional approaches: sDCG[2] and sRBP[4], which we will introduce in the following content.

2.4.1 sDCG. When we obtain the result list for a query, our objective is to prioritize documents with higher relevance scores. Building upon this concept, we consider the relevance score of each document as the value gained from evaluating that specific document. The Cumulative Gain (CG)[1] is the sum of all relevance scores for documents in the result list of a query. The Cumulative Gain(CG) is defined as follows:

$$CG = \sum_{i=1}^{|K|} rel_i, \tag{9}$$

where $|K|$ represents the number of documents in the result list. rel_i is the relevance score of the i-th document.

However, Cumulative Gain does not account for the influence of a document's position. Typically, users are more likely to review documents in higher-ranking positions when examining the SERP. Documents with lower-ranking positions are less valuable since users spend more time and effort reviewing them. When two documents share the same relevance score, the gain from the lower-ranked document should be less. Therefore, in addition to CG, the authors introduce a discounting function to compute the gain of a document while considering its rank. The gain decreases as the sort position of a document decreases. A straightforward method to achieve this is by setting the document's gain equal to its relevance score divided by the logarithm of its rank. By adjusting the base of the logarithm, we can control the discount function to be either flatter or steeper, catering to users with varying levels of patience. The Discounted Cumulative Gain (DCG) is defined as follows:

$$DCG = \sum_{i=1}^{|K|} \frac{\mathrm{rel}_i}{\log_b(i+1)}, \tag{10}$$

where $|K|$ and rel_i is the same as Equation9. b is the base of the logarithm, which can be adjusted to control the discount function. if the b is bigger, the discounting function will be flatter.

DCG can be naturally extended to a session search metric. A straightforward approach is to accumulate the discounted cumulative gain for each query within a session search. Similar to the discounted cumulative gain, it's important to recognize that it consumes time and effort when a user reformulates a query. Therefore, the later a query is formulated within a session, the lower its perceived value. To account for this, the gain obtained from a query should be discounted based on its position in the session. The Session-based Discounted Cumulative Gain (sDCG) is defined as follows:

$$sDCG = \sum_{i=1}^{|S|} \frac{\mathrm{DCG}_i}{\log_{bq}(i+1)}, \tag{11}$$

where $|S|$ is the number of queries in a session search, where each query generates a result list. DCG_i represents the discounted cumulative gain for the i-th query. bq signifies the base of the logarithm, which can be configured to model users with varying levels of patience. If bq is set to a large value, the discount factor for queries at the same position will be small, indicating that users are more patient and tolerant of multiple query reformulations. On the other hand, a small bq suggests that the user is less patient and unlikely to perform numerous query reformulations.

2.4.2 sRBP. Aldo Lipani et al.[4] extend the Rank Biased Precision (RBP) metric, originally designed for single-query search, to the domain of session search by modeling expected user behavior. This novel evaluation metric, known as Session RBP (sRBP), is rooted in users' search behavior and is derived from a user model. Notably, sRBP focuses less on user clicks and places greater emphasis on result examination. In the user model, a user initiates the search by submitting an initial query to a search engine and receiving a result list in response. Subsequently, the user may undertake one of three actions:

(1) Continue examining the next result within the current query.
(2) Reformulate the query and obtain a new result list from the search engine.
(3) Finish the search, either because the user has gathered sufficient information to fulfill their needs or due to dissatisfaction with the search results.

These user behaviors can be formalized through two primary random variables:

(1) $E = \{e, \overline{e}\}$: e means examining the result and $\overline{e}$ denotes the absence of examination.
(2) $N = \{c, qr, l\}$: N represents a random variable associated with the next action. c represents the act of continuing to explore the current search result, qr represents the process of reformulating the query and l indicates the action of leaving the search system.

In addition to this user model, the computation of sRBP incorporates the concept of a cascade model. According to the cascade model, the gain from a single-query search is defined as follows:

$$g(q, r) = \sum_{i=1}^{|K|} dis(r_i) \cdot rel(q, r_i), \tag{12}$$

where q is the query, and r_i corresponds to the i-th result in the result list for query. $dis(\cdot)$ calculates a discount value based on the ranking position of the result. $rel(q, r_i)$ is the relevance score for query q and result r_i.

Expanding Equation12 to session search is as follows:

$$g(q, r) = \sum_{j=1}^{|S|} \sum_{i=1}^{|K|} dis(r_{j,i}) \cdot rel(q_j, r_{j,i}), \tag{13}$$

where q_j is the j-th query within a session search, and $r_{j,i}$ is the i-th result in the result list for the j-th query. The discounting function $dis(\cdot)$ takes into account both the ranking position of the result and the order of the query in the session. $rel(q_j, r_{j,i})$ is the relevance score for query q_j and result $r_{j,i}$.

The result with a higher probability of being examined by users should have a higher discount value. The authors of sRBP define the discount value for a certain result as equal to its probability of being examined by users, which is defined as follows:

$$dis(r_{j,i}) = p(E_{j,i} = e), \tag{14}$$

After deriving this expression[4], the final formula for the discounting function is as follows:

$$dis(r_{j,i}) = (\frac{p - bp}{1 - bp})^{j-1} (bp)^{i-1}, \tag{15}$$

where $b \in [0, 1]$ is named balance parameter, while $p \in [0, 1]$ is known as the persistence parameter.The role of b is to strike a balance between continuing to examine the results and reformulating the query, whereas p plays a role similar to the persistence parameter in RBP. Subsequently, the authors substitute $dis(r_{j,i})$ from Equation 15 into Equation 13 to derive sRBP, as expressed in follows:

$$sRBP(q, r) = (1-p) \sum_{j=1}^{|S|} \sum_{i=1}^{|K|} (\frac{p - bp}{1 - bp})^{j-1} (bp)^{i-1} \cdot rel(q_j, r_{j,i}), \tag{16}$$

when the session search is infinitely long, the sum of the discount functions is equal to $(1 - p)$. This factor can be viewed as a normalization for sRBP.

