Hierarchical Attention Network for Context-Aware Query Suggestion

Xiangsheng Li[†], Yiqun Liu[†⋆], Xin Li[†],Cheng Luo[†], Jian-Yun Nie[‡], Min Zhang[†], Shaoping Ma[†]

†Department of Computer Science and Technology, Institute for Artificial Intelligence,
Beijing National Research Center for Information Science and Technology,
Tsinghua University, Beijing 100084, China
‡Université de Montréal, Montreal, Canada
yiqunliu@tsinghua.edu.cn

Abstract. Query suggestion helps search users to efficiently express their information needs and has attracted many studies. Among the different kinds of factors that help improve query suggestion performance, user behavior information is commonly used because user's information needs are implicitly expressed in their behavior log. However, most existing approaches focus on the exploration of previously issued queries without taking the content of clicked documents into consideration. Since many search queries are short, vague and sometimes ambiguous, these existing solutions suffer from user intent mismatch. To articulate user's complex information needs behind the queries, we propose a hierarchical attention network which models users' entire search interaction process for query suggestion. It is found that by incorporating the content of clicked documents, our model can suggest better queries which satisfy users' information needs. Moreover, two levels of attention mechanisms are adopted at both word-level and session-level, which enable it to attend to important content when inferring user information needs. Experimental results based on a large-scale query log from a commercial search engine demonstrate the effectiveness of the proposed framework. In addition, the visualization of the attention layers also illustrates that informative words and important queries can be captured.

Keywords: Query suggestion, recurrent neural networks, click-through behavior

1 Introduction

Web search queries are usually short and ambiguous [1]. According to the investigations conducted on large-scale commercial search engines, a query often contains less than 3 terms [2] and over 16% of queries are ambiguous [3]. It is therefore challenging for search engines to understand user's search intents. Query suggestion is widely applied by commercial search engines to help users organize their queries and express their information needs more efficiently. It is shown that query suggestion can significantly improve user satisfaction, especially for informational queries [4].

Existing approaches for query suggestion mainly focus on mining the co-occurred queries from query log [5, 6]. The assumption is that a frequently co-occurred query is

[⋆] Corresponding author

likely to be the next query issued by users. These methods suffer from data sparsity and can hardly provide satisfactory results for long-tail queries [7]. To handle this problem, more features based on click-through (e.g., click position, click frequency and dwell time) are exploited [8, 9]. They assumed that different users share common interests with each other when their click behaviors are similar. However, this assumption is not reasonable in many cases since the issued queries are ambiguous. The key problem is to improve the query representation with more clear search intent.

Liu et al. [10] analyzed user's search processes and concluded that in addition to the previous queries, user's search intent can also be reflected by the clicked results. We follow this observation and attempt to incorporate the entire search interactions into query suggestion. Due to the brevity of query, precisely expressing user information needs in queries is intractable. Clicked documents can be regarded as an implicit description of the issued query and enable us to better infer user's information needs. For example, a user submits a query "Tourism" to the search engine, and clicks a document about "Shopping centers". It is intuitive to suggest queries about tourism if we only consider the information of query. But the clicked document reflects that this user is more interested in shopping during the traveling. Existing methods do not consider this information and thus ignore user's diverse and detailed search preference. If we can take advantage of this observation, the suggested query will be closer to the real information needs. Using clicked documents can be helpful to infer the precise search intent behind the short queries.

In this paper, we propose a hierarchical attention network (HAN) which models not only the issued queries but also the clicked documents in a whole session. The network contains three layers of encoder-decoder based on recurrent neural networks (RNN). The first layer concatenates two encoders which model the clicked documents and queries, respectively. Then a session-level encoder summarizes the information of previous search process. The final decoder is utilized to predict the next query according to the session information encoded by the session-level encoder. On the other hand, context information in a session generally has different importance because only a few words contribute to the inference of information needs. To capture the pivot information automatically, we construct two levels of attention mechanisms at word-level and session-level. The word-level attention enables us to focus on informative and important words in clicked documents and queries while the session-level attention aims to recognize the useful queries (e.g., queries with informative words or click feedbacks).

