Latent Graph Recurrent Network for Document Ranking

Qian Dong[1,2(✉)] and Shuzi Niu[1(✉)]

[1] Institute of Software, Chinese Academy of Sciences, Beijing, China
dongqian19@mails.ucas.ac.cn, shuzi@iscas.ac.cn
[2] University of Chinese Academy of Sciences, Beijing, China

Abstract. BERT based ranking models are emerging for its superior natural language understanding ability. The attention matrix learned through BERT captures all the word relations in the input text. However, neural ranking models focus only on the text matching between query and document. To solve this problem, we propose a graph recurrent neural network based model to refine word representations from BERT for document ranking, referred to as **L**atent **G**raph **Re**current Network (LGRe for short). For each query and document pair, word representations are learned through transformer layer. Based on these word representations, we propose masking strategies to construct a bipartite-core word graph to model the matching between the query and document. Word representations will be further refined by graph recurrent neural network to enhance word relations in this graph. The final relevance score is computed from refined word representations through fully connected layers. Moreover, we propose a triangle distance loss function for embedding layers as an auxiliary task to obtain discriminative representations. It is optimized jointly with pairwise ranking loss for ad hoc document ranking task. Experimental results on public benchmark TREC Robust04 and WebTrack2009-12 test collections show that LGRe (The implementation is available at https://github.com/DQ0408/LGRe) outperforms state-of-the-art baselines more than 2%.

Keywords: Ad hoc retrieval · Graph neural network · Transformer

1 Introduction

Neural ranking models focus on learning query-document matching patterns, i.e., knowledge about search tasks. Recently, pretrained neural language models learn such knowledge from an extensive text collection and provide new opportunities for document ranking.

Word embedding [17] applied to document ranking is pretrained with a large corpus based on word co-occurrences within a window of the input text. Unlike such word embedding only encoding the local context, word representations learned from BERT are a function of the entire input text. Taking the concatenation of query and document as input, BERT is naturally fit for the search

C. S. Jensen et al. (Eds.): DASFAA 2021, LNCS 12682, pp. 88–103, 2021.
https://doi.org/10.1007/978-3-030-73197-7_6

task. The reason lies in that the attention matrix contains query-document interaction on the word level. In this sense, BERT based ranking model belongs to interaction based neural ranking models [7].

However, interaction based models only care for matching patterns between query and document. The attention matrix learned from BERT brings about additional word relations in the query-document matching process, such as query-query and document-document word relations. Whether these additional word relations are useful to derive the query-document relevance pattern remains unknown. One interesting observation is that long natural language queries perform better than short keyword queries for BERT based ranking models in document ranking tasks [2]. The problem that additional relations are dominant in the attention matrix is more serious for short queries.

To solve this problem, we propose a graph recurrent neural network based method to refine word embedding learned from BERT in the document ranking task. For each query and document pair, BERT first takes their concatenation as input and obtain word representations. Then, a masking method is adopted to construct a bipartite-core word graph as the matching between query and document, which is a masked attention matrix derived from learned word representations. Distinguished from explicitly defined graphs, the latent graph is learned from BERT for the following word representation refinement layer. To enhance word relations in this graph, word representations are updated through a gated recurrent unit. The final query and document pair representation is summarized from refined word representations and used for prediction. Pairwise ranking loss is a function of relevance scores. Moreover, a triangle distance loss is proposed as function of query, document and query-document pair representations to learn discriminative representations. Both loss functions are optimized jointly in an end-to-end manner. Experiments on public benchmark datasets Robust04 and WebTrack2009-12 are conducted to show the effectiveness of LGRe. Detailed implementations are further analyzed in experiments, such as the effect of additional word relations on query-document relations.

To sum up, our major contributions lie in the following aspects:

- We explore masking strategies applicable to building a bipartite-core word graph as the matching between query and document.
- We refine word representations on this graph through graph recurrent neural network to alleviate useless word relation's effect.
- We propose a triangle distance loss function for the embedding layer, which helps learn discriminative representations for the downstream ranking task.

2 Related Work

Here we briefly review some related studies in terms of neural ranking models without BERT, BERT based ranking models and Graph neural network.

2.1 Interaction Based Neural Ranking Models

Interaction based neural ranking models assume that relevance is in essence about the relation between input texts and it is more effective to learn from interactions rather than individual representations. They focus on designing the interaction function to produce the relevance score. Existing interaction functions are divided into two kinds: non-parametric and parametric interaction functions [7].

Traditional non-parametric interaction functions includes binary indicator, cosine similarity, dot product and radial basis function so on. DRMM [6] converts a local interaction matrix for the query-document word pair to a fixed-length matching histogram for relevance matching. MatchPyramid [16] produces a query-document relevance score by convolution operations over a query-document similarity matrix. Parametric interaction functions are to learn the similarity/distance function from data. For example, Conv-KNRM [4] uses convolutional neural network to represent n-grams of various lengths, matches them in a unified embedding space for the kernel pooling and learning-to-rank layers to generate the final ranking score. Arc-II [8] performs convolution and pooling on the word interaction between two sentences. In this sense, BERT based ranking models can also be treated as parametric interaction based neural ranking models.

