Data Science and Engineering (2022) 7:30–43
https://doi.org/10.1007/s41019-022-00179-3

RESEARCH PAPERS

Disentangled Graph Recurrent Network for Document Ranking

Qian Dong[1,2] · **Shuzi Niu**[1] · **Tao Yuan**[1] · **Yucheng Li**[1]

Received: 2 June 2021 / Revised: 15 September 2021 / Accepted: 16 January 2022 / Published online: 15 February 2022

Abstract
BERT-based ranking models are emerging for its superior natural language understanding ability. All word relations and representations in the concatenation of query and document are modeled in the self-attention matrix as latent knowledge. However, some latent knowledge has none or negative effect on the relevance prediction between query and document. We model the observable and unobservable confounding factors in a causal graph and perform do-query to predict the relevance label given an intervention over this graph. For the observed factors, we block the back door path by an adaptive masking method through the transformer layer and refine word representations over this disentangled word graph through the refinement layer. For the unobserved factors, we resolve the do-operation query from the front door path by decomposing word representations into query related and unrelated parts through the decomposition layer. Pairwise ranking loss is mainly used for the ad hoc document ranking task, triangle distance loss is introduced to both the transformer and refinement layers for more discriminative representations, and mutual information constraints are put on the decomposition layer. Experimental results on public benchmark datasets TREC Robust04 and WebTrack2009-12 show that DGRe outperforms state-of-the-art baselines more than 2% especially for short queries.

Keywords Ad hoc retrieval · Graph neural network · Transformer · Causal inference

1 Introduction

Neural ranking models focus on semantic matching between the query and document with neural networks to solve the ad hoc retrieval problem. Recently, BERT-based ranking models learn latent knowledge for document ranking from large scale text collections. Taking the concatenation of the query and document as input, BERT models their interactions at the word level as the self-attention matrix. In this sense, BERT-based ranking models, which is naturally fit for the ad hoc retrieval task, belong to the interaction-based neural ranking models.

However, interaction-based models only care for defining the interaction function between the query and document [11]. BERT's self-attention matrix is such an interaction function, which models all possible kinds of word relations that are useful for the matching process, such as query–document and document–document word relations. Traditional interaction functions only consider query–document word relations, but BERT takes query–query and document–document word relations into consideration. Whether these additional relations do good to the relevance prediction performance remains unknown.

As mentioned above, all document words are not related to the query. The document representation derived from these words is composed of query related and unrelated parts. Thus, the relevance score of a document to a query is usually determined by the query related part of the document representation instead of the unrelated part [11]. However, it is hard to point out which part is related to the query and which is not, although it is important to disentangle the related part from the document representation to derive the final relevance score.

✉ Shuzi Niu
shuzi@iscas.ac.cn

Qian Dong
dongqian19@mails.ucas.ac.cn

Tao Yuan
yuantao@iscas.ac.cn

Yucheng Li
yucheng@iscas.ac.cn

[1] Institute of Software, Chinese Academy of Sciences, Beijing, China

[2] University of Chinese Academy of Sciences, Beijing, China

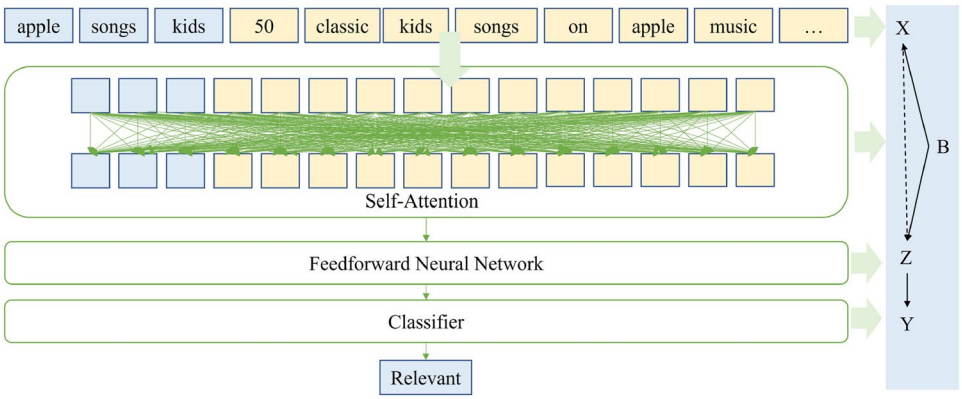


Fig. 1 Illustration of spurious word relations and latent representations in BERT-based ranking model

We probe into how spurious word relations and latent representations have effect on the retrieval performance. BERT's self-attention matrix provides a complete word relation knowledge base like B in Fig. 1. (1) For relevance prediction of a query–document pair, spurious word relations exist in such a base. For query "apple song kids" and a document with title "50 Classic Kid's Songs on Apple Music" in Fig. 1, word relations between "apple" and the other words in the document mislead the model to predict the label of the actually irrelevant document to be relevant. (2) Document representations derived from B contains some unrelated information. Irrelevant words, such as "classic" and "Apple", make the downstream classifier confused and predict the label of the actually irrelevant document to be relevant. Therefore, it is necessary to remove the effect of spurious word relations and unrelated information in the latent representations.

Taking both the spurious word relations and unrelated representations as confounding factors, we depict the causal graph as the right part of Fig. 1. For each query–document pair X and its relevance label Y, those confounding factors correlates X and Y even though there is no direct causation between X and Y. The self-attention matrix B is generated by the input query–document embedding X and measures the word similarity from the document and query. Z represents the latent representations learned from BERT. For the observed confounding factors, i.e. spurious word relations in the self-attention matrix B, we block the unreasonable path, i.e. the back-door path, $X \rightarrow B \rightarrow Z \rightarrow Y$, which has an effect on Y. For the unobserved confounding factors, i.e. unrelated representations in Z, we reduce their effects on Y by resolve the front-door path $X \rightarrow Z \rightarrow Y$.

To reduce effects of confounding factors like the causal graph in Fig. 1, we propose a *D*isentangled *G*raph *Re*current neural network method to decouple word representations learned from BERT for document ranking, referred to as *DGRe*. Specifically, we first design a causal graph for the document ranking task and cast the problem in a causal inference framework. Then an adaptive masking method is proposed to alleviate the observed confounding effect through the transformer layer. After the word refinement layer, a mutual information decomposition layer is finally introduced to disentangle the document representation into query related and unrelated parts owing to the unobserved confounding effects on representations.

For each query and document pair X, DGRe first takes their concatenation as input and obtain word representations through the transformer layer, from which a latent graph is derived as a self attention matrix. Next an adaptive masking method is proposed to disentangle word relations in this latent graph by a sharp activation function ReLu, which aims at keeping relations with higher attention weights and removing relations with lower weights. Then through the word representation refinement layer, word representations are updated with a gated recurrent unit over this disentangled graph to achieve the back-door adjustment $X \rightarrow B \rightarrow Z$ to deal with the observable confounder in B. Afterward for the unobservable confounder in Z, we realize the do-calculus of Z in the front-door adjustment $X \rightarrow Z \rightarrow Y$ through mutual information decomposition layer. It decomposes the derived document representation into query related and unrelated parts according to the query's attention weights.

