Front.Comput.Sci.
DOI

RESEARCH ARTICLE

Understanding the Role of Human-Inspired Heuristics for Retrieval Models

Xiangsheng Li, Yiqun Liu (✉), Jiaxin Mao [1]

1 Department of Computer Science and Technology, Institute for Artificial Intelligence,
Beijing National Research Center for Information Science and Technology,
Tsinghua University, Beijing 100084, China

Abstract Relevance estimation is one of the core concerns of information retrieval (IR) studies. Although existing retrieval models gained much success in both deepening our understanding of information seeking behavior and building effective retrieval systems, we have to admit that the models work in a rather different manner from how humans make relevance judgments. Users' information seeking behaviors involve complex cognitive processes, however, the majority of these behavior patterns are not considered in existing retrieval models. To bridge the gap between practical user behavior and retrieval model, it is essential to systematically investigate user cognitive behavior during relevance judgement and incorporate these heuristics into retrieval models. In this paper, we aim to formally define a set of basic user reading heuristics during relevance judgement and investigate their corresponding modeling strategies in retrieval models. Further experiments are conducted to evaluate the effectiveness of different reading heuristics for improving ranking performance. Based on a large-scale Web search dataset, we find that most reading heuristics can improve the performance of retrieval model and establish guidelines for improving the design of retrieval models with human-inspired heuristics. Our study sheds light on building retrieval model from the perspective of cognitive behavior.

This article is an extension of Li et al. [1]. Compared with the previous conference version, it systematically introduces the reading heuristics for retrieval model. It also includes an extensive study of modeling strategies and experimental results to evaluate different reading heuristics.

E-mail: yiqunliu@tsinghua.edu.cn

Keywords Reading heuristics; Retrieval model; Cognitive behavior;

1 Introduction

Retrieval models lie at the heart of IR system designs. The key idea is to learn representations to model the interactions between query and document based on the inspiration of human cognitive behavior. Therefore, understanding user behavior in relevance judgment is important and necessary for the designing of better retrieval models. Many empirical studies [1, 2] show that good retrieval performance is closely related to the actual user behavior, which implies the possibility of improving retrieval model by exploiting the heuristics from human behavior.

However, existing retrieval models mainly focus on the matching signals between query and document while ignore the heuristics that are inherent in users' relevance judgement behaviors. For example, representation-based models [3, 4] simply encode query and document information into representation vectors and ignore fine-grained information (e.g., passage or sentence-level relevance). These models violate the reading pattern that users' reading attention is not uniformly distribution in a document [5]. Interaction-based models [2, 6] make a strong assumption that sentences in a document are independent of each other, which is inconsistent with users' sequential reading behavior [5]. To build better retrieval models, we argue that it is important to incorporate human-inspired heuristics into the design of retrieval models.

In this paper, we extend our previous work which evaluated the effectiveness of six human-inspired heuristics in retrieval models [1]. Specifically, we conduct a deeper investigation on these heuristics from two aspects. First, we formalize these heuristics from a user behavior dataset and give the intrinsic explanations of how they relate to the design of retrieval models. Second, we discuss different modeling strategies of these heuristics and evaluate their effectiveness for retrieval models on a large-scale Web search dataset. More importantly, by comparing different modeling strategies, we propose specific suggestions for improving the design of retrieval models with different reading heuristics, which reveals new insights on building retrieval model from the perspective of cognitive behavior.

The rest of the paper is organized as follows. The next section discusses related work. Section 3 formalizes the reading heuristics and explain how they relate to the design of retrieval models. Different modeling strategies of the reading heuristics are presented in Section 4. Experimental results and detailed analysis are discussed in Section 5. Section 6 concludes the paper.

2 Related work

2.1 Retrieval model

A variety of retrieval models have been proposed in IR, which can be categorized into two classes: probabilistic models and deep neural models. Probabilistic models (e.g., BM25 [7]) mainly focus on query frequency in a document and ignore semantically relevant information. Deep neural models are categorized into two classes: representation-based models [3] and interaction-based models [6, 8, 9]. Representation-based models aim to learn good representation of query and document while interaction-based models aims to build local interactions between query and document, and then aggregate each interaction to learn a complex pattern for relevance. However, these models violates many users' reading patterns. For example, most of models considers the document as a whole and gives the same weights to each position, which is inconsistent with the finding that users have non-uniform reading attention in a document [5]. To build better retrieval model, it is necessary and important to consider more reading heuritics from user's actual behaviors.

2.2 Cognitive-oriented model

Several studies have shown that cognitive heuristics can effectively guide and improve the computational model in different IR fields. Ding et al [10] proposed a cognitive graph based model in question answering based on the findings of information seeking process. Considering users' varying reading attention in different contexts, Adams et al. [11] and Fu et al. [12] proposed classification models to simulate user cognitive process. However, existing retrieval models still only focus on a few simple assumptions of user behavior and so far there is no research on the relation between users' cognitive behavior and retrieval models. Our study systematically analyzes how different reading heuristics contribute to the improvement of ranking performance and provides novel insights on building retrieval models from the perspective of cognitive behavior.

3 Formal reading heuristics

In this section, we formalize the reading heuristics derived from users' behavior dataset [5] which is a publicly available user study dataset to investigate users' reading attention during relevance judgement. Specifically, they recorded the eye movement of 29 users when making relevance judgment for 60 documents and elaborated the process of users' relevance judgement. Based on the discovered reading patterns in this dataset, we formally define six intuitive and desirable heuristics that are potentially important for the design of retrieval models, which involves *Sequential reading*, *Vertical decaying attention*, *Query centric guidance*, *Context-aware reading*, *Selective attention* and *Early stop reading*.

3.1 Sequential reading

Based on the average first arrival time of each vertical position, Li et al. [5] found that users' reading direction is generally from the top position to the bottom of a document, which implies that the presented order of document content may effect users' perceived relevance. So we define the *Sequential reading* heuristics:

Definition 1. *Let A, B be two pieces of text in a document. If* $d_1 = \{A, B\}$, $d_2 = \{B, A\}$, *then* $rel(d_1) \neq rel(d_2)$.

This heuristic suggests that it is better for retrieval models to split the whole document into several short passages [13] and estimate the relevance on each query-passage pair in

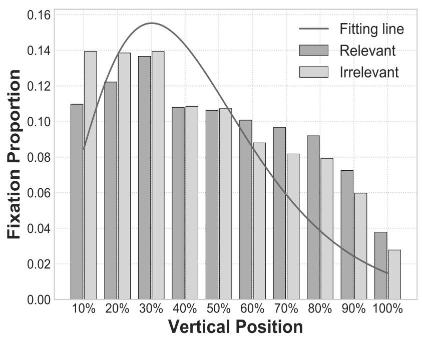


Fig. 1: Users' fixation proportion at each vertical position and its fitting line.

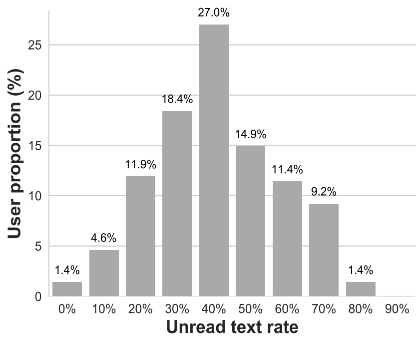


Fig. 2: User proportion in the documents with different unread text rate.

terms of the original order. Specifically, we should model the presented order of content by specific strategies.

