SPM: Structured Pretraining and Matching Architectures for Relevance Modeling in Meituan Search

Wen Zan
zanwencs@gmail.com
Meituan Inc.
Beijing, China

Yaopeng Han
hanyaopeng95@163.com
Meituan Inc.
Beijing, China

Xiaotian Jiang*
jiangxiaotian.work@gmail.com
Meituan Inc.
Beijing, China

Yao Xiao
xiaoyao06@meituan.com
Meituan Inc.
Beijing, China

Yang Yang
yangyang113@meituan.com
Meituan Inc.
Beijing, China

Dayao Chen
chendayao@meituan.com
Meituan Inc.
Beijing, China

Sheng Chen
chensheng19@meituan.com
Meituan Inc.
Beijing, China

ABSTRACT

In e-commerce search, relevance between query and documents is an essential requirement for satisfying user experience. Different from traditional e-commerce platforms that offer products, users search on life service platforms such as Meituan mainly for product providers, which usually have abundant structured information, e.g. name, address, category, thousands of products. Modeling search relevance with these rich structured contents is challenging due to the following issues: (1) there is language distribution discrepancy among different fields of structured document, making it difficult to directly adopt off-the-shelf pretrained language model based methods like BERT. (2) different fields usually have different importance and their length vary greatly, making it difficult to extract document information helpful for relevance matching.

To tackle these issues, in this paper we propose a novel two-stage pretraining and matching architecture for relevance matching with rich structured documents. At pretraining stage, we propose an effective pretraining method that employs both query and multiple fields of document as inputs, including an effective information compression method for lengthy fields. At relevance matching stage, a novel matching method is proposed by leveraging domain knowledge in search query to generate more effective document representations for relevance scoring. Extensive offline experiments and online A/B tests on millions of users verify that the proposed architectures effectively improve the performance of relevance modeling. The model has already been deployed online, serving the search traffic of Meituan for over a year.

*Corresponding Author.

CCS CONCEPTS

• **Information systems** → **Language models**; **Similarity measures**.

KEYWORDS

search relevance; pretraining language model

1 INTRODUCTION

Unlike traditional e-commerce services that mainly provide products [7, 15], most users on life service e-commerce platforms , like Meituan [1], would like to search for service providers instead, such as restaurants, bars, stores, hotels, etc. These two forms of documents differ in that a service provider usually contains a great number of structured information therein. For example, a restaurant may have structured fields like name, address, category, comments, tags, and services like dishes and coupons. Considering all theses different types of structured contents in relevance matching is a challenging task.

To model the semantic relevance between query and document, previous methods tend to use a two-stage training paradigm [8]. At the first stage, a relevance model is first pretraining on a large-scaled dataset with self-supervised tasks. Afterwards, the model is further trained on a downstream dataset for task-specific finetuning.

However, previous e-commerce relevance models are less effective for documents with rich structured contents [22].

Specifically, at pretraining stage, existing relevance methods are normally designed for product search, and each document has only limited fields [24]. However, in life service scenario, there are

[1] https://meituan.com/

various fields in each document, and each field may have its own structure. Under this circumstance, it is impractical to directly input all the contents in a document for pretraining model, let alone the distributional discrepancy among different fields.

At finetuning stage, the models used in relevance task can be categorized into two architectures: cross-encoder and bi-encoder. Cross-encoder models better capture the interactions between query and document, and thus offer better performance. In e-commerce search scenario, bi-encoder models are widely used, since this architecture can cache the representations of query and document, and thus reduce calculations at online inference stage [12]. However, bi-encoder methods usually get lower performance compared to cross-encoder, due to the lack of interactions between query and document.

In this paper, we propose novel pretraining and matching architectures for relevance modeling in e-commerce search, especially for documents with rich structured contents. First, based on the characteristics of relevance task and document structure, we propose an effective pretraining method that employs both query text and multiple fields of document as inputs. Specifically, this method first extract basic fields of a document, such as name, brand, category, address, etc. Then, for lengthy fields like coupons or dishes, a field compression method is designed to effectively extract the pivotal information therein. Considering the vocabulary gap between query text and document, we also add document-related query to make the pretrained model better understand query.