All the representations derived from the BERT layer, word representation refinement layer and mutual information decomposition layer are aggregated through multi-layer perceptrons and classified with a Sigmoid function. 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. Finally, mutual information regularization is proposed to minimize the mutual information between two parts. All 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 DGRe. 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.

(1) A causal graph is designed for BERT-based ranking models, to disentangle the intrinsic reason and confounding factors for the relevance between the query and document.
(2) To reduce the observable confounding effect on word relations, an adaptive masking method is proposed to identify useful word relations from the learned self-attention matrix, and word representation refinement is performed over this disentangled word graph for each query–document pair.
(3) To reduce the unobservable confounding effect on the document word latent representation, mutual information decomposition layer is introduced to decouple the document representation into two parts, i.e. query related and unrelated representations.
(4) Besides the pairwise ranking loss function for the basic document ranking task, a triangle distance loss function for the transformer layer is to learn discriminative representations for the downstream ranking task, and the mutual information regularization for the decomposition layer is to disentangle the document representation.

2 Related Work

Here we briefly review some related studies on interaction-based neural ranking models, BERT-based ranking models, causal inference and other related techniques, such as mutual information 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 [11].

Traditional non-parametric interaction functions include binary indicator, cosine similarity, dot product, radial basis function and so on. DRMM [10] converts a local interaction matrix for the query–document word pair to a fixed-length matching histogram for relevance matching. MatchPyramid [25] 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 [8] 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 [15] 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. However, BERT-based ranking model introduces additional relations while learning the interaction feature between the query and document, which has none or negative effect on the relevance prediction.

2.2 BERT-Based Ranking Models

Pretrained Neural Language Models (PNLMs) have achieved state-of-the-art results in many NLP tasks, and BERT [9] is such a representative PNLM. As mentioned above, it naturally works for the ad hoc ranking because the attention matrix in BERT can be regarded as an interaction function. BERT-based ranking models are supposed to be superior to neural ranking models without BERT.

BERT-MaxP [6] splits a document into overlapping passages. The neural ranker predicts the relevance score of each passage independently, and the relevance score of the document is determined by the passage with the highest relevance score. CEDR [22] incorporates BERT's classification vector into existing neural models, such as DRMM [10] and Conv-KNRM [8]. PARADE [19] leverages passage-level representations to predict a document's relevance score without passage independence assumption. PARADE improves its performance by fine tuning on the MSMARCO passage ranking dataset instead of the Bing search log. Other researches focus on how to improve the efficiency of PNLM in retrieval tasks. PreTTR [23] 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 [7] maps the contextualized term representations from BERT into context-aware term weights for efficient passage retrieval.

Existing BERT-based ranking models focus on how to design the input of BERT layer and take advantage of its output to be adaptable to the document ranking task. In this paper, we explore the underlying reasons inside the BERT layer for the document relevance score to a query and remove possible confounding factors for the BERT layer to derive the relevance score.

2.3 Causal Inference in Neural Network

Causal inference [26] provides researchers with a new methodology to design more robust models. Some studies focus on how to generate counterfactual samples from the perspective of causal inference to improve the performance of the model [1, 16, 30]. Other studies explore how to remove biases in the data sets [31, 35–37]. These studies usually assume that the confounder is observable [31, 36] or domain-specific knowledge [3, 13].

We design the causal graph deep inside the self-attention structure, which is similar to the causal attention [35] in computer vision. Different from the causal attention [35] to alleviate the dataset bias, our designed causal graph is to remove the confounding effects of spurious information on document ranking task. The causal attention [35] is implemented as sampling techniques within one training sample or across several samples. DGRe attempts to reduce the confounding effects by adaptive masking method and the mutual information decomposition layer.

2.4 Other Related Techniques

Mutual information-based methods have been studied for a long history especially in unsupervised representation learning. Benefit from the increasing attention of researchers on mutual information estimation [2, 4, 5, 20], we can efficiently estimate the mutual information of two latent variables through a neural network. DIM [14] introduces a new representation learning loss function by maximizing mutual information in an unsupervised way. GMI [28] brings mutual information into graph representation learning to alleviate the problem of lacking available supervision and avoid potential risks from unreliable labels. In addition, SSD [12] is a disentanglement framework, where mutual information is served as supervision signals for domain adaptation tasks. For DGRe, we utilize mutual information constraints similar to [12], but apply them into the disentanglement of the document representation for document ranking task.

Graph neural network (GNN) has 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 [38]. 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 [32]. Recursive-based GNN attempts to use the gate mechanism like GRU [18] in the propagation step to improve the long-term propagation of information across the graph structure. Here we explore a combination of transformer and recursive-based GNN to refine word representations over a disentangled graph for BERT-based ranking models.

3 Method

To solve the ad hoc document retrieval problem, we first describe the causal inference framework for BERT-based ranking models. Then a network architecture is proposed to performance the causal inference at both the word and document levels. Finally, an additional loss function is introduced to ensure the document representation decomposable.

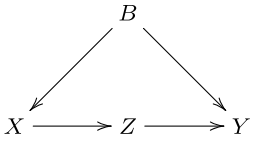


Fig. 2 Causal graph for BERT-based ranking models

3.1 Problem 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 the query and document is key to neural ranking models.

3.2 Causal Inference Framework for Document Ranking

We utilize the causal graph [27] to depict the causal effect in the matching process between the query and document. Due to its intrinsic interaction-based neural model, BERT-based ranking models usually take the concatenation of a query q and document d as the input, i.e. $X = (q, d)$. From the perspective of the matching process, there is redundant information in terms of words and documents, which may lead to the spurious correlation between X and Y. One lies in the self-attention matrix B generated from X, which provides some harmful word relations for the matching process. For example in Fig. 1, the document word relation between "apple" and "song" hinders the model from predicting this document to be irrelevant to the query. The other is that not all words in a document d are related to a query q regardless of the ground truth relevance label of (q, d). When human judge whether q and d is relevant, it is usually determined by the document's query related part instead of the query unrelated part [11].