To validate the effectiveness of our method, we perform our experiments on a query log from a commercial search engine. Comparing to the baselines, our method can significantly improve the performance of query suggestion. Furthermore, we visualize two attention layers and find that the informative words and useful queries in a session can be qualitatively selected. The main contributions of this paper are three-folds:

1. The proposed HAN model encodes both issued queries and the content of clicked documents, which helps us better understanding user information needs.
2. The pivot words and queries in a session can be automatically captured using the attention mechanism without manually selecting pivots.
3. Experiments studies on real-world data show that our model outperforms other baselines on query suggestion.

The reminder of this paper is organized as follows. We review related research studies and compare these work with our approaches in the Section 2. In Section 3, we detail our proposed hierarchical frameworks. Experiments and technical analysis of our models are reported in Section 4. Finally, we conclude this study and highlight the directions of future research work in Section 5.

2 Related work

There have been several studies investigating query suggestion with respect to different search behaviors. Query co-occurrence [11] and term association patterns [12] are common signals used in query suggestion. He et al. [5] proposed a context-aware model called Variable Memory Markov model (QVMM), which builds a suffix tree to model user query sequence. These approaches assume that users share similar search intents with other users who issue similar queries and simply provide query suggestion based on some similarity measures. To provide better query suggestion, click-through behavior [10] is employed along with the issued queries. Liao et al. [13] applied click-through data to build the bipartite graph and clustered queries according to the connections. Jiang et al. [14] exploited query reformulation features to learn users' search behavior and showed the effectiveness for query auto-completion. Different from these methods, we look into users' entire search interaction process by exploiting not only previous queries but also their clicked results, which better satisfy users' search intents.

Other related studies looked into the features resulting from Web search environment. Behaviors such as mouse movements, page scrolls and paginations can also help boost Web search [15, 16]. Zhou et al. [17] aggregated user browsing activities for anchor texts, which improved the performance of Web search. Sun et al. [18] focused on right-click query that is submitted to a search engine by selecting a text string in a Web page and extract the contextual information from the source document to improve search results. These studies show the feasibility of search behavior to improve the performance of Web search.

Joachims et al. [19] applied eye-tracking to analyze users' decision processes in Web search and compared implicit feedback against manual relevance judgments. They concluded that users' clicked documents contained valuable implicit feedback information. Sordoni et al.[2] proposed a hierarchical neural networks for query suggestion but only utilizing users' previous queries. Therefore, we look into the content of users' clicked results and incorporate these results into a hierarchical neural model to mine users' information needs.

3 Models

In this section, we introduce the proposed HAN model for context-aware query suggestion. We first give the problem and notations. Then we present our framework of the HAN model, which consists of two attention layers on word-level and session-level, respectively. Finally, we present the details of different components as well as the training process.

3.1 Problem Definition and notations

We regard query suggestion as a sequential encoding and decoding process. A query session S is considered as a sequence of M queries. For each query $Q_m \in S$, it is followed by a sequence of clicked documents $D_m = \{d_1, ..., d_n\}$ chronologically. Each of query Q_m and documents d_n consists of a sequence of words w. The task of context-aware query suggestion is to predict the next query Q_m given the context $Q_1, ..., Q_{m-1}$ and their clicked documents $D_1, ..., D_{m-1}$. Specifically, we predict the query Q_m by reranking a set of candidate queries based on the predicted ranking scores, which is followed by [2]. V is the size of the vocabulary.

3.2 Hierarchical Attention Networks

User preference on search results reflects user's fine-grained information needs. It is proven to be a useful resource in many IR tasks [20, 18]. To take advantage of user's search behaviors, we propose a hierarchical attention network (HAN), which models the entire search interactions with search engine as shown in Figure 1. Specifically, HAN encodes at query-level and session-level hierarchically to model user's information needs. For query-level encoding, since words contribute unevenly to the representation of query embedding, word-level attention mechanism aims to discover the informative words which can best represent the information needs of the current query. For session-level encoding, due to the noisy query in a session [2], the previous query may not be the best query to reflect user information needs in the whole search process. We adopt the session-level attention mechanism to distinguish the difference of the issued queries. In the following, we will detail how we build the session embedding progressively from word embedding by using the hierarchical structure and utilize it to predict the next query.

