ReprBERT: Distilling BERT to an Efficient Representation-Based Relevance Model for E-Commerce

Shaowei Yao
Alibaba Group
Hangzhou, China
yaoshaowei@alibaba-inc.com

Jiwei Tan*
Alibaba Group
Hangzhou, China
jiwei.tjw@alibaba-inc.com

Xi Chen
Alibaba Group
Hangzhou, China
gongda.cx@taobao.com

Junhao Zhang
Alibaba Group
Hangzhou, China
junhao.zjh@alibaba-inc.com

Xiaoyi Zeng
Alibaba Group
Hangzhou, China
yuanhan@taobao.com

Keping Yang
Alibaba Group
Hangzhou, China
shaoyao@taobao.com

ABSTRACT

Text relevance or text matching of query and product is an essential technique for e-commerce search engine, which helps users find the desirable products and is also crucial to ensuring user experience. A major difficulty for e-commerce text relevance is the severe vocabulary gap between query and product. Recently, neural networks have been the mainstream for the text matching task owing to the better performance for semantic matching. Practical e-commerce relevance models are usually representation-based architecture, which can pre-compute representations offline and are therefore online efficient. Interaction-based models, although can achieve better performance, are mostly time-consuming and hard to be deployed online. Recently BERT has achieved significant progress on many NLP tasks including text matching, and it is of great value but also big challenge to deploy BERT to the e-commerce relevance task. To realize this goal, we propose ReprBERT, which has the advantages of both excellent performance and low latency, by distilling the interaction-based BERT model to a representation-based architecture. To reduce the performance decline, we investigate the key reasons and propose two novel interaction strategies to resolve the absence of representation interaction and low-level semantic interaction. Finally, ReprBERT can achieve only about 1.5% AUC loss from the interaction-based BERT, but has more than 10% AUC improvement compared to previous state-of-the-art representation-based models. ReprBERT has already been deployed on the search engine of Taobao and serving the entire search traffic, achieving significant gain of user experience and business profit.

So this paper is to solve deployment not model perf itself?

Ok, so what is interaction-based BERT?

CCS CONCEPTS

• **Information systems** → **Relevance assessment**; **Similarity measures**; *Document representation*; *Query representation*.

*Jiwei Tan is the corresponding author.

KDD '22, August 14–18, 2022, Washington, DC, USA.

ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539090

KEYWORDS

e-commerce, text matching, knowledge distillation

ACM Reference Format:
Shaowei Yao, Jiwei Tan, Xi Chen, Junhao Zhang, Xiaoyi Zeng, and Keping Yang. 2022. ReprBERT: Distilling BERT to an Efficient Representation-Based Relevance Model for E-Commerce. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3534678.3539090

1 INTRODUCTION

Large e-commerce portals such as Taobao[1] and Amazon[2] are serving hundreds of millions of users with billions of products everyday. Search engine is an important technique for e-commerce to help users find the desirable products they want, according to the queries they enter. Nevertheless, commercial e-commerce search engines are usually optimized to boost users' engagement and conversion, possibly at the cost of relevance in some cases [1]. Presenting products that do not match search query intent will probably degrade customer experience and hamper customers' long-term trust and engagement. Therefore, measuring the relevance between queries and products accurately is crucial to the e-commerce search engine.

Traditional methods relying on hand-crafted features are usually poor at addressing the vocabulary gap [28], and semantic matching is needed to solve this problem. Recently significant progress has been achieved by advanced neural network models, which have been the mainstream for text matching. Neural text matching models can usually be categorized into two classes: *representation-based* models and *interaction-based* models. The representation-based models individually encode both the query and the product title into single embedding vectors, which allows the embeddings to be computed offline. As a result, the representation-based models are online efficient and are mostly applied to industrial search engines. However, individually encoding queries and products into the fixed-dimensional vectors may lose the fine-grained matching information, which causes degraded performance. Relatively, the *interaction-based* models first match different parts of the query with different parts of the products at low level, and then aggregate the partial evidence of relevance to make the final decision. As

[1]http://www.taobao.com
[2]http://www.amazon.com

there are more fine-grained matching information to be considered, the interaction-based models usually perform better on most text matching benchmarks and are more studied recently in the research area. Unfortunately, interaction-based models cannot have embeddings of queries and products pre-computed offline. They are difficult to be deployed to practical online platforms with large traffic and low latency. In addition to low online efficiency, the direct matching at the word-level of traditional interaction-based models also suffers from insufficiency of considering context information, which may result in the mismatch problem. Take the query "apple mobile phone" and a product "delicious apple" as an example. Interaction models usually fail to give a perfect irrelevant score since the "apple" in this query matches the word in the product, whereas the context "mobile phone" should have the most effect.

Recently the popular pre-trained language models like BERT [3] have achieved excellent results in various NLP tasks including text matching. BERT applied to text matching can also be regarded as a kind of interaction-based model. Except for performing fine-grained interaction between queries and products at each layer, BERT also encodes the context information of queries and products, which resolves the disadvantages of interaction-based models and representation-based models at the same time. In addition, the multi-layer architecture of BERT allows the model to perform interaction based on not only high-level semantics but also low-level semantics. However, BERT for text matching also suffers from the disadvantage of interaction-based model as very time-consuming, making it a big challenge to be deployed to practical online service.

Although there are recent efforts trying to deploy BERT online through techniques like knowledge distillation and model simplification [9, 21], it is still impractical to be applied to the relevance task of e-commerce scenario, where thousands of candidates should be measured for a given query at the same time. As a result, deploying BERT to practical information retrieval scenario is challenging despite very valuable. In this paper, we propose a different and desirable solution to this problem. We propose ReprBERT, which distills BERT to a representation-based architecture but can still achieve competitive results, having the advantage of efficiency and scalability for practical online service. Although directly using BERT to encode query and product separately can realize a representation-based model, it unfortunately suffers from great decline of performance. To resolve the problem, we first propose a context-guided attention mechanism to improve the encoder for producing representation. Afterwards, we find a reason of performance decline is the absence of the interaction between query and product representations. Therefore we propose a late interaction strategy which can significantly boost the performance. Moreover, we further reveal another crucial reason is the absence of low-level semantic interaction, as the inherent advantage of interaction-based models. We propose a novel intermediate interaction strategy, which brings further improvement to the ReprBERT model. Targeting at the e-commerce relevance task, we propose the task-oriented distillation that makes the student ReprBERT model approximate the teacher model on our definite relevance task. As a result, the proposed ReprBERT can have close performance to the sophisticated BERT with about 1.5% AUC loss, but has more than 10% AUC improvement compared to state-of-the-art representation-based models. Besides offline evaluation on large-scale real dataset, ReprBERT has also been deployed on the Taobao search engine. Online A/B testing verifies great improvement on the online relevance and Gross Merchandise Value (GMV). The serving latency is as low as previous representation-based model at about 10ms, which can satisfy e-commerce platforms of extremely large traffic like Taobao.