2.2 Pretrained Neural Language Models for IR

Pretrained Neural Language Models (PNLM), e.g., BERT [5] and ELECTRA [1], have achieved state-of-the-art results in many NLP tasks. As mentioned above, it works for the ad hoc ranking because the attention matrix in BERT can be regarded as an interaction function. BERT based ranking models have been shown to be superior to neural ranking models without BERT.

BERT-MaxP [2] splits a document into overlapping passages. The neural ranker predicts the relevance score of each passage independently. Document score is the score of the best passage. CEDR [14] incorporates BERT's classification vector into existing neural models, such as DRMM [6] and Conv-KNRM [4]. PARADE [10] leverages passage-level representations to predict a document relevance score without passage independence assumption. Rather than fine tuning BERT-base on a Bing search log, PARADE improves performance by fine tuning on the MSMARCO passage ranking dataset. Other researches focus on how to improve the efficiency of PNLM in retrieval tasks. PreTTR [13] precomputes part of the document term representations at indexing time, and merge them with the query representation at query time to compute the final ranking score. DeepCT [3] maps the contextualized term representations from BERT into context-aware term weights for efficient passage retrieval.

2.3 Graph Neural Network

Graph neural network (GNN) has recently been widely studied in many fields because of its high-order relation capture ability. The information propagation

step is key to obtain the hidden states of nodes (or edges) for GNN. According to different information propagation methods, GNN can be divided into convolution based, attention based and recursive based models so on [20]. Convolution based GNN, extending convolution operation to the graph domain, includes spectral approaches and spatial approaches. Through the attention mechanism, attention based GNN focuses on important nodes in the graph and important information of these nodes for the sake of improving the signal-to-noise ratio of the original data [18]. Recursive based GNN attempts to use the gate mechanism like GRU [11] in the propagation step to improve the long-term propagation of information across the graph structure.

3 Method

We first formalize the ad hoc document retrieval task. To overcome the inherent weakness of BERT for ranking, the network architecture of our proposed LGRe is described. Additionally, we propose a triangle distance loss function for better representations to aid the downstream ranking task. Both the triangle distance and pairwise ranking loss functions are optimized jointly.

3.1 Formalization

Ad hoc document retrieval task is to produce the ranking of documents in a corpus given a short query. There are Q queries $\{q_i\}_{i=1}^{Q}$ for training. Each query q is represented as a word sequence $s^q = \mathrm{w}_1^q, \mathrm{w}_2^q, \ldots, \mathrm{w}_m^q$ and also associated with a document set $D_q = \{(d_j, y_j)\}_{j=1}^{n_q}$. $y_j \in \{0, 1\}$ is the ground truth relevance label of document d_j. Non-relevant documents from D_q are denoted as D_q^- ($|D_q^-| = n_q^-$) and relevant documents denoted as D_q^+ ($|D_q^+| = n_q^+$). Document $d \in D_q$ is denoted as a word sequence $s^d = \mathrm{w}_1^d, \mathrm{w}_2^d, \ldots, \mathrm{w}_n^d$. How to model the text matching between query and document is key to neural ranking models.

3.2 Architecture

As mentioned before, BERT has such a natural advantage to become a ranker that its learned attention matrices model the query and document interaction. However, its disadvantage is also evident that these learned attention matrices describe all possible word relations without emphasis on the query-document word relations. To solve this problem, we mask these unnecessary word relations and introduce a refinement process over this masked graph. Through this refinement process, word representations will be more suitable to derive the relevance score between query and document. The whole architecture is depicted in Fig. 1.

Transformer Layer. For each query-document pair (q, d), two word sequences are concatenated, i.e. $s^{(q,d)} = [[\mathrm{CLS}], s^q, [\mathrm{SEP}], s^d, [\mathrm{SEP}]]$. Its input embedding $\mathbf{I}^{(q,d)}$ is derived from the sum of the word embedding and its corresponding position embedding. Then $\mathbf{I}^{(q,d)}$ is fed into BERT stacked with L identical layers. For

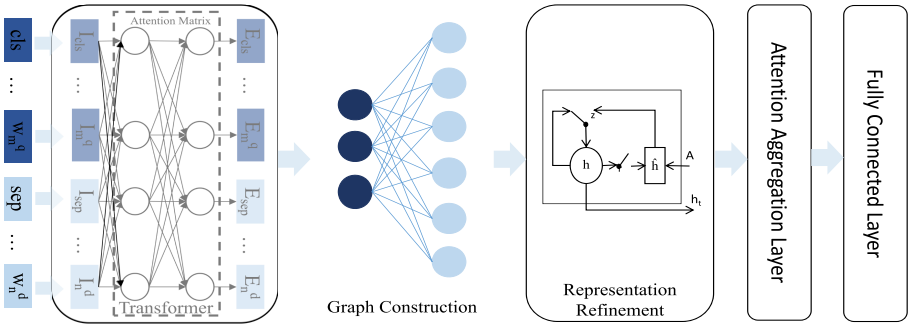


Fig. 1. Latent graph recurrent network architecture

example, $L = 12$ in BERT-base. For each word i at each layer $l = 1, \ldots, L$, its word representation $\mathbf{E}_l^{(q,d)}(i) \in \mathbb{R}^{d_k}$ is obtained by weighted summing representations of the other words in Eq. (2), d_k is the dimension of word representations.