3.2 Vertical decaying attention

Previous retrieval models [8, 9, 14] assume that users have uniform attention when reading a document. From Figure 1, we can observe that users' reading attention is vertically decaying in a document. It motivates us that the content in the beginning of a document plays a more important role compared with other text, as defined in follows:

Definition 2. *Let A, B be two pieces of text in a document. If $d_1 = \{A, B\}$, $d_2 = \{B, A\}$ and $rel(A) > rel(B)$, then $rel(d_1) > rel(d_2)$.*

This heuristic suggests that retrieval models should assign more weights to the text at the beginning of a document.

3.3 Query centric guidance

Users' reading attention is generally influenced by the search intent, which can be reflected by the issued query words explicitly. Previous retrieval models follow this heuristic, and therefore, modeling the interactions between query and document plays an important role in determining the estimated relevance. In particular, Fang et al. [15] proposed several heuristic retrieval constrains that retrieval models should satisfy. These constrains cover the important properties when modeling the interaction between query and document. Liang et al. [13] further improved these heuristics by extending one constrain to consider semantic matching signals. Fan et al. [6] concluded that three important signals should be captured for the query centric guidance, including exact and semantic query matching [16], proximity [17], and term importance [13]. We further discuss how to model these three signals in Section 4.

3.4 Context-aware reading

Users' reading behavior is an sequential process, in which the perceived relevance is influenced by the previously read content. It is observed that users have different reading behaviors (e.g., reading speed, reading attention) after they have different relevance judgment during the reading process [5]. However, existing retrieval models only simply assume that each piece of text in a document is independent of each other. So we propose the context-aware reading heuristic:

Definition 3. *Let A, B be two pieces of text in a document and R is the relevance score, then $err(rel(A) + rel(B|A), R) < err(rel(A) + rel(B), R)$. err is the error function which models the difference between the estimated relevance and ground truth relevance.*

This heuristic implies that retrieval models can better estimate relevance if they can leverage the contextual information. Specifically, if a retrieval model estimates relevance contextually, i.e., $R' = rel(A) + rel(B|A)$, then it is more approximate to the ground truth relevance R compared with independently estimated relevance , i.e., $R' = rel(A) + rel(B)$.

3.5 Selective attention

It is found that there exists a tradeoff between the precision of language understanding (encoding the input accurately) and economy of attention (fixating as few words as possible) [18]. This mental phenomena motivates users to instinctively select important text to read and skip seemingly irrelevant information. We calculate user proportion of different unread text rate in Figure 2. Most of users do not read the full document but only read about 40% of a document. It suggests that retrieval models can ignore the text that has no or little influence on relevance, which is formalized in Definition 4.

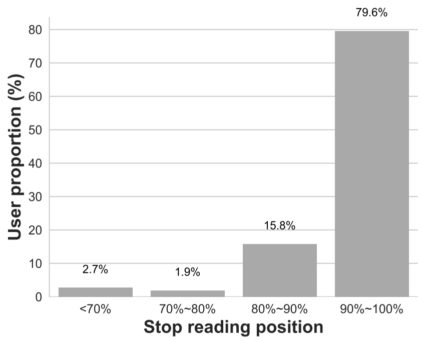


Fig. 3: User proportion in different stop reading positions.

Definition 4. *Let s_i be a sentence in a document $d = \{s_1, s_2, \ldots, s_n\}$, then there exists another document d' which contains part of original sentences, i.e., $d' \subsetneq d$, such that rel(d) = rel(d').*

3.6 Early stop reading

This reading heuristic is similar to *selective attention*, caused by the precision of language understanding and attention effort as well. Once users have a clear understanding of a document, they may stop reading at the current position and ignore the rest part. From the statistic of users' stop reading positions in Figure 3, most of users tend to stop reading nearly at the end of a document and read almost the full document. So we formalize the early stop reading heuristic:

Definition 5. *Let d be a document with several sentences $d = \{s_1, s_2, \ldots, s_n\}$, then there exists a position $k < n$ such that a documentw $d' = \{s_1, s_2, \ldots, s_k\}$ and rel(d) = rel(d').*

The majority stop positions are close the end of a document, which seems to violate the assumption in many related works [19, 20]. Therefore, we will further investigate its effectiveness in the experiment.

4 Reading heuristics modeling in retrieval model

In this section, we will introduce how to model the proposed reading heuristics in the design of retrieval models.

4.1 Overview

In psychology, dual process theory [21] provides an insight that the information seeking process, which consists of an implicit information association process and an explicit reasoning process. This theory implies that information seeking process is conducted by obtaining potentially useful information and relevance reasoning with accumulated knowledge. Following this theory, we divide the information seeking process into two steps: 1) fine-grained information association, and 2) knowledge accumulation. Based on this two-step process, we formalize user reading process during relevance judgement as:

$$\begin{aligned} \mathcal{M}_i &= \mathcal{F}(\mathcal{S}(s_i, p_i|\mathcal{K}_{i-1}), Q) \\ \mathcal{K}_i &= \mathcal{G}(\mathcal{M}_i|\mathcal{K}_{i-1}), i = 1, \ldots, n \end{aligned} \tag{1}$$

where s_i the minimal text unit in a document (e.g., sentence or passage) and p_i is the position of s_i. In this work, we only focus on sentence level. $\mathcal{K}_i$ is the accumulated knowledge up to position i. $\mathcal{S}$ is the select function which controls if s_i should be considered or ignored. $\mathcal{S}(s_i, p_i|\mathcal{K}_{i-1}) = s_i$ if s_i is being considered otherwise it is Φ, which means empty information. $\mathcal{F}$ is the local relevance estimation function which extracts semantic information of s_i and estimate the local relevance between s_i and query Q. $\mathcal{M}_i$ is the interacted semantic information between current text unit s_i and query Q, which contains both obtained knowledge and relevance confidence. $\mathcal{G}$ is the aggregation function which accumulates the current read information. Equation 1 describes the key components of modeling users' reading and relevance judgement process, in which the workflow is shown in Figure 4.

Based on this framework, we can incorporate the user reading heuristics into retrieval model. Note that we can model each reading heuristic with different strategies. In this section, we introduce different methods to incorporate the reading heuristics and implement the corresponding retrieval models defined by Equation 1.

4.2 Sequential reading (p_i)

Sequential reading heuristic defines the reading direction is from top to bottom. Compared with previous retrieval models, this heuristics indicates that the presented order of content may affect users' relevance perception, which motivates us to model the position of each reading content, i.e., p_i. We first compare sequential order with other perturbed orders to illustrate the necessity of content order and then discuss other better position modeling methods.