The contribution of this paper is summarized as follows:

- We propose a method to deploy BERT to online retrieval systems, by distilling BERT to a representation-based architecture. The model has competitive performance compared to complicated BERT but is very computationally efficient.
- We reveal the key reasons of the distillation performance loss are the absence of representation interaction and low-level semantic interaction. Accordingly we propose two novel interaction strategies that can effectively reduce the performance gap.
- We introduce the ReprBERT model[3] which has already been deployed on the largest Chinese e-commerce platform Taobao, serving the entire search traffic for over a year.

2 RELATED WORK

2.1 Text Matching

The e-commerce relevance learning is typically a text matching task. Text matching has long been a hot research topic due to its importance in information retrieval and search engine. Text matching models typically take two textual sequences as input and predict a numerical value or a category indicating their relationship. Early works mostly perform keyword-based matching relying on manually defined features, such as TF-IDF similarity and BM25 [20]. These methods cannot effectively utilize raw text features and usually fail to evaluate semantic relevance.

Recently with the development of deep learning, neural-based text matching models are engaged to solve the semantic matching task and have achieved promising performance. These neural-based methods can be roughly divided into *representation-based* models and *interaction-based* models. The representation-based models are usually Siamese-like architecture, which consists of two identical neural networks, each taking one of the two inputs. DSSM [6] is one representative architecture that employs two separate deep fully-connected networks to encode the query and the document. Meanwhile, more sophisticated architectures can be adopted to enhance the ability of learning semantic representations. For example, ARC-I [5] and CDSSM [22] use CNNs to model the internal structures of language objects and the interaction between them. LSTM-DSSM [16] and LSTM-RNN [17] use RNNs to explicitly model word dependencies in the sentences. Typically dot-product, cosine, or parameterized non-linear layers are used to measure the similarity between query and document representations. Since individually encoding both the queries and documents, the embeddings of them can be pre-computed offline. Therefore, representation-based methods are online efficient and are widely used in industrial search engines. However, the encoding procedures of two inputs are independent with each other, making the final classifier hard to predict their relationship.

difference?

so two tower model is representation bsed model

Assuming interaciton based means not two tower so more interaction at the lower level of neural network

[3]The source code of ReprBERT is available at https://github.com/QAQ-v/ReprBERT.

To overcome the weakness of the representation-based models, *interaction-based* models are proposed which are designed to perform more interaction between two inputs. The interaction-based models first match different parts of the query with different parts of the document at low level, and then aggregate the partial evidence of relevance to make the final decision. Sophisticated techniques can also be introduced in the aggregation procedure. ARC-II [5] and MatchPyramid [18] use CNN to learn rich hierarchical matching patterns over the matching matrix. Match-SRNN [26] further models the recursive matching structure to better capture long-distance dependency between the interactions. DecompAtt [19] leverages attention mechanism for alignment. These methods capture more interactive features between inputs which brings significant improvement on model performance. But the interaction-based models are mostly time-consuming and the embeddings of query and documents cannot be pre-computed offline, which is hard to be deployed to practical online service.

Never heard these before, probably because they can't be productionized due to latency…

More recent studies are built upon pre-trained language models. The most notable example is BERT [3], a pre-trained deep bidirectional Transformers model which can also be seen as the interaction-based model. The typical paradigm of BERT-based relevance model is to feed the query-document pair into BERT and then build a non-linear classifier upon BERT's [CLS] output token to predict the relevance score [14]. Nogueira et al. [15] propose duoBERT that learns the relevance of a pair of texts in a pair-wise fashion. With extremely large corpus for pre-training, these methods can achieve new state-of-the-art performance on various benchmarks but are highly expensive in practice. The e-commerce relevance task can also be viewed as a text matching problem. Unfortunately, commercial e-commerce search engines usually have large traffic and require low latency, making the interaction-based or the more sophisticated BERT models a big challenge to be deployed online. In this work we tackle this problem by proposing to distill the interaction-based BERT to a representation-based architecture, which will be highly efficient for online service.

That's why there's distilled BERT

2.2 E-commerce Relevance Learning

Text relevance is an important technique for e-commerce search engines. Previously there is no commonly-used dataset and benchmark for the e-commerce relevance task, so previous works usually evaluate their models on the online service and the real-world test set constructed by themselves. As human annotation is expensive, some previous e-commerce relevance models use click-through data as implicit feedback label. For example, Jiang et al. [8] propose a typical framework for e-commerce relevance learning. A Siamese network is adopted to learn pair-wise relevance of two products to a query. They investigate training the model with user clicks and batch negatives, followed by fine-tuning with human supervision to calibrate the score by pair-wise learning. Xiao et al. [28] propose a co-training framework to address the data sparseness problem by investigating the instinctive connection between query rewriting and semantic matching. Zhang et al. [31] propose a multi-task learning framework of query intent classification and semantic textual similarity to improve semantic matching efficiency. Liu et al. [11] propose a heterogeneous network embedding method for e-commerce relevance system, which focuses on leveraging contextual information including query/item incidence network and the users' historical behavior sequences. Most recently Yao et al. [30] investigate the weakness of training relevance model with click signals, and address this problem by considering samples of different relevance confidence. They come up with a new training objective to learn a robust relevance model with desirable score distribution from the carefully constructed dataset, and achieve state-of-the-art results. Previous efforts found both the effectiveness and weakness of user behavior data for the e-commerce relevance task. However, how to exploit the noisy user behaviour data well is still a difficult problem. In this work, we explore the superiority of the pre-trained models like BERT, which can leverage the abundant user behaviour data and learn better from the weak signal of relevance.