$$\mathcal{A}_{l-1}^{(q,d)} = \text{softmax}(\frac{(\mathbf{W}_B\mathbf{E}_{l-1}^{(q,d)})(\mathbf{W}_B\mathbf{E}_{l-1}^{(q,d)})'}{\sqrt{d_k}}) \tag{1}$$

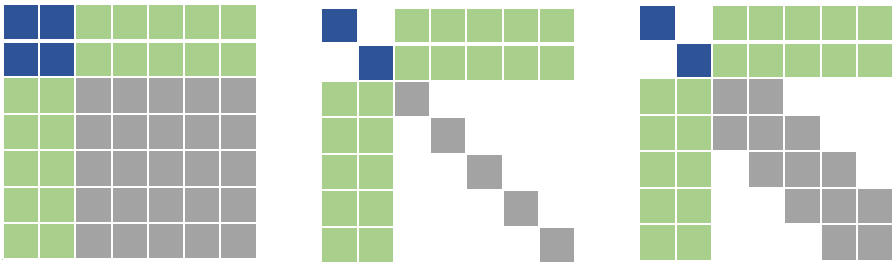
$$\mathbf{E}_l^{(q,d)}(i) = \mathbf{E}_{l-1}^{(q,d)}(i) + \sum_j \mathcal{A}_{l-1}^{(q,d)}(i,j)\mathbf{E}_{l-1}^{(q,d)}(j) \tag{2}$$

where $\mathcal{A}_{l-1}^{(q,d)}$ is the attention matrix learned in the $l-1$-th layer and $\mathbf{E}_0^{(q,d)} = \mathbf{I}^{(q,d)}$. Through this layer, we obtain L attention matrices $\{\mathcal{A}_l^{(q,d)}\}_{l=1}^L$ for the query-document pair (q, d). Each attention matrix $\mathcal{A}_l^{(q,d)}$ naturally models the query-document word interaction.

Bipartite-Core Word Graph Construction. These attention matrices also contain query-query and document-document word interaction, which are not obviously useful for query-document matching in the current ranking task. Particularly, some studies [2] show that these additional interactions may harm the retrieval performance. Here we propose to mask some relations to build a bipartite-core word graph to model the text matching between query and document. Intuitively, there are three different implementations. One extreme case is to keep all the word relations in the attention matrix without masking, which can be treated as the summation of query-document bipartite word adjacent matrix, full document word adjacent matrix and full query word adjacent matrix, referred to as *Full Word Graph*. The other extreme case is to keep only query-document bipartite word relations in the attention matrix and mask these other relations, namely *Query-Document Bipartite Word Graph*. In the middle, we keep some document neighbor word relations and query-document bipartite word relations in the attention matrix and obtain the summation of query-document bipartite word adjacent matrix and document neighbor word

adjacent matrix, namely *Query-Document Bipartite and Neighbor Word Graph*. All three kinds of graphs are bipartite-core graphs. For each transformer layer l, we define a masking matrix $\mathcal{M}_l^{(q,d)}$ for each implementation, and the masked word adjacent matrix of the bipartite core graph is derived as Eq. (3), where ϵ is small enough.

$$\hat{\mathcal{A}}_l^{(q,d)} = \text{softmax}(\frac{(\mathbf{W}_A\mathbf{E}_l^{(q,d)})(\mathbf{W}_A\mathbf{E}_l^{(q,d)})'}{\sqrt{d_k}} + \epsilon(1 - \mathcal{M}_l^{(q,d)})) \tag{3}$$



(a) Full (b) Bipartite (c) Bipartite+Neighbor

Fig. 2. Bipartite-core word graphs constructed from three strategies. Blue, green and grey color represent the word attention score between query and query, document and document, query and document separately. White means no word relation. (Color figure online)

- *Full Word Graph*. We keep all the word relations in $s^{(q,d)}$. No relations will be masked, so $\mathcal{M}_l^{(q,d)} = \mathbf{1}_{(m+n+3)\times(m+n+3)}$, where $\mathbf{1}_{(m+n+3)\times(m+n+3)}$ is a matrix with all elements 1. The masked attention matrix is shown in Fig. 2(a).
- *Query-Document Bipartite Word Graph*. In terms of query-document text matching, there are two types of words. We only keep all the relations between two types of words. As shown in Fig. 2(b), word relations within a query and a document are removed, and white means there are no edges between two corresponding nodes. The up-triangle masking matrix is obtained from Eq. (4) for $i \leq j$, and the down-triangle masking matrix is filled according the symmetry $\mathcal{M}_l^{(q,d)}(j,i) = \mathcal{M}_l^{(q,d)}(i,j)$. Thus the whole masking matrix $\mathcal{M}_l^{(q,d)}$ is applied in Eq. (3) to obtain the query-document bipartite word adjacent matrix.

$$\mathcal{M}_l^{(q,d)}(i,j) = \begin{cases} 1 & 1 \leq i \leq m, m+2 \leq j \leq m+n+2 \\ 1 & i = j \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

