Leveraging Multi-view Inter-passage Interactions for Neural Document Ranking

Chengzhen Fu[1], Enrui Hu[1], Letian Feng[1], Zhicheng Dou[2], Yantao Jia[1]
Lei Chen[3], Fan Yu[1], Zhao Cao[1]
[1] Distributed and Parallel Software Lab, Huawei Technologies Co., Ltd.
[2] Gaoling School of Artificial Intelligence, Renmin University of China
[3] Department of Computer Science and Engineering, Hong Kong University of Science and Technology
{fuchengzhen,huenrui1,fengletian1,jiayantao,fan.yu,caozhao1}@huawei.com

ABSTRACT

The configuration of 512 window size prevents transformers from being directly applicable to document ranking that requires larger context. Hence, recent works propose to estimate document relevance with fine-grained passage-level relevance signals. A limitation of such models, however, is that scoring each passage independently falls short in modeling inter-passage interactions and leads to unsatisfactory results. In this paper, we propose a **M**ulti-view inter-passage **I**nteraction based **R**anking model (**MIR**), to combine intra-passage interactions and inter-passage interactions in a complementary manner. The former captures local semantic relations inside each passage, whereas the latter draws global dependencies between different passages. Moreover, we represent inter-passage relationships via multi-view attention patterns, allowing information propagation at token, sentence, and passage-level. The representations at different levels of granularity, being aware of global context, are then aggregated into a document-level representation for ranking. Experimental results on two benchmarks show that modeling inter-passage interactions brings substantial improvements over existing passage-level methods.

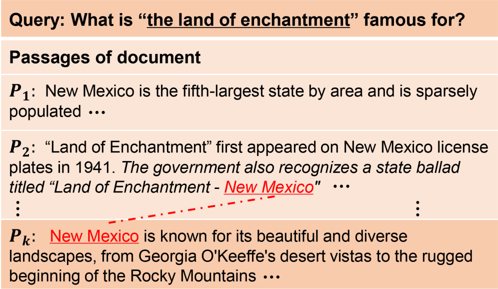
CCS CONCEPTS

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

KEYWORDS

Document ranking; Inter-passage attention; Intra-passage attention

ACM Reference Format:
Chengzhen Fu, Enrui Hu, Letian Feng, Zhicheng Dou, Yantao Jia, Lei Chen, Fan Yu, Zhao Cao. 2022. Leveraging Multi-view Inter-passage Interactions for Neural Document Ranking. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining (WSDM '22), February 21–25, 2022, Tempe, AZ, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3488560.3498412



Query: What is "the land of enchantment" famous for?

Passages of document

P_1: New Mexico is the fifth-largest state by area and is sparsely populated ···

P_2: "Land of Enchantment" first appeared on New Mexico license plates in 1941. *The government also recognizes a state ballad titled "Land of Enchantment - New Mexico"* ···

⋮

P_k: New Mexico is known for its beautiful and diverse landscapes, from Georgia O'Keeffe's desert vistas to the rugged beginning of the Rocky Mountains ···

Figure 1: Examples of query-passage pairs. Despite few lexical overlaps, P_k is relevant by exploiting the information from P_2 that "the land of enchantment" is a nickname.

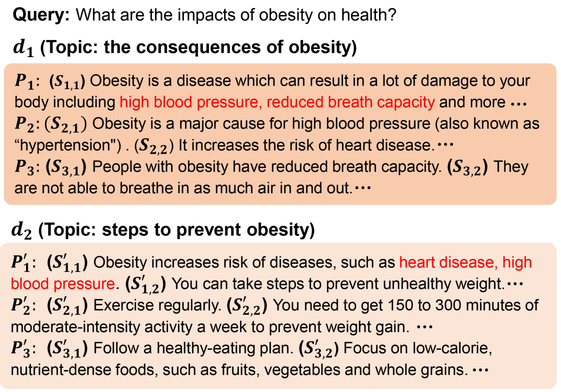
1 INTRODUCTION

Document ranking plays an indispensable role in information retrieval (IR). Recently, exploiting contextual language models, such as BERT [10], has achieved remarkable gains from deeper text understanding. However, the computational and memory requirements of the quadratic attention mechanism limit the input sequence to 512 tokens [2]. Applying BERT directly to document ranking scenarios is confronted with the challenge that the document length is too long to fit into the transformer window.

Consequently, several works propose to estimate document relevance with fine-grained passage-level relevance signals. First, documents are partitioned into passages based on textual discourse units [4] or simply fixed-length windows [16] that do not rely on the document structure. Then, local relevance signals are obtained by applying BERT to each passage individually [8, 40]. Furthermore, they are converted to the document-level relevance score via the score or representation aggregation module [23, 33, 38].

Most of these works are built on the basic hypothesis that passages are independent from each other and the relevance score of each passage could be assigned separately. However, they only consider semantic dependencies inside the passage while neglect the underlying relations among passages. We argue that **handling inter-passage relations are crucial in capturing longer-range word dependencies and generating superior document-aware contextualized representations**. As presented in Figure 1, the query contains a nickname "the Land of Enchantment"(underlined) and all passages come from the same document. Ignoring the information

WSDM '22, February 21–25, 2022, Tempe, AZ, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9132-0/22/02…$15.00
https://doi.org/10.1145/3488560.3498412



Query: What are the impacts of obesity on health?

d_1 **(Topic: the consequences of obesity)**

P_1: $(S_{1,1})$ Obesity is a disease which can result in a lot of damage to your body including high blood pressure, reduced breath capacity and more ···
P_2: $(S_{2,1})$ Obesity is a major cause for high blood pressure (also known as “hypertension") . $(S_{2,2})$ It increases the risk of heart disease.···
P_3: $(S_{3,1})$ People with obesity have reduced breath capacity. $(S_{3,2})$ They are not able to breathe in as much air in and out.···

d_2 **(Topic: steps to prevent obesity)**

P'_1: $(S'_{1,1})$ Obesity increases risk of diseases, such as heart disease, high blood pressure. $(S'_{1,2})$ You can take steps to prevent unhealthy weight.···
P'_2: $(S'_{2,1})$ Exercise regularly. $(S'_{2,2})$ You need to get 150 to 300 minutes of moderate-intensity activity a week to prevent weight gain. ···
P'_3: $(S'_{3,1})$ Follow a healthy-eating plan. $(S'_{3,2})$ Focus on low-calorie, nutrient-dense foods, such as fruits, vegetables and whole grains. ···

Figure 2: Examples of document pairs[1] with the same query. Though both have relevant contents (in red), the topic of d_2 is not aligned with the query and should be ranked lower.

revealed in P_2 that this nickname refers to New Mexico (emphasized by italics), the relevance of P_k will be underestimated due to few lexical overlaps. Conversely, introducing such dependency as a supplement can boost the relevance score of P_2 and thus benefit the ranking of the entire document. By building coreference links on informative tokens (e.g., the red dash line on New Mexico), different passages are connected and their representations are enriched to achieve better query-passage matching. In summary, we claim that introducing **token-level** relations between some selective terms of different passages can promote key information exchange among passages and improve document ranking performance.