2.3 Knowledge Distillation

Knowledge Distillation (KD) [4] is one of the most popular model compression techniques. With the limited computational resource and the strict latency requirement of the online service, KD is an effective way to make real-word applications benefit from expensive models like BERT. The main idea of KD is to first train an expensive and high-performance teacher model, and then learn a shallow student network to mimic the teacher model. During the training process of the student model, the loss function is defined to minimize the distance between outputs of the teacher model and outputs of the student model, i.e., the student model is optimized to learn the soft labels produced by teacher. Such soft labels remove the noise in original training data and are more informative to the student than the 0/1 labels. Recently there have been many works focus on applying KD technique to pre-trained language models like BERT. DistilBERT [21] distills BERT into shallow Transformers in the pre-train stage, while BERT-PKD [23] distills BERT in the fine-tune stage and learns distilled knowledge from intermediate layers besides the output layer. TinyBERT [9] further expands distillation to attention matrices and embedding matrices. These methods mostly distill BERT into a shallow student model with the same interaction-based architecture, which is still computational expensive in some situation with extremely large traffic. Except distilling BERT into a shallow BERT, there are efforts on distillation between different architectures. Tang et al. [24] propose distillation of BERT to a single layer BiLSTM. Lu et al. [12] propose to decouple the two-sentence input during encoding them and produce the embeddings for query and document independently, which allows document embeddings to be pre-computed offline. Xu et al. [29] propose the privileged features distillation that train the teacher model additionally utilizes the privileged features. However, they ignore the interaction between the intermediate representations, which is the critical part of BERT model. In this work we propose an intermediate interaction strategy to help the student model mimic the teacher model better, which is verified to have significant improvement on performance.

3 METHODOLOGY

In this section, we will first describe the ReprBERT encoder, as the basis of our model. Then we will propose two interaction strategies that are crucial to model performance. Finally we introduce how

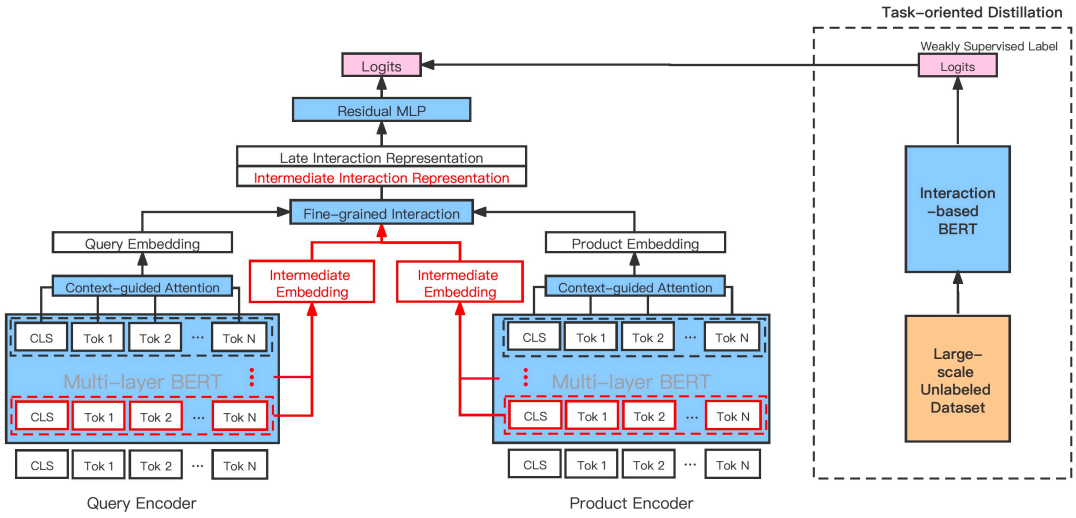

Figure 1: The illustration of ReprBERT. ReprBERT is a Siamese architecture with shared BERT encoders to encode query and product title. Their final embeddings and intermediate embeddings are extracted from the corresponding layers through attention mechanism. These embeddings are fed into the interaction module to get the interaction representations. Finally the residual MLP is used to predict the distribution of target classes.

ReprBERT is trained by knowledge distillation of the teacher model. The architecture of ReprBERT is illustrated in Figure 1.

3.1 ReprBERT Encoder

BERT is used as the encoder to produce query/product representations (embeddings). BERT [3] employs a multi-layer architecture, and each layer consists of two sub-layers: a multi-head attention mechanism and a position-wise feed-forward network (FFN). The multi-head attention mechanism is built on the scaled dot-product attention, and learns dependency from all the words to update the word representations. Formally, given an L-layer BERT and the input sequence $\mathbf{X} = (\mathbf{x}_0, ..., \mathbf{x}_n)$ where $\mathbf{x}_0$ denotes the [CLS] token, the BERT encoder is used to encode the input at l-layer as

$$\begin{aligned} \hat{\mathbf{H}}^{(l-1)} &= \text{MultiHeadAttn}\left(\mathbf{H}^{(l-1)}\right) + \mathbf{H}^{(l-1)} \\ \mathbf{H}^{(l)} &= \text{FFN}\left(\hat{\mathbf{H}}^{(l-1)}\right) + \hat{\mathbf{H}}^{(l-1)} \end{aligned} \tag{1}$$

where $\mathbf{H}^{(0)} = \mathbf{X}$ and $\mathbf{H}^{(L)} = (\mathbf{h}_0^{(L)}, \cdots, \mathbf{h}_n^{(L)})$. $\mathbf{H}^{(l)}$ is the intermediate representation at l layer. The most common way of BERT encoder is to take the [CLS] representation $\mathbf{h}_0^{(L)}$ as the query or product embedding. An improved way can be the pooling-based methods, e.g. mean pooling of the representations of all tokens, which can incorporate more word-level information and result in better representations. Unfortunately, such methods still do not encode the fine-grained word-level information well, due to the lack of context interaction among these token embeddings. For example, there are usually lots of co-occurrences of "case" and "cover", like "airpods case cover" is frequently appeared in many product titles. As a result, the pooling-based method tends to give higher score for the pair of "airpods case" and "pillow cover". The reason is the relevance between "case" and "cover" is learned and incorporated, but the collision between "airpods" and "pillow" can be considered little when using mean pooling to produce the representation.