4.2.1 Sequential order vs perturbed order

Modeling p_i enables retrieval models to consider the presented order of sentences. For sequential modeling, the content of document is split into different fined-grained units

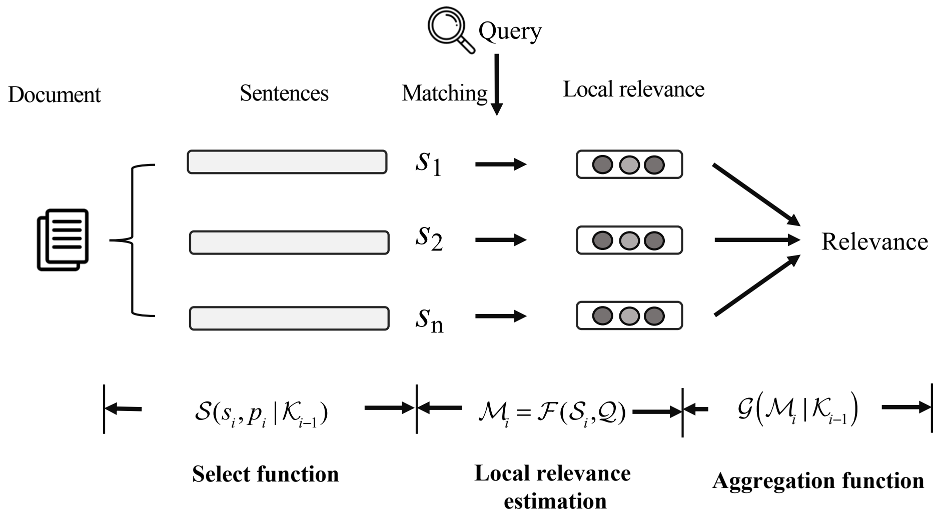


Fig. 4: Overall Framework of modeling reading heuristics.

(e.g., sentence) and we use $p_i \in 1, 2, ..., n$ to denote the position of each unit. According to Equation 1, the input sentence s_i and its position p_i are sequentially fed to the retrieval model for capturing the order information. For perturbed orders, we consider both the inverse order and random order rather than the original sequential order.

4.2.2 Sequential order vs position embedding

To better model the presented order of sentences, we also add position embeddings on the feature level. It provides explicit information about which portion of the sequence is currently processed by the model [22]. We exploit an embedding which encodes the position of the presented order to model p_i in Equation 1. The position embedding vectors are concatenated with text embeddings, i.e., $\mathbf{e} = (s_1 \oplus p_1, \dots, s_n \oplus p_n)$. Specifically, we attempt to exploit two categories of position embedding what consider absolute and relative order relationships, respectively. For absolute position, the general position embedding of a sentence is directly modeled as an indexing function to the embedding space

$$f(p_i) = Emb_{pos}(p_i) \in \mathbb{R}^D \tag{2}$$

where D is the dimension of position embedding. Adjacent positions are not guaranteed to be close in their specific embedding space. To model the inner sequential and adjacent relationships of position, we also apply relative position embedding as in [23]. Instead of an independent vector, the position embedding is defined as a continuous function over the position. The formula for positional embedding is as follows:

$$\begin{aligned} Emb_{pos}(p_i, 2d) &= sin(\frac{p_i}{10000^{2d/D}}) \\ Emb_{pos}(p_i, 2d+1) &= cos(\frac{p_i}{10000^{2d/D}}) \end{aligned} \tag{3}$$

where d is the dimension index. Compared with absolute position embedding, relative position embedding shift smoothly with incremental positions and correlate with each other in the embedding space.

4.3 Vertical decaying attention ($\mathcal{G}$)

Vertical decaying attention suggests that users' reading attention is decaying vertically in a document. Compared with previous retrieval models, this heuristics motivates us to focus on the beginning of a document, i.e., to let the aggregation function $\mathcal{G}$ assign more weights to the initial content of a document. We try two strategies to model users' vertical decaying attention in retrieval models.

4.3.1 Vertical decaying coefficient

This strategy directly multiply each sentence embedding with a decaying coefficient. The coefficient is generated from users' general distribution of reading attention. We utilize a Gamma distribution to fit users' general fixation distribution, i.e., the line in Figure 1:

$$\alpha(v) = \frac{(v-l)^{k-1}}{\Gamma(k) \cdot \theta^k} \exp(-\frac{v-l}{\theta}) \tag{4}$$

where v is the vertical position in a document, l, k, θ is the location parameter, shape parameter and scale parameter, respectively. After fitting the data, we have $l = 1.36, k = 4.37$

and $\theta = 1.36$. The aggregation function $\mathcal{G}$ is then defined as the product of the semantic information of sentence $\mathcal{M}_i$ and their decaying coefficient $\alpha(v)$.

$$\mathcal{G}(\mathcal{M}_i|\mathcal{K}_{i-1}) = \mathcal{M}_i \cdot \alpha(v(\mathcal{M}_i)) \tag{5}$$

To exploit this strategy, we need to measure users' reading attention and use it to get an estimated decaying distribution. The fitted distribution is then used to control the weight of each sentence's semantic information.

4.3.2 Unsupervised decaying Attention learning

Attention mechanism [23] can be broadly interpreted as a vector of importance weights, which are learned automatically during the optimization process. Motivated by the reading heuristic that users' reading attention is vertically decaying, we regularize the attention function to have similar decaying distribution, i.e., $\alpha(v_i) > \alpha(v_j), \forall i < j$. The aggregation function $\mathcal{G}$ is formalized as follows:

$$\begin{aligned} \mathcal{G}(\mathcal{M}_i|\mathcal{K}_{i-1}) &= \mathcal{M}_i \cdot \hat{\alpha}(v_i) \\ \alpha(v_i) &= \tanh(W \cdot \mathcal{M}_i + b) \\ \hat{\alpha}(v_i) &= \frac{\alpha(v_i)}{\sum_i \alpha(v_i)} \end{aligned} \tag{6}$$

In particular, the attention weights are regularized to decease over the vertical positions as follows:

$$\mathcal{L}_{un-decay} = \sum_{i<j} \max(0, \gamma - \hat{\alpha}(v_i) + \hat{\alpha}(v_j)) \tag{7}$$

where γ is the boundary distance between any adjacent pair. The objective function is to let $\alpha(v_i) - \alpha(v_j) > \gamma, \forall i < j$, which forms a decaying distribution as we observed in users' reading behavior. In the training process, Equation 7 is optimized with ranking loss by multiplying an additional trade-off parameter λ, i.e., $\mathcal{L} = \mathcal{L}_{rank} + \lambda \cdot \mathcal{L}_{decay}$.

4.3.3 Supervised decaying Attention learning

The regularization function in Equation 7 only forces the attention weights to have a decaying distribution, however, this distribution is not certainly correlated with the real distribution of users' reading attention. In order to approximate the observed attention distribution from user behavior, we exploit the fitted Gamma distribution in Equation 4 as the supervision signals and minimize the squared error between each attention weight and users' attention value in the specific vertical position:

$$\mathcal{L}_{su-decay} = \sum_i (\hat{\alpha}(v_i) - \overline{\alpha}(v_i))^2 \tag{8}$$

where $\overline{\alpha}(v_i)$ is the general decaying coefficient from user behavior data. Similarly, additional trade-off parameter λ is incorporated to avoid over-fitting, i.e., $\mathcal{L} = \mathcal{L}_{rank} + \lambda \cdot \mathcal{L}_{decay}$. In this method, users' reading attention provides a useful inductive bias on attention weight in retrieval models and inspires the model to work like a real user.