Query encoding: For each query $Q_m = \{w_{m,1}, ..., w_{m,N_m}\} \in S$ in a session, the content of the responding clicked documents is represented as an aggregation of words $D_m = \{d_1, ..., d_n\} = \{w^{'}_{m,1}, ..., w^{'}_{m,K_m}\}$, where N_m and K_m are the number of words in the query and the corresponding clicked documents. The clicked documents under the same query are concatenated chronologically to form a pseudo document of length K_m. According to the statistic of our dataset, a query generates 1.46 clicks on average and 45.45% queries are submitted without any interactions. We then adopt the variant recurrent neural network called gated recurrent unit (GRU) [21] to learn word representation. In particular, query and clicked document are encoded by a query GRU and a click GRU, respectively. If there is no click following the query ($K_m = 0$), click encoder is idle. The representation is obtained by summarizing the contextual information as follows:

$$\begin{aligned} h^c_{m,n} &= GRU_c(h^c_{m,n-1}, w^{'}_{m,n}), n = 1, ..., K_m \\ h^q_{m,n} &= GRU_q(h^q_{m,n-1}, w_{m,n}), n = 1, ..., N_m \end{aligned} \tag{1}$$

where $h^c_{m,n} \in \mathbb{R}^{d_h}$ and $h^q_{m,n} \in \mathbb{R}^{d_h}$ are the output recurrent state of click GRU and query GRU, respectively. Click GRU is cascaded to query GRU as in [22], i.e., the initial recurrent state $h^c_{m,0} = 0$ and $h^q_{m,0} = h^c_{m,K_m}$. Each recurrent state stores the order-sensitive information to that position. Then we introduce a word-level attention mechanism to extract the informative words that are important to express the information needs of current query.

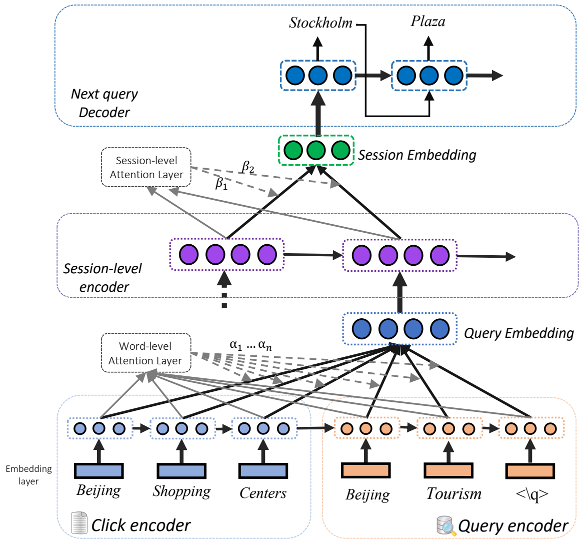

Fig. 1: The architecture of hierarchical attention network for query suggestion. (HAN)

Word-level attention: A document is much longer than a query, and many words in it are not useful to inferring user information needs. An example is shown in Figure 1: the word “ *Shopping*” is the most important word from the clicked document to understand the information need behind the current query “*Beijing Tourism*”. Thus if we can suggest the next query “*Beijing Plaza*”, which may better express what the user is interested in, thus a possible next query that the user will use. These informative words from the clicked documents are aggregated in the representation of current query embedding as follows:

$$\begin{aligned} l_{m,t} &= sigmoid(W_w h_{m,t} + b_w) \\ \alpha_{m,t} &= \frac{exp(l_{m,t})}{\sum_j exp(l_{m,j})} \\ q_m &= \sum_t \alpha_{m,t} h_{m,t} \end{aligned} \tag{2}$$