Furthermore, modeling inter-passage relationships at higher levels of granularity, e.g., **sentence and passage-level**, enables us to identify the key topic of the entire document. As shown in Figure 2, in conjunction with passage-level interactions in d_2 (namely, $P'_1P'_2$ or $P'_1P'_3$), we realize that d_2 is more concerned with another topic, i.e., “steps to prevent weight gain”. Though d_2 contains terms (highlighted in red) relevant to the query, the document-level topic it mainly expounds deviates from the information need the query represents (i.e., “consequences of obesity”). In contrast, by interacting P_1 and P_2 (or P_3) in d_1, we discover contents in d_1 is coherently focused on the topic “the harmness of obesity”. Thus, d_1 is more topically consistent with the user search intents and should be ranked higher than d_2. In a word, we envision that **higher level inter-passage interactions help capture global semantic coherence of all passages and induce document-level topics implicitly**, which inherently complement existing passage-level methods.

In this paper, we propose a **M**ulti-view inter-passage **I**nteraction based **R**anking model (**MIR**), to augment the transformer [31] with the ability to capture inter-passage correlations for document ranking. Due to the complexity in interacting arbitrary pair of tokens from different text pieces, we pre-select a subset of informative “pivot” tokens (§3.1.2), and subsequently construct graphs following their inherent relations (§3.1.3). Specifically, as document semantics usually present the word-sentence-passage hierarchy, we represent inter-passage relationships via multi-view attention patterns. Hence, graphs on different granularity levels (i.e., token, sentence, passage) are constructed, with pivot tokens resembling nodes and edges representing semantic links. **First**, token-level edges are built to characterize global syntactic (e.g., word dependencies) or semantic features for sailent terms. **Then**, sentence and passage-level links are drawn to capture global semantic coherency and induce the topical structure of the entire document. By this means, MIR models rich contextual information beyond the segment length. Meanwhile, the pivot token selection provides a balance between efficiency and model representation capacity (§5.2).

Overall, MIR is composed of stacking attention blocks and an multi-view aggregation layer on top to generate a global document representation. Each block has two sub-layers, i.e., an intra-passage attention layer and an inter-passage attention layer. The former captures local semantic relations inside each passage, whereas the latter compensates for the limited intra-attention span and draws global dependencies between different passages based on pre-constructed graphs. In each block, pivot tokens serve as a conduit for information flow and their representations are enriched via graph-informed inter-passage attention. Coupled with intra-passage attention, information introduced by pivot tokens are further propagated to regular tokens. In this way, the combination of intra-passage and inter-passage attention upgrades the model to be aware of the global document context. Finally, the aggregation layer combines these document-aware representations at different levels of granularity, to form a comprehensive document embedding for ranking.

To summarize, our contributions are three-fold: **1)** We investigate the problem of modeling interactions among passages within a long document for better document ranking. It is the first time that the inter-passage dependencies are seriously studied for neural document ranking. **2)** We integrate intra-passage and inter-passage interactions into a unified framework, which enables document-aware contextual representations. Meanwhile, interactions among different passages are conditioned on a small subset of pivot tokens, making a trade-off between efficiency and effectiveness. **3)** We design multi-view attention patterns to allow information propagation at different levels of granularity, which help capture longer-range dependencies and induce topical structure of documents.

2 RELATED WORK

Passage based Document Ranking. Many approaches have been developed to address the 512 limit of transformer models. A common idea is to split the document into (overlapping) passages [4, 29], process each passage separately [1, 8, 16, 24–26], then combine the signals with a sophisticated aggregation model [23, 25, 38]. Some works explore different strategies to combine the passage ranking scores. Dai and Callan [8] took the maximum (BERT-MaxP), first (BERT-FirstP), and summation (BERT-SumP) of matching scores of query-passage pairs as document-level ranking scores. To deal with documents with varying length, IDCM [16] further used an intra-document cascade ranking model with a fast passage selection module. However, all these approaches are usually trained on passages, without any information flow across passages, limiting the contextualization within the current passage. Others also investigate sophisticated representation aggregation methods to obtain a document embedding for ranking. PCGM [33] performed

[1] $S_{i,j}$ represents the j-th sentence in the i-th passage.

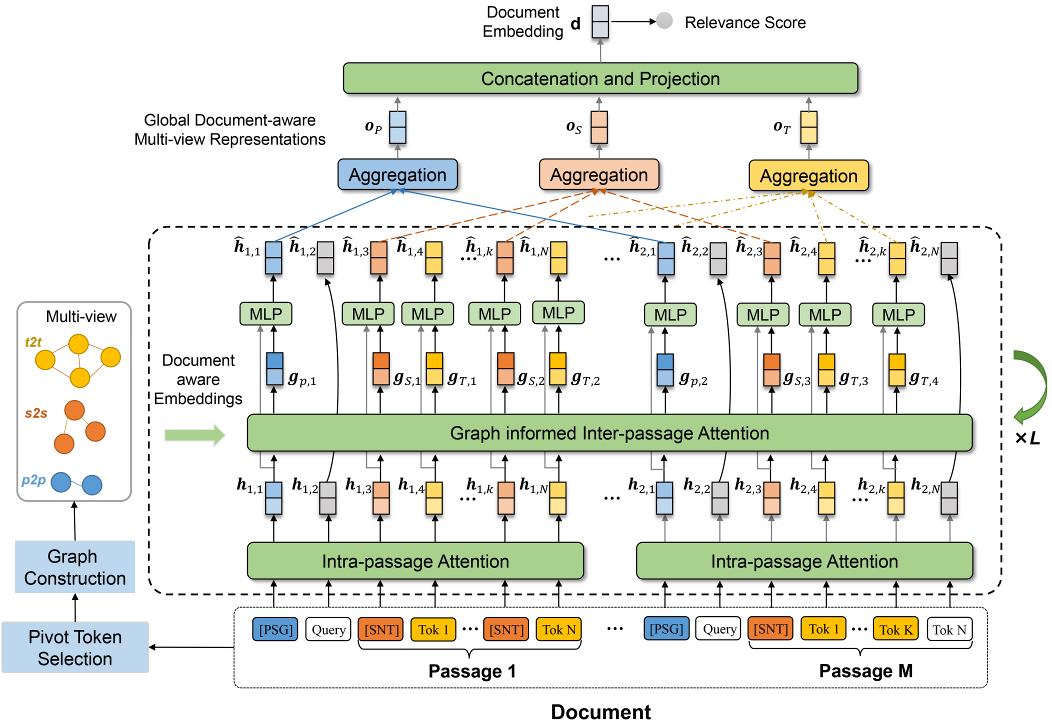

Document Embedding
d
Relevance Score
Concatenation and Projection
Global Document-aware Multi-view Representations
o_P
o_S
o_T
Aggregation
Aggregation
Aggregation
$\hat{h}_{1,1}$ $\hat{h}_{1,2}$ $\hat{h}_{1,3}$ $\hat{h}_{1,4}$ $\hat{h}_{1,k}$ $\hat{h}_{1,N}$ $\hat{h}_{2,1}$ $\hat{h}_{2,2}$ $\hat{h}_{2,3}$ $\hat{h}_{2,4}$ $\hat{h}_{2,k}$ $\hat{h}_{2,N}$
MLP
Multi-view
t2t
s2s
p2p
Document aware Embeddings
$g_{p,1}$ $g_{S,1}$ $g_{T,1}$ $g_{S,2}$ $g_{T,2}$ $g_{p,2}$ $g_{S,3}$ $g_{T,3}$ $g_{T,4}$
Graph informed Inter-passage Attention
×L
$h_{1,1}$ $h_{1,2}$ $h_{1,3}$ $h_{1,4}$ $h_{1,k}$ $h_{1,N}$ $h_{2,1}$ $h_{2,2}$ $h_{2,3}$ $h_{2,4}$ $h_{2,k}$ $h_{2,N}$
Graph Construction
Intra-passage Attention
Intra-passage Attention
Pivot Token Selection
[PSG] Query [SNT] Tok 1 ··· [SNT] Tok N
[PSG] Query [SNT] Tok 1 ··· Tok K Tok N
Passage 1
Passage M
Document