4.4 Query centric guidance ($\mathcal{F}$)

This heuristic aims to perceive the relevance between document text and query. Previous study shows that users' reading attention is significantly higher in the context around the query terms [5], which motivates us to follow IR heuristics in [16] and define how to model the interactions between query and document, i.e., the relevance estimation function $\mathcal{F}$. Specifically, such heuristics include exact query matching and semantic query matching [16], proximity [17] and term importance [13]. To cover these three IR heuristics, existing methods mainly model the relationship between query and document text by an interaction matrix. Specifically, for a given query $\mathbf{q} = [w_1, w_2, ..., w_m]$ and a document $\mathbf{d}$ with T sentences, where each sentence is $\mathbf{s} = [v_1, v_2, ..., v_n]$, exact query matching and semantic query matching are modeled by a semantic matching matrix M^{cos} and an exact matching matrix M^{xor}, respectively.

$$M_{ij}^{cos} = cos(w_i, v_j), \tag{9}$$

$$M_{ij}^{xor} = \begin{cases} 1, & w_i = v_j \\ 0, & otherwise \end{cases} \tag{10}$$

Two matrices provides critical signals for information retrieval as suggested by [13,16,24], which are utilized by most of existing neural retrieval models. Some of retrieval models only utilize semantic matching signals [9, 14] and some of them [2, 6] further extend each element M_{ij} to a three-dimensional representation vector $S_{ij} = [x_i, y_j, M_{ij}]$ by concatenating two term embeddings to capture term importance, where $x_i = w_i * \boldsymbol{W_c}$ and $y_j = v_j * \boldsymbol{W_c}$. $\boldsymbol{W_c}$ is a compressed matrix to be learned during training. The proximity is learned by different modeling strategies on the interaction M (or S). In the following part, we discuss two popular modeling strategies in terms of spatial proximity and semantic proximity.

4.4.1 Spatial proximity modeling

Spatial proximity is the connectivity within a certain range, which aims to model interaction representation with the central and adjacent semantic interactions. It provides effective contextual signals for constructing interaction representation. Classic spatial proximity modeling strategies include multi-level convolutional neural network [25] and spatial recurrent neural network [26]. Multi-level convolutional neural network generate *n*-gram semantic representation and then aggregates into a interaction representation. Spatial recurrent neural network sequentially models the interaction matrix from top left to bottom right, which aims to summarize all the semantic interactions in the last hidden representation. Spatial proximity is exploited by most existing neural retrieval models.

4.4.2 Semantic proximity modeling

Compared with spatial proximity, semantic proximity clusters the different strengths of semantic matching and aims to represent the level of each local interactions' groups. Classic semantic proximity modeling strategies include matching histogram mapping [8] and kernel pooling [9]. Matching histogram mapping is first proposed in Deep Relevance Matching Model(DRMM), which directly takes the count of local interactions in each bin as the histogram value. This method can clearly distinguishes the degree of different matching signals but is highly time-consuming for feature processing. Instead, kernel pooling is an end-to-end modeling method, which exploits RBF kernel to calculate how word pair similarities are distributed around it. The feature extraction process is as follows:

$$
\begin{aligned}
K_k(M_i) &= \sum_j \exp(-\frac{(M_{ij}-\mu_k)^2}{2\sigma_k^2}) \\
\vec{K}(M_i) &= \{K_1(M_i), ..., K_K(M_i)\} \\
\phi(M) &= \sum_i \log \vec{K}(M_i)
\end{aligned}
\tag{11}
$$

where K is the RBF kernel. This method softly counts word matches at different similarity levels and provide soft-TF ranking features.

Finally, we use $\hat{\boldsymbol{s}}_{\boldsymbol{i}}$ to denote the sentence-level interacted semantic state between query $\boldsymbol{q}$ and sentence s_i by using the methods in this section, i.e., $\mathcal{M}_i$ in Equation 1. The matrix in Equation 10 can be transferred into vectors by using a MLP layer.

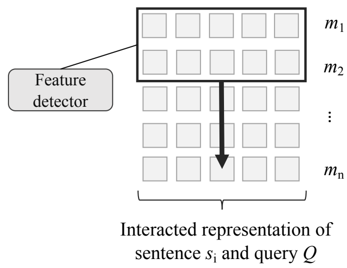


Fig. 5: Framework of sentence-level convolution.

4.5 Context-aware reading ($\mathcal{G}$)

This heuristic suggests that users' local relevance perception is not simply produced in a single sentence but gradually accumulated based on the previous read text. Reviewing the Equation 1, this heuristic focuses on modeling $\mathcal{G}$ such that $\mathcal{G}(\mathcal{M}_i|\mathcal{K}_{i-1}) \propto \mathcal{K}_{i-1}$. Existing context-aware modeling methods include sentence-level convolution neural network and sequential model.

4.5.1 Sentence-level convolution neural network

Sentence-level context modeling aims to generate semantic representation not only based on current sentence but also previous sentences. Similar to *n*-gram in NLP technique, it models contiguous sequence of *n* sentences from a document. In retrieval model, we model the interacted semantic representation between sentence and query, i.e., $\mathcal{M}_i$ in Equation 1. 1-D convolution neural network can model this case well, as shown in Figure 5. For each sliding window, feature detector (or filter) extract semantic information based on a continuous sequence of sentences, which considers contextual information.

4.5.2 Sequential model

A drawback of sentence-level convolution neural network is that it only considers a part of previous sentences at each position. Sequential model such as recurrent neural network can use the internal state (memory) to process all the previous inputted sequences, which is able to consider a large range of contextual information. $\mathcal{G}$ can be context independent if we replace the RNN module with a simple non-linear layer.

4.6 Selective attention ($\mathcal{S}$)

This heuristic indicates that users may skip some seemingly irrelevant text during relevance judgement. In other words, not every sentence is necessary for retrieval model. In Equation 1, we focus on modeling the select function $\mathcal{S}$ which controls if a sentence should be considered or not. Existing retrieval models mainly include two kinds of strategies to draw selectively attention: query centric attention and reinforcement learning.

4.6.1 Query centric attention

Wu et al. [27] proposed a query-centric assumption to define where the relevance occurs. It assumes that the relevant information for a query only locates in the contexts around query terms [8], which motivates the retrieval model to only focus on query-centric context. This attention is fixed and only concentrate on the context with query terms. Exact/semantic matching signals and proximity heuristic are captured in the interaction matrix between query and each query-centric context.