- *Query-Document Bipartite and Neighbor Word Graph.* Word order information plays an important role in the search task, especially for the long document. Thus we take some local document word relations into consideration and keep its left and right r neighbors word relations. As shown in Fig. 2(c), the sliding window size is $2r+1$ over the document and nearly $2r+1$ diagonals are colored here. In bipartite attention matrix in Fig. 2(b), there is only one diagonal. The up-triangle masking matrix is set as Eq. (5) for $i \leq j$, and the down-triangle masking matrix is filled with $\mathcal{M}_l^{(q,d)}(j,i) = \mathcal{M}_l^{(q,d)}(i,j)$ according to its symmetric property. Thus the whole masking matrix $\mathcal{M}_l^{(q,d)}$ is applied in Eq. (3) to obtain the query-document bipartite and neighbor word adjacent matrix.

$$\mathcal{M}_l^{(q,d)}(i,j) = \begin{cases} 1 & 1 \leq i \leq m,\ m+2 \leq j \leq m+n+2 \\ 1 & m+2 \leq i \leq m+n+2, j = i, \ldots, i+r \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Word Representation Refinement. Through the above layer, we remove the unnecessary information from the word graph $\mathcal{A}_l^{(q,d)}$ and obtain the bipartite core graph $\hat{\mathcal{A}}_l^{(q,d)}$. Similarly, word representations $\mathbf{E}_l^{(q,d)}$ learned from BERT also need to be refined to separate the relevant information from these noisy relations. We use Gated Graph Neural Networks (GGNN) [11] to update word representations over the bipartite-core graph $\hat{\mathcal{A}}_l^{(q,d)}$. At each propagation step, GGNN aggregates neighbor word representations for each word in the graph $\hat{\mathcal{A}}_l^{(q,d)}$ and concatenates word representations from the last iteration and from neighborhood aggregation this iteration as the input embedding of Gated Recurrent Unit (GRU) in Eq. (7). This will help utilize high-order word relations to obtain fine-grained representations. Word attention matrix is computed according to Eq. (8). The query-document pair representation is aggregated as Eq. (9).

$$\mathbf{h}_0 = \mathbf{E}_l^{(q,d)} \tag{6}$$

$$\mathbf{h}_t = \text{GRU}([\mathbf{h}_{t-1}, \hat{\mathcal{A}}_l^{(q,d)} \mathbf{h}_{t-1}]) \tag{7}$$

$$\mathbf{h}_T^{att} = (\mathbf{W}_a \mathbf{h}_T) \cdot (\mathbf{W}_h \mathbf{h}_T)' \tag{8}$$

$$\mathbf{h}_l^{G_{q,d}} = [\text{sum}(\mathbf{h}_T^{att}) + \max(\mathbf{h}_T^{att}), \mathbf{E}_l^{(q,d)}(0)] \tag{9}$$

After T propagation steps, a final graph level representation for each query-document pair is learned denoted as $\mathbf{h}_l^{G_{q,d}}$ for each transformer layer l. Then it is fed into the last fully connected layer with weight matrix W_s to predict the relevance score $s_l(q,d)$ in Eq. (10). The final relevance score $f(q,d)$ is determined by the linear combination of all the relevance scores $\{s_l(q,d)\}_{l=1}^{L}$ in Eq. (11).

$$s_l(q,d) = \mathbf{W}_s \mathbf{h}_l^{G_{q,d}} + \mathbf{b}_s \tag{10}$$

$$f(q,d) = \mathbf{w}_f (s_l(q,d))_{1 \times L} + b_f \tag{11}$$

3.3 Loss Function

To derive a robust ranking function, the pairwise ranking loss is usually used for optimization. Additionally, we introduce a metric learning task as an auxiliary task to learn discriminative representations.

From the embedding perspective, we propose a triangle distance loss to place constraints on query, document and query-document representations. Cosine distance [9] was first introduced to make examples with different labels separated from each other in the classification problem. Similarly treating query-document pair as an instance, we define the distance between query-document representations with different labels as this cosine distance, referred to as **pairwise cosine distance**. The pairwise cosine distance is computed for transformer and refinement layer respectively, whose query-document representations are $\mathbf{E}_L^{(q,d)}(0) = \mathbf{e}_L^{(q,d)}$ and $\mathbf{h}_L^{G_{q,d}}$ correspondingly. The distance summation of both layers is shown in Eq. (12). It only puts constraints on query-document representations in Fig. 3(b). We further split this unified query-document representation $\mathbf{E}_L^{(q,d)}$ into query representation $\mathbf{E}_L^q$ and document representation $\mathbf{E}_L^d$. Moreover, we define the **pointwise cosine distance** between a query and document representations in this ranking scenario as Eq. (13). This pointwise distance only puts constraints between a query and document representations without documents of different labels as Fig. 3(a). Neither pairwise nor pointwise distance will produce compact representations for query, document and query-document representations. So we propose a **triangle distance** to combine both pairwise and pointwise cosine distance as Eq. (14). As shown in Fig. 3(c), this triangle distance place constraints not only on the distance between a query and document representations but also on the distance between different documents.

$$\mathcal{C}_{\text{pair}}(q, D_q) = \frac{1}{2n_q^+ n_q^-} \sum_{\substack{d_+ \in D_q^+ \\ d_- \in D_q^-}} 2 + \cos(\mathbf{e}_L^{(q,d_+)}, \mathbf{e}_L^{(q,d_-)}) + \cos(\mathbf{h}_L^{G_{q,d_+}}, \mathbf{h}_L^{G_{q,d_-}}) \quad (12)$$