To alleviate this problem, we propose a context-guided attention mechanism to produce better representations. Following the self-attention definition in [25], the context-guided attention regards the projected embedding of the sentence (i.e., the [CLS] token) $\mathbf{H}_{0,q}^{(L)} W_q$ as Query vector Q_q (q denotes query and p denotes product), other projected token embeddings $\mathbf{H}_{1...n,q}^{(L)} W_q$ are viewed as the same Key and Value vectors $K_q = V_q$. Here W_q and W_p are learnable parameter matrix which project $\mathbf{H}_{\mathbf{q}}^{(\mathbf{L})}$ or $\mathbf{H}_{\mathbf{p}}^{(\mathbf{L})}$ to a specific dimension d. Therefore, $\mathbf{H}_{0,q}^{(L)} W_q$ attends to the matrix of all other vectors $\mathbf{H}_{1...n,q}^{(L)} W_q$ to generate their attention weights and produce the query embedding $\mathbf{q}$, as Equation 2:

$$\begin{aligned} \mathbf{q} &= (Q_q K_q^T) V_q = (\mathbf{H}_{0,q}^{(L)} W_q (\mathbf{H}_{1...n,q}^{(L)} W_q)^T) \mathbf{H}_{1...n,q}^{(L)} W_q \\ \mathbf{p} &= (Q_p K_p^T) V_p = (\mathbf{H}_{0,p}^{(L)} W_p (\mathbf{H}_{1...n,p}^{(L)} W_p)^T) \mathbf{H}_{1...n,p}^{(L)} W_p \end{aligned} \tag{2}$$

Compared to the commonly used [CLS] token embedding or pooling-based methods, the proposed context-guided attention could better encode the context information and model the relevance and collision relations between tokens in word-level.

3.2 Late Interaction

As discussed above, one major weakness of the representation-based model is lacking interaction between the representations of query and product. Unfortunately, the Siamese architecture will be destroyed if the model performs interaction when encoding query and product. Inspired by Khattab and Zaharia [10], we propose a late interaction module in ReprBERT to introduce interaction after the embeddings of query $\mathbf{q}$ and product $\mathbf{p}$. As Equation 3, two

embeddings are combined via summation, subtraction and max pooling to get the late interaction representation $\mathbf{r}_{\text{Late}}$:

$$\begin{aligned}\mathbf{r}_{\text{Late}} &= \text{Finegrained-Interaction}(\mathbf{q}, \mathbf{p}) \\ &= \text{Concat}(\text{MaxPool}(\mathbf{p}, \mathbf{q}), \mathbf{p} - \mathbf{q}, \mathbf{p} + \mathbf{q})\end{aligned} \tag{3}$$

where $\mathbf{r}_{\text{Late}} \in \mathbb{R}^{3d}$ represents the matching representation of the query and product. Compared to most representation-based methods that directly use cosine similarity as the metric, more finegrained interaction information can be encoded into the representation $\mathbf{r}_{\text{Late}}$ through the late interaction, which has also been demonstrated in previous study [30] that can help the model make more accurate decision about relevance.

3.3 Intermediate Interaction

With the late interaction, ReprBERT is enhanced with the interaction between the encoded representations of query and product. However, there is still considerable performance gap compared to the original BERT. We consider one major reason is there still exists essential difference between them: the BERT performs self-attention at each layer, and hence all the intermediate representations of the query have interaction with the intermediate representations of the product at each layer. While ReprBERT separately encodes the query and the product, if only computes the interaction at the late interaction module, the intermediate representations of the query and the product are independent with each other. To resolve this problem and further reduce the performance gap, we propose a novel intermediate interaction strategy. The intermediate interaction module first extracts *all* the intermediate representations of query $\mathbf{q}^{(l)}$ and product $\mathbf{p}^{(l)}$ at each layer, and then performs the interaction the same as the late interaction module, respectively:

$$\begin{aligned}\mathbf{r}^{(l)} &= \text{Finegrained-Interaction}(\mathbf{p}^{(l)}, \mathbf{q}^{(l)}) \\ \mathbf{r}_{\text{Intermediate}} &= \text{WeightedPool}(\mathbf{r}^{(1)}, ..., \mathbf{r}^{(L-1)})\end{aligned} \tag{4}$$

where WeightedPool is the weighted average pooling layer. In this way, ReprBERT can not only perform the interaction between the intermediate representations of inputs like BERT, but also keep the advantage of the representation-based architecture to pre-compute the embeddings offline. The main cost is the storage space of the intermediate representation. Fortunately the dimension of the intermediate representation is fixed no matter how many intermediate layers there are. Afterwards, $\mathbf{r}_{\text{Intermediate}}$ and $\mathbf{r}_{\text{Late}}$ are combined through max-pooling according to Equation 5, which is further provided to a Multi-Layer Perceptron (MLP) with residual connection, to project into a 2-dimensional vector $\mathbf{y}$ for binary classification. The final relevance score is taken as the value on the positive class of distribution $\mathbf{y}$ after the Softmax function:

$$\begin{aligned}\mathbf{r} &= \text{MaxPool}(\mathbf{r}_{\text{Intermediate}}, \mathbf{r}_{\text{Late}}) \\ \mathbf{y} &= \text{Softmax}(\text{MLP}(\text{MLP}(\mathbf{r}) + \mathbf{r}))\end{aligned} \tag{5}$$

3.4 Knowledge Distillation

To make ReprBERT approximate the performance of BERT, knowledge distillation is introduced to train the model. To get the best possible ReprBERT, the first step is to realize a state-of-the-art teacher model. Therefore, instead of using the original BERT [3] simply, we use a StructBERT [27] model as the teacher, which has been verified better than BERT by improving the training process and training target. Moreover, StructBERT proposes a continue training strategy that can improve the performance a lot by introducing in-domain language data for specific tasks. As shown in our experiments, the StructBERT continue-trained with e-commerce data can achieve much better results on the task. Therefore in our work we take the best StructBERT as the teacher model. Fortunately, StructBERT have the same model architecture with BERT, and hence the only difference of using BERT or StructBERT as the teacher for our task is the different teacher performance.

3.4.1 Teacher model construction. We employ the 12-layer StructBERT as the teacher model. StructBERT is also built upon the same BERT architecture as a multi-layer bidirectional Transformer network, but incorporates the language structure information in the pre-training process. Besides pre-training on the general corpus, StructBERT is continue-trained on large unlabeled e-commerce dataset according to two auxiliary well-designed tasks: shuffle mask language model (LM) and sequential sentence prediction. These two auxiliary pre-training tasks are more difficult than the pre-training tasks of BERT, which can help the model better exploit inherent language structures. Meanwhile, continue-training on the e-commerce data helps the model obtain more in-domain knowledge and perform better on downstream e-commerce relevance task. The StructBERT model is further finetuned on the task-specific annotation data to achieve a best possible teacher model.