4.6.2 Reinforcement learning

Fixed query-centric attention only focuses on the context with query terms and ignores other sentences, which may lose important message for relevance judgement. To better select important information in a document, we can also exploit reinforcement learning to learn which sentence is more important. The policy action is defined as the sentence selection and it will be sampled to maximize the expected reward during the training process. The decision policy is defined as follows:

$$\pi(a_i^s|\hat{\boldsymbol{s}}_i, \boldsymbol{h}_{i-1}^s, p_i) = \sigma(\boldsymbol{W}_s * [\hat{\boldsymbol{s}}_i, \boldsymbol{h}_{i-1}^s, posEmb(p_i)] + b_s) \tag{12}$$

where $\boldsymbol{h}_i^s$ is the hidden state of accumulated knowledge up to position p_i, i.e., $\mathcal{K}_i$ in Equation 1. $\hat{\boldsymbol{s}}_i$ is the sentence-level interacted semantic state between query $\mathbf{q}$ and sentence s_i by using the methods in Section 4, i.e., $\mathcal{M}_i$ in Equation 1. We then use a RNN module to model the aggregation function $\mathcal{G}$:

$$\boldsymbol{h}_t^s = RNN(\boldsymbol{h}_{t-1}^s, \hat{\boldsymbol{s}}_t), i = 1, ..., N' \tag{13}$$

N' is the number of selected sentences, i.e., only the selected sentence can be transferred into the aggregation function $\mathcal{G}$. The reward is defined as the performance of relevance prediction and it guides the model to learn a good representation and sample appropriate actions, which is based on the offline evaluation metric in learning to ranking. We have three different reward types:

$$R = \begin{cases} -\sum_{k}^{K} MSE(y_k, \tilde{y_k}), & pointwise \\ -\sum_{d^+}\sum_{d^-} max(0, 1 - y_{d^+} + y_{d^-}), & pairwise \\ NDCG(\boldsymbol{y}_{1:K}, \tilde{\boldsymbol{y}}_{1:K}), & listwise \end{cases} \tag{14}$$

where K is the document number of a result list, y and $\bar{y}$ are the predicted relevance score and the ground truth, respectively. Pairwise reward is based on a pair of positive and negative samples d^+ and d^-. Listwise reward is the list-level evaluation measure $NDCG$.

The model is optimized by the policy gradient strategy [28], aiming to maximize the expected reward. The gradient of the policy is given by

$$\begin{aligned} \nabla\mathcal{J}_\Theta(\Theta) &= \mathbb{E}_{\pi_\Theta}\left[\sum_{k=1}^{K}\sum_{i=1}^{N'} \nabla \log \pi(a_{k,i}^s|\Theta) \cdot R\right] \\ &\approx \frac{1}{M}\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{i=1}^{N'} \nabla \log \pi(a_{(m,k),i}^s|\Theta) \cdot R_m \end{aligned} \tag{15}$$

where Θ denotes all the model parameters, M is the sampled number. Policy gradient strategy can backpropagate reward signals to optimize the parameters of retrieval model so that the important sentences can be selected. More importantly, the seeming irrelevant text will be skipped and the retrieval model can also obtain good ranking performance based on only a part of document text.

4.7 Early stop reading ($\mathcal{S}$)

Similar to selective attention, this heuristic also focused on modeling the select function $\mathcal{S}$ in Equation 1. Users will stop reading once the read text is enough to make relevance judgement, which motivates retrieval models to estimate relevance only based on a part of document content. To validate if retrieval models can work well in partial data, we exploit a fixed partial data modeling strategy and a dynamic stop reading strategy.

4.7.1 Partial data modeling

According to the findings in Li et al. [5], users have a preliminary relevance judgement according to the content at the top position. We thus use this strategy to determine if retrieval models can also estimate relevance based on only a part of document content. We partition the document with different ratios and remove the rest content, i.e., keep the beginning content with 20%, 40%, 60% and 80%.

Algorithm 1: Reinforcement learning of Selective attention and Early stop reading.

Input: Query q, document $d = \{s_1, s_2, ..., s_n\}$ (the position of each sentence s_i is p_i), initial state of selective attention $\boldsymbol{h}_0^s$ and early stop reading $\boldsymbol{h}_0^f$
Output: Relevance score R
/* Continue reading while stop reading signal is not positive */
while $a_i^f = 0$ **do**
  Compute sentence-level semantic interacted state $\hat{\boldsymbol{s}}_i$ between query q and sentence s_i.
  Compute selective reading signal $\pi(a_i^s|\hat{\boldsymbol{s}}_i, \boldsymbol{h}_{i-1}^s, p_i)$ based on Equation 12.
  Compute early stop reading signal $\pi(a_i^f|\hat{\boldsymbol{s}}_i, \boldsymbol{h}_{i-1}^s, p_i)$ based on Equation 16.
  if $a_i^f = 0$ ***and*** $a_i^s = 1$ **then**
    /* Continue reading and select the current senetence */
    Update $\boldsymbol{h}_t^s$ based on Equation 13
  Update parameters by $\nabla\mathcal{J}_\Theta(\Theta)$ in Equation 15 in terms of action a_i^s.
  Update parameters by $\nabla\mathcal{J}_\Theta(\Theta)$ in Equation 15 in terms of action a_i^f. // Replace a^s with a^f in Equation 15
Obtain the last aggregated semantic state $\boldsymbol{h}_{N'}^s$.
Estimate the relevance score R by using a MLP layer on $\boldsymbol{h}_{N'}^s$.

4.7.2 Reinforcement learning

Compared with fixed spitting strategy, reinforcement learning is a good strategy to dynamic select the stop reading postion. In this strategy, the agent is used to decide whether the collected information is enough to stop reading. Similar to the decision policy in selective attention modeling, MLP is used to determine the probability of different actions:

$$\pi(a_i^f|\hat{\boldsymbol{s}}_i, \boldsymbol{h}_{i-1}^s, p_i) = \sigma(\boldsymbol{W}_f * [\hat{\boldsymbol{s}}_i, \boldsymbol{h}_{i-1}^s, posEmb(p_i)] + b_f) \tag{16}$$

Once the agent produces a signal for stopping reading (i.e., $a_i^f = 1$), the rest text will be ignored. The reward is also based the ranking performance of retrieval model, as shown in Equation 14. In practical, Equation 12 and 16 can be learned together in a single retrieval model. This heuristic can help retrieval model improve efficiency as well as obtaining good ranking performance.

To better understand the procedure of reinforcement learning in *Selective attention* and *Early stop reading*, we give a workflow algorithm in Algorithm 1. Two heuristics are modeled together in a single model for simplification. If we only need to use one heuristic, we can force the other one to be a fixed value, e.g., let $a^f = 0$ and only compute a^s during the procedure.

Table 1: Statistics of the dataset(QCL) in our experiments.

	QCL-Train	QCL-Test
# query	534,655	2,000
# doc	7,682,872	50,150
# doc per query	14.37	25.08
Vocabulary Size	821,768	445,885

5 Experiment

5.1 Dataset

We validate the effectiveness of modeling different reading heuristics based on a large-scale public available benchmark dataset(QCL) [29]. The dataset is sampled from a real query log of a Chinese commercial search engine *Sogou.com*. It contains weak relevance labels (i.e., click relevance labels [9] derived by five different click models for over 12 million query-document pairs. Prior studies [9, 30] have shown that weak relevance labels derived from click models can be used to train and evaluate retrieval models better than vanilla click signals due to the reduced behavior bias, e.g., position bias. Since PSCM [31] achieves the best relevance estimation performance among these six alternatives, we employ PSCM labels as the ground truth in our experiment.