where click recurrent state $\boldsymbol{h}_m^c \in \mathbb{R}^{K_m \times d_h}$ and query recurrent state $\boldsymbol{h}_m^q \in \mathbb{R}^{N_m \times d_h}$ are concatenated into $\boldsymbol{h}_m = [\boldsymbol{h}_m^c, \boldsymbol{h}_m^q] \in \mathbb{R}^{(K_m+N_m) \times d_h}$. The attention weights are estimated by feeding the recurrent state $\boldsymbol{h}_m$ into a one-layer network to get $l_{m,t}$ and further normalized through a softmax function. Finally, we calculate the query embedding as a weighted sum of each click and query recurrent state and use this vector to represent the information needs of current query behavior. The attention layer is learned end-to-end and gradually assign more attention to reliable and informative words.

Session-level encoding: Inferring user information needs requires us to consider the whole search behaviors in a session. The previously issued queries and clicked results are both useful to infer user search intents. Therefore, we adopt a session-level GRU, which encodes the previous query embedding to current position:

$$h^s_m = GRU_s(h^s_m, q_m), m = 1, ..., M \tag{3}$$

Each recurrent state $h^s_m \in \mathbb{R}^{d_s}$ incorporates the information of both its surrounding context queries and itself. Then, we apply a session-level attention mechanism to select important queries in the context.

Session-level attention: It has been found that a query session may contain queries that are not strongly related to the user's information need, which is called noisy queries (e.g. a user may wonder around in a session) [2]. Therefore, it is necessary to use the previous queries selectively. The importance of queries should be measured correctly in order to better understand user information needs in a session. To do this, we adopt the session-level attention mechanism, which assigns to each query embedding with an attention value.

$$\begin{aligned} g_{m,t} &= \mathit{sigmoid}(W_s h^s_{m,t} + b_s) \\ \beta_{m,t} &= \frac{\mathit{exp}(g_{m,t})}{\sum_j \mathit{exp}(g_{m,j})} \\ s_m &= \sum_t \beta_{m,t} h^s_{m,t} \end{aligned} \tag{4}$$

Similarly, by combining session recurrent state $\boldsymbol{h}^s_m$ with a one-layer network and normalizing them by a softmax function, we obtain an attention vector β and further an aggregated session embedding $s_m \in \mathbb{R}^{d_s}$, which summarizes the context queries.

Decoding: As shown in Figure 1, the next query decoder decodes the session embedding s_m to produce the next candidate query. First, the session embedding s_m is transformed to the initial state of the decoder:

$$d_{m,0} = tanh(Ds_m + b) \tag{5}$$

where $|D| = |d_{m,0}| \times |s_m|$ is a projection matrix and b is the bias. The words of next query are decoded by another GRU:

$$d_m = GRU_{dec}(d_{m,n-1}, w_n), n = 1, ..., N_{m+1} \tag{6}$$

and the probability of the next word is:

$$\begin{aligned} p(w_n|w_{1:n-1}, S) &= \mathit{softmax}(w_n f(d_{m,n-1}, w_{n-1})), \\ f(d_{m,n-1}, w_{n-1}) &= Hd_{m,n-1} + Ew_{n-1} + b_o \end{aligned} \tag{7}$$

where f is a dense layer with parameters H,E and b_o. The softmax layer loops over the vocabulary size V to find next possible word.

In our experiments, we construct a candidate query set for each session and rerank these queries instead of generating the next query directly. The score of a candidate query Q is the probability of being decoded given the session context S:

$$s(Q) = \sum_n \log p(w_n|w_{1:n-1}, S) \tag{8}$$

Table 1: Statistics of the dataset in our experiments.

Dataset	Background	Train	Valid	Test
# Sessions	13,877,582	6,938,508	1,734,596	1,730,773
# Queries	24,572,310	12,286,783	3,070,747	3,062,958

We utilize this score as an additional feature to combine with a learning-to-rank algorithm to evaluate the reranking performance. It will also be compared with the scores from Sordoni et al [2], which only models previously issued queries.