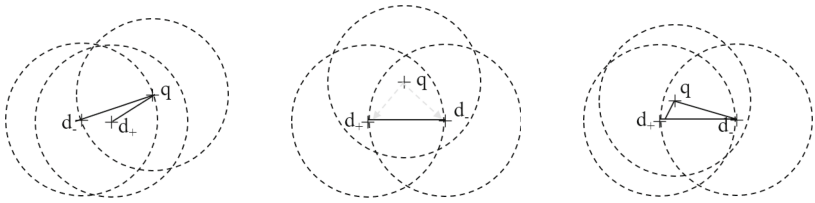
$$\mathcal{C}_{\text{point}}(q, D_q) = \frac{1}{n_q} \sum_{j=1}^{n_q} 1 + (1 - 2y_j) \cos(\mathbf{E}_L^q, \mathbf{E}_L^{d_j}) \quad (13)$$

$$\mathcal{L}_{\text{triangle}}(q, D_q) = \mathcal{C}_{\text{point}}(q, D_q) + \mathcal{C}_{\text{pair}}(q, D_q) \quad (14)$$

From the ranking perspective, we introduce a margin based pairwise ranking loss as follows

$$\mathcal{L}_{\text{rank}}(q, D_q) = \frac{1}{|D_q^+||D_q^-|} \sum_{d_+ \in D_q^+} \sum_{d_- \in D_q^-} \max(0, 1 - f(q, d_+) + f(q, d_-)). \quad (15)$$

We train both tasks in a multi-task learning framework with the optimization of $\lambda \mathcal{L}_{\text{triangle}}(q, D_q) + \mathcal{L}_{\text{rank}}(q, D_q)$.



(a) Pointwise Distance (b) Pairwise Distance (c) Triangle Distance

Fig. 3. Illustration of different constraints' effect on learned query/document representations

4 Experiments

We compare our proposed model LGRe with state-of-the-art baselines to investigate its effectiveness on two public benchmark datasets. Moreover, ablation studies for each component of LGRe are also explored.

4.1 Experimental Setting

Datasets. We use two TREC collections, Robust04 and WebTrack2009-12. Robust04 uses TREC discs 4 and 5[1], and WebTrack 2009-12 uses ClueWeb09b[2] as document collections. Note that the statistics are obtained only from the documents returned by BM25. Both data sets are white-space tokenized, lowercased, and stemmed using the Krovetz stemmer. Consistent with the baselines of the corresponding dataset, Robust04 uses Indri[3] for indexing, and WebTrack2009-12 uses Anserini [19] for indexing. Table 1 provides detailed information on these two data sets.

Table 1. Statistics of datasets.

	#Docs	Avg. Doc. Len	#Queries	Avg. Query Len	#Docs/Query
Robust04	37,500	428.2	250	3.62	150
WebTrack2009-12	19,590	1,393.0	200	2.64	100

Baselines. Three kinds of baselines are compared over these two datasets. (1) BM25: Candidate documents for each query are usually generated by BM25 in the first stage ranking. (2) Interaction based Neural Ranking Models (without BERT): DRMM [6] and ConvKNRM [4]. (3) BERT based Neural Ranking Models: Vanilla BERT, BERT-MaxP [2], CEDR-KNRM [14], and PARADE [10].

[1] 520k documents, https://trec.nist.gov/data_disks.html.
[2] 50M web pages, https://lemurproject.org/clueweb09/.
[3] http://www.lemurproject.org/indri.php.

Training Setting. For all BERT based baselines in our experiments, we make domain adaptation on MSMARCO.[4] Simple domain adaptation of BERT leads to a pre-trained model with both types of knowledge that can improve related search tasks where labelled data are limited [2]. Some performance results on Robust04 come from the paper aggregation site "Papers With Code".[5] Since WebTrack2009-12 does not have a unified data preprocessing pipeline similar to Robust04, we compare all baselines based on our data preprocessing pipeline.

Evaluation Setting. With the same division on both datasets, we use five fold cross validation with three folds for training, one fold for validation and one fold for test. The number of training epochs is 20 with batch size 32. The learning rate of BERT fine-tuning and LGRe is $1e-5$ and $5e-5$ respectively. λ is $1e-2$. All these hyperparameters are chosen according to performances in terms of the P@20 and nDCG@20 on the validation set, which are computed using script *trec_eval*.[6]

4.2 Effectiveness Analysis

The ranking performance of LGRe (bipartite and neighbor masking strategy + triangle distance) on both document ranking datasets is shown in Table 2. All the performances are averaged on five test sets for each dataset. Imp.% column in the table corresponds to the relative performance improvement of LGRe compared with each baseline. From Table 2, we observe the following phenomena.

Table 2. Ranking performance comparison among different models on Robust04 and WebTrack2009-12. Best results are in bold. The relative performance improvement is statistically significant with $p < 0.01$ in two-tailed paired t-test.