5.2 Experimental setting

To fairly compare the ranking performance of different retrieval models, we exploit the same experimental settings when modeling different reading heuristics. The parameters are optimized by Adadelta, with a batch size of 80 and a learning rate of 0.1. We trained word embedding based on a Chinese Wikipedia dataset [1)] by word2vec and set the dimension as 50. The dimension of hidden vectors is all set as 128 while the dimension of position embedding is 3. The filter windows sizes of the CNN layer used in this paper are 2 to 5s and 64 feature maps for each filter. The sequential model we used is Gated Recurrent Unit (GRU). The boundary distance γ in unsupervised decaying attention is set as 0.01. For the reinforcement learning in the heuristics of *Selective attention* and *Early stop reading*, we exploit pointwise reward during the training process since it obtains better ranking perfor-

1) http://download.wikipedia.com/zhwiki

mance than pairwise and listwise rewards. The number of sampled action sequences for each document is 5, with an exploration rate 0.2 to control the agent's greed in accumulated rewards. We adopt early stopping with a patience of 10 epochs for training to avoid overfitting. The *Original* model used in our experiment is a baseline, which is implemented by only using multi-level CNN in $\mathcal{F}$ and RNN in $\mathcal{G}$.

5.3 Experimental results

In this section, we exploit different strategies to model the six reading heuristics drawn from human behaviors and evaluate their effectiveness for retrieval tasks. Suggestions for the design of retrieval models are then given based the comparison.

5.3.1 Sequential reading

Table 2: Ranking performance when using different strategies to model the heuristic of ***Sequential Reading***. †, ‡, § denote the significant difference compared with *Random*, *Inverse* and *Sequential*, respectively. (p-value ≤ 0.05)

	NDCG@1	NDCG@3	NDCG@5	NDCG@10
Random	0.6799	0.7099	0.7355	0.7981
Inverse	0.6803	0.7091	0.7328	0.7941
Sequential	$0.6988^{\dagger,\ddagger}$	$0.7198^{\dagger,\ddagger}$	$0.7418^{\dagger,\ddagger}$	0.8008
Ab-posEmb	$0.7104^{\dagger,\ddagger,\S}$	$\mathbf{0.7328}^{\dagger,\ddagger,\S}$	$\mathbf{0.7632}^{\dagger,\ddagger,\S}$	$\mathbf{0.8090}^{\dagger,\ddagger,\S}$
Re-posEmb	$\mathbf{0.7154}^{\dagger,\ddagger,\S}$	$0.7311^{\dagger,\ddagger,\S}$	$0.7593^{\dagger,\ddagger,\S}$	$0.8072^{\dagger,\ddagger}$

This heuristic suggests that the presented order of document content plays an important role for retrieval models. To test whether this implication holds, we change the order of the sentences in document and exploit two position embedding to model the sequential order.

It is observed that when using different presented order, retrieval model perform differently over different evaluation metrics. Specifically, the *Sequential* order outperforms other two presented order significantly, which illustrates that reading direction is important for retrieval models. When applying position embedding to the retrieval model, the ranking performances of two strategies significantly increases. This indicates that position embedding is a good strategy to model the heuristic of *Sequential reading*. Between absolute and relative position embedding, we find that their difference is not significant.

Suggestions: Based on the inspiration in [13], documents are better to be split into several short passages. Then retrieval model can apply on each query-passage pair and aggregate them as the final relevance. We further conclude that the presented order of each fine-grained passage (or sentences) should also be considered. More importantly, position embedding can effectively model the heuristic of *Sequential reading* and improve ranking performance.

5.3.2 Vertical decaying attention

Table 3: Ranking performance when using different strategies to model the heuristic of ***Vertical Decaying Attention***. †, ‡, §, ¶ denote the significant difference compared with *Original*, $+\alpha(v)$, *+attention* and $+\mathcal{L}_{un-decay}$ respectively. (p-value ≤ 0.05)

	NDCG@1	NDCG@3	NDCG@5	NDCG@10
Original	0.6988	0.7198	$0.7418^{\ddagger}$	0.8008
$+\alpha(v)$	0.6869	0.7064	0.7299	0.7915
+attention	$0.7145^{\dagger,\ddagger}$	$0.7260^{\ddagger}$	$0.7533^{\dagger,\ddagger}$	$0.8081^{\dagger,\ddagger}$
$+\mathcal{L}_{un-decay}$	$0.7012^{\dagger,\ddagger}$	$0.7121^{\ddagger}$	0.7346	0.7918
$+\mathcal{L}_{su-decay}$	$\mathbf{0.7221}^{\dagger,\ddagger,\S,\P}$	$\mathbf{0.7348}^{\dagger,\ddagger,\S,\P}$	$\mathbf{0.7712}^{\dagger,\ddagger,\S,\P}$	$\mathbf{0.8164}^{\dagger,\ddagger,\S,\P}$

This heuristic comes from the findings that users' reading attention vertically decays [5]. We attempt to exploit different strategies to model users' decaying attention in retrieval models. The results are shown in Table 3.

It is observed that adding the decaying coefficient does not improve the ranking performance compared with the original retrieval model. It suggests that it may be not suitable to incorporate this heuristic directly by adding a decaying coefficient. As the attention mechanism with decaying regularization, we find that unsupervised decaying attention learning method cannot bring improvement compared with the original attention mechanism. However, when we apply human attention as a supervised signal, the model performs significantly better than that of original attention mechanism. This indicates that directly forcing the weights to be decaying is not an appropriate modeling strategy. Instead, if we use human attention as an additional signal to guide the attention learning, the ranking performance will increase significantly. This is consistent with the finding in other NLP researches [32].

Suggestions: Human attention can provide a good inductive bias on attention learning in retrieval task. However, simply adding decaying coefficient or forcing the attention value to decay are not good strategies for retrieval performance. Instead, we should use the estimated human attention derived from user behavior to regularize attention functions in retrieval models, which can help improve ranking performance.

5.3.3 Query centric guidance

This heuristics indicates that modeling the interaction between query and document is the most important for retrieval

Table 4: Ranking performance when using different strategies to model the heuristic of ***Query Centric Guidance***. †, ‡, § denote the significant difference compared with *Multi-level CNN*, *Spatial RNN* and *Matching histogram*, respectively. (p-value ≤ 0.05)

	Strategy	NDCG@1	NDCG@3	NDCG@5	NDCG@10
Spatial Proximity	Multi-level CNN	0.6988‡,§	0.7198‡,§	0.7418‡,§	0.8008‡,§
	Spatial RNN	0.6745	0.6982	0.7237	0.7874
Semantic Proximity	Matching histogram	0.6869‡	0.7012	0.7288	0.7918
	Kernel pooling	**0.7259**†,‡,§	**0.7404** †,‡,§	**0.7524** †,‡,§	**0.8112**†,‡,§

Table 5: Ranking performance when using different strategies to model the heuristic of ***Context-aware Reading***. †, ‡ denote the significant difference compared with *MLP* and *1-D CNN*, respectively. (p-value ≤ 0.05)

	NDCG@1	NDCG@3	NDCG@5	NDCG@10
MLP	0.6650	0.6684	0.6823	0.7693
1-D CNN	0.6870†	0.7132†	0.7308†	0.7934†
RNN	**0.6988**†,‡	**0.7198**†	**0.7418**†,‡	**0.8008**†,‡

models. As discussed in Section 4, three important heuristics, i.e., exact and semantic query matching, proximity and term importance should be covered. Specifically, since exact and semantic query matching and term importance are included when we construct an interaction matrix, only proximity needs to be modeled by different strategies. We compare the performances between spatial proximity and semantic proximity in Table 4.