To emphasize the common cause of X and Y, we extends the causal graph in Fig. 1 and derive the graph in Fig. 2 to describe two confounding factors mentioned above. Based on this causal graph, the document ranking task is to answer the do-operation query $P(Y|do(X))$. Y is the binary relevance label, i.e. relevant or not. In practice,

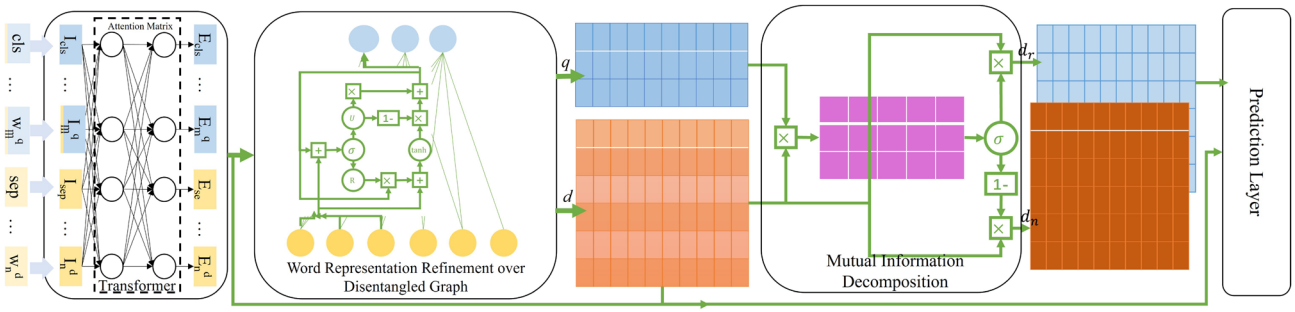


Fig. 3 Disentangled graph recurrent network architecture

the probability is usually computed through a sigmoid layer. For simplicity, we suppose $P(Y|do(X)) \propto \exp(g(\cdot))$. Similarly, other probabilities here are also supposed to be computed with a softmax/sigmoid layer in proportion to the exponential form $\exp(\cdot)$.

To remove the query unrelated part of the document representation, we block the front-door path $X \rightarrow Z \rightarrow Y$ by the unobservable confounder Z. According to the front-door path adjustment [27], we deconfound the factor Z by Eq. (1). Different from traditional front-door adjustment [35], we resolve the do-calculus of Z by decomposing it into query related and unrelated parts, i.e. Z_r and Z_n. To calculate the expectation $\mathbb{E}_Z[Z]$ in Eq. (4), we introduce a mutual information decomposition layer in Fig. 3 to split the document representation into two independent parts.

$$\begin{aligned} P(Y|do(X)) &= \sum_{z} P(Y|do(Z))P(Z|do(X)) \\ &= \sum_{Z_j \in \{Z_r, Z_n\}} P(Y|Z_j)P(Z_j|do(X)) \end{aligned} \tag{1}$$

To eliminate harmful word relations from the self-attention matrix, we block the back-door path $X \leftarrow B \rightarrow Y$ by the confounding factor B. Specifically, we estimate the do-operation query $P(Z|do(X))$ in Fig. 2 by Eq. (2) to keep useful word relations in the self-attention matrix B. Suppose word relations with the positive similarity in B, denoted as B_+, have positive effect on the performance. The do-calculus of X is resolved with disentangling useful word relations from spurious ones in B. To estimate the expectation $\mathbb{E}_B[X]$ in Eq. (4), we design an adaptive masking method to obtain the disentangled graph and perform message passing over this disentangled word graph to refine word representations in Fig. 3.

$$P(Z_j|do(X)) = \sum_{B_i \in \{B_+, B_-\}} P(Z_j|X, B_i)P(B_i) \tag{2}$$

Replacing $P(Z_j|do(X))$ in Eq. (1) with Eq. (2), the prediction function $P(Y|do(X))$ is obtained as Eq. (3). The expectation of an exponential function can be approximated by Weighted Geometric Mean [29, 33, 35]. So, the approximation of Eq. (3) is the weighted geometric mean of $P(Y|X, Z)$, which can be further approximated by exchange the order of exponential and expectation operator as Eq. (4). The sigmoid layer will be used for normalization to derive the probability $P(Y|do(X))$.

$$P(Y|do(X)) = \mathbb{E}_Z \mathbb{E}_B[P(Y|X, Z)] \tag{3}$$

$$\propto \exp\left(g(\mathbb{E}_Z[Z], \mathbb{E}_B[X])\right) \tag{4}$$

3.3 Architecture

Given a query–document pair $X = (q, d)$, self-attention mechanism in the Transformer layer of Fig. 3 provides us a natural way to model their interaction and at the same time the confounding factor to predict its label $P(Y|X)$. We introduce the causal graph to reduce its negative effect on the performance and resolve the do-query $P(Y|do(X))$ based on this graph by both back-door and front-door adjustments. And we arrive at the do-free form $\exp(g(\mathbb{E}_B[X], \mathbb{E}_Z[Z]))$, which can be implemented as neural network layers. Next, the disentangled word representations $\mathbb{E}_B[X]$ is refined over the word graph generated from the transformer layer under the supervision of the adaptive masking method in Fig. 3. Then, we further decompose the document word representations into two parts with the query attention mechanism and derive the query related document word representations to approximate $\mathbb{E}_Z[Z]$. Finally, multi-layer perceptrons (MLP) are utilized to aggregate all these word representations and followed by a sigmoid layer to predict the relevance probability of (q, d), which is shown in the rightmost Fig. 3.

3.3.1 Transformer Layer

For each query–document pair (q, d), two word sequences are concatenated, i.e. $X^{(q,d)} = [[\text{CLS}], s^q, [\text{SEP}], s^d, [\text{SEP}]]$.

Its input embedding $\mathbf{I}^{(q,d)}$ is derived from the sum of the word embedding and its corresponding position embedding of $X^{(q,d)}$. Then $\mathbf{I}^{(q,d)}$ is fed into BERT stacked with L identical layers. For 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 the other word representations in Eq. (6), d_k is the dimension of word representations.

$$\mathcal{A}_{l-1}^{(q,d)} = \text{softmax}\left(\frac{(\mathbf{W}_B\mathbf{E}_{l-1}^{(q,d)})(\mathbf{W}_B\mathbf{E}_{l-1}^{(q,d)})'}{\sqrt{d_k}}\right) \tag{5}$$

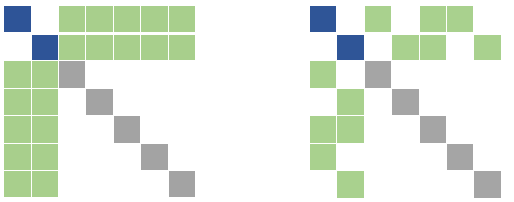
$$\mathbf{E}_l^{(q,d)}(i) = \mathbf{E}_{l-1}^{(q,d)}(i) + \sum_j \mathcal{A}_{l-1}^{(q,d)}(i,j)\mathbf{E}_{l-1}^{(q,d)}(j) \tag{6}$$

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 $B = \{\mathcal{A}_l^{(q,d)}\}_{l=1}^L$ for each query–document pair (q, d). Each attention matrix $\mathcal{A}_l^{(q,d)}$ naturally models the query–document word interaction.

3.3.2 Word Representation Refinement over Disentangled Graph

To explore the confounding factor in $\mathcal{A}_l^{(q,d)} \in B$, we find word relations in each self-attention matrix $\mathcal{A}_l^{(q,d)}$ fall into three categories: document–document, query–query and query–document word relations. Intuitively, only query–document word relations are useful for query–document matching, and other additional interactions may harm the retrieval performance [6]. A simple method is to mask these relations and obtain only a bipartite word graph. However, this is not flexible for different query–document pairs for not all the document words are useful for the matching. Thus, here we propose an adaptive masking method to separate good word relations from spurious relations for retrieval performance. We perform message passing over this disentangled word graph to remove spurious relations' negative effect on word representations.