Figure 3: The overview of MIR. Blue, orange and yellow represent tokens associated with passage (P), sentence (S), token (T) level graphs, respectively. Grey denotes regular tokens that do not participate in graph construction.

sequential passage representation aggregation using a LSTM [15]. However, its training relies on passage-level cumulative gain annotations [34]. Several works achieve representation aggregation via max or attention pooling [23, 44], or in a complex hierarchical manner [32, 36, 37, 41]. Particularly, Transformer-XH [42] modeled text sequences by linking them with eXtra Hop attention paths. Though these methods can alleviate the lack of global document context to some extent, how to guide information propagation at different levels of granularity still remains underexplored.

Long-Document Transformers. Thinking of the full attention model as a complete graph, another line of works modifies the quadratic full attention by designing sparse attention patterns (graphs) [6, 17, 21, 28, 30, 43], making it feasible to process long documents. For example, Child et al. [6] used a form of dilated sliding window of blocks of size 8x8 to achieve sparsification. Subsequent works further explore the idea that combines local windowed attention with a task motivated global attention [2, 3, 13, 21, 39]. Global attention are added to *global tokens* to attend to all tokens, whereas regular tokens only attend to to a local neighborhood. Taking Longformer [3] for instance, global attention in QA is provided on all query tokens. For memory efficiency, they implement custom CUDA kernels using Tensor Virtual Machine (TVM) [5].

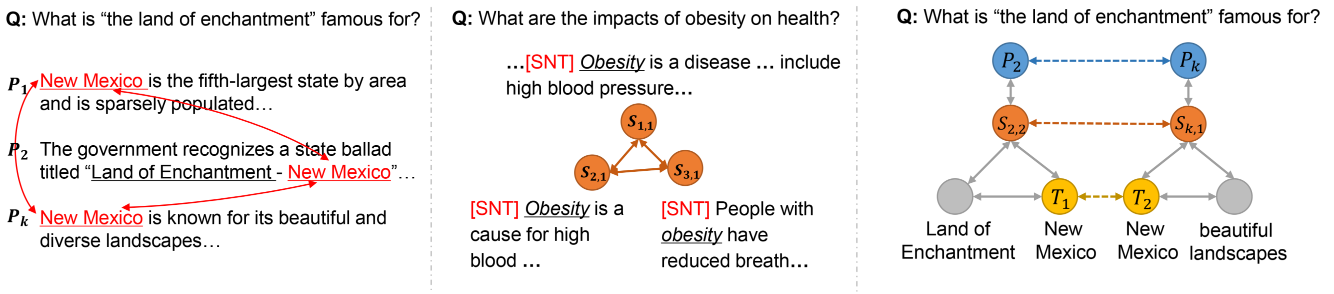
In this work, pivot tokens share some similarities with global tokens. Nevertheless, pivot tokens are carefully selected based on importance and graphs at different levels of granularity are constructed to capture diverse relations. These designs are more suited for the document ranking task.

3 METHODOLOGY

A naive solution to modeling inter-passage interactions is to interact arbitrary pair of tokens from different passages. However, it is usually infeasible due to limited resources. This motivates us to construct inter-passage interactions on a small subset of "pivot" tokens that play a prominent role in document semantics. To model relationships comprehensively, graphs among pivot tokens are built at different granularity levels. **Figure 3** shows the overall architecture of MIR. MIR is comprised of two main components: (1) An iterative attention stack which includes L stacked transformer blocks to incorporate the intra-passage and inter-passage interactions. It learns document-level context aware token, sentence, and passage representations. In each block, a basic **intra-passage attention sub-layer** that provides localized representations is in conjunction with an **inter-passage attention sub-layer** that enables flexible information exchange and integration between passages based on multi-view graphs. This design ensures global context information accessible to regular tokens, with pivot tokens serving as information transfer stations. (2) An multi-view aggregation layer generates an overall document representation for ranking based on the learned token, sentence, and passage representations.

3.1 Passage Partition and Graph Construction

3.1.1 Passage Partition. Following [16], given a document d with length l, we partition it into overlapping windows of size w with the stride of size k. It brings a set of approximately $\lceil l/k \rceil$ passages:


Q: What is “the land of enchantment” famous for?
P1 New Mexico is the fifth-largest state by area and is sparsely populated…
P2 The government recognizes a state ballad titled “Land of Enchantment - New Mexico”…
Pk New Mexico is known for its beautiful and diverse landscapes…
Q: What are the impacts of obesity on health?
…[SNT] Obesity is a disease … include high blood pressure…
S1,1
S2,1
S3,1
[SNT] Obesity is a cause for high blood …
[SNT] People with obesity have reduced breath…
Q: What is “the land of enchantment” famous for?
P2
Pk
S2,2
Sk,1
T1
T2
Land of Enchantment
New Mexico
New Mexico
beautiful landscapes

(a) token-level links

(b) sentence-level links

(c) stacked attentions

Figure 4: Illustration of attention patterns for (a) token-level and (b) sentence-level. (c) shows the effectiveness of stacked attentions. Solid lines indicate intra-passage attention that resembles a complete graph (some edges are omitted for simplicity). Dash lines represent the inter-pasaage attention. Different colors of nodes and edges denote different views.

$P_d = \{d_{0:w-1}, d_{k:k+w-1}, d_{2k:2k+w-1}, \cdots\}$. For clarity, we define $\alpha = 1 - k/w$ as the overlapping ratio. Particularly, we prepend special tokens, [PSG] and [SNT], to each passage and each sentence. They can be viewed as tokens representing a summary of passages and sentences respectively. We use the following input format:

$$\texttt{[PSG] query [SNT] sent}_1\ \texttt{[SNT] sent}_2\ \texttt{[SNT]} \cdots .$$

3.1.2 Pivot Token Selection. Due to the complexity in interacting arbitrary pair of tokens from different passages, we pre-select several salient “pivot” tokens as representatives. Based on the granularity of information they carry, pivot tokens are further grouped into three sets, i.e., $\mathbb{P}$ (passage-level), $\mathbb{S}$ (sentence-level) and $\mathbb{T}$ (token-level). **First**, special tokens added to the inputs, i.e., [PSG] and [SNT], represent the summary of passage and sentence level semantics. Accordingly, we add them to $\mathbb{P}$ and $\mathbb{S}$, respectively. **Second**, the token-level set $\mathbb{T}$ is comprised of informative entities and ordinary words that play a prominent role in document semantics. Entities are annotated by an open-source framework TagMe [11]. As for ordinary words, inspired by [9], we experiment with a pseudo-relevance feedback (PRF) based weak-supervision method to estimate their importance.

