BERM: Knowledge Distillation for Contextual Relevance Matching in E-commerce Product Search

1 IMPLEMENTATION DETAILS

Here we introduce the implementation details of the whole knowledge distillation framework as follows:

- **Teacher model**. For the teacher model, we adopt BERT-Base[1] with a 12-layer Transformer encoder where the hidden size is 768 and head size is 12. We pre-train BERT-Base on a human-labeled dataset with 380,000 query-item pairs. The fine-tuned BERT-Base is then used as an expert to refine noisy click behavior data from S_{train}. The refinement rule is: if the prediction score of BERT-Base is less than α (default value is 0.3), then the raw edge is deleted; if the score is larger than β (default value is 0.7), then a new edge is added.
- **Student model**. For the student model, we adopt the proposed BERM model. We implement BERM in TensorFlow 2.0 with the high-level Estimator API. For the input query phrase (item title) BERM, we split it into several words and then truncate or pad its length to 10 (65) words. Each word embedding is acquired by the lookup operation on a static vocabulary table whose total size is 39,846. This table is generated by pre-training two billion search data with the tool of Word2Vec. The size of embedding is 128. In BERM, the size of the first-layer (second-layer) feed-forward neural network is 1024*256 (256*64) and the activation function is ReLU.
- **Training details**. We use Lazy-Adam as the optimizer and its learning rate is 1.0e-3. To reduce overfitting on the training data, we use L2 regularization on each layer of neural networks. For Data-E, we set the training epoch as 20. For Data-A and Data-S, we set the training epoch as 3. All results reported are selected according to the best AUC values on the testing set.

2 ABLATION STUDY

In the ablation study, we use Data-E and Data-A as the experimental datasets.

2.1 Integration of embedding

$$\boldsymbol{E}_{all} = \text{CONCAT}(\boldsymbol{E}^{Q}_{seq}, \boldsymbol{E}^{I}_{seq}, \boldsymbol{E}_{int}, \boldsymbol{E}_{Q-I-Q}, \boldsymbol{E}_{I-Q-I}). \quad (1)$$

To further examine the importance of each component in the final embedding of BERM, we remove one or two components from it (Equation 1) at a time and examine how the change affects its overall performance. There are three components in the complete BERM: representation-based embedding ($\boldsymbol{E}^{Q}_{seq}$, $\boldsymbol{E}^{I}_{seq}$), interaction-based embedding ($\boldsymbol{E}_{int}$), and metapath embedding ($\boldsymbol{E}_{Q-I-Q}$, $\boldsymbol{E}_{I-Q-I}$).

The corresponding results on Data-E and Data-A are reported in Table 1 and 2. We have the following empirical observation and analysis:

- In general, the both-component setting outperforms the single-component setting but is worse than the triple-component setting (i.e., BERM). It demonstrates that different components in BERM have different positive effects on the overall performance and they cannot replace each other.
- The introduction of k-order relevance modeling can bring stable advancement to each 0-order relevance model. For example, "$\boldsymbol{E}^{Q}_{seq}, \boldsymbol{E}^{I}_{seq}, \boldsymbol{E}_{Q-I-Q}, \boldsymbol{E}_{I-Q-I}$" surpasses "$\boldsymbol{E}^{Q}_{seq}, \boldsymbol{E}^{I}_{seq}$" 6.83% according to the metric of AUC on Data-A. This demonstrates that applying metapath embedding to relevance matching can make effective use of the neighboring nodes' information in the user behavior graph.

Table 1: Ablation study on Data-E. Different variants are obtained by removing one or two components from BERM.