This part is similar to the approach of Sordoni et al [2], except that s_m is enriched with clicked feedback and two levels attention weighting. We will see in our experiments that these additions help to produce better suggestions.

Model training: Our model is trained end-to-end by using the whole sessions in the query log. Given the issued queries $Q_{1:M}$ and corresponding sets of clicked documents $D_{1:M}$ in a session, each query Q_m is treated as the ground truth based on the context $Q_{1:m-1}$ and $D_{1:m-1}$. The training is conducted by maximizing the log-likelihood of a session S:

$$\mathcal{L}(S) = \sum_{m}^{M} \sum_{w_n \in Q_m} \log p(w_n | w_{1:n-1}, S) \tag{9}$$

4 Experiments

In this section, we empirically evaluate the performance of the proposed HAN model.

4.1 Dataset

We conduct experiments on the query log from a popular commercial search engine, each entry of which is made up of user ID, issued query, document titles of clicked URLs, and timestamps. Since we are not able to obtain the body text of each document, we only consider the title as the document content. Publicly available query log, e.g., AOL [2], do not contain the detailed content thus are not suitable for our experiment. Queries are split into sessions based on 30 minutes gap. We shuffle and split them into background, training, validation, and test set with the ratio of 8:4:1:1. The detailed statistics of the dataset is listed in Table 1.

The background set is used to train our model and generate baseline features for a learning-to-rank framework, which follows the prior work [2]. The ranker with only the baseline features is considered as a **Base ranker**. The candidate queries to be reranked are the top 20 most frequent queries based on the co-occurrence with the input query sequence in the background set. A ranking by co-occurrence frequency turned out to be a strong baseline [2]. We call this method the Most Popular Suggestions (**MPS**). Finally, the likelihood score derived from our model is used as an additional feature to produce a new ranker. The likelihood score derived from the baseline **HRED** [2], which only models the previously issued queries, is also used as an additional feature and compared with our model.

We compare the effectiveness of this ranker with other rankers over the training, validation and test set. In testing, we take the last query Q_M and the prior context $Q_{1:M-1}$ and $D_{1:M-1}$ as the ground truth and inputs, respectively. The metric to evaluate the performance of query suggestion is Mean Reciprocal Rank (MRR).

Table 2: The MRR performance of the method. * indicates the statistical significant improvements over each of the baselines.

# Model	MRR@3	MRR@5	MRR@20
1 MPS	0.5475	0.5678	0.5893
2 Base Ranker	0.5831	0.6077	0.6265
3 + HRED	0.5913	0.6175	0.6349
4 + HAN	**0.6042***	**0.6289***	**0.6475***

4.2 Experiment setup

To make a fair comparison of our model with the baseline HRED [2], we use the same parameters as HRED for the common RNN architectures. The dimensionality of query encoder, click encoder, session-level encoder and decoder are set at 300, 300, 600 and 300, respectively. The most frequent $90K$ words in the background set form our vocabulary V. The word embedding, with a dimensionality of 256, is randomly initialized by a standard Gaussian distribution and is trainable during the training. We apply $Adam$ to optimize the parameters with mini-batch size 40. The gradients are normalized if their norm exceeds a threshold 1 to stabilize the learning. Early stopping on the validation set is performed during the training process.

LambdaMART is employed as our learning-to-rank algorithm, which is a state-of-the-art supervised ranker that won the Yahoo! Learning to Rank Challenge (2010) [23]. We used the default setting for LambdaMART's prior parameters and the parameters are learned using standard separate training and validation set. The details of baseline features (17 in total) used to train the baseline ranker are listed as follows:

1. Features that describe each suggestion: the suggestion frequency in the background set and the length of the suggestion in terms of number of words and characters.
2. Features that describe the anchor query: frequency of the anchor query in the background set, the times that the suggestion follows the anchor query in the background set and the Levenshtein distance between the anchor query and the suggestion.
3. Features that describe the whole session: character n-gram similarity between the suggestion and 10 most recent queries in the context, the average Levenshtein distance between the suggestion and each query in the context and the estimated scores using the context-aware Query Variable Markov Model (QVMM) [5].