The heuristic masking method is visualized as Fig. 4a. Word relations within a query and within a document are removed, and white means there no edges between corresponding nodes. The up-triangle masking matrix of $\mathcal{M}_l^{(q,d)}$ is obtained from Eq. (7) for each transformer layer l. According to the symmetry of $\mathcal{M}_l^{(q,d)}$, its down-triangle matrix is filled. Based on this simple masking method, the masked self-attention-like matrix $G_l^{(q,d)}$ is defined as Eq. (8), where ϵ is small enough. ReLU is introduced to filter all possible spurious relations with the word similarity smaller than 0. This is referred to as Adaptive Masking.



(a) Heuristic Masking (b) Adaptive Masking

Fig. 4 Bipartite word graphs constructed from two strategies. Blue, green and gray color represent the word attention score between query and query, document and document, query and document separately. White means no word relation

$$\mathcal{M}_l^{(q,d)}(i,j) = \begin{cases} 1 & 1 \le i \le m, m+2 \le j \le m+n+2 \\ 1 & i = j \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

$$G_l^{(q,d)} = \text{ReLU}\left(\frac{(\mathbf{W}_A\mathbf{E}_l^{(q,d)})(\mathbf{W}_A\mathbf{E}_l^{(q,d)})'}{\sqrt{d_k}} + \epsilon(1 - \mathcal{M}_l^{(q,d)})\right) \tag{8}$$

To derive the rigorous masked self-attention matrix, we first normalize each element in the self-attention-like matrix $G^{(q,d)_l}$ with its infinite norm to avoid overflow. Then a modified softmax function $\text{softmax}_m(x)$ for a vector $x \in \mathbb{R}^{1\times n_x}$ in Eq. (9) is introduced to obtain the probability distribution over all other words, where the probability is 0 for the entry 0. In other words, negative word relations should be completely filtered out. The masked self-attention matrix through the adaptive masking method, namely the disentangled word graph $\hat{\mathcal{A}}_l^{(q,d)}$ for transformer layer l, is defined as Eq. (10).

$$\text{softmax}_m(x) = \left(\frac{\exp(x_i) - 1}{\sum_j(\exp(x_j) - 1)}\right)_{1\times n_x} \tag{9}$$

$$\hat{\mathcal{A}}_l^{(q,d)} = \text{softmax}_m\left(\frac{G_l^{(q,d)}}{\|G_l^{(q,d)}\|_\infty}\right) \tag{10}$$

With the adaptive masking method, the disentangled graph is derived denoted as $\hat{\mathcal{A}}_l^{(q,d)}$. To distill the useful word representations from all the word representations, we perform message passing over this disentangled graph. The process is called word representation refinement. We use gated graph neural networks (GGNN) [18] to update word representations over the bipartite-core graph $\hat{\mathcal{A}}_l^{(q,d)}$. At each propagation step t, 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. (11). This will

help utilize high-order word relations to obtain fine-grained representations.

$$\begin{aligned}\mathbf{h}_0^l &= \mathbf{E}_l^{(q,d)} \\ \mathbf{h}_t^l &= \mathrm{GRU}([\mathbf{h}_{t-1}, \hat{\mathcal{A}}_l^{(q,d)}\mathbf{h}_{t-1}])\end{aligned} \tag{11}$$

After T propagation steps, a final graph level representation for each query–document pair is learned denoted as $\mathbf{Z}_T^l$ for each transformer layer l. Self-attention mechanism is again applied in Eq. (12) to the derived word representations $\mathbf{Z}_T^l$. The softmax function in Eq. (12) is an approximation probability distribution. Thus, the expectation $\mathbb{E}_B[X]$ is in proportion to $\mathbf{Z}_T^l$.

$$\mathbb{E}_B[X] \propto \mathbf{Z}^l = \mathrm{softmax}((\mathbf{W}_a\mathbf{h}_T^l)\cdot(\mathbf{W}_h\mathbf{h}_T^l)')\cdot\mathbf{h}_T^l \tag{12}$$

3.3.3 Mutual Information Decomposition Layer

From the perspective of the query–document interaction, word representation refinement layer is proposed to eliminate the spurious query–document word relations through the disentangled word graph. From the perspective of the document representation, not all document words are necessary for the query–document matching process. Naturally, the document word importance is dependent on how the query representation attends to it. Here we introduce a conventional attention mechanism to put more weights on the document words in terms of the query representation. According to this attention mechanism, the document word representation is decomposed into query-related part and its complement. To obtain good decomposed representations, we add mutual information constraints to minimize the overlapped information between two parts. These constraints will be introduced in the loss function section.

$$\mathbf{Z}^l = [\mathbf{Z}_{\mathrm{CLS}}^l, \mathbf{Z}_q^l, \mathbf{Z}_{\mathrm{SEP}}^l, \mathbf{Z}_d^l, \mathbf{Z}_{\mathrm{SEP}}^l] \tag{13}$$

Through the word representation refinement layer, we obtain all word representations as Eq. (13). Based on query word representations $\mathbf{Z}_q^l$, we use a sigmoid function to decide the probability that this document word is important for the current query word. With this word probability, we split the document word representations $\mathbf{Z}_d^l$ into query related part $\mathbf{Z}_{d_r}^l$ as Eq. (14) and query unrelated part $\mathbf{Z}_{d_n}^l$ as Eq. (15) two parts. For simplicity, we assume only the query related part $\mathbf{Z}_{d_r}$ has effect on the retrieval performance. So, the target expectation $\mathbb{E}_Z[Z]$ is calculated as $1\times\mathbf{Z}_{d_r}^l + 0\times\mathbf{Z}_{d_n}^l = \mathbf{Z}_{d_r}^l$.

$$\mathbf{Z}_{d_r}^l = \sigma((\mathbf{W}_q\mathbf{Z}_q^l)\cdot(\mathbf{W}_d\mathbf{Z}_d^l)')\cdot\mathbf{Z}_d^l \tag{14}$$

$$\mathbf{Z}_{d_n}^l = (1-\sigma((\mathbf{W}_q\mathbf{Z}_q^l)\cdot(\mathbf{W}_d\mathbf{Z}_d^l)'))\cdot\mathbf{Z}_d^l \tag{15}$$

3.3.4 Prediction Layer

We add skip connections from BERT layer, word refinement layer and mutual information decomposition layer separately to the prediction layer to avoid unnecessary information loss for prediction. The $g(\cdot)$ of the final do-operation query $P(Y|do(X))$ in Eq. (4) is estimated as the linear combination of these aggregated information above in Eq. (16). Then a sigmoid function is employed to estimate $P(Y|do(X))$ in Eq.(17).

$$g(q,d) = \mathbf{w}_f(\mathbf{W}_s[\mathbf{Z}^l, \mathbf{Z}_{d_r}^l, \mathbf{E}_l^{(q,d)}(0)] + \mathbf{b}_s)_{1\times L} + b_f \tag{16}$$

$$P(Y|do(X)) \approx f(q,d) = \sigma(g(q,d)) \tag{17}$$

3.4 Loss Function

To obtain the optimal model parameters, we add the triangle distance loss, decomposition loss and pairwise ranking loss separately to the corresponding transformer layer, decomposition layer and prediction layer.