Models	Data-E AUC	Data-E F1-score	Data-E FNR↓
$\boldsymbol{E}^{Q}_{seq}, \boldsymbol{E}^{I}_{seq}$	0.8537	0.8044	0.3560
$\boldsymbol{E}_{int}$	0.8595	0.8037	0.3464
$\boldsymbol{E}_{Q-I-Q}, \boldsymbol{E}_{I-Q-I}$	0.8430	0.8173	0.2995
$\boldsymbol{E}^{Q}_{seq}, \boldsymbol{E}^{I}_{seq}, \boldsymbol{E}_{int}$	0.8638	0.8086	0.3331
$\boldsymbol{E}_{int}, \boldsymbol{E}_{Q-I-Q}, \boldsymbol{E}_{I-Q-I}$	0.8761	0.8221	**0.2758**
$\boldsymbol{E}^{Q}_{seq}, \boldsymbol{E}^{I}_{seq}, \boldsymbol{E}_{Q-I-Q}, \boldsymbol{E}_{I-Q-I}$	0.8656	0.8190	0.2922
BERM	**0.8785**	**0.8256**	0.2966

Table 2: Ablation study on Data-A.

Models	Data-A AUC	Data-A F1-score	Data-A FNR↓
$\boldsymbol{E}^{Q}_{seq}, \boldsymbol{E}^{I}_{seq}$	0.8067	0.8660	0.3697
$\boldsymbol{E}_{int}$	0.8289	0.9084	0.4459
$\boldsymbol{E}_{Q-I-Q}, \boldsymbol{E}_{I-Q-I}$	0.8500	**0.9114**	0.4110
$\boldsymbol{E}^{Q}_{seq}, \boldsymbol{E}^{I}_{seq}, \boldsymbol{E}_{int}$	0.8776	0.9070	0.4743
$\boldsymbol{E}_{int}, \boldsymbol{E}_{Q-I-Q}, \boldsymbol{E}_{I-Q-I}$	0.8824	0.9099	0.3705
$\boldsymbol{E}^{Q}_{seq}, \boldsymbol{E}^{I}_{seq}, \boldsymbol{E}_{Q-I-Q}, \boldsymbol{E}_{I-Q-I}$	0.8750	0.9094	0.3753
BERM	**0.8862**	0.9107	**0.3673**

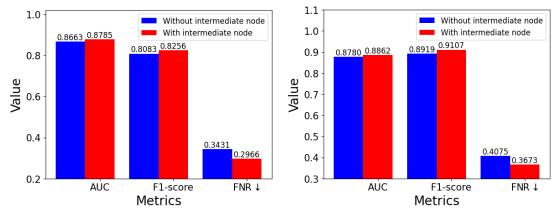
2.2 Effect of the intermediate node

The metapath defined in BERM includes the intermediate node. To further investigate the effect of the intermediate node, we compare the performances of BERM with the intermediate node (i.e., "Q-I-Q" and "I-Q-I") and BERM without intermediate node (i.e., "Q-Q" and "I-I") in Fig. 1. We observe that BERM with intermediate node performs better than the other one on Data-E and Data-A. So we conclude that the intermediate node has strong semantic closeness to the anchor node and cannot be ignored in our method.

3 SENSITIVITY ANALYSIS

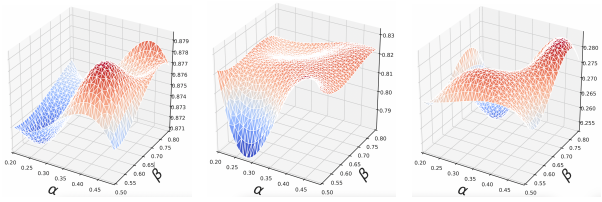
In sensitivity analysis, we use Data-E and Data-A as the experimental datasets.

[1] https://storage.googleapis.com/bert_models/2020_02_20/uncased_L-12_H-768_A-12.zip

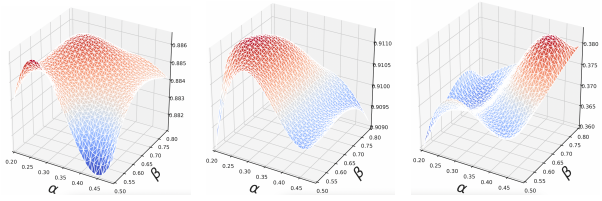


(a) Data-E (b) Data-A

Figure 1: Effect of the intermediate node. The red (blue) bar represents BERM with (without) the intermediate node.