After that, we design different mask strategies for each field to better adopt the word distribution of their own. Experiment shows that the proposed pretraining method brings better performance on relevance scoring with cross-encoder models, proving the effectiveness of modeling the rich structured information in documents. However, when we use bi-encoder model instead, the performance was not improved, and even worse than its original version that use query and name field only. It means that the document representation obtained by this model cannot effectively capture the rich information contained in the document. To solve this problem, in this paper we use query intent signals in search engine and design information extractors to generate better document representation. This method enables the query and document information to interact earlier, i.e, in the process of generating document representation. Therefore, it can effectively extract the information most beneficial for relevance matching.

To summarize, the contributions of this paper are as follows:

- We propose an effective multi-field pretraining architecture for e-commerce search with rich structured documents, and design a field compression algorithm to extract pivotal information from lengthy fields.
- We propose a novel matching architecture for structured documents with effective document extractors to generate more informative document representations for relevance scoring, utilizing domain knowledge in search engines.
- We conduct extensive experiments to evaluate the effectiveness of the proposed methods. Results on offline evaluation and online A/B test show that the methods can significantly improve the performance of relevance scoring in e-commerce search.

2 RELATED WORKS

2.1 Pretraining Models

In recent years, pretrained models like BERT [2], RoBERTa [11], XL-Net [21] have brought great performance improvements on many NLP tasks. These models are first pretrained on large-scale unsupervised data, and then fine-tuned on task-specific data for downstream tasks. BERT is a representative work among them. It obtains text representations by using transformer [18] architecture and training on two tasks - MLM (Mask Language Model) and NSP (Next Sentence Prediction). Recently, [11, 21] prove that the NSP task is not indispensable.

Since pretrained models usually do not model knowledge of specific downstream tasks, many works have begun to study how to encode domain knowledge into these models. Gururangan *et al.* [3] pointed out that continuing pretraining with domain data can gain performance improvements in domain-specific downstream tasks. Zou *et al.* [25] proposed a pretraining paradigm for relevance modeling in web search. Zhang *et al.* [23] proposed a pretraining work incorporating multi-modal information in e-commerce scenarios. Zhang *et al.* [24] proposed a e-commerce pretrained model with phrase mask and document neighbor for augmentation. Despite the success these models made in their domains, there is still lack of pretraining methods designed for documents with rich structured contents, which is very common in life service e-commerce scenario.

2.2 Text Matching

Text matching (or text relevance) task computes the similarity between two sentences. In the recent decade, neural methods have been flourishing for their better capability to model semantic similarity. Neural text matching methods can be categorized into two types: cross-encoder methods [1, 13, 16, 20], and bi-encoder methods [4, 5, 14, 19]. They differs in that the former type concatenates two sentences together before feeding them into the model to calculate similarity, while the latter one first gets representations for each of the two sentence before calculating similarity score between them. The advantage of cross-encoder methods lies in that information of the two sentences can be well interacted, so these models normally have better performance. The advantage of bi-encoder models lies in that the trained sentence representations can be offline cached, which greatly speeds up online inference of the model. Recently, BERT has shown its superiority on text matching task. To increase interactions between sentences for bi-encoder BERT methods, Reimers *et al.* [17] proposed Siamese BERT networks that use a shared transformer for both of the sentences. Khattab *et al.* [9] proposed ColBERT that uses a sum-of-max operator to facilitate interaction between representations of the two sentences. Humeau *et al.* [6] use additional parameters to extract multiple representations of the longer sentence. However, these methods treat two sentences as plain text, while in industrial search engines both side of the match have rich additional knowledge to be used.

3 METHODOLOGY

In this section, we demonstrate the details of the proposed methods. First, we introduce the pretraining model designed for structured
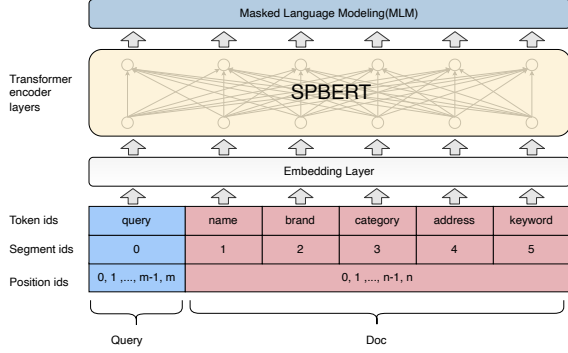
Figure 1: Structured Pretraining BERT (SPBERT) model.