First, we use BM25 to retrieve top$k$ (set to 10) documents for each query. Then we collect a document's pseudo-relevant queries $|Q_d|$, and generate the weight of term t using the percentage of queries that mention t, i.e.,

$$y_{t,p} = \frac{|Q_{d,t}|}{|Q_d|}, p \in P_d. \tag{1}$$

3.1.3 Multi-view Graph Construction. On the basis of pivot token types, graphs on different granularity levels are then constructed, with pivot tokens resembling nodes and edges representing semantic links. Hence, graphs are split into three separate pieces: passage to passage (p2p), sentence to sentence (s2s), token to token (t2t). Traditional self-attention can be viewed as a complete graph with the identity matrix as the adjacency matrix. Instead of making edges fully connected, we define some informative connections, to characterize (1) syntactic (e.g., coreference), (2) discourse (e.g., co-occurrences) and (3) semantic (e.g., similarities) relations.

Suppose n_t, n_s, n_p are the number of nodes in $\mathbb{T}$, $\mathbb{S}$ and $\mathbb{P}$, adjacency matrices $\boldsymbol{A}^{\text{t2t}} \in \mathbb{R}^{n_t \times n_t}$, $\boldsymbol{A}^{\text{p2p}} \in \mathbb{R}^{n_s \times n_s}$, $\boldsymbol{A}^{\text{s2s}} \in \mathbb{R}^{n_p \times n_p}$ are defined as: **1)** $A^{\text{t2t}}_{i,j} = 1$ if they are mentions of the same entity or they are the same ordinary words; **2)** $A^{\text{s2s}}_{i,j} = 1$ if the pair of sentences have overlapping terms defined in $\mathbb{T}$; **3)** $A^{\text{p2p}}_{i,j} = 1$ if the similarity score of j-th passage is ranked in the top$k$ (set to 5) for the i-th passage or vice versa. The score is computed by the cosine similarity of tf-idf representations.

The reasons are as follows: **(1)** The first graph (token-level) ensures that entities or sailent terms across multiple passages are connected via coreference links. In **Figure 4a**, mentions linked to “New Mexico” are connected. Thus, different properties (e.g., population, nickname, landscape) of New Mexico are jointly encoded to enrich the representation, which mitigates the lexical mismatch between query and P_k. **(2)** The second graph (sentence-level) models discourse relations. It captures global semantic coherency by extracting word concurrence patterns.

In **Figure 4b**, interacting sentences containing “obesity” (underlined) from different passages help induce the document-level topic and benefit the query-document matching. Passage-level graph has similar effect in discovering topical structure of documents.

3.2 Iterative Attention Stacks

Overall, MIR is composed of stacking attention blocks. Each block has an intra-passage and an inter-passage attention layer. Since representations of pivot tokens are updated by the graph based inter-passage attention and are aware of the global context, to integrate such information into their original contextual representations, each block will aggregate the outputs of two sub-layers into the unified representations, as the inputs to the next block.

Formally, for the l-th block, we denote the outputs of the intra-passage attention for τ-th passage as $\boldsymbol{H}^l_\tau = \left\{\mathbf{h}^l_{\tau,0}, \cdots, \mathbf{h}^l_{\tau,N}\right\} \in \mathbb{R}^{N \times E}$, and the outputs of inter-passage attention for passage, sentence and token-level graphs as $\boldsymbol{G}^l_P \in \mathbb{R}^{n_p \times E}$, $\boldsymbol{G}^l_S \in \mathbb{R}^{n_s \times E}$, $\boldsymbol{G}^l_T \in \mathbb{R}^{n_t \times E}$, respectively. Here, N and E denote the number of tokens for each passage and the dimensions of the representation. The block output for the i-th token is computed as,

$$\hat{\mathbf{h}}^l_{\tau,i} = \begin{cases} \boldsymbol{W}^C \left[\mathbf{h}^l_{\tau,i} ; \mathbf{g}^l\right], & \text{if it is the pivot token;} \\ \mathbf{h}^l_{\tau,i}, & \text{otherwise,} \end{cases} \tag{2}$$

where $\mathbf{g}^l$ is the corresponding node representation extracted from one of $\boldsymbol{G}_P^l$, $\boldsymbol{G}_S^l$, $\boldsymbol{G}_T^l$, if the i-th token belongs to or is part of a pivot token. $\boldsymbol{W}^C \in \mathbb{R}^{E\times 2E}$ is the projection matrix. The l-th block outputs for τ-th passage are denoted as $\hat{\boldsymbol{H}}_\tau^l = \left\{\hat{\mathbf{h}}_{\tau,0}^l, \cdots, \hat{\mathbf{h}}_{\tau,N}^l\right\}$.

Discussion. When attention layers are stacked, pivot tokens serve as a bridge for information transportation. As shown in **Figure 4c**, regular tokens can indirectly attend to all other relevant tokens in the document via pivot tokens. Hence, connection between "Land of Enchantment" and "beautiful landscapes" is built, which is critical for query matching. Moreover, the information flow among different text pieces induces a natural structure. Specifically, low-level pivot tokens are linked to help characterize global syntactic features (e.g., word dependencies) and high-level pivot tokens are linked to summary and propagate query-aware topic information by injecting word-sentence-passage hierarchy.

3.3 Intra-passage Attention Sub-Layer

To obtain $\boldsymbol{H}_\tau^l$, this sub-layer applies a self-attention mechanism and acts as the transformer layer. For each token, the representation is computed as a weighted sum of embeddings of all other tokens. The weights are assigned by the attention (Att) function as follows,

$$\text{Att}\left(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}\right) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{E}}\right)\boldsymbol{V}, \tag{3}$$

where $\boldsymbol{Q}$, $\boldsymbol{K}$, $\boldsymbol{V} \in \mathbb{R}^{N\times E}$ are different projections of inputs. The multihead (MH) strategy further projects the inputs into h different subspaces and performs attention on each split in parallel. The outputs are computed as,

$$\text{MH}\left(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}\right) = \text{concat}\left(\left[\text{Att}\left(\boldsymbol{Q}\boldsymbol{W}_i^Q, \boldsymbol{K}\boldsymbol{W}_i^K, \boldsymbol{V}\boldsymbol{W}_i^V\right)\right]_{i=1}^h\right)\boldsymbol{W}^O, \tag{4}$$

where $\boldsymbol{W}_i^Q, \boldsymbol{W}_i^K, \boldsymbol{W}_i^V \in \mathbb{R}^{E\times E/h}$ are parameters for the i-th subspace. $\boldsymbol{W}^O \in \mathbb{R}^{E\times E}$ is the projection matrix to obtain final outputs.