	Robust04				WebTrack2009-12			
Model	P@20	Imp.%	nDCG@20	Imp.%	P@20	Imp.%	nDCG@20	Imp.%
BM25	0.3123	53.38	0.4140	31.96	0.2805	27.95	0.1772	53.78
DRMM	0.2892	65.63	0.3040	79.70	0.3077	16.64	0.2015	35.24
Conv-KNRM	0.3408	40.55	0.3871	41.13	0.3155	13.76	0.213	27.93
Vanilla BERT	0.4042	18.51	0.4541	20.30	0.3253	10.32	0.254	7.28
BERT-MaxP	0.4277	11.99	0.4931	10.79	0.3373	6.40	0.2613	4.28
CEDR-KNRM	0.4667	2.64	0.5381	1.52	0.3481	3.10	0.2653	2.71
PARADE	0.4604	4.04	0.5399	1.19	–	–	–	–
LGRe	**0.479**	–	**0.5463**	–	**0.3589**	–	**0.2725**	–

[4] https://microsoft.github.io/TREC-2019-Deep-Learning.
[5] https://paperswithcode.com/sota/ad-hoc-information-retrieval-on-trec-robust04.
[6] https://trec.nist.gov/trec_eval.

(1) Compared with the best state-of-the-art baseline on each dataset, LGRe's relative performance gain is not less than 2% in terms of Precision@20. This improvement is statistically significant in the ranking task.
(2) Among all three kinds of baselines, BERT based ranking models achieve the best performance. One reason is that these interaction based ranking models without BERT usually derive the interaction matrix based on shallow pre-trained word embedding, such as word2vec [15]. These shallow word embedding only capture the local context, such as synonym, but cannot obtain complex or global patterns among words. This problem is solved by BERT with global word interactions, which makes it possible. The other reason is that interaction based ranking models like DRMM [6] predefine the query-document interaction matrix as input ignoring the query and document representation learning. All the interaction matrix, query and document representations are dynamically learned from data for BERT based ranking models. These learnable parameters make ranking models more flexible and suitable for different datasets.
(3) Compared with Vanilla BERT, LGRe's performance improvement agrees with our motivation that vanilla BERT has an inherent weakness though it naturally considers with the document ranking task. LGRe is mainly composed of BERT and word representation refinement process based on BERT. To a certain degree, LGRe's performance improvement also indicates the necessity of the following word refinement process in its architecture as Fig. 1.
(4) For all methods in Table 2 except DRMM [6], the ranking performance is higher on Robust04 than it on WebTrack2009-12. Dataset statistics show that the averaged query length is shorter and the averaged document number of each query is fewer on WebTrack2009-12. Fewer training instances may be one reason. So we will make a further study to verify the effect of query length on the ranking performance.

4.3 Ablation Study for Masking Strategy

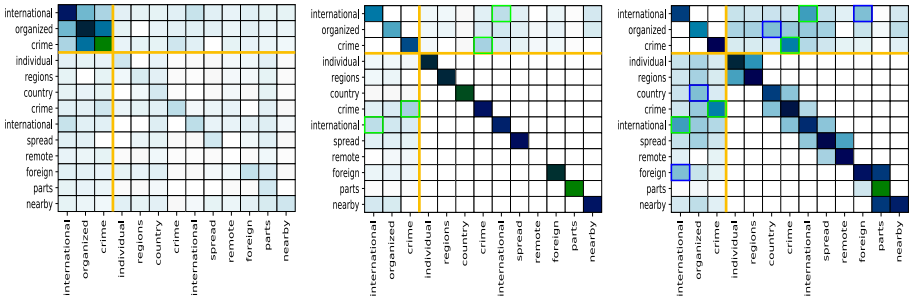
Three candidate strategies for the bipartite-core word graph construction are compared: (1) Full Word Graph: denoted as LGRe(Full). (2) Query-Document Bipartite Word Graph: denoted as LGRe(Bipartite). (3) Query-Document Bipartite Core and Neighbor Word Graph: denoted as LGRe(Bipartite+Neighbor). Note that all the methods in Table 3 have the same setting except the masking strategy, such as adopting the pairwise ranking loss plus the triangle cosine distance loss as the loss function. Imp.% column means the relative performance improvement of each other method compared with LGRe(Full), i.e. no masking.

The primary comparison result in Table 3 is that masking some word relations in the attention matrix will bring about the performance gain. The relative performance gain is statistically significant, at least 0.5%. It indicates that some word relations, such as query-query and document-document, learned from BERT are noise for the query-document text matching problem. The masking

Table 3. Ranking performance comparisons with different masking strategies on Robust04

Model	P@20	Imp.%	nDCG@20	Imp.%
LGRe (Full)	0.471	–	0.5359	–
LGRe (Bipartite)	0.4764	1.15	**0.5447**	1.64
LGRe (Bipartite+Neighbor)	**0.4771**	1.3	0.5403	0.82

strategy for graph construction is essential for LGRe. Additionally, keeping document neighbor word relations does not always promote the ranking performance. The relative NDCG@20 decreases by 0.82% due to document neighbor word relations, although the relative P@20 increases by 0.15%. The introduction of document neighbor word relations makes adjacent word representations learned from the word graph much closer. This leads to a smaller distinction between relevant documents' representation, which are originally near to each other. That is why the addition of document neighbor word relation will increase the hit rate of relevant documents, and hurt the ranking of relevant documents.