3.4.1 Triangle Distance

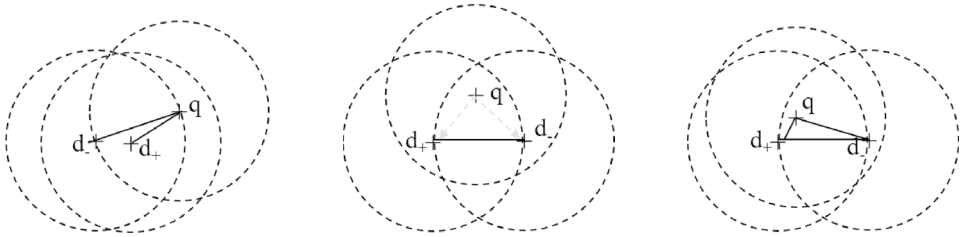
From the embedding perspective, we propose a triangle distance loss to place constraints on query, document and query–document representations. Cosine distance [17] was first introduced to make examples with different labels separated from each other in the classification problem. Given two samples a and b with representation x_a and x_b , respectively, the cosine distance is defined as Eq. (18), where $\mathbb{I}(a,b) = 1$ if a and b have the same label and 0 otherwise.

$$s(x_a, x_b) = 1 + 2\mathbb{I}(a,b)\cos(x_a, x_b) \tag{18}$$

We split the unified query–document word representations $\mathbf{E}_L^{(q,d)}$ into query word representations $\mathbf{E}_L^q$ and document word representations $\mathbf{E}_L^d$. Moreover, we define the *pointwise cosine distance* as Eq. (19), which only puts constraints between query and document word representations in Fig. 5a.

$$\mathcal{C}_{\mathrm{point}}(q, D_q) = \frac{1}{n_q}\sum_{j=1}^{n_q} s(\mathbf{E}_L^q, \mathbf{E}_L^{d_j}) \tag{19}$$

Similarly treating each 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 mutual information decomposition layer, respectively, whose query–document representations are $\mathbf{e}_d^L = \mathbf{E}_L^{(q,d)}(0)$. The distance summation



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

Fig. 5 Illustration of different constraints' effect on learned query/document representations

of both layers is shown in Eq. (20). It only puts constraints on query–document representations in Fig. 5b.

$$\mathcal{C}_{\text{pair}}(q, D_q) = \sum_{\substack{d_+ \in D_q^+ \\ d_- \in D_q^-}} \frac{(s(\mathbf{e}^L_{d^+}, \mathbf{e}^L_{d^-}) + s(\mathbf{Z}^L_{d_r^+}, \mathbf{Z}^L_{d_r^-}))}{2n_q^+ n_q^-} \tag{20}$$

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. (21). As shown in Fig. 5c, this triangle distance places constraints not only on the distance between a query and document representations but also on the distance between different documents.

$$\mathcal{L}_{\text{triangle}}(q, D_q) = \mathcal{C}_{\text{point}}(q, D_q) + \mathcal{C}_{\text{pair}}(q, D_q) \tag{21}$$

3.4.2 Decomposition Loss

It is reasonable to decompose the document word representations into two parts satisfying the following three conditions: (1) minimizing the interdependency between query related and unrelated document word representations; (2) minimizing the interdependency between query and query unrelated document word representations; (3) maximizing the interdependency between query and query related document word representations.

Here the interdependency is measured by mutual information, which is computed as the KL-divergence between the joint distribution and the production of two marginal distributions. As the marginal distributions are hard to estimated, so we approximate the mutual information in the dual representation of KL-divergence, which is proposed by Mutual Information Neural Estimator (MINE) [2]. Specifically, three constraints in terms of mutual information are expressed as $\mathcal{L}_{rn}(\mathbf{Z}^L_{d_r}, \mathbf{Z}^L_{d_n}, \phi)$ in Eq. (22), $\mathcal{L}_{qn}(\mathbf{Z}^L_q, \mathbf{Z}^L_{d_n}, \phi)$ in Eq. (23) and $\mathcal{L}_{qr}(\mathbf{Z}^L_q, \mathbf{Z}^L_{d_r}, \phi)$ in Eq. (24) separately. ϕ denotes parameters in the mapping function as Eq. (14) from $\mathbf{Z}^L_d$ to $\mathbf{Z}^L_{d_r}$ and $\mathbf{Z}^L_{d_n}$. The overall mutual information constraint $\mathcal{L}_{mi}(\mathbf{Z}^L_q, \mathbf{Z}^L_{d_r}, \mathbf{Z}^L_{d_n}, \phi)$ is computed as $\mathcal{L}_{rn} + \mathcal{L}_{qn} - \mathcal{L}_{qr}$.

$$\mathbb{E}_{P(\mathbf{Z}^L_{d_r}, \mathbf{Z}^L_{d_n})}[\phi] - \log(\mathbb{E}_{P(\mathbf{Z}^L_{d_r})P(\mathbf{Z}^L_{d_n})}[e^{\phi}]) \tag{22}$$

$$\mathbb{E}_{P(\mathbf{Z}^L_q, \mathbf{Z}^L_{d_n})}[\phi] - \log(\mathbb{E}_{P(\mathbf{Z}^L_q)P(\mathbf{Z}^L_{d_n})}[e^{\phi}]) \tag{23}$$

$$\mathbb{E}_{P(\mathbf{Z}^L_q, \mathbf{Z}^L_{d_r})}[\phi] - \log(\mathbb{E}_{P(\mathbf{Z}^L_q)P(\mathbf{Z}^L_{d_r})}[e^{\phi}]) \tag{24}$$

3.4.3 Ranking Loss

From the ranking perspective, we introduce a margin-based pairwise ranking loss $\mathcal{L}_{\text{rank}}(q, D_q)$ as Eq. (25).

$$\frac{1}{n_q^+ n_q^-} \sum_{\substack{d_+ \in D_q^+ \\ d_- \in D_q^-}} \max(0, 1 - f(q, d_+) + f(q, d_-)) \tag{25}$$

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

4 Experiments

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

4.1 Experimental Setting

4.1.1 Datasets

We use two TREC collections, Robust04 and WebTrack 2009–12. 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,

[1] 520k documents, https://trec.nist.gov/data_disks.html.

[2] 50M web pages, https://lemurproject.org/clueweb09/.