The output for a transformer (TRM) layer is denoted as,

$$\begin{aligned} &\text{TRM}\left(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}\right) = \text{LayerNorm}(\boldsymbol{O} + \text{FFN}\left(\boldsymbol{O}\right)), \\ &\qquad \text{where } \boldsymbol{O} = \text{LayerNorm}\left(\boldsymbol{Q} + \text{MH}\left(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}\right)\right), \end{aligned} \tag{5}$$

where FFN $(\cdot)$ is a fully connected two layer feed-forward network, and LayerNorm $(\cdot)$ denotes the layer normalization.

For the l-th intra-passage attention layer, all of the $\boldsymbol{Q}$, $\boldsymbol{K}$, $\boldsymbol{V}$ come from unified representations of previous layer, i.e., $\hat{\boldsymbol{H}}_\tau^{l-1}$. In this case, contextual representations of the τ-th passage are denoted as:

$$\boldsymbol{H}_\tau^l = \text{TRM}(\hat{\boldsymbol{H}}_\tau^{l-1}, \hat{\boldsymbol{H}}_\tau^{l-1}, \hat{\boldsymbol{H}}_\tau^{l-1}). \tag{6}$$

3.4 Inter-passage Attention Sub-Layer

This sub-layer models relationships among selected pivot tokens. Their initial graph representations at the l-th layer are extracted from corresponding hidden states of intra-passage attention layer, which we denote as $\boldsymbol{H}_P^l \in \mathbb{R}^{n_p\times E}$, $\boldsymbol{H}_S^l \in \mathbb{R}^{n_s\times E}$, $\boldsymbol{H}_T^l \in \mathbb{R}^{n_t\times E}$ for passage, sentence and token-level graphs respectively. To handle cases where entities or terms are comprised of multiple subtokens, we use the mean pooling of subwords to denote the embedding.

We also follow a multi-head attention based neighborhood aggregation strategy, which is consistent with the mechanism in §3.3. Note that the intra-passage attention defined in Eq.(3) can be viewed as a complete graph, whereas inter-passage attention are built on the pre-defined adjacency matrix (§3.1.3). Taking passage-level graph for instance, the attention function should be modified as,

$$\text{Att}\left(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}\right) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{E}} - \left(1 - \boldsymbol{A}^{\text{p2p}}\right)C\right)\boldsymbol{V}, \tag{7}$$

where C is a large constant to mask out illegal connections (setting to $-\infty$). Thus, representations of passage-level graphs are then updated as follows:

$$\boldsymbol{G}_P^l = \text{TRM}(\boldsymbol{H}_P^l, \boldsymbol{H}_P^l, \boldsymbol{H}_P^l). \tag{8}$$

Attention for sentence and passage level graph is similar. We use two sets of projections, to compute attention scores of intra-passage attention and inter-passage attention, to provide flexibility to model the different types of attention. We also experiment with other variants, such as GCN [20] and GIN aggregator [35] (§5.3.3).

3.5 Multi-view Aggregation Layer

After stacking L layers of attention, this layer analyzes how to summarize outputs into an overall document representation. Prior works, such as PARADE [23], simpy combine representations corresponding to `[PSG]`. Nevertheless, additionally exploiting features from other views may enrich the representational power and generalize better(§5.4). Hence, we apply pooling at multiple levels of granularity, to generate global document-aware vector from passage view ($\mathbf{o}_P$), sentence view ($\mathbf{o}_S$) and token view ($\mathbf{o}_T$). They are aggregated to the document representation as,

$$\mathbf{d} = \boldsymbol{W}^A\left[\mathbf{o}_P; \mathbf{o}_E; \mathbf{o}_T\right], \tag{9}$$

where $\boldsymbol{W}^A \in \mathbb{R}^{E\times 3E}$ denotes the projection matrix.

Taking passage-view pooling for instance, $\mathbf{o}_P$ are obtained as,

$$\mathbf{o}_P = \text{MHP}\left(\mathbf{w}_P, \boldsymbol{G}_P^L, \boldsymbol{G}_P^L\right), \tag{10}$$

where $\mathbf{w}_P \in \mathbb{R}^E$ is a weight vector learned during training.

Formally, the single-head based pooling function is computed as,

$$\text{Pool}\left(\mathbf{w}_P, \boldsymbol{G}_P^L, \boldsymbol{G}_P^L\right) = \left(\boldsymbol{G}_P^L\right)^\top \text{softmax}\left(\boldsymbol{G}_P^L\,\mathbf{w}_P\right). \tag{11}$$

Similar to Eq. (4), to make it more expressive, we further extend to multi-head pooling (MHP) by linearly projecting them h times to yield h outputs from different subspaces. The output is denoted as,

$$\begin{aligned} &\text{MHP}\left(\mathbf{w}_P, \boldsymbol{G}_P^L, \boldsymbol{G}_P^L\right) = \boldsymbol{W}^O\left[\mathbf{o}_i; \cdots; \mathbf{o}_h\right], \\ &\qquad \text{where } \mathbf{o}_i = \text{Pool}\left(\boldsymbol{W}_i^q\,\mathbf{w}_P, \boldsymbol{G}_P^L\,\mathbf{W}_i^K, \boldsymbol{G}_P^L\,\mathbf{W}_i^V\right), \end{aligned} \tag{12}$$

where $\boldsymbol{W}_i^q \in \mathbb{R}^{E/h\times E}$, $\boldsymbol{W}_i^K \in \mathbb{R}^{E\times E/h}$ and $\boldsymbol{W}_i^V \in \mathbb{R}^{E\times E/h}$ are projection matrices for the i-th subspace.

3.6 Training

For each query q, we select a positive document and several negative documents, to form a group G_q. We place a linear layer on top of the document representation to obtain the relevance score, and define the contrastive loss [12] for one query q as,

$$\mathcal{L}_q := -\log\frac{\exp\left(\mathbf{v}^\top\mathbf{d}^+\right)}{\sum_{d\in G_q}\exp\left(\mathbf{v}^\top\mathbf{d}\right)}, \tag{13}$$

where $\mathbf{v} \in \mathbb{R}^E$ is the weight vector that projects the representation into a scalar score. $\mathbf{d}$ is the document representation for a document $d \in G_q$ and $\mathbf{d}^+ \in \mathbb{R}^E$ denotes the document representation for the selected positive sample.

4 EXPERIMENTAL DESIGN

4.1 Datasets and Baselines

Following prior work [16], we use two query sets: (1) **MS MARCO**: It consists of 3.2 million documents [7] with 367,013 training queries.

The official evaluation metric is MRR, we also report MAP and nDCG. (2) **TREC 2019 Deep Learning Track**: It uses the same document collection and its test set consists of 43 queries. The official evaluation metric is nDCG@10, we also report nDCG@100 and MAP. We compare MIR against lots of neural baselines.

BM25 [27] is a widely-used unsupervised bag-of-words retrieval model based on IDF-weighted counting.

BERT-FirstP [8] predicts the relevance of each passage independently and uses the score of the first passage as relevance score.

BERT-MaxP [8] encodes short paragraphs with BERT and combines scores with a max-pooling layer.

IDCM [16] uses an intra-document cascade ranking model with a fast passage selection module for efficiency.