4.3 Overall Accuracy

Table 2 shows the overall performance of our model and the baselines. In addition to MPS and Base Ranker, we also compare our model with HRED [2], which is similar to ours, except that our session embedding is enriched by user click feedback and two levels of attention weighting. It is observed that the proposed HAN model consistently achieves the best performance on MRR at top 3, 5 and 20.

The improvement due to the addition of a hierarchical encoder-decoder can be seen by comparing HRED and HAN to the base ranker. This result is consistent with [Sordoni et al. 2015]. The comparison between HAN and HRED is particularly interesting. The difference between them is due to the utilization of the content of clicked document and to the attention mechanism. We can see that these elements contributed in improving the suggestions. According to our statistics, about 7.4% queries contain words in

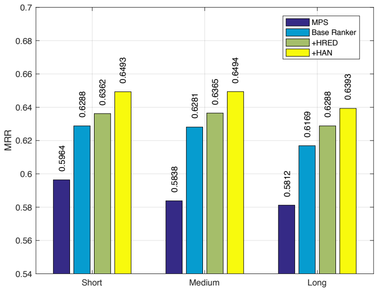

Fig. 2: Performance on sessions with different lengths.

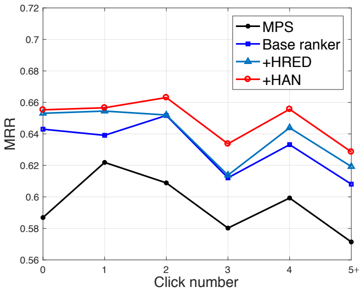

Fig. 3: Performance on sessions with different click frequency.

previous clicked documents but not in previous queries. Globally, query suggestion incorporating clicked documents is more effective and precise.

In addition, our model can automatically capture the pivot information with hierarchical attention mechanism. We observe that most of the words in clicked documents are duplicate or even useless to infer user information needs. Attention mechanism enables us to distinguish these words and assign higher weights to those contributing to the next query. Similarly, the last query is not necessarily the most important nor related query to user information needs. Our model is able to capture the important queries and predict the next query. We will show a detailed case study in Section 4.6.

4.4 Effectiveness of session length

In order to investigate the effect of session length on our context-aware model, we separate the test set into three categories (The proportion is reported in the brackets):

1. **Short**: Sessions with only 2 queries. (47%)
2. **Medium**: Sessions with 3-4 queries. (34%)
3. **Long**: Sessions with more than 4 queries. (17%)

In Figure 2, we report the results for different session lengths. We observe that the proposed HAN model outperforms the baselines over different session lengths. As the session becomes longer, MPS performs worse. It is because context information becomes more important, frequency-based method suffers from the noise and sparse signals on longer sessions. Except for MPS, when a model is used for short and medium sessions, its performance is similar. However, we generally observe a lower performance on long sessions. We explain the decrease in performance on long sessions by the fact that the sessions contain more noise. Indeed, in a long session, the searches of the user may be less focused on a specific information need, and queries about unrelated topics could appear. This will create additional difficulties for any methods. The difference in session length does not affect the comparison of our model with the others: In all the three groups of sessions, our model outperforms the others in a similar way.

Table 3: Examples of HAN's query suggestions. The bold words reflect the information needs behind the queries. The purple color represents the session-level attention while the blue color represents the word-level attention. Deeper color implies larger attention weights.

Context 1	*Context* 2
$\mathbf{Q}_1$: Food helps to loss weight	$\mathbf{Q}_1$: How to recover sight?
$C_{1,1}$: Diet to lose weight	$C_{1,1}$: LASIK surgery
$\mathbf{Q}_2$: Chinese medical cosmetology	$\mathbf{Q}_2$: Bad Sight
$C_{2,1}$: Chinese medical cosmetology **hospital**	$\mathbf{Q}_3$: LASIK surgery
	$C_{3,1}$: **HongKong** LASIK hospital
Suggested queries	
1: Plastic surgery **hospital**	1. **HongKong** LASIK surgery
2: Beauty Health	2. **HongKong** LASIK surgery hospital
3: Skin Beauty	3. What if LASIK surgery fails

4.5 Effectiveness of click feedback

This experiment further evaluates the effectiveness of click feedback to our model. We split the test set into six categories according to the click frequency in a session, i.e., 0 to 4 and more than 5. Their proportion are 12%,10%,16%,13%,14% and 35%, respectively. The result is shown in Figure 3.

