MM6 at the NTCIR-16 Session Search Task

Shengjie Ma
Gaoling School of Artificial Intelligence, Renmin University of China
sm8807@nyu.edu

Chuwei Zeng
Gaoling School of Artificial Intelligence, Renmin University of China
zengchw3@mail2.sysu.edu.cn

Jiaxin Mao
Gaoling School of Artificial Intelligence, Renmin University of China
maojiaxin@gmail.com

ABSTRACT

A single query may hardly satisfy the user's information needs, so that users will continuously submit more queries to the search system until they are satisfied or stop trying. This search process is called session search. The MM6 team participated in the IR subtask of the NTCIR-16 Session Search Task. This paper reports our three approaches for FOSS task and one approach for POSS task. We display and discuss the official results at the end. Please also refer to the past NTCIR proceedings[1].

KEYWORDS

session search, document ranking, session graph

TEAM NAME

MM6

SUBTASKS

Fully Observed Session Search (FOSS). Partially Observed Session Search (POSS)

1 INTRODUCTION

As users increasingly rely on search engines to access useful information, complex search scenarios emerge in endlessly. A single query may hardly satisfy the user's information needs, so that users will continuously submit more queries to the search system until they are satisfied or stop trying. This search process is called session search. During a search session, users' initial queries might deviate from the true intention, and their behavior and decisions will evolve according to historical search result. In addition, their search intentions may change. Therefore search systems are expected to handle these problems for better ranking.

The MM6 team participated in the IR sub-tasks (FOSS and POSS) of the Session Search Task [3]. The FOSS subtask aims to re-tank the candidate documents for the last query of a session. While the POSS subtask aims to re-rank the documents for the last $k - m$ queries(query) according to the partially observed contextual information in previous search rounds, where $1 \leq m \leq k - 1$.

To model users' on-task search behaviors for FOSS subtask, we experimented three approaches.The first method we choose is Context Attentive document Ranking and query Suggestion (CARS)[1], a hierarchical recurrent neural network (RNN) based multi-task model. Then considering the outstanding performance of BERT[4] on NLP tasks, we tested a BERT based model Contrastive learning for Context Aware document ranking (COCA)[10]. The last method for FOSS subtask is named Session Graph (SG), which is a graph based neural model motivated by the heterogeneous graph pooling (HG-Pool) method[7].

Detailed description of our approach is in section 3 and we discuss the experimental and official results in section 4-5.

[1]https://research.nii.ac.jp/ntcir/workshop/OnlineProceedings15/NTCIR/toc_ntcir.html

2 RELATED WORK

The development of neural networks evolved various solutions for in-session document ranking. Some researchers proposed a hierarchical neural structure with RNNs to model historical queries and suggest the next query[6]. This model is further extended with the attention mechanism to better represent sessions and capture user-level search behavior [11]. Recently, researchers found that jointly learning query suggestion and document ranking can boost the model's performance on both tasks[1]. In addition to leveraging historical queries, the historical clicked documents are also reported to be helpful in both query suggestion and document ranking[2]. More recently, large-scale pretrained language models, such as BERT[4], are utilized in session search[8]. Based on BERT, [10]use data augmentation strategies and contrastive learning to pretrain the model in a self-supervised manner.

3 METHODS

In this section, we introduce our approaches for FOSS and POSS task respectively.

3.1 FOSS SUBTASK

To handle FOSS task, we experimented with three approaches, Context Attentive document Ranking and query Suggestion (CARS)[1], Contrastive learning for Context Aware document ranking (COCA)[10] and Session Graph (SG). We will briefly introduce them as follows.

3.1.1 Context Attentive document Ranking and query Suggestion. Given that the form of every user search history is a sequence. We first implemented a hierarchical recurrent neural network (RNN) based multi-task model CARS, which maintains a two-level RNN structure for learning in-task search con text representation. At the lower level, RNN-based query and document encoders encapsulate information in a user's query formulation and click actions into continuous embedding vectors; and at the upper level, another set of RNN-based query- and document-session encoders take the embeddings of each search action as input and summarize past on-task search context on the fly. Then, the learned representations from both levels are utilized to rank documents under the current query and suggest the next query[1]. For the text length constraint, we only use document titles and the first clicked document of every query to build user search historical sequences.