3.4.2 Student model training. ReprBERT is initialized using the first several layers' parameters of the teacher model to get the general knowledge. We also tried initializing ReprBERT using the jump layers (e.g. 1, 3, 5 layers or 2, 4, 6 layers) of the teacher model following Jiao et al. [9] but did not get better results. The most common way to distill knowledge is first unsupervisedly pre-training a shallow student model on a large unlabeled dataset, and then training the student by minimizing the cross entropy loss between the teacher Softmax output and the student Softmax output on the human-annotated dataset, besides minimizing the cross entropy between the student prediction and the actual label. Since our downstream task is definite, and the main intention is to make the outputs of ReprBERT and teacher model as similar as possible, we directly use the teacher model to annotate a large-scale unlabeled dataset. Then we get the teacher Softmax outputs as the soft labels y^{soft}, and minimize the cross entropy loss between y^{soft} and the student prediction y, which can be seen as a cross-structure distillation:

$$loss = -\sum_{i=1}^{N}(y_i^{\text{soft}}\log(y_i) + (1 - y_i^{\text{soft}})\log(1 - y_i)) \tag{6}$$

The large-scale unlabeled dataset is constructed from real online search logs, containing tens of millions of query-product pairs. After the cross-structure distillation, we finetune ReprBERT on the human-annotated dataset with hard label y^{hard} and soft label y^{soft}:

$$\begin{aligned}loss = -\sum_{i=1}^{N}&(y_i^{\text{soft}}\log(y_i) + (1 - y_i^{\text{soft}})\log(1 - y_i) \\ &+ \lambda(y_i^{\text{hard}}\log(y_i) + (1 - y_i^{\text{hard}})\log(1 - y_i)))\end{aligned} \tag{7}$$

Dataset	#sample	#query	#product	#good	#bad
Train	960,938	81,191	988,578	804,513	156,425
Valid	143,571	15,365	138,693	122,835	20,736
Test	207,965	26,201	193,108	166,028	41,937

Table 1: Statistics of the human annotation dataset.

where N is the number of samples, λ is a small scalar that weights down the hard-label loss.

4 EXPERIMENTS

4.1 Dataset

The large unlabeled dataset used for knowledge distillation is collected by randomly sampling from the search logs of Taobao within a year, which contains about 50 million query-product pairs. The samples are then annotated by the teacher model to generate soft labels for knowledge distillation. For the finetuning and evaluation of the proposed model, we use a large-scale human-annotated dataset. The dataset contains query-product pairs also sampled from the search logs, and then labeled Good (relevant) or Bad (irrelevant) by experienced human annotators. This is a daily task running in Taobao, which has accumulated more than one million labeled samples. The average length of the query and title is 7.3 and 32.6 Chinese characters on the whole annotated data, respectively. The dataset is split to training, validation and test set, as detailed in Table 1.

4.2 Experimental Setup

4.2.1 Teacher model setup. The teacher model used in this paper is the best possible model we explored in our extensive experiments, namely the 12-layer StructBERT [27] that is first pre-trained on general domain data, and then continue-trained on the e-commerce data, and finally fine-tuned using about 1 million annotated training data in Table 1, to achieve the best results on the validation data. The word vocabulary size is 21128, which is the same with BERT in Chinese language. The first special classification token [CLS] is fed into the 2-layer MLP to generate the 2-dimensional classification vector for binary classification. The batch size is set to 32 and the learning rate equals to $3e-5$. The teacher model learns query-product relevance from the binary labels based on the cross-entropy loss. AdamW is used to optimize the model with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-6$, decay_rate = 0.01. We train the model on 1 Tesla P100 GPU card and the model can converge in 10 hours.

4.2.2 ReprBERT setup. The encoders of ReprBERT are architecture like BERT or StructBERT, but the 12 layers of the encoders can be reduced to improve efficiency. After balancing the effectiveness and efficiency, ReprBERT adopts 2 layers which can still achieve competitive performance. Before the context-guided-attention, the dimensions of the query and product embeddings are projected into 128, which further improves the efficiency and reduces the cost of storage in practice. The parameter size of the first residual MLP used to project the final matching representation is (128×128). The parameter size of the second MLP used to output the binary classification vector is (128×2). The batch size is set to 128 and the learning rate equals to $2e$-5 with decay rate of 0.8 for every $1e7$ steps. The weight $\lambda = 0.5$ to weight down the hard-label loss. ReprBERT is also trained based on the cross-entropy loss and optimized by Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-8$. These hyperparameters are chosen from the experimental results on the validation set. We train the model on 20 Tesla P100 GPU cards and the model can converge in a day. The convergence is reached when the ROC-AUC score does not improve on the validation set.

4.3 Baselines and Evaluation Metrics

4.3.1 Baselines. We adopt several state-of-the-art methods for comparison as follows. All baselines are finetuned with the training set in Table 1 to achieve the best results on the validation set. Among these baselines, Siamese BERT, MASM and Poly-encoders belong to the representation-based architecture which is also known as the Bi-encoder architecture. BERT, StructBERT and DistilBERT belong to interaction-based architecture which is also known as the Cross-encoder architecture. All baselines and our models are implemented by Tensorflow with version of 1.12.