PARADE [23] generates an overall document representation to obtain the relevance score by aggregating passage representations. **PARADE$_{\text{Max}}$** utilizes a max pooling operation. **PARADE$_{\text{TF}}$** applies the transformer to passage representations.

Transformer-XH [42] models a group of text sequences and aggregates them with an extra-hop attention layer.

Longformer [3] uses a combination of a windowed local-context self-attention and an task motivated global attention. Here, we report the results of two variants, i.e., whether or not global attention is provided on all question tokens, which we refer as "**Longformer (+global)**" and "**Longformer**" respectively.

QDS-Transformer [18]: It further tailors Longformer to the ranking task with query-directed sparse attention.

4.2 Training Configurations

We use the first stage retrieval results open sourced by HDCT [9]. All models are trained for two epochs with a batch size of 16. During training, for each query, we use one positive samples and seven negative samples randomly sampled from top 100 documents ranked by HDCT. We use Adam [19] with learning rate of 1e-5, l_2 weight decay of 0.01, learning rate warmup over the first 10% of steps. For PRF-based term importance estimation (§3.1.2), we follow the setting in the original paper [9] and select at most 10 unique terms. Following [16], max input length is set to 2,048 tokens. Through statistical analysis, pre-constructed graphs of nearly 90% documents have less than 64 sentence-level nodes and 256 token-level nodes. Hence, we limit the maximum nodes in the sentence graph to 64 and the maximum nodes in token graph to 256. We set the number of layers L to 12 and intra-passage attention layers in MIR are initialized by BERT base model. Parameters of inter-passage attention layer are trained from scratch. Unless otherwise specified, the rest of paper report results with a window size of 128 and an overlapping factor of $\alpha = 0.25$. Larger window size does not further improves the effectiveness, also stated in [16, 18].

5 EXPERIMENTAL RESULTS

5.1 Overall Results

Table 1 summarizes experimental results. In general, we find that MIR significantly outperforms existing models under different settings of document lengths. More observations are as follows.

(1) Compared to models within 512 tokens, **even with a smaller window size (i.e., 128), MIR using inter-passage attention performs as powerfully as BERT-FirstP (Line 4 vs. 2).** In contrast, due to the lack of inter-passage attention, BERT-MaxP with the window size of 128 is inferior to FirstP (Line 3 vs. 2). It confirms that the inter-passage attention is helpful to capture the global context and preserve the expressivity of the quadratic full Transformers.

(2) Compared to models beyond 512 tokens, **MIR outperforms existing approaches in terms of all evaluation metrics.** MIR improves performance with a large margin over passage-level methods PARADE and Transformer-XH, which confirms the effectiveness of multi-view inter-passage interactions. Concretely, on MS MARCO, its NDCG@10 has 2.1% absolute improvement over BERT-MaxP, 1.3% over PARADE$_{\text{TF}}$, and 1.4% over Transformer-XH. Furthermore, MIR outperforms longformer and QDS-Transformer by a remarkable margin (Line 13 vs. 11). This progress can be attributed to the design of multi-view inter-passage graphs, which models more comprehensive and informative relationships.

(3) Among all passage-level models, **the ones adopting representation aggregation perform better.** Specifically, PARADE-style methods surpass score aggregation methods such as BERT-MaxP (Line 8 vs. 5). We argue that aggregating passage-level representations in a lightweight layer can alleviate the severe lack of document-level context and lead to gains. However, merely interacting multiple passages in the final layer is insufficient to model inter-passage relationships. Hence, MIR further achieves significant improvements compared to representation aggregation baselines.

(4) **Exploiting longer document input (512 vs. 2K) consistently brings a substantial boost in performance**. Specifically, MIR 2K setting outperforms 512 setting by a remarkable margin (Line 13 vs. 4). Similar patterns are observed with BERT-FirstP and BERT-MaxP(Line 3 vs. 5). This indicates longer context contains more abundant information, which benefits the text understanding and thus improves the effectiveness.

5.2 Efficiency-Effectiveness Analysis

We also evaluate how the efficiency (y-axes) and the effectiveness (x-axes) vary with the length (the maximum tokens we make use of) from 512 to 2K. All experiments are conducted on an Nvidia DGX-1 server with 512 GB memory and a single Tesla V100 GPU using fp16 training. We compare MIR with BERT-MaxP using the same window size of 128. In Figure 5, as expected, modeling interactions brings computational cost and performance improvement at the same time. However, **MIR with the length of 1024 or 1280 can exceed the best effectiveness of BERT-MaxP with 2048 tokens and provide a higher throughput simultaneously (highlighted in red)**. Moreover, MIR shows huge advantages over longformer in both effectiveness and efficiency. This is due to the fact that global tokens attend to all tokens across the sequence. On the contrary, pivot tokens only interact with each other, which reduces the complexity and improves efficiency. In summary, MIR with 1024 tokens

Table 1: Evaluation results for two benchmark. “†” denotes MIR is significantly better than other methods from the same setting in t-test with $p < 0.05$ level. Best results in each setting are in bold. The second best results in each setting are underlined.

Doc. Length	Window Size		Model	TREC DL 2019			MSMARCO		
				nDCG@10	nDCG@100	MAP@100	nDCG@10	MRR@10	MAP@100
-	-	1	BM25	0.488	0.501	0.234	0.311	0.252	0.265
512	512	2	BERT-FirstP	0.652	0.537	0.256	0.497	0.425	0.432
512	128	3	BERT-MaxP	0.634	0.531	0.246	0.477	0.409	0.417
		4	MIR (ours)	**0.649**†	**0.547**†	**0.257**†	**0.498**†	**0.427**†	**0.431**†
2K	128	5	BERT-MaxP	0.657	0.547	0.259	0.497	0.427	0.434
		6	IDCM	0.665	0.567	0.265	0.497	0.426	0.436
		7	PARADE$_{\text{Max}}$	0.659	0.547	0.267	0.503	0.429	0.437
		8	PARADE$_{\text{TF}}$	0.660	0.554	0.271	0.505	0.435	0.441
		9	Transformer-XH	0.656	0.566	0.274	0.504	0.434	0.439
		10	Longformer	0.659	0.554	0.265	0.487	0.419	0.426
		11	+ global	0.668	0.562	0.274	0.505	0.434	0.441
		12	QDS-Transformer	0.667	0.560	0.278	0.504	0.435	0.440
		13	MIR (ours)	**0.697**†	**0.578**†	**0.294**†	**0.518**†	**0.447**†	**0.454**†

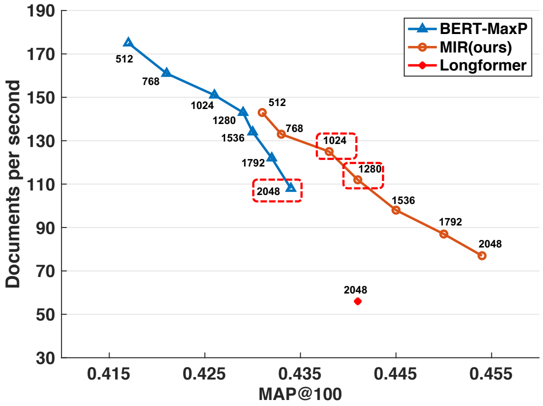


Figure 5: Throughout and MAP@100 on MS MARCO.

is preferable in efficiency-sensitive scenarios, while MIR with 2048 token further improves results at the expense of computation.