Figure 2: Matching framework

document (**S**tructured **P**retraining BERT, i.e.,SPBERT), including two parts: data construction and pretraining strategy. Then we introduce the matching architecture for structured information, and propose two information extractors to better match query and structured documents in e-commerce scenario.

3.1 Pretraining Model for Structured Document

We formally define the i_{th} document with structured information as $D_i = \{\mathbb{W}_i^{F_1}, \mathbb{W}_i^{F_2}, \cdots, \mathbb{W}_i^{F_K}\}$, where K denotes the number of input fields. Let $\mathbb{W}_i^{F_k} = \{w_{i1}^{F_k}, w_{i2}^{F_k}, \cdots, w_{i|F_k|}^{F_k}\}$ denotes all tokens in the k_{th} field in D_i, $w_{ij}^{F_k}$ denotes the j_{th} token in $\mathbb{W}_i^{F_k}$, and $|F_k|$ denotes the number of tokens in $\mathbb{W}_i^{F_k}$.

Considering the fact that fields (such as name, brand, address, category, products, etc.) of a document in e-commerce scenario are normally not independent of each other, we define the following objective function for pretraining:

$$\max_{\theta} \sum_{i=1}^{|D|} \sum_{k=1}^{K} \sum_{j=1}^{|F_k|} m_{ij}^{F_k} \log p_{\theta}(w_{ij}^{F_k} | \mathbb{W}_i^{F_1}, \cdots, \mathbb{W}_i^{F_{k-1}}, \cdots, \mathbb{W}_i^{F_{k+1}}, \cdots, \mathbb{W}_i^{F_K},$$
$$w_{i1}^{F_k}, \cdots, w_{i(j-1)}^{F_k}, w_{i(j+1)}^{F_k}, \cdots, w_{i|F_k|}^{F_k}) \quad (1)$$

where $m_{ij}^{F_k} \geq 0$ denotes how many times token $w_{ij}^{F_k}$ is sampled to be masked for MLM[2] pretraining task.

The model structure is illustrated in Figure 1. In order to distinguish different fields which are concatenated together as input, we use a separate segment embedding for each of them. Besides, we use position embedding starting from 0 for query and document respectively, so that the words in the same position of query and name will get the same position embedding.

In this section, we introduce the proposed matching framework with structured information extractors.

3.1.1 Matching Framework.
As depicted in Figure 2, the framework uses a bi-encoder architecture to encode tokens of query and document respectively and obtains their representations. The advantage of bi-encoder architecture is that the output embeddings of the document can be calculated and cached in advance to speed up online serving [22]. A representation extraction layer is designed to extract the most important information beneficial for matching from the sophisticated structured input. Details of the structured
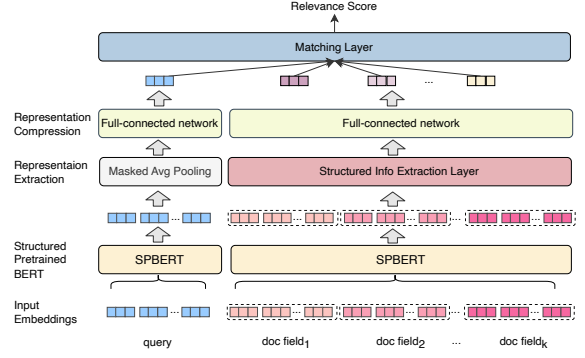
information extractors are elaborated in section 3.1.2. After that, a representation compression layer is adopted to reduce dimension of the extracted representations. Then, relevance score between query and doc can be calculated through the matching layer.

3.1.2 Intention-Guided Extractor(IGE).
In this section, we further propose an intention-guided information extractor, as depicted in Figure 3. It utilizes the intention signal of the query to be matched in advance, and extracts information according to the guidance of this knowledge. Specifically, we use the **q**uery **u**nderstanding (QU) module in search engine to provides query intention signal, and then extract the information in the structured document according to the signal.

