THUIR_SS at the NTCIR-17 Session Search (SS) Task

Xinyan Han
DCST, Tsinghua University
Quan Cheng Laboratory
Beijing 100084, China
hanxinya20@mails.tsinghua.edu.cn

Yiteng Tu
Renmin University of China
Beijing 100872, China
yitengtu16@gmail.com

Haitao Li
DCST, Tsinghua University
Quan Cheng Laboratory
Beijing 100084, China
liht22@mails.tsinghua.edu.cn

Qingyao Ai
DCST, Tsinghua University
Quan Cheng Laboratory
Beijing 100084, China
aiqy@tsinghua.edu.cn

Yiqun Liu
DCST, Tsinghua University
Zhongguancun Laboratory
Beijing 100084, China
yiqunliu@tsinghua.edu.cn

ABSTRACT

Session Search holds significant importance in the field of information retrieval and user experience. In this paper, we detail the approach of the THUIR_SS team in the NTCIR17 Session Search (SS-2) task. Specifically, we submit five runs for FOSS and POSS tasks respectively. We try different approaches for feature fusion, including learning to rank and linearly combination. The final report of the SS-2 task demonstrate the effectiveness of our method, significantly outperforming other competitors.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; **Retrieval models and ranking**.

KEYWORDS

session search, dense retrieval, document ranking

ACM Reference Format:
Xinyan Han, Yiteng Tu, Haitao Li, Qingyao Ai, and Yiqun Liu. 2023. THUIR_SS at the NTCIR-17 Session Search (SS) Task. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

TEAM NAME

THUIR_SS

SUBTASKS

FOSS, POSS

1 INTRODUCTION

Session Search is becoming an increasingly important topic in the information retrieval community. In traditional ad-hoc retrieval scenarios, search engines only consider one-shot query information [8?]. However, in the multi-turn interactions between users and search engines, there is often a wealth of user search behavior information [1], which can help search engines better understand user search intent, thereby providing more satisfying search results for users. In a session-based search setting, the search engine takes into account not only the current query but also the context of prior interactions or queries made by the user in the same session [10, 13]. This information is beneficial for user intent understanding and generating higher-quality search results.

As a well-known competition in NTCIR, the Session Search (SS) task provides extensive session data and contributes to the advancement of session search techniques. SS-2 is the second year of the session search competition, which consists of three subtasks: *Fully Observed Session Search* (FOSS), *Partially Observed Session Search* (POSS) and *Session-level Search Effectiveness Estimation* (SSEE).

In this paper, we introduce the solution of the THUIR_SS team for the Session Search task (SS-2), which achieves the best performance in both FOSS and POSS subtasks. To be specific, we submit five runs for the FOSS and POSS tasks respectively. We use both traditional methods and pre-trained language models to extract features. These features are assembled via various methods, including learning to rank algorithms, RRF method [4], and linear combination. Surprisingly, the linear combination method achieves the best performance in both subtasks. We suspect that is due to the change in retrieval intent present in the session, and that linear combination is affected to a lower degree.

2 METHODS

First, we introduce the definition of notation in this paper. In FOSS subtask, a search context is denoted as $C = [q_1, d_1^+, ..., q_{k-1}, d_{k-1}^+, q_k]$, where q_i is the i-th query in a session and d_i^+ is the first corresponding clicked document title of q_i. If there is no user clicked document for q_i, we skip d_i^+, i.e., there might be two or more consecutive q in C. In POSS subtask, the user interaction information for the last $k - n$ queries is not provided, we just skip d^+ and concatenate q. As a result, a search context can be denoted as $C = [q_i, d_1^+, ..., q_n, d_n^+, q_{n+1}, ..., q_j]$, where $n + 1 \leq j \leq k$. In fact, the FOSS session context can be seen as a special case of POSS session context where $n + 1 = k$. Thus we get a unified representation of C: $C = [q_i, d_1^+, ..., q_n, d_n^+, q_{n+1}, ..., q_j]$.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2.1 Feature Extraction

Conventional information retrieval approaches involve the assessment and ranking of (q, d) pairs based on a limited set of factors including term frequency (tf), inverse document frequency (idf), and document length. While these methodologies have been instrumental in the past, they often overlook the complexities of modern information retrieval challenges such as contextual information. As proven in [11, 12, 18], by using the combination of more features including neural signals rather than a small set of features, we can get more effective ranking results. In our approaches, we incorporate a total of 11 features which include both term-level features from traditional sparse retrieval methods and semantic-level features obtained through deep neural networks. The details of these features are shown in Table 1.