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	1393.0	200	2.64	100

Table 2 Ranking performance comparison among different models on Robust04 and WebTrack2009-12

Model	Robust04				WebTrack2009-12			
	P@20	Imp.%	nDCG@20	Imp.%	P@20	Imp.%	nDCG@20	Imp.%
BM25	0.3123	54.24	0.4140	33.57	0.2805	28.77	0.1772	57.11
DRMM	0.2892	66.56	0.3040	81.91	0.3077	17.39	0.2015	38.16
Conv-KNRM	0.3408	41.34	0.3871	42.86	0.3155	14.48	0.213	30.7
Vanilla BERT	0.4042	19.17	0.4541	21.78	0.3253	11.04	0.254	9.61
BERT-MaxP	0.4277	12.63	0.4931	12.14	0.3373	7.09	0.2613	6.54
CEDR-KNRM	0.4667	3.21	0.5381	2.77	0.3481	3.76	0.2653	4.94
PARADE	0.4604	4.62	0.5399	2.43	–	–	–	–
LGRe	0.479	0.56	0.5463	1.22	0.3589	0.64	0.2725	2.17
DGRe	**0.4817**	–	**0.553**	–	**0.3612**	–	**0.2784**	–

Best results are in bold. The relative performance improvement is statistically significant with $p < 0.01$ in two-tailed paired t-test

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 [34] for indexing. Table 1 provides detailed information on these two data sets.

4.1.2 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 [10] and ConvKNRM [8]. (3) BERT-based Neural Ranking Models: Vanilla BERT, BERT-MaxP [6],CEDR-KNRM [22] and PARADE [19].

4.1.3 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 [6]. 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.

4.1.4 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 DGRe 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 DGRe[7] 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 DGRe compared with each baseline. From Table 2, we obtain the following observations.

Compared with the best state-of-the-art baseline on each dataset, DGRe's relative performance gain is not less than

[3] http://www.lemurproject.org/indri.php.

[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.

[7] The codes are available at https://github.com/DQ0408/DGRe.

2% in terms of Precision@20. This improvement is statistically significant in the ranking task.

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 pretrained word embedding, such as word2vec [24]. 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. The other reason is that interaction-based ranking models like DRMM [10] predefine the query–document interaction matrix as input ignoring the query and document representation learning. The interaction matrix, query and document representations are all dynamically learned from data for BERT-based ranking models. These learnable parameters make ranking models more flexible and suitable for different datasets.

Compared with vanilla BERT, DGRe's performance improvement agrees with our motivation that vanilla BERT has an inherent weakness though it naturally considers with the document ranking task. DGRe is mainly composed of BERT and word representation refinement process based on BERT. To a certain degree, DGRe's performance improvement also indicates the necessity of the following word refinement process in its architecture as Fig. 3.

The major differences between our proposed LGRe and DGRe lie in the adaptive masking layer and mutual information decomposition layer, which makes DGRe performs consistently better than LGRe in terms of two evaluation metrics. This performance improvement of DGRe shows that it is necessary to disentangle query and document word representations for document ranking. Which layer plays a more important role in the performance improvement among two layers will be explored next.

For all methods in Table 2 except DRMM [10], 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 Adaptive Masking Layer

To explore the role of the adaptive masking layer, we compared performances from the following three scenarios: (1) Without Masking: keep all the word relations in a query and document pair. (2) Heuristic Masking: only keep the relations between the query and document words as Eq. (7). (3) Adaptive Masking: keep word relations as Eq. (8). Note that all the methods in Table 3 have the same setting except the masking strategy, such as adopting the pairwise ranking loss without the triangle cosine distance supervision and no mutual information regularization terms. Imp.% column means the relative performance improvement of each other method compared with DGRe without masking and best results are in bold.

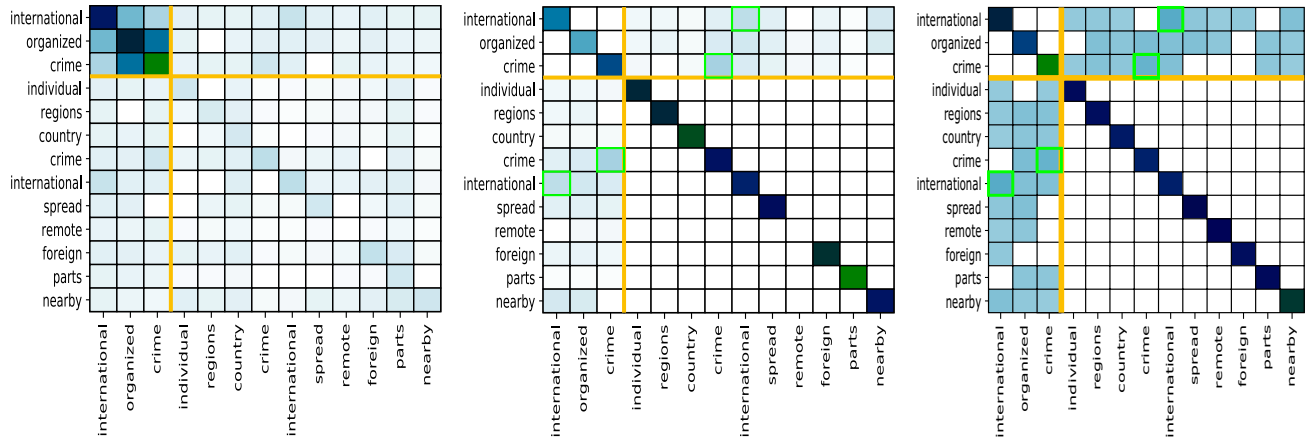
Table 3 Ranking performance comparisons with different masking strategies on Robust04

Model	P@20	Imp.%	nDCG@20	Imp.%
Without masking	0.471	–	0.5359	–
Heuristic masking	0.4764	1.15	0.5447	1.64
Adaptive masking	**0.478**	1.49	**0.5464**	1.96

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 1%. 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 strategy for graph construction is essential for DGRe. Additionally, adaptively filtering out the negative relations between query and document will continue to improve the performance of DGRe. This also indicates that there do exist spurious word relations between the query and document, which have a negative effect on the retrieval performance.

In absence of useless word relations, word representations are learned from the remaining word relations and are representative of the relevant part for both the query and document. Thus, the relevance scores are more discriminative among relevant documents, which leads to a higher nDCG@20 improvement than P@20 improvement.

For an intuitive understanding, we choose a specific query and document from Robust04 to show these spurious word relations. 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. 6. As we know, the meaning of short queries are vague, and forms of short queries are incomplete. For the graph without masking in Fig. 6a, the exact matching signals on "international" and "crime" are overwhelmed by many relations in documents. For the graph with heuristic masking in Fig. 6b, the exact matching signals on "international" and "crime" are obviously enhanced by masking word relations within a query and document. For Fig. 6c, the exact matching signals on "international" and "crime" are further improved. Meanwhile, negative and dispensable relations such as "individual" and "organized", "parts" and "international", "foreign" and "crime", etc., are all filtered out.