Formally, a group of intention-guided extractors $h^{I_1}, \cdots, h^{I_M}$, $h^{I'_1}, \cdots, h^{I'_M}$ is used to extract query intention specific representation from document with K fields. Here, M denotes the total number of intention types, h^{I_i} and $h^{I'_i}$ respectively denotes the corresponding extractors that the i_{th} intention is hit or not, $h_i^{F_k}$ denotes the representation after SPBERT encoder of the i_{th} token from k_{th} field. The resultant representations of extractor h^{I_i} and $h^{I'_i}$ are denoted as $\hat{h}^{I_i}$ and $\hat{h}^{I'_i}$ respectively, which are defined as follows:

$$[\hat{h}^{I_i}, \hat{h}^{I'_i}] = attention([h^{I_i}, h^{I'_i}], [h_1^{F_1}, \cdots, h_{|F_K|}^{F_K}]) \quad (2)$$

Where $attention(X, Y) = softmax(XY^T)Y$. Additionally, we also use a group of T global extractors $[h^{G_1}, \cdots, h^{G_T}]$ to extract information extract global document representations $[\hat{h}^{G_1}, \cdots, \hat{h}^{G_T}]$ in the same way in Eq.2. The output of the intention-guided extractor layer is $[\hat{h}^{I_1}, \cdots, \hat{h}^{I_M}, \hat{h}^{I'_1}, \cdots, \hat{h}^{I'_M}, \hat{h}^{G_1}, \cdots, \hat{h}^{G_T}]$.

Then, a compression layer is utilized to obtain a memory-friendly version of representation for deployment. Specifically, we adopt a full-connected network to project the extracted representation from a high dimension to a much lower dimension. For simplicity, we use the same notations for representations before and after compression layer. Note that these are embeddings we will pre-calculate and cache when we deployment the model.

In matching layer, we first index out intention related representations from $[\hat{h}^{I_1}, \cdots, \hat{h}^{I_M}, \hat{h}^{I'_1}, \cdots, \hat{h}^{I'_M}]$ according to the corresponding intention signals, indicate each predefined intention is hit or not hit by the input query, obtained from query understanding
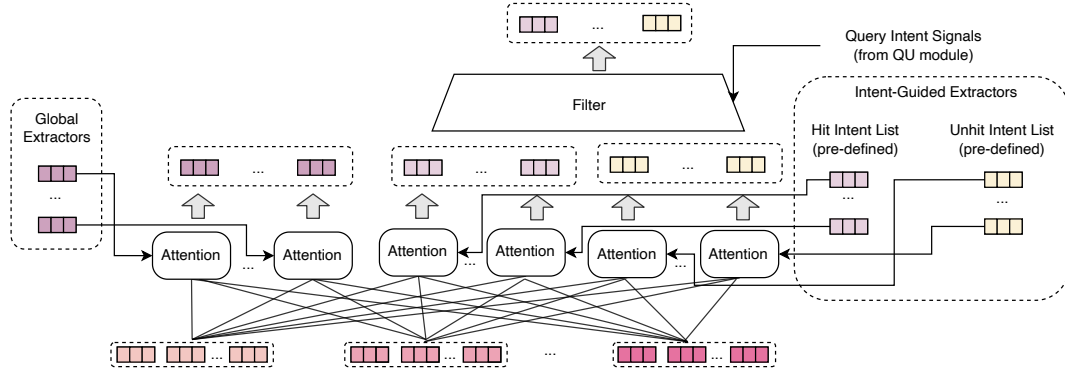
Figure 3: Illustration of the proposed intent-guided information extractor.

module in search engine. The selected intention related representations are noted as $[\hat{h}^{\tilde{I}_1}, \cdots, \hat{h}^{\tilde{I}_M}]$. Note that although we will cache $2*M+T$ embeddings for each document, only $M+T$ embeddings are involved during matching. By aggregating these representations using attention, we get the final document representation:

$$[\bar{h}^d] = attention([h^q], [\hat{h}^{\tilde{I}_1}, \cdots, \hat{h}^{\tilde{I}_M}, \hat{h}^{G_1}, \cdots, \hat{h}^{G_T}]) \qquad (3)$$

Where h^q is the query representation after the compression layer.