2.1.1 Sparse Retrieval. A sparse retrieval model ranks the documents focusing on the lexical matching scores between the query and the documents. It is usually based on the inverted index structure with term-level matching signals like tf-idf. In this paper, we choose two classic but effective sparse retrieval methods, BM25 [15] and QLD [14]. Specifically, given a (q, d) pair, we obtain four relevance features through the BM25 and QLD algorithms. We first calculate the BM25 and QLD scores of the pair, then we conduct BM25 and QLD again with RM3 [7] pseudo-relevance feedback. The formulas for BM25 and QLD are shown in Equation (1) and (2):

$$BM25(d, q) = \sum_{t \in q} \frac{IDF(t) * TF(t, d) * (k_1 + 1)}{TF(t, d) + k_1 * (1 - b + b * \frac{len(d)}{avgdl}})) \tag{1}$$

$$QLD(d, q) = \prod_{t \in q} P(t|d)^{c(t,q)} \tag{2}$$

In Equation (1), $TF(t, d)$ is the occurrence frequency of term t in d, $avgdl$ is the average length of documents in the collection. $IDF(t)$ can be calculated via Equation (3):

$$IDF(t) = log \frac{N - n(t) + 0.5}{n(t) + 0.5} \tag{3}$$

where N is the number of documents, and $n(t)$ is the number of documents that contains term t. In Equation (2), $c(t, q)$ is the occurrence frequency of term t in q, and $P(t|d)$ is the probability of term t occurring in document d.

On one hand, we solely feed the single-turn query q_k to the aforementioned method to obtain four ad-hoc relevance scores. On the other hand, we incorporate contextual information into the query and regard the context information C as the input so as to extract another four features at the session level.

2.1.2 Dense Retrieval. In dense retrieval (DR) approaches, a dual-encoder model projects the query and document texts into dense vectors, and the similarity score is then obtained via a simple inner-product or cosine-similarity calculation:

$$r_q = BERT_{[CLS]}(q) \tag{4}$$

$$r_d = BERT_{[CLS]}(d) \tag{5}$$

$$s(q, d) = r_q \cdot r_d \tag{6}$$

To train a DR model, contrastive learning has been proven to be a highly effective approach [5, 19]. We utilize the InfoNCE Loss as the loss function:

$$\mathcal{L} = -log \frac{\exp\left(s(q, d^+)\right)}{\exp\left(s(q, d^+)\right) + \sum_j \exp\left(s(q, d_j^-)\right)} \tag{7}$$

where d^+ and $\{d_j^-\}$ separately represent the clicked document and unclicked documents as well as in-batch negatives. Similar to Sec 2.1.1, we also use both the single-turn query and the context as input to obtain two scores on different levels.

2.1.3 Context-aware Ranking Model. Apart from DR models, we also train a fine-grained context-aware ranking model, DCL [21], with a curriculum learning framework. According to Zhu et al. [21], in the session search scenario, the difficulty of learning from different positive pairs and negatives varies, thus the sequence of easy and hard samples shouldn't be randomly selected in the model training process. In the framework, we sort positive pairs and negative pairs to make the model training process more robust. The model is composed of an MLP head after the pre-trained BERT encoder. We concatenate the session sequence and feed it to DCL:

$$X = [CLS]q_1[EOS]d_1[EOS]...q[EOS][SEP]d[EOS][SEP] \tag{8}$$

$$r = BERT(X)_{[CLS]} \tag{9}$$

$$OUTPUT = MLP(r) \tag{10}$$

where r is the dense representation of the session sequence and the $OUTPUT$ score is the session level relevance feature as well as our last feature.