5.3 Effects of Multi-view Attention Patterns

5.3.1 Different attention patterns. Table 2 explores the influence of attention patterns with different levels of granularity. As expected, each individual pattern contributes to the whole. Specifically, removing the token-level links causes the most decline, which confirms their indispensable role in incorporating fine-grained dependency signals into contextual representations. Without inter-passage interactions, the contextualization is limited to the current passage. Therefore, the model fails to take full advantage of rich contextual information and drops significantly in terms of all metrics.

5.3.2 Different model variations. In the middle of Table 3, we test three variants: (1) **shared**: all parameters for the inter-passage and intra-passage attention are shared; (2) **without stacks**: the iterative stacking pattern is removed and only one inter-passage attention layer is placed on top of consecutive intra-passage attention layers; (3) **fully-connected**: all pairs of pivot tokens are connected.

Table 2: Ablations of attention patterns on MS MARCO. “w/o graph” refers to removing all inter-passage interactions.

Models	nDCG@10	MRR@10	MAP@100
MIR	**0.518**	**0.447**	**0.454**
w/o p2p	0.510 (-0.8%)	0.443 (-0.6%)	0.448 (-0.6%)
w/o s2s	0.511 (-0.7%)	0.439 (-0.8%)	0.447 (-0.7%)
w/o t2t	0.506 (-1.2%)	0.437 (-1.0%)	0.443 (-1.1%)
w/o graph	0.504 (-1.4%)	0.427 (-2.0%)	0.436 (1.8%)

Table 3: Results with different attention configurations on MS MARCO. Middle:different model variations. Bottom: different neighborhood aggregators.

Models	nDCG@10	MRR@10	MAP@100
MIR	**0.518**	**0.447**	**0.454**
shared	0.499 (-1.9%)	0.428 (1.9%)	0.435 (1.9%)
w/o stacks	0.497 (-2.1%)	0.429 (1.8%)	0.435 (-1.9%)
full-connected	0.511(-0.7%)	0.439(-0.8%)	0.447(0.8%)
MIR-GCN	0.509 (-0.9%)	0.439 (1.8%)	0.445 (-0.9%)
MIR-GIN	0.508 (-1.0%)	0.441 (-0.6%)	0.446 (-0.8%)
MIR-GraphSAGE	0.489 (-2.9%)	0.420 (2.7%)	0.426 (2.8%)

We observe that all variants hurt the performance. Sharing parameters significantly make the performance worse, which shows that empowering model to maintain dedicated representations for different types of attention is critical. Without the iterative stacking pattern, the information flow across multiple passages is insufficient and the performance drops sharply. The result degradation caused by full-connected graph shows explicit graph structure imposes constraints on relations (e.g., semantic similarities and discourse relations) and MIR benefits from such relational inductive bias.

5.3.3 Different neighborhood aggregators. We use attention to convey messages in the graph (§3.4). In the bottom of Table 3, we examine three alternatives: (1) **GCN** [20]: the convolutional propagation rule is applied; (2) **GraphSAGE** [14]: we use an elementwise max-pooling operation after transformation; (3) **GIN** [35]: we use multi-layer perceptrons to model relationships. The results show that replacing attention with three variants all harms the effectiveness. This verifies that attention, equipped with the multi-head strategy, has the strongest representational power. In comparison, GraphSAGE using max pooling severely underfits. The performance deterioration is due to the fact that simple pooling falls short in selecting salient information and characterizing semantic features.

5.4 Effects of Multi-view Aggregation

5.4.1 Effects of multiple views. In the middle of Table 4, we inspect the effect of different views on the aggregation layer. Analogous trends hold for it. Results deteriorate after removing each view, verifying the necessity of summarizing documents from multiple views. Moreover, we observe that the influence of removing token view is relatively smaller. We argue that maybe the token-level semantics have been absorbed into the summary tokens. Compared to utilizing [PSG] purely, the supplementation of another two views further enrich the representation and improve the results.

5.4.2 Different pooling schemes. We use attention based pooling to do aggregation (§3.5). In the bottom of Table 4, we experiment with different aggregation methods following [23]: (1) **MIR-Max**: element-wise max pooling operation are utilized; (2) **MIR-CNN**: multiple Convolutional Neural Network (CNN) [22] layers are stacked; (3) **MIR-Transformer**: all embeddings are fed into a Transformer layer. We observe that max-pooling causes a decline in performance, which confirms that max-pooling performs poorly in identifying salient information. Unexpectedly, other complex (i.e, Transformers) or hierarchical methods (i.e., CNN) almost yield no additional gain. We argue that stacked attention layers have thoroughly promoted inter-passage interactions and the representations have been aware of the global context. Additionally exploring dependencies in the aggregation layer may not obtain improvements.

5.5 Experiment with Document Lengths

According to the length L of corresponding positive documents, we divide the whole query set on MS MARCO to four groups: (a) All; (b) $0 < L \leq 512$; (c) $512 < L \leq 2K$; (d) $L > 2K$. From Figure 6, we have the following observations: **(1)** The gap between MIR and others is widening when changing groups from (b) to (c). The reason is that methods except MIR limit the contextualization within the current passage and fail to utilize longer-range semantic relationships. Additionally capturing dependencies across all segments makes MIR more competitive under the longer document setting; **(2)** Surprisingly, though Group (b) mainly tests short-term dependency, MIR dramatically improves the BERT-FirstP from 47.1% to 47.6%.

Table 4: Results with different aggregation configurations on MS MARCO. Middle: ablations studies on multiple views. Bottom: different pooling schemes.

Models	nDCG@10	MRR@10	MAP@100
MIR	0.518	**0.447**	**0.454**
w/o passage	0.513 (-0.5%)	0.441 (-0.6%)	0.448 (-0.6%)
w/o sentence	0.514 (-0.4%)	0.442 (-0.5%)	0.447 (-0.7%)
w/o token	0.515 (-0.3%)	0.445 (-0.2%)	0.451 (-0.3%)
only passage	0.509 (-0.9%)	0.439 (-0.8%)	0.445 (-0.9%)
MIR-Max	0.511 (-0.7%)	0.440 (-0.7%)	0.449 (-0.5%)
MIR-CNN	**0.519** (+0.1%)	0.446 (-0.1%)	0.454 (0)
MIR-Transformer	0.518 (0)	0.447 (0)	0.453 (-0.1%)

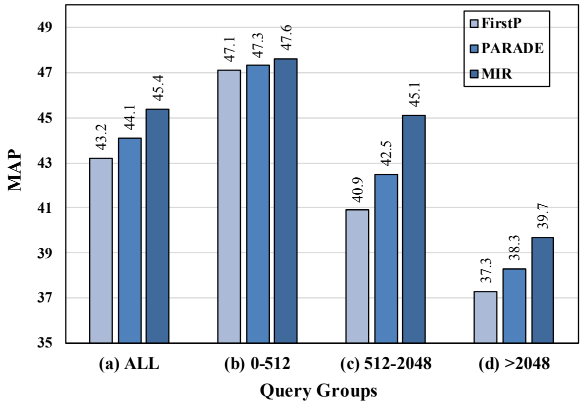


Figure 6: Performance on different query groups.