We can observe that kernel pooling method outperforms other strategies significantly, which suggests that end-to-end soft-matching can capture more important semantic signals for retrieval model. In spatial proximity, multi-level CNN performs significantly better rather than spatial RNN. More importantly, the computational complexity of spatial RNN is much larger than that of multi-level CNN, which illustrates multi-level CNN is better for practical retrieval model. For semantic proximity, we observe that matching histogram does not perform well compared to other strategies, which suggests that simply taking the count of local interactions in each histogram bin cannot capture semantic matching well. More importantly, the processing of matching histogram brings in much additional computational cost, which is also a disadvantage.

Suggestions: When modeling the interaction between query and document, we find that multi-level CNN and kernel pooling method are better strategies since they can capture better semantic signals for retrieval models. Compared with them, spatial RNN and matching histogram suffer from lower ranking performance and higher computational cost.

5.3.4 Context-aware reading

This heuristic suggests that when modeling the local relevance (or semantic representation), we should also consider the contextual information in retrieval models. We compare the context-aware modeling method (i.e., 1-D CNN and RNN) with context-independent modeling method (i.e., MLP).

It is observed that context-aware modeling method is significantly better than context-independent modeling method, which indicates that show the heuristic of *Context-aware reading* is important in retrieval models. In particular, RNN achieves significantly better ranking performance than 1-D CNN over most of evaluation metrics, which suggests that RNN is better to model contextual information.

Suggestions: The heuristic of *Context-aware reading* is important for retrieval model since it can help improve the ranking performance effectively. It's suggested to consider contextual information when modeling the location relevance (or semantic representation) of each fine-grained sentence.

5.3.5 Selective attention

Table 6: Ranking performance when using different strategies to model the heuristic of ***Selective Attention***. †, ‡ denote the significant difference compared with *Original* and *Query-Centric*, respectively. (p-value ≤ 0.05)

	NDCG@1	NDCG@3	NDCG@5	NDCG@10
Original	0.6988	0.7198	0.7418	0.8008
Query-Centric	0.7058	0.7227	0.7452	0.8059
RL-select	**0.7359**†,‡	**0.7604**†,‡	**0.7609**†,‡	**0.8130**†,‡

This heuristic indicates that retrieval models can ignore the text that has no or little influence on relevance. We compare the fixed query-centric attention and dynamic selective attention with reinforcement learning. The result is shown in Table 6.

It is observed that two modeling strategies of *Selective attention* perform significantly different. Query centric attention performs similar to the original retrieval model without

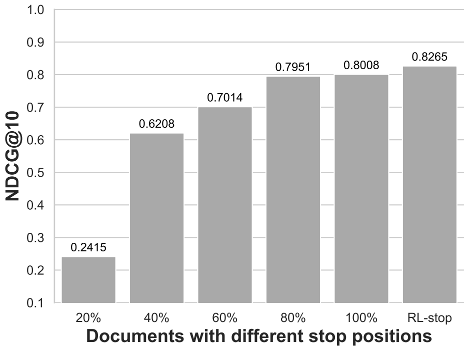


Fig. 6: Ranking performance of when applying different stop positions (***Selective Attention***). Percentage means only using the specific part of document. RL-stop indicates using reinforcement learning to select the position dynamically.

significantly difference. However, when applying dynamic selective attention with reinforcement learning, the model achieves significantly better ranking performance, which illustrates that dynamic selective attention can help retrieval model to selectively capture important information and obtain better ranking performance.

Suggestions: The heuristic of *Context-aware reading* can help retrieval model to selectively capture important information. Query centric attention can not improve ranking performance as effectively as dynamic selective attention. More importantly, considering that reinforcement learning will bring in additional computational cost, we should also carefully balance the tradeoff between effectiveness and efficiency when building practical retrieval models.

5.3.6 Early stop reading

This heuristic is similar to *Selective attention*, which helps retrieval model select information to estimate relevance and ignore others. Due to the tradeoff between the precision of language understanding and attention effort, users tends to stop reading before the end of a document. Thus, we compare the ranking performance when using a fixed partial data modeling strategy and a dynamic stop reading strategy. The results are shown in Figure 6. We only list the result of NDCG@10 due to the space limitation and the performances of other metrics are similar.

It is observed that dynamic stop reading strategy wit reinforcement learning outperforms other strategies significantly (p-value < 0.05). It illustrates that using reinforcement learning can indeed help retrieval model to stop in a proper position and further improve ranking performance. As for partial data modeling, we find that the ranking performance gets better as the percentage of data increases, which is self-explanatory. Interestingly, we observe that 80% of data can achieve quite similar ranking performance as the full data, even though the difference is significant (p-value < 0.05) because of the large data size. This means that retrieval model can consider this tradeoff if the system prefers to improve computational efficiency.

Suggestions: This heuristic puts forward a tradeoff between computational cost and ranking performance. If one retrieval system prefers the ranking performance, it can exploit RL to implement this heuristic. On the other hand, we find 80% of document content can achieve quite similar ranking performance as the whole content, which implies that retrieval model can by reducing unnecessary contents within documents to improve computational efficiency.

6 Conclusion

This paper investigates the relationship between users' reading heuristics and the design of retrieval models. Based on six reading heuristics derived from search user behaviors, we give different modeling strategies of each heuristics and compare their effectiveness in practical retrieval models. For *Sequential reading*, it's better to split the whole document into several find-grained content and use position embedding to model the text order. For *Vertical decaying attention*, human attention signals can provide a good inductive bias on attention learning to improve ranking performance. Considering different methods to model the interactions between query and document, we find multi-level CNN and kernel pooling method can provide both good ranking performance and computational efficiency. *Context-aware reading* motivates retrieval models that deploying sequential models to estimate local relevance based on contextual information can help improve ranking performance. For *Selective attention* and *Early stop reading*, we find that reinforcement learning can effectively improve the ranking performance but also bring in addition computational cost. In practical system, we should carefully consider this tradeoff when using these two reading heuristics. In summary, our study sheds lights on cognitive-oriented retrieval model and provides better insights on building retrieval model from the perspective of cognitive behavior.

In this work, we only independently focus on each heuristic and evaluate the interleaving effectiveness. However, these heuristics are relative to each other to some extent. For instance, *Sequential reading* is the fundamental of *Context-*

aware reading and *Early stop reading*. It's interesting to see if it's necessary to model sentence position for *Sequential reading* when modeling *Early stop reading*. More importantly, these heuristics can be incorporated into a single model together as discussed in our original paper [1]. It's also interesting to see which combination of all modeling strategies can perform best in retrieval model. In the future, we aim to focus on these problems and have deeper insights into the relationship between cognitive behaviors and retrieval model.