In Figure 3, we observe that a ranker outperforms MPS on all different click frequencies. HRED still outperforms base ranker, but the differences are marginal on frequencies 2 and 3. The proposed HAN model outperforms the base ranker with a quite large margin on all the frequencies. Compared to HRED, HAN produces only marginal improvements when there are limited click information (frequencies 0 and 1). From frequency 2, we can see larger differences between them. This indicates that HAN can benefit more when more click information is available. However, when query sessions become very long, HAN also faces more difficulties to determining what could be the next query due to the problem of noise we discussed in Section 4.4. More research is required to infer the topic of the next query from a noisy history.

4.6 Case study

To better illustrate the effectiveness of our model, we provide two example sessions in Table 3. Based on the context, we predict the next query using standard word-level decoding techniques such as beam-search [24]. This method is able to obtain a predetermined number of best partial suggestions. We list top 3 suggested queries in Table 3.

It is observed that by incorporating the clicked documents, our model can better understand the information needs behind the queries. In $Context$ 1, we can only understand that this user is going to learn about beauty by the issued queries. However, the responding clicked document reflects more detailed information that this user is probably looking for a cosmetology hospital. The suggested queries from our model cover this potential information needs and are more likely to be clicked. The second example shows that the clicked document helps our model to understand that the user is interested in LASIK surgery in Hong Kong.

Looking at the attentions paid to queries and words, we can see that in general, those corresponding to the key concepts in the session are captured with more attention. For example, in the second session, the user is likely looking for a place for LASIK surgery, rather than general information explaining bad sight. So Q2 captures less intention than Q1 and Q3. The generated query suggestions reflect this. We can also see that even if any query in the session can obtain some attention weight, in general, the latest query tends to have a higher weight, which is intuitive: the search intent in a session can evolve and the last query better reflects the current intent than an early one.

The above observations show that the attention mechanisms on queries and words can successfully capture the most important elements. However, as we explained earlier, the mechanisms can be fooled by the noise queries in a session, especially when it is long. More investigations are required to detect the true intent of the user. Our model is able to figure out these queries and assign them with lower attention weights. As Q2 in $Context$ 2, bad sight is short and vague to infer the information needs. It is not helpful for search intent modeling and thus assigned with a low attention weight by our model.

5 Conclusion

In this paper, we propose a hierarchical attention network (HAN) to explicitly model user search behavior by using not only the issued queries but also the content of clicked documents. HAN encodes queries and clicked documents with two recurrent neural networks and produces a context-aware session embedding hierarchically. The predicted next query is therefore expected to better reflect information needs in the suggestions.

An essential problem of incorporating clicked documents lies in how to select the pivot and informative words. Similarly, identifying strongly relevant queries in a session is another important challenge to infer user information needs. To address these problems, two levels of attention mechanisms are employed to automatically capture the differences without manually selecting pivots. Experiments conducted on a large-scale commercial query log demonstrate the effectiveness of our model. Compared to the model that only uses the issued queries (HRED), our model obtains better performance due to the utilization of click feedback and the attention mechanism.

For future work, we plan to integrate query suggestion to the existing ranking models. By actively rewriting the issued queries with our query suggestion model, the ranking of search results may produce better search results. Another important issue we will investigate is how to better determine the pivot words from the most relevant queries in a noisy session. A more sophisticated topical similarity measure could be integrated in the attention mechanism.

6 Acknowledgements

This work is supported by Natural Science Foundation of China (Grant No. 61622208, 61732008, 61532011) and National Key Basic Research Program (2015CB358700).

References

1. Byron J Gao, David C Anastasiu, and Xing Jiang. Utilizing user-input contextual terms for query disambiguation. In *ACL*, 2010.