We hypothesis that learning from longer contexts strengthens the ability of MIR to capture more general matching patterns.

6 CONCLUSION

Existing passage-level methods limit the contextualization within the current passage and fail to utilize longer-range relationships. In this work, we present MIR, a multi-view inter-passage interaction based ranking model. It integrates intra-passage and inter-passage interactions into a unified framework, which enable global document-aware contextual representations. Moreover, we design multi-view inter-passage attention patterns, to help capture syntactic features with token-level interactions and global coherency with higher-level interactions. We conduct extensive experiments to verify the effectiveness of modeling multi-view inter-passage interactions. In the future, we will explore more sophisticated relation patterns that particularly tailored to document ranking tasks.

ACKNOWLEDGEMENT

This work was supported by National Natural Science Foundation of China No. 61872370, and Intelligent Social Governance Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China.

REFERENCES

[1] Qingyao Ai, Brendan O'Connor, and W Bruce Croft. 2018. A neural passage model for ad-hoc document retrieval. In European Conference on Information Retrieval. Springer, 537–543.

[2] Joshua Ainslie, Santiago Ontanon, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. 2020. ETC: Encoding Long and Structured Inputs in Transformers. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, Online, 268–284. https://doi.org/10.18653/v1/2020.emnlp-main.19

[3] Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. arXiv preprint arXiv:2004.05150 (2020).

[4] James P Callan. 1994. Passage-level evidence in document retrieval. In SIGIR'94. Springer, 302–310.

[5] Tianqi Chen, Thierry Moreau, Ziheng Jiang, Lianmin Zheng, Eddie Yan, Haichen Shen, Meghan Cowan, Leyuan Wang, Yuwei Hu, Luis Ceze, et al. 2018. {TVM}: An automated end-to-end optimizing compiler for deep learning. In 13th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 18). 578–594.

[6] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. arXiv preprint arXiv:1904.10509 (2019).

[7] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Ellen M Voorhees. 2019. Overview of the trec 2019 deep learning track. TREC (2019).

[8] Zhuyun Dai and Jamie Callan. 2019. Deeper text understanding for IR with contextual neural language modeling. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. 985–988.

[9] Zhuyun Dai and Jamie Callan. 2020. Context-aware document term weighting for ad-hoc search. In Proceedings of The Web Conference 2020. 1897–1907.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[11] Paolo Ferragina and Ugo Scaiella. 2010. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In Proceedings of the 19th ACM international conference on Information and knowledge management. 1625–1628.

[12] Luyu Gao, Zhuyun Dai, and Jamie Callan. 2021. Rethink training of BERT rerankers in multi-stage retrieval pipeline. European Conference on Information Retrieval (2021).

[13] Ankit Gupta and Jonathan Berant. 2020. Gmat: Global memory augmentation for transformers. arXiv preprint arXiv:2006.03274 (2020).

[14] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In Proceedings of the 31st International Conference on Neural Information Processing Systems. 1025–1035.

[15] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. Neural computation 9, 8 (1997), 1735–1780.

[16] Sebastian Hofstätter, Bhaskar Mitra, Hamed Zamani, Nick Craswell, and Allan Hanbury. 2021. Intra-Document Cascading: Learning to Select Passages for Neural Document Ranking. SIGIR (2021).

[17] Sebastian Hofstätter, Hamed Zamani, Bhaskar Mitra, Nick Craswell, and Allan Hanbury. 2020. Local self-attention over long text for efficient document retrieval. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2021–2024.

[18] Jyun-Yu Jiang, Chenyan Xiong, Chia-Jung Lee, and Wei Wang. 2020. Long Document Ranking with Query-Directed Sparse Transformer. EMNLP Findings (2020).

[19] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014).

[20] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. ICLR (2016).

[21] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. ICLR (2020).

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems 25 (2012), 1097–1105.

[23] Canjia Li, Andrew Yates, Sean MacAvaney, Ben He, and Yingfei Sun. 2020. PARADE: Passage representation aggregation for document reranking. arXiv preprint arXiv:2008.09093 (2020).

[24] Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, Nazli Goharian, and Ophir Frieder. 2020. Efficient document re-ranking for transformers by precomputing term representations. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. 49–58.

[25] Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. 2019. CEDR: Contextualized embeddings for document ranking. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. 1101–1104.

[26] Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage Re-ranking with BERT. arXiv preprint arXiv:1901.04085 (2019).

[27] Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. Now Publishers Inc.

[28] Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. 2021. Efficient Content-Based Sparse Attention with Routing Transformers. Transactions of the Association for Computational Linguistics 9 (2021), 53–68. https://doi.org/10.1162/tacl_a_00353

[29] Koustav Rudra and Avishek Anand. 2020. Distant Supervision in BERT-based Adhoc Document Retrieval. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management. 2197–2200.

[30] Sainbayar Sukhbaatar, Edouard Grave, Piotr Bojanowski, and Armand Joulin. 2019. Adaptive Attention Span in Transformers. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Florence, Italy, 331–335. https://doi.org/10.18653/v1/P19-1032

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in neural information processing systems. 5998–6008.

[32] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2021. Hi-Transformer: Hierarchical Interactive Transformer for Efficient and Effective Long Document Modeling. ACL (2021).

[33] Zhijing Wu, Jiaxin Mao, Yiqun Liu, Jingtao Zhan, Yukun Zheng, Min Zhang, and Shaoping Ma. 2020. Leveraging passage-level cumulative gain for document ranking. In Proceedings of The Web Conference 2020. 2421–2431.

[34] Zhijing Wu, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2019. Investigating passage-level relevance and its role in document-level relevance judgment. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. 605–614.

[35] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How powerful are graph neural networks? ICLR (2019).

[36] Liu Yang, Mingyang Zhang, Cheng Li, Michael Bendersky, and Marc Najork. 2020. Beyond 512 tokens: Siamese multi-depth transformer-based hierarchical encoder for long-form document matching. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management. 1725–1734.

[37] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies. 1480–1489.

[38] Zeynep Akkalyoncu Yilmaz, Shengjin Wang, Wei Yang, Haotian Zhang, and Jimmy Lin. 2019. Applying BERT to document retrieval with birch. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations. 19–24.

[39] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big Bird: Transformers for Longer Sequences.. In NeurIPS.

[40] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2020. An analysis of BERT in document ranking. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. 1941–1944.

[41] Xingxing Zhang, Furu Wei, and Ming Zhou. 2019. HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. arXiv preprint arXiv:1905.06566 (2019).

[42] Chen Zhao, Chenyan Xiong, Corby Rosset, Xia Song, Paul Bennett, and Saurabh Tiwary. 2020. Transformer-xh: Multi-evidence reasoning with extra hop attention. ICLR (2020).

[43] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In Proceedings of AAAI.

[44] Jie Zhou, Xu Han, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2019. GEAR: Graph-based evidence aggregating and reasoning for fact verification. ACL (2019).