DistilBERT$_{\text{StructBERT}}$ is an implementation of the well-known distilled BERT model [21] which leverages knowledge distillation during the pre-training phase. The DistilBERT could reduce the size of a BERT model by 40%, while retaining 97% of its language understanding capabilities. The DistilBERT is also interaction-based architecture but with less layers in order to reduce computation. In our experiments we use StructBERT as the teacher of DistilBERT which refers to **DistilBERT**$_{\text{StructBERT}}$ and we set the total layers to 6. The implementation follows Sanh et al. [21]. **Siamese BERT** is a Siamese architecture that uses pre-trained BERT to separately produce 768-dimensional query and product embeddings (from the [CLS] token). The query and the product title are encoded independently using a shared BERT encoder. Afterwards the final relevance score is computed by the cosine distance of the query embedding and the title embedding. We initialize the BERT encoder of Siamese BERT from StructBERT pre-trained weights. **MASM** [30] is the state-of-the-art e-commerce relevance model, which is representation-based and is not pre-trained language model like BERT. The model is trained from the weak supervision of click-through data, by a new designed training objective on an elaborately constructed dataset with different relevance confidence of samples. We use the best **MASM+LWR+Finetune** model as described in [30] but we denote it as MASM in this paper for simplicity. The model is also finetuned using the data in Table 1. This model was the online deployed model and has served the entire Taobao search traffic for over a year. **Poly-encoders** [7] is another state-of-the-art framework to improve BERT online efficiency. It has an additional learnt attention mechanism that represents more global features from which to perform self-attention, resulting in performance gains over Bi-encoders and large speed gains over Cross-encoders with the representation-based architecture. The number of the context codes is set to 64 and the embedding dimension is 768 (from the [CLS] token). Similarly, we also apply the KD technique to the Poly-encoders which refers to **Poly-encoders**$_{\text{StructBERT}}$ with the same distill strategy as ReprBERT. **StructBERT** [27] is the best teacher model we explored. By pre-training on general domain language data and continue-training on e-commerce data of extremely

large amount, together with finetuning on the task-specific annotation data, StructBERT achieves new state-of-the-art performance on the e-commerce relevance task. Unfortunately it is impractical to be deployed online due to heavy interaction computation. It can be viewed as the upper-bound for ReprBERT.

4.3.2 Evaluation metrics. We use both offline and online metrics to evaluate our model. In offline evaluation, since the human annotation is binary, the task is evaluated as a classification task. The Area Under Curve (AUC) is used as the evaluation metric, which is widely adopted in industrial tasks [8, 30]. Receiver Operator Characteristic (ROC) curve is most commonly used for measuring the results of binary decision problems. Meanwhile, considering actual relevance datasets are usually highly skewed, we also use the Precision-Recall curve for evaluation, which is more informative in this scenario [2]. Note that in the e-commerce relevance task, most instances are positive and we are more concerned about negative instances. Therefore the PR-AUC used in this paper is the negative PR-AUC that treats Bad as 1 and Good as 0 following Yao et al. [30]. The two metrics are denoted as ROC-AUC and Neg PR-AUC. Besides performance, we also evaluate the different model complexity of parameters and online computation efficiency. The FLOPs / token is computed according to Molchanov et al. [13] which shows the floating-point operations per second (FLOPs) when there is only 1 token being considered. The "+" sign separates the online and offline calculation FLOPs, which means the former part of computation can be pre-computed offline. The Memory metric indicates the online memory overhead for storing pre-computed query and product vectors where we use vector size for comparison. In online evaluation, we use the rate of *Good* annotated by human annotators and the number of transactions as the evaluation metrics. The query-product pairs for human relevance judgment are randomly sampled from the online search logs according to the amount of Page View (PV) as the sample weight.

4.4 Results

Table 2 presents the comparison of different models. The first block is the state-of-the-art models for the e-commerce relevance task. BERT shows its effectiveness as a powerful interaction-based model and it can outperform state-of-the-art representation-based models Siamese BERT, Poly-encoders and MASM. Even initialized from the weights of StructBERT, Siamese BERT that employs BERT as separate encoders in a representation-based architecture has significant performance decline compared to BERT. It demonstrates the challenge to realize a representation-based BERT model. StructBERT has considerable improvement over BERT, indicating the continue-training with the in-domain data is very effective to improve the model performance.

The second block presents the models distilled from StructBERT. From the improvement of Siamese $\text{BERT}_{\text{StructBERT}}$ and $\text{Poly-encoders}_{\text{StructBERT}}$ compared with their original models, we can see knowledge distillation is an effective way to improve model performance. However, the representation-based models Siamese $\text{BERT}_{\text{StructBERT}}$ and $\text{Poly-encoders}_{\text{StructBERT}}$ still have large gap from the teacher StructBERT model. It further verifies direct distillation across different structures will cause serious performance degradation. For comparison, $\text{DistilBERT}_{\text{StructBERT}}$ is an interaction architecture same as the StructBERT, which demonstrates much better performance than Siamese $\text{BERT}_{\text{StructBERT}}$ and $\text{Poly-encoders}_{\text{StructBERT}}$. However, all the computation of DistilBERT has to do at run-time so it is still hard to be deployed online. Finally, our proposed ReprBERT achieves the best distillation performance, outperforming all representation-based models and even the interaction-based $\text{DistilBERT}_{\text{StructBERT}}$. As a result, ReprBERT achieves more than 10% AUC improvement over previous state-of-the-art e-commerce relevance model MASM [30].

4.5 Model Complexity

Table 2 also compares the parameters, computation and memory consumption of each model. It can be found ReprBERT has the least parameters since the BERT encoders are condensed to 2 layers. The FLOPs indicates computation efficiency, which affects the online latency. MASM has the least FLOPs, as it is not a BERT-based model, and also online efficient as a representation-based model. BERT has the most FLOPs, and all the computation should be computed online. Siamese BERT and Poly-encoders are both online efficient, but their performance is far behind ReprBERT. Meanwhile, their memory needed for pre-computed vectors is 768, which is 3x of ReprBERT. In addition, to save the memory overhead and accelerate inference in online service, we use the float16 dtype instead of float32 dtype to store the pre-computed embeddings, which can reduce the memory required to a half, while almost no performance drop in AUCs.

To verify the online efficiency, we also compare the average inference time of ReprBERT and StructBERT on 1000 queries in Table 3. QEL refers to the number of Query Encoding Loops. The experiments are performed on a local CPU platform. We report the average inference time of the model to score 1000 products per query. The first row presents the inference time of the 12-layer StructBERT. As the embeddings of the query and products cannot be decoupled, the query also has to be encoded for 1000 times and the products' embeddings cannot be pre-computed offline. Therefore, the whole computation process needs to be done at the run-time, which makes BERT very time-consuming. For ReprBERT which has only 2 layers, the representation-based architecture allows to pre-compute all the embeddings of products. There are only the query-side encoder and interaction module to be computed at run-time. If we encode the query every time in the same way as StructBERT, there is still 1 order of magnitude faster. In fact there is no need to regenerate the query embedding at each run-time because the embedding of query is independent from the embedding of product. Therefore, without regenerating the query embeddings, i.e., the query encoding process is only computed once, the inference time is only about 0.1% of StructBERT.