2. Alessandro Sordoni, Yoshua Bengio, Hossein Vahabi, Christina Lioma, Jakob Grue Simonsen, and Jian-Yun Nie. A hierarchical recurrent encoder-decoder for generative context-aware query suggestion. In *CIKM*, 2015.
3. Ruihua Song, Zhenxiao Luo, Jian-Yun Nie, Yong Yu, and Hsiao-Wuen Hon. Identification of ambiguous queries in web search. *IPM*, 2009.
4. Yang Song, Dengyong Zhou, and Li-wei He. Post-ranking query suggestion by diversifying search results. In *SIGIR*, 2011.
5. Qi He, Daxin Jiang, Zhen Liao, Steven CH Hoi, Kuiyu Chang, Ee-Peng Lim, and Hang Li. Web query recommendation via sequential query prediction. In *ICDE,09*.
6. Van Dang and Bruce W Croft. Query reformulation using anchor text. In *WSDM*, 2010.
7. Zhipeng Huang, Bogdan Cautis, Reynold Cheng, and Yudian Zheng. Kb-enabled query recommendation for long-tail queries. In *CIKM*, 2016.
8. Liangda Li, Hongbo Deng, Anlei Dong, Yi Chang, Ricardo Baeza-Yates, and Hongyuan Zha. Exploring query auto-completion and click logs for contextual-aware web search and query suggestion. In *WWW*, 2017.
9. Wanyu Chen, Fei Cai, Honghui Chen, and Maarten de Rijke. Personalized ery suggestion diversification. In *SIGIR*, 2017.
10. Yiqun Liu, Junwei Miao, Min Zhang, Shaoping Ma, and Liyun Ru. How do users describe their information need: Query recommendation based on snippet click model. *Expert Systems with Applications*, 38(11):13847–13856, 2011.
11. Chien-Kang Huang, Lee-Feng Chien, and Yen-Jen Oyang. Relevant term suggestion in interactive web search based on contextual information in query session logs. *JAIST*, 2003.
12. Xuanhui Wang and ChengXiang Zhai. Mining term association patterns from search logs for effective query reformulation. In *CIKM*, 2008.
13. Zhen Liao, Daxin Jiang, Enhong Chen, Jian Pei, Huanhuan Cao, and Hang Li. Mining concept sequences from large-scale search logs for context-aware query suggestion. *ACM Transactions on Intelligent Systems and Technology*, 2011.
14. Jyun-Yu Jiang, Yen-Yu Ke, Pao-Yu Chien, and Pu-Jen Cheng. Learning user reformulation behavior for query auto-completion. In *SIGIR*, 2014.
15. Fernando Diaz, Ryen White, Georg Buscher, and Dan Liebling. Robust models of mouse movement on dynamic web search results pages. In *CIKM*, 2013.
16. Yiqun Liu, Chao Wang, Ke Zhou, Jianyun Nie, Min Zhang, and Shaoping Ma. From skimming to reading: A two-stage examination model for web search. In *CIKM*, 2014.
17. Bo Zhou, Yiqun Liu, Min Zhang, Yijiang Jin, and Shaoping Ma. Incorporating web browsing activities into anchor texts for web search. *IR*, 2011.
18. Aixin Sun and Chii-Hian Lou. Towards context-aware search with right click. In *SIGIR*, 2014.
19. Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, Filip Radlinski, and Geri Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *TOIS*, 2007.
20. Olivier Chapelle and Ya Zhang. A dynamic bayesian network click model for web search ranking. In *WWW*, pages 1–10, 2009.
21. Junyoung Chung, Caglar Gulcehre, Kyung Hyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *Eprint Arxiv*, 2014.
22. Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiský, and Phil Blunsom. Reasoning about entailment with neural attention. *ICLR*, 2015.
23. Qiang Wu, Christopher JC Burges, Krysta M Svore, and Jianfeng Gao. Adapting boosting for information retrieval measures. *Information Retrieval*, 2010.
24. Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*, 2014.