(a) AUC of Data-E (b) F1-score of Data-E (c) FNR of Data-E



(d) AUC of Data-A (e) F1-score of Data-A (f) FNR of Data-A

Figure 2: Effect of different values of α and β. (a), (b), and (c) are the results of Data-E; (d), (e), and (f) are the results of Data-A. The red (blue) color corresponds to the high (low) value.

3.0.1 Thresholds α and β. In BERM, α decides how many edges of the noisy click behavior should be deleted and β decides how many hidden useful edges should be retrieved. To investigate the sensitivity of α and β, we conduct experiments with 16 different hyper-parameter settings where α ranges from 0.2 to 0.5 and β ranges from 0.5 to 0.8. We apply the three-order curve interpolation method to show the final results in Fig. 2. In general, the results of BERM are robust to the change of hyper-parameter α and β on either Data-E or Data-A. For example, the maximum error of AUC is no more than 1%. So we conclude that user behaviors play a major role in the performance of BERM and the knowledge from BERT provides auxiliary effects for it.

3.0.2 The selection of neighbor structure. In BERM, the selection of neighbor structure directly affects which context information is transmitted to the anchor node. A good selection strategy can aggregate valuable neighboring node information to enrich the anchor node's representation. To investigate the effect of different neighbor structure selection strategies on BERM and seek a relatively optimal solution, we use different values of hyper-parameter λ to control the

Table 3: Effect of different neighbor selection strategies on Data-E.

Rate	Click+BERT's score			Purchase+BERT's score		
	AUC	F1-score	FNR↓	AUC	F1-score	FNR↓
$\lambda = 0.0$	**0.8785**	**0.8256**	0.2966	**0.8785**	**0.8256**	0.2966
$\lambda = 0.2$	0.8779	0.8237	**0.2834**	0.8777	0.8236	**0.2810**
$\lambda = 0.4$	0.8770	0.8200	0.3198	0.8756	0.8214	0.3068
$\lambda = 0.6$	0.8670	0.8085	0.4000	0.8696	0.8045	0.3694
$\lambda = 0.8$	0.8679	0.8101	0.3901	0.8660	0.8105	0.3262
$\lambda = 1.0$	0.8656	0.8091	0.3850	0.8671	0.8109	0.3727

Table 4: Effect of different neighbor selection strategies on Data-A.

Rate	Click+BERT's score			Purchase+BERT's score		
	AUC	F1-score	FNR↓	AUC	F1-score	FNR↓
$\lambda = 0.0$	**0.8862**	0.9107	**0.3673**	**0.8862**	0.9107	**0.3673**
$\lambda = 0.2$	0.8849	**0.9113**	0.3794	0.8830	**0.9117**	0.3831
$\lambda = 0.4$	0.8821	0.9112	0.4016	0.8818	0.9100	0.4131
$\lambda = 0.6$	0.8779	0.9068	0.4793	0.8783	0.9064	0.4844
$\lambda = 0.8$	0.8802	0.9076	0.4676	0.8790	0.9084	0.4683
$\lambda = 1.0$	0.8792	0.9077	0.4671	0.8787	0.9079	0.4716

ratio between user behavior and BERT's score. Specifically, we calculate a new score $Score_{new}(Q, I) = \lambda * User(Q, I) + (1-\lambda) * Score(Q, I)$ where $User(Q, I)$ is the user behavior feature (e.g., for click behavior, $User(Q, I) = 1$ if click behavior happens between query Q and item I). The addition and deletion of edges refer to $Score_{new}(Q, I)$, rather than $Score_(Q, I)$. We report the results with different λ in Table 3 and 4. From them, we can conclude that:

- Using BERT's score is better than using user behaviors for the selection of neighbors. Therefore, the value of AUC or F1-score gradually decreases with the increase of λ; the value of FNR increases with the increase of λ. The optimal $\lambda \in [0.0, 0.2]$.
- According to the metric of FNR, the purchase behavior is better than the click behavior on Data-E. The reason for it is that purchase behaviors reveal more accurate semantic relevance information than click behaviors. However, the purchase behavior is worse than the click behavior on Data-A. We think that it is caused by the sparsity of purchase behaviors in the dataset of all categories.