3 EXPERIMENTS

3.1 Metrics

The official metric of the FOSS subtask is Normalized Discounted Cumulative Gain (NDCG). The metrics of POSS are RsDCG and RsRBP. The official method to evaluate the reasonability of SSEE measures is by comparing their consistency with golden user satisfaction labels, using Pearson's r and Spearman's ρ.

3.2 Results and Analysis

3.2.1 FOSS and POSS. Because the methods we use are the same in the FOSS task and the POSS task, we only discuss the results in the FOSS task in this section. Results can be divided into two parts, test results and official results.

The test results of our approaches are evaluated on the Tiangong-ST dataset providing an annotated relevance score (0-4) for the last query of each session. The results are shown in Table 1. We observe that context-aware document ranking models can better understand user intent to make accurate retrievals.

Table 1: Results of Our Runs in testset of Tiangong-ST.

Model Name	nDCG@3	nDCG@5	nDCG@10
BM25	0.3117	0.3076	0.3222
HEXA	0.7819	0.8083	0.9053

Table 2: Official results in FOSS.

Run Name	nDCG@3	nDCG@5
BITIR-FOSS-NEW-2	0.0014880	0.0021785
BITIR-FOSS-NEW-1	0.0014391	0.0019561

Table 3: Evaluation of Our Runs in SSEE Subtask.

Run Name	Description	Pearson	Spearman
BITIR-SSEE-REP-2	sRBP	0.4326276	0.4376210
BITIR-SSEE-REP-1	sDCG	0.3878581	0.4076994

The official results are shown in Table 2. The official results and the test results have a significant discrepancy. We think the possible reasons are as follows:

- **(1)** We used the entire document to match the query, because of too much noise, BM25 retrieved a lot of unrelated documents. The difference in results between the Tiangong-ST dataset and the official dataset may be due to shorter queries and more accurate intents on the Tiangong-ST dataset
- **(2)** Due to limited computational resources, before using HEXA, we employed BM25 to initially filter 50 candidate documents for each query from a pool of 1 million documents. However, the size of the candidate set is too small, resulting in the exclusion of some relevant documents.
- **(3)**The method of graph linking is simple and does not consider the semantic information between tokens. As a result, most queries and documents are not linked to relevant nodes in the graph, resulting in the loss of a large amount of information.

3.2.2 SSEE. In the SSEE subtask, We set b and bq equal to 2 for sDCG and $b = 0.64$, $p = 0.86$ for sRBP. We use the usefulness for each result provided in the dataset as rel_i. The results of the SSEE subtask are presented in Table 3.

Under our parameter settings, sRBP outperforms sDCG. However, it's important to note that since Session Scores are not provided in the dataset, we did not optimize the parameters, and the parameter values used may not be optimal. sDCG emphasizes result ranking and relevance scores without explicitly modeling user search behavior. It uses a discount function based on result ranking and relevance scores to compute the gain for each query result. In contrast, sRBP implements a more detailed user model, considering whether the user will continue browsing, reformulate the query, or exit the search system. Additionally, sDCG does not incorporate normalization factors, which means it does not account for the influence of session length and tends to favor longer sessions. Both methods have their specific focuses and should be chosen based on the specific situation. In this subtask, our comparison primarily involves assessing the consistency of each metric with the golden user satisfaction labels. This could be attributed to the fact that sRBP places a greater emphasis on user behavior, which likely contributed to its higher consistency.

4 CONCLUSIONS

This paper presents our participation in the Session Search task at NTCIR-17. We apply the traditional retrieval model and graph-based model for the FOSS subtask and POSS subtask. We implement two existing metrics for the SSEE task but are not able to design more innovative and effective metrics. Although the official results of these tasks are terrible, we can accumulate more experimental experience for improving the performance of the model.

REFERENCES

[1] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

[2] Kalervo Järvelin, Susan L Price, Lois ML Delcambre, and Marianne Lykke Nielsen. 2008. Discounted cumulated gain based evaluation of multiple-query IR sessions. In *Advances in Information Retrieval: 30th European Conference on IR Research, ECIR 2008, Glasgow, UK, March 30-April 3, 2008. Proceedings 30*. Springer, 4–15.

[3] Haitao Li, Jiannan Wang, Jia Chen, Weihang Su, Beining Wang, Fan Zhang, Qingyao Ai, Jiaxin Mao, and Yiqun Liu. 2023. Overview of the NTCIR-17 Session Search (SS) Task. *Proceedings of NTCIR-17*.

[4] Aldo Lipani, Ben Carterette, and Emine Yilmaz. 2019. From a user model for query sessions to session rank biased precision (sRBP). In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*. 109–116.

[5] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389. https://doi.org/10.1561/1500000019

[6] Shuting Wang, Zhicheng Dou, and Yutao Zhu. 2023. Heterogeneous Graph-based Context-aware Document Ranking. *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining* (2023). https://api.semanticscholar.org/CorpusID:257079778