3.1.2 Contrastive learning for Context Aware document ranking. Considering the outstanding performance of BERT[4] on NLP tasks,

we tested a BERT based model COCA. In this work, [10] propose a data augmentation approach to generate possible variations from a search log. More specifically, COCA[10] use three strategies to mask some terms in a query or document, delete some queries or documents, or reorder the sequence. These strategies reflect some typical variations in user's behavior sequences and generate more training data from search logs. Based on the augmented data, contrastive learning is implemented with a pre-trained language model BERT[4] through encoding a sequence and its variants into a contextualized representation with a contrastive loss. The document ranking is then learned by a linear projection on top of the optimized sequence representation. Similarly for the text length constraint, we only use document titles and the first clicked document of every query to build user search historical sequences.

3.1.3 Session Graph. Traditional methods model a session as a sequence. However a user's search history includes various types of useful information, for example queries, document titles, document contents, clicks, and other additional information. Simply compressing a user history data to a flat sequence might ignore natural topological relationships existing between them. Apart

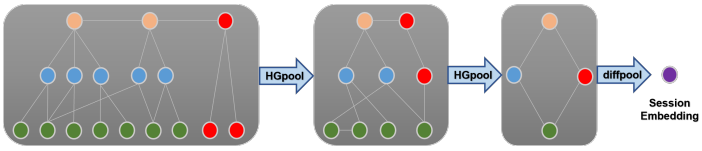

Figure 1: The session graph and pooling.

from that, session data is heterogeneous. Thus motivated by the heterogeneous graph pooling (HG-Pool) method[7], we consider experimenting heterogeneous graph method on session search task, which aims to learn more fine-grained session features.

Graph Construction. From the Session Search Task Data, each session is constructed as a graph, which contains four types of nodes, including history queries, clicked documents, keywords and the last query. Since the document content is significant but always too long. We believe the keywords is an efficient way to utilize document contents. Meanwhile, aiming to underline the importance of the last query, we set the last query and its keywords as an extra node type. We regard each query and corresponding clicked documents as nodes and link each query-document pair. We also link every two queries that are adjacent in time. In addition, the keywords nodes are extracted by TD-IDF from corresponding documents. Every keyword is linked to the documents where it appears.

Node vectors Learning. We use a a multi-head self-attention network to learning token vectors from query and document titles and then use an inner-attention network to generate query and document node vectors. The keywords vectors are initialized from pre-trained Chinese word vectors.

Graph Representation learning. Motivated by the heterogeneous graph pooling (HG-Pool) method[7], which can consider the varied characteristics of different kinds of nodes for graph representation learning, we adjust HG-pool method to handle four types of nodes. After the pooling operation, we get four vectors of each node type, which condense the information of its corresponding type of nodes in the original graph(as show in Fig.1.). Then we apply a diff-pool[9] to get the graph representation.

Ranking and Training. We apply a multi-head self-attention network and an inner-attention network to encode candidates title and an inner-attention network to encode their keywords. Then we use a dense layer to gain final candidates embeddings, which are scored by a classifier with its corresponding session graph representation. We train the model by BCE-Loss.

Table 1: The training results of our FOSS runs

METRICS	CARS	COCA	SG
MAP	0.5856	0.6756	0.8240
MRR	0.6016	0.6930	0.8430
NDCG@1	0.4388	0.5488	0.7330
NDCG@3	0.5536	0.6585	0.8359
NDCG@5	0.6145	0.7141	0.8638
NDCG@10	0.6920	0.7613	0.8747

3.2 POSS SUBTASK

We adopted a Hierarchical Behavior Aware Transformers (HBA-Transformers) model [5] to this task. This model uses Bert encoder to get contextualized representations of input tokens and then uses intra-behavior and inter-behavior attention to get the final representations. Then the representation of the special token [cls] is fed to the document ranker to get a relevance score

4 FOSS SUBTASK RESULT

In this section, we will first discuss the training results and then discuss the final official evaluation results.

During the training process, training data is divided into training, validation, and test sets at a ratio of 8:1:1. In all sets, there are 10 candidate documents for each query in the session and we use the clicked documents as the satisfied clicks. For the experiment settings, training epoch is 3. Training batch-size for CARS, COCA and SG are 32, 128 and 32 respectively. Learning rate is 5e-5 and linearly decayed during the training. All hyperparameters are tuned based on the performance on the validation set. An interesting phenomenon is that when the batch-size is smaller than 128, the experimental performance of COCA is poor.