References

1. Li X S, Mao J X, Wang C, Liu Y Q, Zhang M, and Ma S P. Understanding reading attention distribution during relevance judgement. *In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM Press*, 2019, 795-804.
2. Pang L, Lan Y Y, Guo J F, Xu J, and Cheng X Q. Deeprank: A new deep architecture for relevance ranking in information retrieval. *In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017.
3. Hu B T, Lu Z D, Li H, and Chen Q C. Convolutional neural network architectures for matching natural language sentences. *In: Advances in neural information processing systems*, 2014, 2042-2050.
4. Huang P S, He X D, Gao J F, Deng L, Acero A, and Heck L. Learning deep structured semantic models for web search using clickthrough data. *In: Proceedings of the 22nd ACM international conference on Information and Knowledge Management*, 2013, 2333-2338.
5. Li X S, Liu Y Q, Mao J X, He Z X, Zhang M, and Ma S P. Understanding reading attention distribution during relevance judgement. *In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management. New York: ACM Press*, 2018, 733-742.
6. Fan Y X, Guo J F, Lan Y Y, Xu J, Zhai C X, and X Q Cheng. Modeling diverse relevance patterns in ad-hoc retrieval. *In: Proceedings of the 41th ACM International Conference on Information and Knowledge Management. New York: ACM Press*, 2018, 375-384.
7. Robertson S E and Walker S. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. *In: Proceedings of the 14th ACM international conference on Information and Knowledge Management*, 1994, 232-241.
8. Guo J F, Fan Y X, Ai Q Y, and Croft W B. A deep relevance matching model for ad-hoc retrieval. *In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, 2016, 55-64.
9. Xiong C, Dai Z, Callan J, Liu Z, and Power R. End-to-end neural adhoc ranking with kernel pooling. *In: Proceedings of the 40th ACM International Conference on Information and Knowledge Management. New York: ACM Press*, 2017.
10. Ding M, Zhou C, Chen Q B, Yang H X, and Tang J. Cognitive graph for multi-hop reading comprehension at scale. *arXiv preprint arXiv:1905.05460*, 2019.
11. Yu A W, Lee H, and Le Q V. Learning to skim text. *arXiv preprint arXiv:1704.06877*, 2017.
12. Fu T J and Ma W Y. Speed reading: Learning to read forbackward via shuttle. *In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, 4439-4448.
13. Pang L, Lan Y Y, Guo J F, Xu J, and Cheng X Q.
14. Hui K, Yates A, Berberich K, and de Melo G. Pacrr: A position-aware neural ir model for relevance matching. *arXiv preprint arXiv:1704.03940*, 2017.
15. Fang H, Tao T, and Zhai C X. Diagnostic evaluation of information retrieval models. *Transactions on Information Systems (TOIS). ACM*, 2011, 29(2): 7.
16. Fang H, Tao T, and Zhai C X. A formal study of information retrieval heuristics. *In: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. New York: ACM Press*, 2004, 49-56.
17. Tao T and Zhai C X. An exploration of proximity measures in information retrieval. *In: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, 2007, 295-302.
18. Hahn M and Keller F. Modeling human reading with neural attention. *arXiv preprint arXiv:1608.05604*, 2016.
19. Liu X G, Mou L L, Cui H T, Lu Z D, and Song S. Jumper: Learning when to make classification decisions in reading. *arXiv preprint arXiv:1807.02314*, 2018.
20. Yu K Y, Liu Y, Schwing A G, and Peng J. Fast and accurate text classification: Skimming, rereading and early stopping. 2018.
21. Wason P C and Evans J S B. Dual processes in reasoning? *Cognition. Elsevier*, 1974, 141-154.
22. Gehring J, Auli M, Grangier D, Yarats D, and Dauphin Y N. Convolutional sequence to sequence learning. *In: Proceedings of the 34th International Conference on Machine Learning. JMLR*, 2017, 1243-1252.
23. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, and Polosukhin I. Attention is all you need. *In: Advances in neural information processing systems.*, 2017, 5998-6008.
24. Pang L, Lan Y Y, Guo J F, Xu J, Wan S X, and Cheng X Q. Text matching as image recognition. *In: Proceedings of the 30th AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI*, 2016.
25. Nie Y F, Li Yan L, and Nie J Y. Empirical study of multi-level convolution models for ir based on representations and interactions. *In: Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval*, 2018, 59-66.
26. Graves A and Schmidhuber J. Offline handwriting recognition with multidimensional recurrent neural networks. *In: Advances in neural information processing systems*, 2009, 545-552.
27. Wu H C, Luk R W P, Wong K F, and Kwok K L. A retrospective study of a hybrid document-context based retrieval model. *Information processing and management*, 2007, 43(5): 1308-1331.

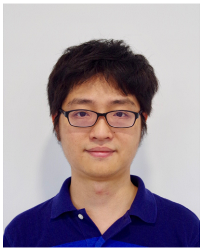
28. Sutton R S, McAllester D A, Singh S P, and Mansour Y. Policy gradient methods for reinforcement learning with function approximation. *In: Advances in neural information processing systems*, 2000,1057-1063.
29. Zheng Y K, Fan Z, Liu Y Q, Luo C, Zhang M, and Ma S P. Sogou-qcl: A new dataset with click relevance label. *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM*, 2018, 1117-1120.
30. Dehghani M, Zamani H, Severyn A, Kamps J, and Croft W B. Neural ranking models with weak supervision. *In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM*, 2017, 65-74.
31. Wang C, Liu Y Q, Wang M, Zhou K, Nie J Y, and Ma S P. Incorporating non-sequential behavior into click models. *In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM*, 2015, 283-292.
32. Barrett M, Bingel J, Hollenstein N, Rei M, and Søgaard A. Sequence classification with human attention. *In: Proceedings of the 22nd Conference on Computational Natural Language Learning*, 2018, 302-312.

Xiangsheng Li

Xiangsheng Li is now a 3rd-year PhD student at the Department of Computer Science and Technology in Tsinghua University, Beijing, China. His research interests include Web Search, User Behavior Analysis and Search Ranking Modeling. He is mainly working on using human-inspired heuristics (e.g., behavior patterns and knowledge information) to design and improve retrieval models.

Yiqun Liu

Yiqun Liu is now working as professor and co-chair at the Department of Computer Science and Technology in Tsinghua University, Beijing, China. His major research interests include Web Search, User Behavior Analysis, and Web Data Mining. He is senior members of ACM and IEEE, distinguished member of CCF, a visiting professor of National Institute of Informatics (NII) in Japan and a visiting research professor of National University of Singapore (NUS).



Jiaxin Mao

Ph.D. Jiaxin Mao works as a post-doctoral research fellow at the Department of Computer Science and Technology at Tsinghua University. His research interests are Information Retrieval, Web Mining, and Applied Machine Learning. He has published more than 30 papers in top-tier conferences and journals including SIGIR, TOIS, WWW, KDD, IJCAI, WSDM, and CIKM. He is the recipient of the SIGIR 2018 Best Short Paper Honourable Mention Award, ICTIR 2019 Best Short Paper Honorable Mention Award, and the Chinese Journal of Computers Best Paper Award (2014-2018). He also serves as the ACM SIGIR Student Affairs Co-Chair.