2.2 Learning-to-rank

Learning-to-rank is a machine learning technique that constructs ranking models for information retrieval systems [2, 9]. Its goal is to rank new, unseen lists in a similar way to rankings in the training data. We feed the 11 features (Table 1) extracted in Sec 2.1 into two widely-used learning-to-rank models, LightGBM [6] and LambdaMART [17].

	feature
1	ad-hoc score of BM25
2	ad-hoc score of QLD
3	ad-hoc score of BM25 with RM3
4	ad-hoc score of QLD with RM3
5	session score of BM25
6	session score of QLD
7	session score of BM25 with RM3
8	session score of QLD with RM3
9	ad-hoc score of DR model
10	session score of DR model
11	sesion score of DCL model

Table 1: features of learning-to-rank model

2.2.1 LambdaMART. LambdaMART [17] is a classic learning-to-rank model based on the gradient boosting decision tree (GBDT) algorithm. Its advantage is that it can directly optimize information retrieval evaluation metrics such as NDCG.

2.2.2 LightGBM. Light Gradient Boosting (MachineLightGBM) [6] is another machine learning framework that implements GBDT. Compared with traditional GBDT methods, it has faster training speed, lower memory consumption, better accuracy, distributed support, and handling of large-scale data.

2.3 RRF

Reciprocal Rank Fusion (RRF) is a simple but effective method to combine the document rank of several from multiple IR systems [4]. It's an unsupervised method and doesn't need any training sample. The formula of RRF is:

$$RRF(d \in D) = \sum_{r \in R} \frac{1}{k + r(d)} \tag{11}$$

where D is a set of documents to be ranked and R is a set of ranking lists. $r(d)$ is the rank of d in ranking list r. k is a hyperparameter. We sort documents according to the score of features. Since we have extracted 11 features, we get 11 rankings sorted by each feature score. Then we use all 11 rankings to calculate RRF scores.

2.4 Linear Combination

We use the linear combination of two scores of (C, d) pair to generate LS score.

$$LS(C, d) = a * Norm_1(C, d) + (1 - a) * Norm_2(C, d) \tag{12}$$

$$Norm_i(C, d) = \frac{S_i(C, d) - min(S_i(C))}{max(S_i(C)) - min(S_i(C))} \tag{13}$$

where $S_i(C, d)$,$i \in 1, 2$ is the score of (C, d) pair in method i. $max(S_i(C))$ is the maximum score of $S_i(C, d)$ for all d, and $min(S_i(C))$ is the minimum score of $S_i(C, d)$ for all d considering C. $a \in [0, 1]$ is a hyperparameter and is set to 0.5 in our experiment.

In THUIR_SS-FOSS-NEW-3 and THUIR_SS-POSS-NEW-3, we choose the ad-hoc score of BM25 as S_1 and the score computed by DCL model as S_2. In submission THUIR_SS-FOSS-NEW-6 and THUIR_SS-POSS-NEW-6, we replace S_1 with QLD ad-hoc score.

3 EXPERIMENTS

3.1 Evaluation Metrics

The evaluation metrics of FOSS subtask are NDCG@3 and NDCG@5. NDCG@k can be formalized as follows:

$$DCG@k = \sum_{i}^{k} \frac{2^{r(i)} - 1}{log_2(i + 1)}$$

$$NDCG@k = \frac{DCG@k}{IDCG} \tag{14}$$

where $r(i)$ is the relevance of the i-th document in the ranking list. IDCG is the DCG calculated with ranking list sorted by relevance from high to low.

The evaluation metrics of POSS subtask are RS_DCG and RS_RBP [20].

$$RS\_DCG = \sum_{m=1}^{M} mem_m \sum_{n=1}^{N} g(r_{m,n}, q_m) \cdot d_{m,n}(sDCG)$$

$$RS\_RBP = \sum_{m=1}^{M} mem_m \sum_{n=1}^{N} g(r_{m,n}, q_m) \cdot d_{m,n}(sRBP)$$

$$mem_m = FF(M - m) = e^{-\lambda(M-m)}$$

where M is the number of queries in a session and N is the number of documents. $g(r_{m,n}, q_m)$ is the gain function. $d_{m,n}(sRBP)$ and $d_{m,n}(sRBP)$ is the session level discount function.