At last, cosine similarity is used to calculate the relevance score between query representation and document representation:

$$s = cosine(h^q, \bar{h}^d) \qquad (4)$$

We use the following loss function to characterize relevance at a finer granularity, since the label is multi-leveled:

$$loss = max(0, (s-y)^2 - \epsilon) \qquad (5)$$

where $y \in [0, 1]$ denotes the normalized relevance label. ϵ is the margin hyper-parameter, which is set to 0.1 in our experiments.

4 EXPERIMENT

To demonstrate the effectiveness of the proposed method, we conduct extensive experiments on an industrial life service e-commerce search engine. Experimental results show that for e-commerce search with rich structured documents, the proposed pretraining and matching architectures can significantly improve the performance of relevance modeling.

4.1 Datasets

4.1.1 Pretraining Dataset. The dataset at this stage is mainly composed of structured documents from search engine results. Specifically, we sampled 50 million documents from Meituan search, a large-scale real-world life service e-commerce search engine, to build the training corpus. These documents include fields like name, brand, category, address, keywords. In addition, for each frequency-visited document, we sample a top query and add it as a new field to the training corpus to better adapt to the relevance task.

4.1.2 Relevance Dataset. The relevance dataset consists of more than 1 million human-labeled query-document pairs from historical search results from Meituan Search. With the help of Meituan's crowd-sourcing platform, each of these query-document pairs is annotated as one of the three relevance levels: strong relevant, weak relevant, not relevant. Specifically, there are 1,068,135 strong relevant samples, 169,454 weak relevant samples, and 249,695 irrelevant samples. The labels y used for the strong relevant, weak relevant, and irrelevant samples in Equation (5) are 1, 0.7, and 0, respectively. We sample 10,000 samples for devset and 10,000 for testset to evaluate models, and the rest 1,467,284 samples are used for training model.

4.2 Evaluation Metrics

We use the following evaluation metrics to evaluate the performance of the proposed method.

The Area Under Curve(**AUC**) is widely used as the evaluation metric in e-commerce scenarios[7, 22] for evaluating relevance model . If the relevance score given by the model is ideally all higher on the relevant samples than the irrelevant ones, AUC will get the maximum value of 1. Traditional AUC formula is only suitable for binary classification, but our relevance matching task has three relevance levels. So we employ a multiclass AUC formula:

$$AUC = \frac{1}{N} \sum_i^N \frac{\sum_j \sum_k \mathbb{1}(y_{ij} > y_{ik}) \cdot \mathbb{1}(s_{ij} > s_{ik})}{\sum_j \sum_k \mathbb{1}(y_{ij} > y_{ik})} \qquad (6)$$

where $y_{ij} \geq 0$ is the relevant label for the j_{th} returned document of the i_{th} query, $s_{ij} \in [0, 1]$ is the model score for the j_{th} returned document of the i_{th} query and N is the total number of test queries.

The **Badcase@5** is an important metric we used to evaluate the quality of the top search results of the online system. Specifically, it calculates the proportion of queries having irrelevant cases in the top 5 ranking results:

$$Badcase@5 = \frac{1}{N} \sum_i^N \mathbb{1}(\sum_j^5 y_{ij} = 0) \qquad (7)$$

4.3 Competitor System

To evaluate the effect of the proposed structured pretraining method on relevance task, we compare the following **C**ross-**E**ncoder methods. We use [+] to denote that the models use a query and multiple fields of a document as inputs. Otherwise, the inputs consist only of the query and the name field of the document.

- **BERT-Large-CE**: a cross-encoder matching model based on BERT-Large (24 layers) with query and document name as input.
- **BERT-Large-CE[+]**: the same model structure with BERT-Large-CE, but with query and structured document as input. Fields of the document are concatenated with [SEP], using the same segment embedding.
- **BERT-CE[+]**: a cross-encoder matching model based on a 6-layer BERT model distilled from BERT-Large, using query and structured document as input.
- **SPBERT-Large-CE[+]**: a cross-encoder matching model based on SPBERT-Large, the 24-layer structured pretraining model. Using query and structured document as input.
- **SPBERT-CE[+]**: a cross-encoder matching model based on a 6-layer SPBERT model distilled from SPBERT-Large, using query and structured document as input.