(a) Without Masking (b) Heuristic Masking (c) Adaptive Masking

Fig. 6 Attention matrices learned from DGRe with different masking strategies. The green box represents exact term matching. The yellow line is the dividing line between query and document

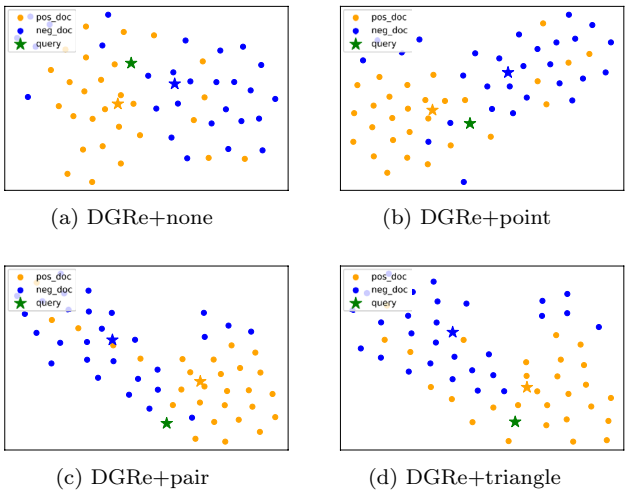
Table 4 Ranking performance comparisons among DGRe with different distance definitions on Robust04

Model	P@20	Imp.%	nDCG@20	Imp.%
DGRe+none	0.478	–	0.5464	–
DGRe+point	0.4785	0.1	0.5477	0.24
DGRe+pair	0.4789	0.19	0.5486	0.4
DGRe+triangle	**0.4811**	0.65	**0.5498**	0.99

4.4 Ablation Study for Triangle Distance

We introduce the cosine distance learning task as the auxiliary task for document ranking in DGRe. 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 DGRe with different loss functions on Robust04: (1) DGRe+none: training models with only $\mathcal{L}_{\text{rank}}$. (2) DGRe+point: training models with $\mathcal{L}_{\text{rank}} + \lambda\mathcal{C}_{\text{point}}$. (3) DGRe+pair: training models with $\mathcal{L}_{\text{rank}} + \lambda\mathcal{C}_{\text{pair}}$. (4)DGRe+triangle: training models with training models with $\mathcal{L}_{\text{rank}} + \lambda(\mathcal{C}_{\text{point}} + \mathcal{C}_{\text{pair}})$. Experimental results are shown in Table 4. Imp.% column corresponds to the relative performance improvement of each method compared with DGRe+none and best results are in bold.

The auxiliary task, i.e. cosine distance learning task, always plays a positive role in the document ranking problem in Table 4, although the improvement of DGRe+point under the P@20 evaluation is not significant. Obviously, the relative performance gain for both DGRe+point and DGRe+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 synergy effect on ranking performances shows the advantage of triangle cosine distance loss. The triangle distance loss put the constraints on document representations which keep relevant documents away from each other, so there is a better performance on nDCG@20. 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 [21] shown in Fig. 7.



(a) DGRe+none (b) DGRe+point

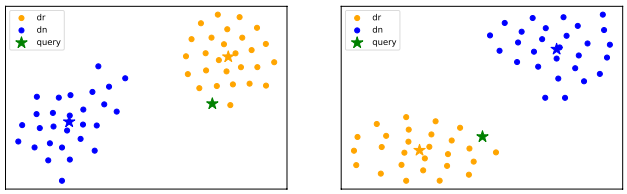
(c) DGRe+pair (d) DGRe+triangle

Fig. 7 Query and document representations from DGRe with different losses. The pentagram means the mass center of each group

Several results are obtained from Fig. 7. (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

Table 5 Ablation study for mutual Information Regularization on Robust04

Model	P@20	Imp.%	nDCG@20	Imp.%
DGRe-$\mathcal{L}_{mi}$	0.4811	–	0.5498	–
DGRe	0.4817	0.12	0.553	0.58



(a) DGRe-$\mathcal{L}_{mi}$ (b)DGRe

Fig. 8 Query and document representations from DGRe with/without $\mathcal{L}_{mi}$. The pentagram means the mass center of each group

ranking loss takes indirectly effect on learned representations. (2) (b) v.s. (d). DGRe+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. 7b that some relevant and non-relevant document points are mixed together. (3) (c) v.s. (d). DGRe+pair only defines a relevant and non-relevant document point distance and requires non-relevant document points that are far from relevant document points. This may lead to the problem in Fig. 7c 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 disriminative representation for all the query, relevant and non-relevant documents.

4.5 Ablation Study for Mutual Information Regularization

To study the role of mutual information regularization term $\mathcal{L}_{mi}$, we conducted ablation experiments on Robust04 by keeping the whole architecture of DGRe optimized without $\mathcal{L}_{mi}$. Experimental results are shown in Table 5. Imp.% column corresponds to the relative performance improvement of DGRe compared with DGRe without mutual information regular terms.

It is worth noting that the performance gain is at least 0.1% achieved by mutual information regularization in Table 5. The result indicates the regularization is essential for retrieval. The other interesting observation is that the NDCG improvement is higher than the precision improvement. One reason is that the decomposition layer makes the document representations more discriminative especially among the relevant documents. To verify this analysis, we randomly select a query, and plot query and its relevant and irrelevant document's query related representations through dimension reduction with t-sne [21] shown in Fig. 8.

The qualitative result suggests relevant document points in Fig. 8b that are scattered more widely than those in Fig. 8a, and at the same time, relevant and irrelevant document nodes are well separated. The representation distinctions of relevant documents from DGRe are larger than those from DGRe without mutual information regularization. In other words, the mutual information regularization term makes the relevant document representations more discriminative, which coincides with the comparison result in Table 5.

4.6 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 6. Imp.% column represents the relative performance improvement of each other method compared with BM25 and best results are in bold.

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 performance gain on shorter queries than these on longer queries in Table 6. Due to the addition of the word representation refinement layer and mutual information decomposition layer, DGRe'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 DGRe makes the query part emerge in the query–document representation.

One interesting observation is that nDCG@20 of DGRe is higher on short queries than it on long queries while P@20 is slightly lower on short queries than that on long

Table 6 Ranking performance comparisons on two subsets of Robust04 with different query lengths

Model	QLEN ≤ 3				QLEN > 3			
	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
DGRe	**0.4372**	13.35	**0.5124**	9.28	**0.4483**	5.48	**0.499**	2.87

queries. This inconsistent result is owing to the small average query length difference between two groups. The major reason lies in the DGRe's bias toward both short queries and relevant documents. Thus, when P@20 is more or less the same, nDCG@20 will be higher on short queries. Generally, DGRe's absolute performances on long queries are higher than those on short queries, but DGRe's performance improvement on long queries is smaller than that on short queries compared with baselines.

5 Conclusion

To reduce the effects of spurious information, we propose to remove useless word relations of BERT and disentangle the query related part of the document representation for the document ranking task, namely DGRe. To alleviate the observable confounder in word pair relations, we make the back-door adjustment on the causal graph and refine the word representations over the disentangled graph generated from our proposed adaptive masking method. To resolve the unobservable confounder in document word representations, we do the front-door adjustment in the causal graph and decompose the document word representations into query related and unrelated parts minimizing the mutual information between them. For optimization, we introduce triangle distance loss function to constrain the transformer and refinement layer and mutual information regularization to penalize the decomposition layer.

Experiments are comprehensively conducted on two public benchmark datasets, and we obtain the following results. (1) Reducing the spurious information's effects, DGRe outperforms state-of-the-art methods about 2% in terms of P@20 and nDCG@20. (2) Both masking strategies and the mutual information decomposition layer play essential roles in the performance improvement. (3) DGRe mainly prompts performances of short queries.