4.6 Ablation Study

Table 4 presents the results of ablation experiments. We first compare the effect of layer numbers. *w/ 4 layers* indicates a deeper layer structure can bring slight performance improvement, but will significantly increase model parameters and computation cost. *w/o context-guided attention* directly uses the sentence embedding from the [CLS] token as the encoded representation instead of the context-guided-attention among all token embeddings we propose.

why is structBERT so high??

Model	ROC-AUC	Neg PR-AUC	Params	FLOPs / token	Memory
Siamese BERT	0.765	0.565	101.2M	91M+**1.5K**	768
MASM (Yao et al. [30])	0.793	0.582	76.8M	674K	640
Poly-encoders (Humeau et al. [7])	0.808	0.623	101.2M	182M+97.5K	768
$\text{BERT}_{\text{base}}$ (Devlin et al. [3])	0.850	0.662	101.2M	182M	0
StructBERT (Wang et al. [27])	**0.908**	**0.711**	101.2M	182M	0
$\text{DistilBERT}_{\text{StructBERT}}$ [21]	0.892	0.698	58.7M	91M	0
$\text{Siamese BERT}_{\text{StructBERT}}$	0.864	0.668	101.2M	91M+**1.5K**	768
$\text{Poly-encoders}_{\text{StructBERT}}$ (Humeau et al. [7])	0.870	0.680	101.2M	182M+97.5K	768
$\text{ReprBERT}_{\text{StructBERT}}$ (Ours)	0.894	0.702	**30.6M**	**30.4M**+297K	256

Table 2: Comparison results of different methods on the the test set. Best scores are in bold.

Model	QEL	Product scored	Inference time (ms)
StructBERT	1000	1000	321,408
ReprBERT	1000	1000	29,164
ReprBERT	1	1000	412

Table 3: Inference time of different models. QEL refers to the number of query encoding loops.

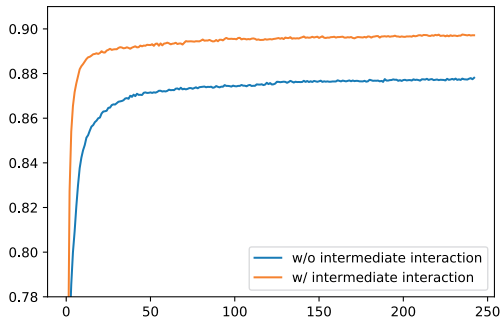

Figure 2: The effect of intermediate interaction. The vertical axis is the ROC-AUC score on the validation set and the horizontal axis is the training steps.

ReprBERT	ROC-AUC	FLOPs / token
w/ 2 layers	0.894	30.4M+297K
w/ 4 layers	0.896	60.9M+299K
w/o context-guided attention	0.890	30.3M+296K
w/o late interaction	0.850	30.2M+296K
w/o intermediate interaction	0.877	30.2M+296K

Table 4: Ablation study of replacing single component of ReprBERT. "w/ 2 layers" represents the proposed ReprBERT in this paper.

Model	ROC-AUC	Neg PR-AUC
$\text{StructBERT}_{\text{e-commerce}}$	0.908	0.711
$\text{StructBERT}_{\text{base}}$	0.885	0.693
ReprBERT	0.894	0.702
ReprBERT w/o FT	0.886	0.695

Table 5: The effect of extra training data on the test set. E-commerce refers to the constructed in-domain e-commerce data for pre-training of the teacher model. FT refers to the finetuning of ReprBERT on the human-annotated data.

It shows considering the context interaction will be more effective when producing representations. Finally we verify the effects of two interaction modules. The late interaction between the final representations shows more important than the interaction between the intermediate representations. Both interaction techniques can effectively improve the model performance with only small increase of computation cost. To further verify the effect of the intermediate interaction, we illustrate the learning process of ReprBERT with and without the intermediate interaction in Figure 2. We can see that with the intermediate interaction, ReprBERT converges faster and can achieve better results, resulting in better training efficiency and model performance.

Moreover, in Table 5 we investigate the effect of extra training data, including the large-scale in-domain e-commerce data for the continue training of the StructBERT (teacher model), and the human-annotation dataset for the finetuning of the ReprBERT (denotes as FT). The $\text{StructBERT}_{\text{base}}$ refers to that the model is only pre-trained on the general corpus, while the $\text{StructBERT}_{\text{e-commerce}}$ denotes the model is continue-trained on the e-commerce data. We can see that the $\text{StructBERT}_{\text{e-commerce}}$ has better performance than $\text{StructBERT}_{\text{base}}$. It demonstrates continue-training on the task-related data is helpful for improving model performance. In addition, we also investigate the effect of finetuning on the human-annotated data using the loss function as Equation 7. ReprBERT w/o FT refers to the ReprBERT that is not finetuned on the human-annotated dataset. It can be seen the finetuning process is also helpful for the model to get better results.

4.7 Online Evaluation

Online A/B testing is also conducted to evaluate ReprBERT, by replacing the previous MASM [30] model with ReprBERT, and all other factors are the same. Both experiments take about 2% proportion of Taobao search traffic, and the A/B testing lasts for two weeks. As a result, ReprBERT improves the number of transactions by about 0.6% on average. The daily human annotation results show

that ReprBERT also improves the rate of relevance by 0.5%. Online A/B testing verifies the proposed ReprBERT is superior to previous state-of-the-art models, and can achieve significant online profit considering the extremely large traffic of Taobao every day.

ReprBERT has already served the entire Taobao search traffic. After pre-computing the representations of queries and products, the online serving latency can be optimized to as low as 10ms on the distributed computing system with CPUs. This is close to the previous representation-based serving model MASM [30] and can satisfy the requirement for the extremely large traffic of Taobao.

5 CONCLUSION AND FUTURE WORK

In this paper, we study an industrial task of measuring the semantic relevance for queries and products in e-commerce. We propose ReprBERT, which introduces knowledge distillation from the original BERT to a representation-based BERT model. By proposing a context-guided attention and two interaction techniques, ReprBERT achieves close performance to the original BERT model. As a result, we make it possible to deploy the BERT model to online service of extremely large traffic with low latency requirement. ReprBERT achieves promising results on both offline and online experiments, and has been deployed to serve the entire Taobao search traffic for over a year.