To improve model efficiency on online serving, relevance models in e-commerce scenario often adopt **Bi-Encoder** architecture. This architecture can cache representations of queries and documents in advance and thus reduce online calculation. We further evaluate the performance of the pretraining model by comparing it with methods of this architecture.

- **BERT-BE**: a bi-encoder matching model based on BERT with query and document name as input.
- **SPBERT-BE**: a bi-encoder matching model based on SP-BERT with query and document name as input.
- **SPBERT-BE[+]**: the same model structure with SPBERT-BE with query and structured document as input.

To evaluate the effectiveness of the proposed matching methods with structured extractors, we compare them with state-of-the-art late interaction methods for comparison.

- **SPBERT-SBERT[+]**: SBERT [17] architecture using SPBERT as pretrained model. As one of late interaction methods, SBERT concatenate input query embedding, document embedding, and simple mathematical operation result between them before maching. In this way, this method usually gets better results compared with other methods directly matching query embedding and document embedding.
- **SPBERT-ColBERT[+]**: ColBERT [9] architecture using SP-BERT as pretrained model. Since ColBERT model calculates the matching score of each token of a query to all tokens of a document, it tends to gain better performance than other bi-encoder methods. However, this method needs to cache the token representations of all documents beforehand, which is memory expensive and hard to deploy when documents have long length.
- **SPBERT-PolyEncoder[+]**: PolyEncoder [6] architecture using SPBERT as pretrained model. PolyEncoder uses multiple codes to interact with document to obtain effective document representations. By controlling the number of codes, this method can flexibly trade off between memory usage and model performance.
- **SPBERT-IGE[+]**: a matching method with the intent-guided extractor elaborated in section 3.1.2, using SPBERT as pretrained model. This method extracts the information of structured documents for all types of intent signals.

4.4 Experimental Setting

For all the experiments, we set the learning rate to 2e-5, warm-up ratio to 10%, dropout rate to 0.1, and use the Adam [10] optimizer.

In the pretraining stage, the model parameters are set to 24 hidden layers, 16 attention heads, hidden size of 1024, and feed-forward layers with dimension 4096; the batch size is set to 20 during training, and the model is distributedly trained on 64 NVIDIA V100 GPUs for 10 epochs.

In the distillation stage, we use the pretraining model as the teacher. The parameters of the student model are set to 6 hidden layers, 12 attention heads, a hidden size of 384, and feed-forward layers with dimension 1200. During training, the batch size is set to 8 and the model is distributedly trained on 32 NVIDIA V100 GPUs for 3 epochs.

In the finetuning stage for relevance task, we set the sequence length to 160 for cross-encoder models. As to the bi-encoder models, we set the maximum query length to 32, and the maximum document length to 128. Before the matching layer, we use a one-layer fully-connected network as a compression layer to project the dimension of embeddings from 384 to 32. For SPBERT-IGE[+], the number of intent-guided extractors is set to 3 corresponding to the number of query intent types we use, and we adjust the number of global extractors to meet the requirements on the total number of extractors in each experiment. The embeddings of all extractors are random initialized before training. ϵ in equation (5) is set to 0.01. Batch size is set to 256 during training. Early stop is performed when AUC does not improve within 5 epochs.

To the best of our knowledge, there is no public e-commerce relevance dataset that has highly structured documents as well as query intent signals. Therefore, we report the evaluation results of the proposed method on the dataset described in 4.1.2 for all offline experiment.