(a) Full (b) Bipartite (c) Bipartite+Neighbor

Fig. 4. Attention matrices learned from LGRe with different masking strategies. The green box represents exact term matching. The blue box represents synonym matching. The yellow line is the dividing line between query and document. (Color figure online)

For an intuitive understanding, we choose a specific query and document from Robust04. Query: "international, organized, crime". Document (stop words removed): "individual, regions, country, crime, international, spread, remote, foreign, parts, nearby". Attention matrices learned from different masking strategies are shown in Fig. 4. As we know, the meaning of short queries are vague, and forms of short queries are incomplete. For the full word graph in Fig. 4(a), the exact matching signals on "international" and "crime" is overwhelmed by many relations in documents. For the bipartite word graph in Fig. 4(b), the exact matching signals on 'international" and "crime" are obviously enhanced

by masking query and document word relations. For Fig. 4(c), the addition of document neighbor word relations will promote the exact matching signals and strengthen the relation of semantically similar words, such as "international" and "foreign", "organized" and "country". Meanwhile, relating some unrelated words may become possible noise for the final ranking. This case study gives a better explanation of the performance gain of both LGRe (Bipartite) and LGRe (Bipartite+Neighbor) in Table 3.

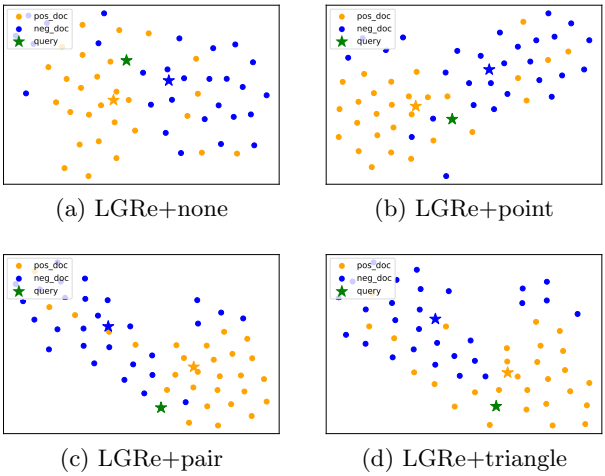
4.4 Ablation Study for Distance Learning Task

We introduce the cosine distance learning task as the auxiliary task for document ranking in LGRe. Whether this task is an essential part will be studied here. If it is necessary, which distance definition among three kinds in the Loss function section is the best choice. We compare LGRe with different loss functions on Robust04: (1) LGRe+none: only the pairwise ranking loss without any distance loss. (2) LGRe+point: the linear combination of pairwise ranking loss and pointwise cosine distance loss. (3) LGRe+pair: the linear combination of pairwise ranking loss and pairwise cosine distance loss. (4) LGRe+triangle: the linear combination of pairwise ranking loss and triangle cosine distance loss. Experimental results are shown in Table 4. Imp.% column corresponds to the relative performance improvement of each method compared with LGRe+none.

Table 4. Ranking performance comparisons among LGRe with different distance definitions on Robust04

Model	P@20	Imp.%	nDCG@20	Imp.%
LGRe+none	0.4771	–	0.5403	–
LGRe+point	0.4769	−0.04	0.5419	0.30
LGRe+pair	0.4778	0.15	0.5427	0.44
LGRe+triangle	0.479	0.39	**0.5463**	1.11

In most cases, the auxiliary task, i.e. cosine distance learning task, plays a positive role in the document ranking problem in Table 4. The only exception is LGRe+point under the P@20 evaluation. Obviously, the relative performance gain for both LGRe+point and LGRe+pair is limited. However, the performance improvement from the combination of pointwise and pairwise cosine distance loss, i.e. triangle distance loss, is much higher than the summation of performance gains from pointwise and pairwise distance loss separately. This $1 + 1 > 2$ effect on ranking performances shows the advantage of triangle cosine distance loss. Whether the cosine distance loss will help learn discriminative and compact representations remains unknown. Thus, we analyze a specific query, and plot query and document representations through dimension reduction with t-sne[12] shown in Fig. 5.



(a) LGRe+none (b) LGRe+point

(c) LGRe+pair (d) LGRe+triangle

Fig. 5. Query and document representations from LGRe with different losses. The pentagram means the mass center of each group.

Several results are obtained from Fig. 5. (1) (a) v.s. (b) and (c) and (d). The cosine distance learning task makes query, relevant and non-relevant document representations apart from each other. The reason lies that the embedding loss constrains representations directly, while the pairwise ranking loss takes indirectly effect on learned representations. (2) (b) v.s. (d). LGRe+point only defines a query and document point distance, and requires non-relevant document point far from and relevant document point near by the query point. This may lead to the problem in Fig. 5(b) that some relevant and non-relevant document points are mixed together. (3) (c) v.s. (d). LGRe+pair only defines a relevant and non-relevant document point distance, and requires non-relevant document points are far from relevant document points. This may lead to the problem in Fig. 5(c) that two kinds of distances from query to relevant and non-relevant document points respectively are not distinguishable. Generally, it is better to choose the triangle distance learning task as the auxiliary task to learn a discriminative representation for all the query, relevant and non-relevant documents.