The training result is shown in Table 1. We found that compared with the RNN-based multi-task learning models (CARS), BERT-based methods (COCA) achieve better performance. Among all models, SG achieves the best results, which demonstrates the effectiveness on modeling session data as heterogenous graphs.

However, the official evaluation results of FOSS task are not as good as expected. As shown in Table 2, the official result of CARS is not in the list and COCA ranks low (6/12) but relatively higher than SG (9/12). Possible reasons could be over-fitting during training or the click label and human label are inconsistent in distribution.

5 POSS SUBTASK RESULT

The POSS subtask aims to re-rank the documents from the last several queries in a POSS session. We adopted a Hierarchical Behavior Aware Transformers (HBA-Transformers) model [5] to this task.

Table 2: The official evaluation results of our FOSS runs

METRICS	CARS	COCA(REP − 1)	SG(NEW − 21)
NDCG@3	-	0.4253	0.3642
NDCG@5	-	0.4420	0.3850
NDCG@10	-	0.4572	0.4029

Table 3: The official evaluation results of our POSS runs

RUN NAME	RS_RBP	RS_DCG
MM6-POSS-REP-2	0.326737	0.423102
MM6-POSS-REP-1	0.299660	0.379261

However, the original model is not designed for POSS task, since there used to be only one query, rather than several, that needs to re-rank the corresponding documents. So we have to modify the model before use it. We choose to split a POSS session into several foss session so that we can directly use the model. The reason we do not consider the unobserved queries as history is that we think this will introduce extra bias to the results. Based on this model, we generated 2 runs for this subtask. The first run sets the history window size to 3, which means the model takes 3 history queries into account, since we think queries that are too far from the current query provides little information and may mislead the model. The second run sets the history window size to 0, which means we do not use any history information to help the re-ranking process. We adopt this setting because of our observation of the POSS dataset that there exists many session in which the queries have no dependency with each other. It seems that the users just select trending queries recommended by the search engine to form a search session. In this case, considering history interactions do no good to the re-ranking process of the current query. The official results show that run2 performs better than run1, as shown in Table 3.

6 CONCLUSIONS

Although unsatisfactory, the results we made still show the potential of graph neural networks, especially heterogeneous graphs, in search system. In the future we should improve the theoretical knowledge for model optimization and accumulate more experimental experience.

REFERENCES

[1] Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. 2018. Multi-Task Learning for Document Ranking and Query Suggestion. In *International Conference on Learning Representations*. https://openreview.net/forum?id=SJ1nzBeA-

[2] Wasi Uddin Ahmad, Kai-Wei Chang, and Hongning Wang. 2019. Context attentive document ranking and query suggestion. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 385–394.

[3] Jia Chen, Weihao Wu, Jiaxin Mao, Beining Wang, Fan Zhang, and Yiqun Liu. 2022. Overview of the NTCIR-16 Session Search (SS) Task. *Proceedings of NTCIR-16. to appear* (2022).

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[5] Chen Qu, Chenyan Xiong, Yizhe Zhang, Corby Rosset, W Bruce Croft, and Paul Bennett. 2020. Contextual Re-Ranking with Behavior Aware Transformers. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1589–1592.

[6] Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. 2015. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *proceedings of the 24th ACM international on conference on information and knowledge management*. 553–562.

[7] Chuhan Wu, Fangzhao Wu, Yongfeng Huang, and Xing Xie. 2021. User-as-Graph: User Modeling with Heterogeneous Graph Pooling for News Recommendation. In *IJCAI*.

[8] Chenyan Xiong, Chen Qu, Yizhe Zhang, Corby Rosset, W. Bruce Croft, and Paul Bennett. 2020. Contextual Re-Ranking with Behavior Aware Transformers. In *SIGIR 2020*. ACM, 1589–1592. https://www.microsoft.com/en-us/research/publication/contextual-re-ranking-with-behavior-aware-transformers/

[9] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. 2018. Hierarchical graph representation learning with differentiable pooling. *Advances in neural information processing systems* 31 (2018).

[10] Yutao Zhu, Jian-Yun Nie, Zhicheng Dou, Zhengyi Ma, Xinyu Zhang, Pan Du, Xiaochen Zuo, and Hao Jiang. 2021. Contrastive Learning of User Behavior Sequence for Context-Aware Document Ranking. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2780–2791.