4.5 Offline Experimental Results

Table 1 reports the experimental results of the base model BERT-Large, the structured pretraining model SPBERT-Large, as well as the corresponding distilled models BERT and SPBERT, using all fields or name field as document input respectively. Comparing BERT-Large-CE with BERT-Large-CE[+], we can observe that adding structured inputs to unstructured pretraining models is harmful for their performance. This may be attributed to the distributional discrepancy among different fields of structured document, which can not be distinguished by unstructured pretraining models. On the contrary, the comparison between BERT-Large-CE and SPBERT-Large-CE[+] informs us that using structured input with structured pretraining model does not lower model performance, but instead improves AUC by as much as 1.8%. This result reveals that adding structured information is beneficial for relevance tasks. Moreover, this benefit can only be achieved by using structured pretraining. Finally, by comparing the distilled model BERT-CE and SPBERT-CE[+], it can be seen that the advantages of structured pretraining are still preserved after distillation.

To improve model efficiency, e-commerce relevance methods often use bi-encoder architecture to reduce real-time computation. Table 1 reports the performance of BERT and SPBERT of this form. First, comparing SPBERT-BE with BERT-BE, it can be observed that

SPBERT can still achieve better results even using unstructured input. This is because the term-weight-based masking strategy employed on name field at pretraining stage enables the words important to the matching task to be better learned. Note that by comparing SPBERT-BE with SPBERT-BE[+], we can see that the performance of bi-encoder model decreases when using structured input. This is due to the fact that structured document includes multiple fields with different importance to the relevance task, so it is difficult to extract enough effective information by just using pooling. This observation inspires us to design the more effective information extraction methods based on the characteristics of structured document.

Table 1: Performance of structured pretraining models on relevance task

Model	AUC(%)
BERT-Large-CE	87.88
BERT-Large-CE[+]	86.48
SPBERT-Large-CE[+]	89.60
BERT-CE	86.52
SPBERT-CE[+]	88.51
BERT-BE	82.19
SPBERT-BE	82.64
SPBERT-BE[+]	81.91

To prove the effectiveness of the proposed matching architectures, we compare them with state-of-the-art late interaction bi-encoders. In consideration of fairness, the number of codes in PolyEncoder and extractors in SPBERT-IGE[+] are all set to 8. As shown in Table 2, late interaction methods alleviate the problem of performance degradation after adding structured information in bi-encoder. Moreover, we observe that SPBERT-IGE[+] has better performance than SPBERT-SBERT[+] and SPBERT-PolyEncoder[+]. This indicates the proposed extractor can better extract informative representations for matching than those that extract information completely on the model itself. Note that SPBERT-IGE[+] has reached close performance to SPBERT-ColBERT[+], while it at the same time needs much fewer document representations that need to be cached than the latter, meaning that the proposed method is both effective and memory-friendly.

Table 2: Comparison of the proposed matching methods.

Model	AUC(%)	Doc Embedding Storage
SPBERT-BE	82.64	1x
SPBERT-BE[+]	81.91	1x
SPBERT-SBERT[+]	82.37	1x
SPBERT-ColBERT[+]	85.46	128x
SPBERT-PolyEncoder[+]	84.93	8x
SPBERT-IGE[+]	85.37	11x (8x is used during matching)

4.6 Online Experimental Results

4.6.1 Deployment. Meituan search faces tens of millions of queries every day, so the online system has strict restriction on latency.

Since BERT uses multiple layers of transformers, its efficiency decrease as the number of layers increases. For the sake of efficiency, we deploy the distilled BERT for online serving. As illustrated in Figure 4, to further lower the latency, we cache the extracted multiple embeddings of the documents offline. We also cache embeddings of the top queries offline, while calculate that of the long-tail queries online. The intent signal of queries are obtained from query understanding module in search engine. Then, the matching layer uses these results to calculate relevance score. Finally, this score is discretized into relevance level, and used by the search system for stratification, namely sorting the search results stably by the relevance level in descending order.
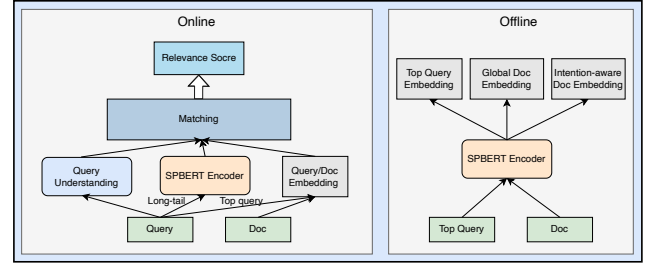

Figure 4: Deployment of the proposed relevance model

4.6.2 Online A/B Test. We conduct an online A/B test for one week to ensure that our new proposed model (i.e.,SPBERT-IGE[+]) will improve the system performance compared with the old one (i.e., BERT-BE). The results show that the new model can largely improve the overall user experience of the e-commerce search system. In particular, the Badcase@5 metric has decreased by 1.12% which is statistically significant with $p < 0.05$. This show that the structured pretrain and matching architectures with intent-guided extractors are very helpful for improving the relevance of the search system.