4.5 Query Length Analysis

As mentioned before, one possible reason for the lower performance on WebTrack 2009-12 is shorter queries. To further explore the effect of query length on the ranking performance of BERT based ranking models, we conduct a group study on different query lengths. Robust04's queries are divided into two groups: one group with query length ≤ 3, the other group with query length > 3. The number of queries in two groups is 144 and 106 respectively. We randomly select 100 queries from each group, and randomly divide them into training, validation,

and test set with a ratio of 8 : 1 : 1. Performance comparisons on the test set with vanilla BERT and BM25 are shown in Table 5. Imp.% column represents the relative performance improvement of each other method compared with BM25.

Table 5. Ranking performance comparisons on two subsets of Robust04 with different query lengths.

	QLEN≤3				QLEN>3			
Model	P@20	Imp.%	nDCG@20	Imp.%	P@20	Imp.%	nDCG@20	Imp.%
BM25	0.3857	–	0.4689	–	0.425	–	0.4851	–
Vanilla BERT	0.3935	2.02	0.4729	0.85	0.4291	0.96	0.4876	0.52
LGRe	**0.4357**	12.96	**0.5058**	7.89	**0.44**	3.53	**0.493**	1.63

For all the methods, absolute performances on the shorter query subset are usually lower than these on the longer query subset. This suggests that document ranking for shorter queries is more difficult. Due to the concatenation of query and document pair as input, BERT models the global word interaction over the query-document text. This helps query words find their related words, which will alleviate the difficult short query problem to some degree. In this sense, both BERT based ranking models obtain higher performances gain on shorter queries than these on longer queries in Table 5. Due to the addition of the word representation refinement process, LGRe's relative performance improvement is much higher than vanilla BERT's. Compared with longer queries, the global word interaction learned from BERT is easier to generate a query-document representation submerging the query information. The refinement process of LGRe makes the query part emerge in the query-document representation.

5 Conclusion

To overcome the inherent weakness of BERT in the ranking task, we propose LGRe to refine word representations. We propose to mask the attention matrix from BERT to construct a bipartite-core word graph as the text matching between query and document. Then, word representations are updated through recurrent propagation steps to remove the useless information from original word embedding. Additional, triangle distance learning task is proposed to serve as the auxiliary task for document ranking.

Acknowledgement. This research work was funded by the National Natural Science Foundation of China under Grant No. 62072447.

References

1. Clark, K., Luong, M.T., Le, Q.V., Manning, C.D.: ELECTRA: pre-training text encoders as discriminators rather than generators. arXiv preprint arXiv:2003.10555 (2020)

2. Dai, Z., Callan, J.: Deeper text understanding for IR with contextual neural language modeling. In: Proceedings of the 42nd International ACM SIGIR, pp. 985–988 (2019)
3. Dai, Z., Callan, J.: Context-aware term weighting for first stage passage retrieval. In: Proceedings of the 43rd International ACM SIGIR, pp. 1533–1536 (2020)
4. Dai, Z., Xiong, C., Callan, J., Liu, Z.: Convolutional neural networks for soft-matching N-grams in ad-hoc search. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, pp. 126–134 (2018)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
6. Guo, J., Fan, Y., Ai, Q., 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, pp. 55–64 (2016)
7. Guo, J., et al.: A deep look into neural ranking models for information retrieval. Inf. Process. Manage. **57**, 102067 (2019)
8. Hu, B., Lu, Z., Li, H., Chen, Q.: Convolutional neural network architectures for matching natural language sentences. In: Advances in Neural Information Processing Systems, pp. 2042–2050 (2014)
9. Li, Baoli, Han, Liping: Distance weighted cosine similarity measure for text classification. In: Yin, H., et al. (eds.) IDEAL 2013. LNCS, vol. 8206, pp. 611–618. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41278-3_74
10. Li, C., Yates, A., MacAvaney, S., He, B., Sun, Y.: PARADE: passage representation aggregation for document reranking. arXiv preprint arXiv:2008.09093 (2020)
11. Li, Y., Tarlow, D., Brockschmidt, M., Zemel, R.: Gated graph sequence neural networks. arXiv preprint arXiv:1511.05493 (2015)
12. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**(Nov), 2579–2605 (2008)
13. MacAvaney, S., Nardini, F.M., Perego, R., Tonellotto, N., Goharian, N., Frieder, O.: Efficient document re-ranking for transformers by precomputing term representations. arXiv preprint arXiv:2004.14255 (2020)
14. MacAvaney, S., Yates, A., Cohan, A., Goharian, N.: CEDR: contextualized embeddings for document ranking. In: Proceedings of the 42nd International ACM SIGIR, pp. 1101–1104 (2019)
15. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
16. Pang, L., Lan, Y., Guo, J., Xu, J., Wan, S., Cheng, X.: Text matching as image recognition. arXiv preprint arXiv:1602.06359 (2016)
17. Pennington, J., Socher, R., Manning, C.D.: GloVe: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
18. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
19. Yang, P., Fang, H., Lin, J.: Anserini: enabling the use of Lucene for information retrieval research. In: Proceedings of the 40th International ACM SIGIR, pp. 1253–1256 (2017)
20. Zhou, J., et al.: Graph neural networks: a review of methods and applications. arXiv preprint arXiv:1812.08434 (2018)