3.2 Implementation Details

The DR model is trained on TianGong-ST dataset [3]. The dual-encoder of DR model is initialized with bert-base-chinese. We set the learning rate to 2e-5 and the number of training epochs to 60. The DCL model is also trained on TianGong-ST with parameters set to default values. When training LambdaMART, we choose lambdarank as the training objective and NDCG@10 as the training metric. The learning rate is 0.01. For the LightGBM model, we choose LambdaMART as the ranker. We set the number of trees to 1000 and the number of leaves for each tree to 10. The hyperparameter k in the RRF formula is left to the default value of 60 in our experiment.

Run Name	Description	NDCG@3	NDCG@5	Rank
THUIR_SS-FOSS-NEW-1	learning-to-rank (LightGBM)	0.1547940	0.2038491	5
THUIR_SS-FOSS-NEW-3	linear combination	0.5853154	0.6745773	1
THUIR_SS-FOSS-NEW-4	learning-to-rank (LambdaMART)	0.2506041	0.3309875	4
THUIR_SS-FOSS-NEW-5	RRF	0.3931865	0.4768206	3
THUIR_SS-FOSS-NEW-6	linear combination	0.5643186	0.6569274	2

Table 2: Preliminary Evaluation of Our Runs in FOSS Subtask

Run Name	Description	RS_DCG	RS_RBP	Rank
THUIR_SS-POSS-NEW-1	learning-to-rank (LightGBM)	0.023533	0.048312	5
THUIR_SS-POSS-NEW-3	linear combination	0.174898	0.367266	2
THUIR_SS-POSS-NEW-4	learning-to-rank (LambdaMART)	0.068510	0.143386	4
THUIR_SS-POSS-NEW-5	RRF	0.136628	0.288760	3
THUIR_SS-POSS-NEW-6	linear combination	0.181201	0.379338	1

Table 3: Preliminary Evaluation of Our Runs in POSS Subtask

run name	S_1	S_2
THUIR_SS-FOSS-NEW-3 THUIR_SS-POSS-NEW-3	BM25 ad-hoc score	DCL score
THUIR_SS-FOSS-NEW-6 THUIR_SS-POSS-NEW-6	QLD ad-hoc score	DCL score

Table 4: linear combination

3.3 Results

The descriptions and final evaluation results on FOSS and POSS subtasks are shown in Tabel 2 and Table 3.

In both FOSS and POSS subtasks, the linear combination method achieves the best performance of all submissions. It's surprising that this simple and unsupervised method beats learning-to-rank models that are expected to have better performance. We think our learning-to-rank method still has room for improvement. First, more features can be selected to feed into the model, such as user interaction information (clicks and timestamp) and the length of query and document, according to the previous work [16]. Second, we find that in FOSS and POSS subtasks, sometimes the query at the beginning of a session and the query at the end of a session are not very semantically related. It is more reasonable to consider them in different sessions. These unrelated queries bring noise to our results. Our linear superposition method only uses ad-hoc scores and neural reranking scores, which reduces the noise brought by unrelated queries in a session to some extent. Also, we think that designing a new method for the POSS subtask, rather than simply treating its context as the context of FOSS without clicked documents, would improve performance.

4 CONCLUSIONS

Our team (THUIR_SS) participates in the FOSS and POSS subtask of the NTCIR-17 Session Search (SS) Task. We try learning to rank model, RRF method, and linear combination method. The submission using linear combination achieves the best performance in both FOSS and POSS subtasks.

REFERENCES

[1] Eugene Agichtein, Ryen W White, Susan T Dumais, and Paul N Bennet. 2012. Search, interrupted: understanding and predicting search task continuation. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval.* 315–324.

[2] Jia Chen, Haitao Li, Weihang Su, Qingyao Ai, and Yiqun Liu. 2023. THUIR at WSDM Cup 2023 Task 1: Unbiased Learning to Rank. arXiv:2304.12650 [cs.IR]

[3] Jia Chen, Jia Xin Mao, Yinqun Liu, Min Zhang, and Ma Shaoping. 2019. TianGong-ST: A New Dataset with Large-scale Refined Real-world Web Search Sessions. In *Proceedings of the 28th ACM International on Conference on Information and Knowledge Management.*

[4] Gordon V. Cormack, Charles L. A. Clarke, and Stefan Büttcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *International Acm Sigir Conference on Research Development in Information Retrieval.*

[5] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906* (2020).