5 CONCLUSION

In this paper, we describe novel pretraining and matching architectures for relevance matching with rich structured documents. To make the method aware of the distributional discrepancy among fields and the different importance of each field to relevance task, we propose a structured pretraining architecture that uses multiple fields to pretrain a language model. To generate informative representations for relevance scoring, we propose a matching architecture with query intent guided document information extractors. Extensive offline and online experiments are conducted to verify the effectiveness of the proposed architectures and the corresponding model are successfully implemented and deployed on Meituan Search to boost search relevance.

REFERENCES

[1] Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 2018. Convolutional neural networks for soft-matching n-grams in ad-hoc search. In *WSDM*.
[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.
[3] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: adapt language models to domains and tasks. In *ACL*.

[4] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *NIPS*.

[5] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *CIKM*.

[6] 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. In *ICLR*.

[7] 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 *DLP-KDD*.

[8] Yunjiang Jiang, Yue Shang, Ziyang Liu, Hongwei Shen, Yun Xiao, Sulong Xu, Wei Xiong, Weipeng Yan, and Di Jin. 2020. Bert2dnn: Bert distillation with massive unlabeled data for online e-commerce search. In *ICDM*.

[9] Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *SIGIR*.

[10] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *arXiv preprint arXiv:1412.6980*.

[11] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. In *arXiv preprint arXiv:1907.11692*.

[12] Yiqun Liu, Kaushik Rangadurai, Yunzhong He, Siddarth Malreddy, Xunlong Gui, Xiaoyi Liu, and Fedor Borisyuk. 2021. Que2search: Fast and accurate query and document understanding for search at facebook. In *KDD*.

[13] Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2017. Learning to match using local and distributed representations of text for web search. In *WWW*.

[14] Paul Neculoiu, Maarten Versteegh, and Mihai Rotaru. 2016. Learning text similarity with siamese recurrent networks. In *Rep4NLP@ACL*.

[15] Priyanka Nigam, Yiwei Song, Vijai Mohan, Vihan Lakshman, Weitian Ding, Ankit Shingavi, Choon Hui Teo, Hao Gu, and Bing Yin. 2019. Semantic product search. In *KDD*.

[16] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text matching as image recognition. In *AAAI*.

[17] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP-IJCNLP*.

[18] 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 *NIPS*.

[19] Shengxian Wan, Yanyan Lan, Jiafeng Guo, Jun Xu, Liang Pang, and Xueqi Cheng. 2016. A deep architecture for semantic matching with multiple positional sentence representations. In *AAAI*.

[20] Yu Wu, Wei Wu, Chen Xing, Ming Zhou, and Zhoujun Li. 2017. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. In *ACL*.

[21] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NIPS*.

[22] 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 *WWW*.

[23] Boxuan Zhang, Chao Wei, Yan Jin, and Weiru Zhang. 2021. ACE-BERT: Adversarial Cross-modal Enhanced BERT for E-commerce Retrieval. In *arXiv preprint arXiv:2112.07209*.

[24] Denghui Zhang, Zixuan Yuan, Yanchi Liu, Zuohui Fu, Fuzhen Zhuang, Pengyang Wang, Haifeng Chen, and Hui Xiong. 2020. E-BERT: A phrase and product knowledge enhanced language model for e-commerce. In *arXiv preprint arXiv:2009.02835*.

[25] Lixin Zou, Shengqiang Zhang, Hengyi Cai, Dehong Ma, Suqi Cheng, Shuaiqiang Wang, Daiting Shi, Zhicong Cheng, and Dawei Yin. 2021. Pre-trained language model based ranking in Baidu search. In *KDD*.