In the future work, we will focus on building a more powerful teacher model, and will explore more sophisticated knowledge distillation techniques to further improve the model performance.

REFERENCES

[1] David Carmel, Elad Haramaty, Arnon Lazerson, Liane Lewin-Eytan, and Yoelle Maarek. 2020. Why Do People Buy Seemingly Irrelevant Items in Voice Product Search?: On the Relation between Product Relevance and Customer Satisfaction in eCommerce. In *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*. ACM, 79–87. https://doi.org/10.1145/3336191.3371780

[2] Jesse Davis and Mark Goadrich. 2006. The Relationship between Precision-Recall and ROC Curves. In *Proceedings of the 23rd International Conference on Machine Learning* (Pittsburgh, Pennsylvania, USA) *(ICML '06)*. 233–240. https://doi.org/10.1145/1143844.1143874

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

[4] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. arXiv:1503.02531 [stat.ML]

[5] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional Neural Network Architectures for Matching Natural Language Sentences. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2* (Montreal, Canada) *(NIPS'14)*. 2042–2050.

[6] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry P. Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013*. ACM, 2333–2338. https://doi.org/10.1145/2505515.2505665

[7] Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Transformer Architectures and Pre-training Strategies for Fast and Accurate Multi-sentence Scoring. arXiv:1905.01969 [cs.CL]

[8] Yunjiang Jiang, Yue Shang, Rui Li, Wen-Yun Yang, Guoyu Tang, Chaoyi Ma, Yun Xiao, and Eric Zhao. 2019. A Unified Neural Network Approach to E-Commerce Relevance Learning. In *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data (DLP-KDD '19)*. Article 10, 7 pages. https://doi.org/10.1145/3326937.3341259

[9] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. TinyBERT: Distilling BERT for Natural Language Understanding. arXiv:1909.10351 [cs.CL]

[10] Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, China) *(SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 39–48. https://doi.org/10.1145/3397271.3401075

[11] Ziyang Liu, Zhaomeng Cheng, Yunjiang Jiang, Yue Shang, Wei Xiong, Sulong Xu, Bo Long, and Di Jin. 2021. Heterogeneous Network Embedding for Deep Semantic Relevance Match in E-commerce Search. *arXiv preprint arXiv:2101.04850* (2021).

[12] Wenhao Lu, Jian Jiao, and Ruofei Zhang. 2020. TwinBERT: Distilling Knowledge to Twin-Structured BERT Models for Efficient Retrieval. arXiv:2002.06275 [cs.IR]

[13] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. 2017. Pruning Convolutional Neural Networks for Resource Efficient Inference. In *5th International Conference on Learning Representations*.

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

[15] Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-Stage Document Ranking with BERT. arXiv:1910.14424 [cs.IR]

[16] H. Palangi, L. Deng, Y. Shen, J. Gao, X. He, J. Chen, X. Song, and R. Ward. 2014. Semantic Modelling with Long-Short-Term Memory for Information Retrieval. arXiv:1412.6629 [cs.IR]

[17] Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab Ward. 2016. Deep Sentence Embedding Using Long Short-Term Memory Networks: Analysis and Application to Information Retrieval. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.* 24, 4 (April 2016), 694–707. https://doi.org/10.1109/TASLP.2016.2520371

[18] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text Matching as Image Recognition. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence* (Phoenix, Arizona) *(AAAI'16)*. AAAI Press, 2793–2799.

[19] Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A Decomposable Attention Model for Natural Language Inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, 2249–2255. https://doi.org/10.18653/v1/D16-1244

[20] Stephen Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. 1995. Okapi at TREC-3. In *Overview of the Third Text REtrieval Conference (TREC-3)*. Gaithersburg, MD: NIST, 109–126.

[21] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv:1910.01108 [cs.CL]

[22] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. 2014. A Latent Semantic Model with Convolutional-Pooling Structure for Information Retrieval. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. 101–110. https://doi.org/10.1145/2661829.2661935

[23] Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient Knowledge Distillation for BERT Model Compression. *arXiv preprint arXiv:1908.09355* (2019).

[24] Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. 2019. Distilling Task-Specific Knowledge from BERT into Simple Neural Networks. arXiv:1903.12136 [cs.CL]

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

[26] Shengxian Wan, Yanyan Lan, Jun Xu, Jiafeng Guo, Liang Pang, and Xueqi Cheng. 2016. Match-SRNN: Modeling the Recursive Matching Structure with Spatial RNN. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. AAAI Press, 2922–2928.

[27] Wei Wang, Bin Bi, Ming Yan, Chen Wu, Jiangnan Xia, Zuyi Bao, Liwei Peng, and Luo Si. 2020. StructBERT: Incorporating Language Structures into Pre-training for Deep Language Understanding. In *International Conference on Learning Representations*.

[28] Rong Xiao, Jianhui Ji, Baoliang Cui, Haihong Tang, Wenwu Ou, Yanghua Xiao, Jiwei Tan, and Xuan Ju. 2019. Weakly Supervised Co-Training of Query Rewriting AndSemantic Matching for e-Commerce. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining (WSDM '19)*. 402–410. https://doi.org/10.1145/3289600.3291039

[29] Chen Xu, Quan Li, Junfeng Ge, Jinyang Gao, Xiaoyong Yang, Changhua Pei, Fei Sun, Jian Wu, Hanxiao Sun, and Wenwu Ou. 2020. Privileged Features Distillation at Taobao Recommendations. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery Data Mining* (Virtual Event, CA, USA) *(KDD '20)*. Association for Computing Machinery, New York, NY, USA, 2590–2598. https://doi.org/10.1145/3394486.3403309

[30] Shaowei Yao, Jiwei Tan, Xi Chen, Keping Yang, Rong Xiao, Hongbo Deng, and Xiaojun Wan. 2021. Learning a Product Relevance Model from Click-Through Data in E-Commerce. In *Proceedings of the Web Conference 2021* (Ljubljana, Slovenia) *(WWW '21)*. Association for Computing Machinery, New York, NY, USA, 2890–2899. https://doi.org/10.1145/3442381.3450129

[31] Hongchun Zhang, Tianyi Wang, Xiaonan Meng, and Yi Hu. 2019. Improving Semantic Matching via Multi-Task Learning in E-Commerce. In *eCOM@SIGIR*.