[6] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17).* Curran Associates Inc., Red Hook, NY, USA, 3149–3157.

[7] Victor Lavrenko and W Bruce Croft. 2001. Relevance-based language models. (2001), 120–127.

[8] Haitao Li, Qingyao Ai, Jia Chen, Qian Dong, Yueyue Wu, Yiqun Liu, Chong Chen, and Qi Tian. 2023. SAILER: Structure-aware Pre-trained Language Model for Legal Case Retrieval. arXiv:2304.11370 [cs.IR]

[9] Haitao Li, Jia Chen, Weihang Su, Qingyao Ai, and Yiqun Liu. 2023. Towards Better Web Search Performance: Pre-training, Fine-tuning and Learning to Rank. *arXiv preprint arXiv:2303.04710* (2023).

[10] Haitao Li, Jia Chen, Jiannan Wang, Weihang Su, Qingyao Ai, Xinan Hu, Beining Wang, and Yiqun Liu. 2023. Overview of the NTCIR-17 Session Search (SS-2) Task.

[11] Haitao Li, Weihang Su, Changyue Wang, Yueyue Wu, Qingyao Ai, and Yiqun Liu. 2023. THUIR@ COLIEE 2023: Incorporating Structural Knowledge into Pre-trained Language Models for Legal Case Retrieval. *arXiv preprint arXiv:2305.06812* (2023).

[12] Haitao Li, Changyue Wang, Weihang Su, Yueyue Wu, Qingyao Ai, and Yiqun Liu. 2023. THUIR@ COLIEE 2023: More Parameters and Legal Knowledge for Legal Case Entailment. *arXiv preprint arXiv:2305.06817* (2023).

[13] Shengjie Ma, Chong-Yee Chen, Jiaxin Mao, Qi Tian, and Xuhui Jiang. 2023. Session Search with Pre-trained Graph Classification Model. *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2023). https://api.semanticscholar.org/CorpusID:259949880

[14] Jay M. Ponte and W. Bruce Croft. 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* (Melbourne, Australia) *(SIGIR '98).* Association for Computing Machinery, New York, NY, USA, 275–281. https://doi.org/10.1145/290941.291008

[15] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at TREC-3. In *Text Retrieval Conference.* https://api.semanticscholar.org/CorpusID:3946054

[16] Weihang Su, Xiangsheng Li, Yiqun Liu, Min Zhang, and Shaoping Ma. 2023. THUIR2 at NTCIR-16 Session Search (SS) Task. arXiv:2307.00250 [cs.IR]

[17] Qiang Wu, Christopher JC Burges, Krysta M Svore, and Jianfeng Gao. 2010. Adapting boosting for information retrieval measures. *Information Retrieval* 13 (2010), 254–270.

[18] Shenghao Yang, Haitao Li, Zhumin Chu, Jingtao Zhan, Yiqun Liu, Min Zhang, and Shaoping Ma. 2022. THUIR at the NTCIR-16 WWW-4 Task. *Proceedings of NTCIR-16. to appear* (2022).

[19] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 1503–1512.

[20] Fan Zhang, Jiaxin Mao, Yiqun Liu, Weizhi Ma, and Shaoping Ma. 2020. Cascade or Recency: Constructing Better Evaluation Metrics for Session Search. In *SIGIR '20: The 43rd International ACM SIGIR conference on research and development in Information Retrieval.*

[21] Yutao Zhu, Jianyun Nie, Yixuan Su, Haonan Chen, Xinyu Zhang, and Zhicheng Dou. 2022. From Easy to Hard: A Dual Curriculum Learning Framework for Context-Aware Document Ranking. *Proceedings of the 31st ACM International Conference on Information & Knowledge Management* (2022). https://api.semanticscholar.org/CorpusID:251719021