In the real world applications, two stage ranking paradigm, i.e. retrieval and re-ranking, is common for modern information retrieval systems. Our proposed method DGRe is mainly employed in the re-ranking stage to sort the retrieved documents according to their relevance scores to the query.

Two major limitations of DGRe are considered to be improved. Due to its low computational efficiency, DGRe cannot be directly applied to the retrieval stage. Next, we will try to improve the model efficiency and apply it to the dense retrieval scenario. Simple ReLU function is used for adaptive masking to remove useless word relations, where the decision threshold of useless word relations is fixed for different scenarios. For future work, we will use optimal transportation technique to improve the masking strategy in the transformer layer.

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

Declarations

Conflict of interest Avoid reviewers from Chinese Academy of Sciences.

Consent to participate All authors consent to participate in this work.

Consent for publication All authors consent to publish the paper.

References

1. Abbasnejad E, Teney D, Parvaneh A, Shi J, Hengel AVD (2020) Counterfactual vision and language learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 10044–10054
2. Belghazi MI, Baratin A, Rajeshwar S, Ozair S, Bengio Y, Courville A, Hjelm D (2018) Mutual information neural estimation. In: International conference on machine learning. PMLR, pp 531–540
3. Bolukbasi T, Chang KW, Zou J, Saligrama V, Kalai A (2016) Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. arXiv preprint arXiv:1607.06520

4. Chan C, Al-Bashabsheh A, Huang HP, Lim M, Tam DSH, Zhao C (2019) Neural entropic estimation: a faster path to mutual information estimation. arXiv preprint arXiv:1905.12957
5. Choi K, Lee S (2020) Regularized mutual information neural estimation. arXiv preprint arXiv:2011.07932
6. Dai Z, Callan J (2019) Deeper text understanding for IR with contextual neural language modeling. In: Proceedings of the 42nd international ACM SIGIR, pp 985–988
7. Dai Z, Callan J (2020) Context-aware term weighting for first stage passage retrieval. In: Proceedings of the 43rd international ACM SIGIR, pp 1533–1536
8. Dai Z, Xiong C, Callan J, Liu Z (2018) 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
9. Devlin J, Chang MW, Lee K, Toutanova K (2018) Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805
10. Guo J, Fan Y, Ai Q, Croft WB (2016) 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
11. Guo J, Fan Y, Pang L, Yang L, Ai Q, Zamani H, Wu C, Croft WB, Cheng X (2019) A deep look into neural ranking models for information retrieval. Inf Process Manag 102067
12. Hao Z, Lv D, Li Z, Cai R, Wen W, Xu B (2021) Semi-supervised disentangled framework for transferable named entity recognition. Neural Netw 135:127–138
13. Hendricks LA, Burns K, Saenko K, Darrell T, Rohrbach A (2018) Women also snowboard: overcoming bias in captioning models. In: Proceedings of the European conference on computer vision (ECCV), pp 771–787
14. Hjelm RD, Fedorov A, Lavoie-Marchildon S, Grewal K, Bachman P, Trischler A, Bengio Y (2018) Learning deep representations by mutual information estimation and maximization. arXiv preprint arXiv:1808.06670
15. Hu B, Lu Z, Li H, Chen Q (2014) Convolutional neural network architectures for matching natural language sentences. In: Advances in neural information processing systems, pp 2042–2050
16. Kocaoglu M, Snyder C, Dimakis AG, Vishwanath S (2017) Causalgan: learning causal implicit generative models with adversarial training. arXiv preprint arXiv:1709.02023
17. Li B, Han L (2013) Distance weighted cosine similarity measure for text classification. In: International conference on intelligent data engineering and automated learning. Springer, Berlin, pp 611–618 (2013)
18. Li Y, Tarlow D, Brockschmidt M, Zemel R (2015) Gated graph sequence neural networks. arXiv preprint arXiv:1511.05493
19. Li C, Yates A, MacAvaney S, He B, Sun Y (2020) Parade: passage representation aggregation for document reranking. arXiv preprint arXiv:2008.09093
20. Lin X, Sur I, Nastase SA, Divakaran A, Hasson U, Amer MR (2019) Data-efficient mutual information neural estimator. arXiv preprint arXiv:1905.03319
21. Maaten LVD, Hinton G (2008) Visualizing data using t-SNE. J Mach Learn Res 9:2579–2605
22. MacAvaney S, Yates A, Cohan A, Goharian N (2019) CEDR: contextualized embeddings for document ranking. In: Proceedings of the 42nd international ACM SIGIR, pp 1101–1104
23. MacAvaney S, Nardini FM, Perego R, Tonellotto N, Goharian N, Frieder O (2020) Efficient document re-ranking for transformers by precomputing term representations. arXiv preprint arXiv:2004.14255
24. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781
25. Pang L, Lan Y, Guo J, Xu J, Wan S, Cheng X (2016) Text matching as image recognition. arXiv preprint arXiv:1602.06359
26. Pearl J (1995) Causal diagrams for empirical research. Biometrika 82(4):669–688
27. Pearl J (2014) Probabilistic reasoning in intelligent systems: networks of plausible inference. Elsevier, Amsterdam
28. Peng Z, Huang W, Luo M, Zheng Q, Rong Y, Xu T, Huang J (2020) Graph representation learning via graphical mutual information maximization. In: Proceedings of the web conference 2020, pp 259–270
29. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958
30. Tang K, Niu Y, Huang J, Shi J, Zhang H (2020) Unbiased scene graph generation from biased training. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 3716–3725
31. Veitch V, Sridhar D, Blei D (2020) Adapting text embeddings for causal inference. In: Conference on uncertainty in artificial intelligence. PMLR, pp 919–928 (2020)
32. Veličković P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y (2017) Graph attention networks. arXiv preprint arXiv:1710.10903
33. Xu K, Ba J, Kiros R, Cho K, Courville A, Salakhudinov R, Zemel R, Bengio Y (2015) Show, attend and tell: neural image caption generation with visual attention. In: International conference on machine learning. PMLR, pp 2048–2057 (2015)
34. Yang P, Fang H, Lin J (2017) Anserini: enabling the use of Lucene for information retrieval research. In: Proceedings of the 40th international ACM SIGIR, pp 1253–1256
35. Yang X, Zhang H, Qi G, Cai J (2021) Causal attention for vision-language tasks. arXiv preprint arXiv:2103.03493
36. Yue Z, Zhang H, Sun Q, Hua XS (2020) Interventional few-shot learning. arXiv preprint arXiv:2009.13000
37. Zhang D, Zhang H, Tang J, Hua X, Sun Q (2020) Causal intervention for weakly-supervised semantic segmentation. arXiv preprint arXiv:2009.12547
38. Zhou J, Cui G, Zhang Z, Yang C, Liu Z, Wang L, Li C, Sun M (2018) Graph neural networks: a review of methods and applications. arXiv preprint arXiv:1812.08434