Reproducibility Analysis and Enhancements for Multi-Aspect Dense Retriever with Aspect Learning

Keping Bi[1,2][0000−0001−5123−4999], Xiaojie Sun[1,2][0009−0006−4570−6359], Jiafeng Guo[1,2⋆][0000−0002−9509−8674], and Xueqi Cheng[1,2][0000−0002−5201−8195]

[1] CAS Key Lab of Network Data Science and Technology, ICT, CAS
[2] University of Chinese Academy of Sciences
{bikeping, sunxiaojie21s, guojiafeng, cxq}@ict.ac.cn

Abstract. Multi-aspect dense retrieval aims to incorporate aspect information (e.g., brand and category) into dual encoders to facilitate relevance matching. As an early and representative multi-aspect dense retriever, MADRAL learns several extra aspect embeddings and fuses the explicit aspects with an implicit aspect "OTHER" for final representation. MADRAL was evaluated on proprietary data and its code was not released, making it challenging to validate its effectiveness on other datasets. We failed to reproduce its effectiveness on the public MA-Amazon data, motivating us to probe the reasons and re-examine its components. We propose several component alternatives for comparisons, including replacing "OTHER" with "CLS" and representing aspects with the first several content tokens. Through extensive experiments, we confirm that learning "OTHER" from scratch in aspect fusion is harmful. In contrast, our proposed variants can greatly enhance the retrieval performance. Our research not only sheds light on the limitations of MADRAL but also provides valuable insights for future studies on more powerful multi-aspect dense retrieval models. Code will be released at: https://github.com/sunxiaojie99/Reproducibility-for-MADRAL.

Keywords: Multi-aspect Retrieval · Dense Retrieval · Aspect Learning.

1 Introduction

Standing on the shoulders of pre-trained language models (PLMs)[5,23], dense retrieval models have exhibited impressive performance in the first stage of information retrieval [6,7,15,11]. Most dense retrieval models concentrate on unstructured textual data, while much less attention has been paid to structured item retrieval such as product search and people search. These scenarios have a wide population of users and the aspect information like brand (e.g., "Apple") and affiliation (e.g., "Microsoft") can be pivotal to enhance relevance matching. Nonetheless, it remains largely unexplored how to effectively integrate these aspects within dense retrieval models.

⋆ Corresponding author

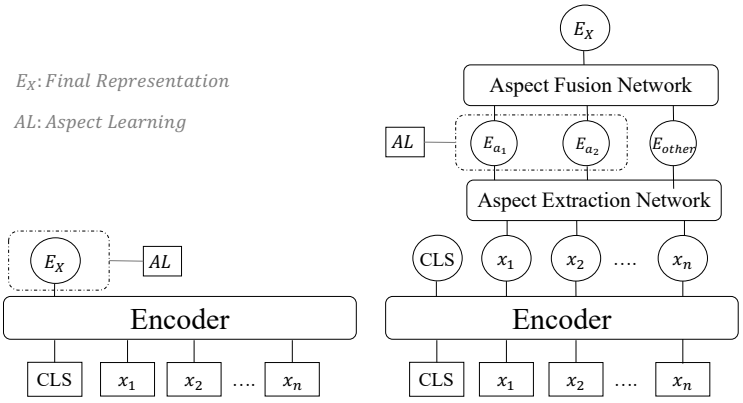

Fig. 1: Two multi-aspect dense retrieval models proposed by Kong et al. [10].

Recently, Kong et al. [10] initiated such a study by proposing a Multi-Aspect Dense Retriever with Aspect Learning, named MADRAL, and a simpler yet competitive baseline MTBERT. As illustrated in Figure 1, MADRAL has three major components, i.e., aspect extraction, aspect learning, and aspect fusion, to produce the final representation E_X. Specifically, this model employs an aspect extraction network to extract extra aspect embeddings alongside the initial BERT parameters and conducts aspect learning by predicting the value IDs of an aspect (e.g., the ID of "Beauty" in the vocabulary of the product category). Notably, a special aspect "OTHER" is included to capture the implicit semantics that the explicit aspects cannot cover. Then for relevance matching, these aspect embeddings are integrated using an aspect fusion network to produce the final query/item representation. In contrast, MTBERT only conducts aspect learning on the CLS token, which is also used for relevance matching. Both models significantly outperform the original BERT and MADRAL can achieve much more compelling performance. The framework of MADRAL is insightful for the research on multi-aspect dense retrieval.

Although claimed to be effective, MADRAL has been experimented on proprietary data (i.e., Google shopping) that is not accessible to the public. The code of MADRAL has not been released either, which makes it even harder to reproduce the experimental results in [10]. Since Google shopping data has aspect information of both queries and items, it is also unknown whether MADRAL will be effective on other datasets of different properties. We have tried to reproduce its performance on the public MA-Amazon data, which has large-scale real-world queries and multiple aspect information associated with the items, but surprisingly find that MADRAL [3] has significantly worse performance than

[3] The authors have not provided their code upon our request but verified our implementation of MADRAL.

its backbone BERT. This has motivated us to study why it does not work and how to enhance it to work effectively.

We speculate that there are two potential reasons for its unsatisfactory performance: 1) the brand-new embedding of aspect "OTHER" may not learn the implicit semantics well during fine-tuning; 2) it is challenging to learn extra aspect embeddings sufficiently from scratch during pre-training. To validate the reasons, we propose several alternative methods for aspect fusion and aspect representation. Specifically, instead of learning implicit semantics with a new token "OTHER", we propose to fuse "CLS", which is designated to capture global content semantics explicitly, in the final representation. For aspect representation, we reuse the first several content tokens to represent aspects whose embeddings only need to be adjusted with the aspect learning objectives. Extensive experiments show that both versions of enhancements can yield significantly better retrieval results when replacing the original counterparts of MADRAL, confirming the existence of the above issues. Our studies pave the way for future research on this topic that uses MADRAL as a benchmark and also provide valuable insights into the development of more powerful multi-aspect dense retrievers.

2 RELATED WORK

Dense Retrieval. Dense retrieval models typically adopt a bi-encoder architecture, which encodes a query and a document into two vectors and uses a similarity function like a dot product to measure their relevance. Karpukhin et al. [8] have explored pre-trained language models (PLMs) for information retrieval by using the BERT as the encoder and training it with in-batch negatives. This achieves superior performance compared to the models before the PLM era. Subsequently, researchers delved into various fine-tuning techniques to improve dense retrieval such as mining hard negatives [27,17], distilling the knowledge from cross encoders [24], and representing documents with multi-vector representations [14,9,29].

Most research efforts on dense retrieval have been spent on unstructured text until recently Kong et al. [10] proposed an effective method MADRAL that incorporates the structured aspect information of queries or items into the dense retrievers. This work leverages a typical way of injecting the aspect information [1,2] to the item representation, i.e., predicting the values associated with the aspects as an auxiliary training objective. Following this paradigm, Sun et al. [22] studied how to capture fine-grained semantic relations between different aspect values. As MADRAL [10] is the first multi-aspect dense retriever and adopts a typical manner of modeling the aspects, our reproducibility study on it will pave the way for future research that uses MADRAL as a benchmark in this direction.

Multi-Field Retrieval. It has been a longstanding research topic on how to effectively utilize multi-field information such as titles, keywords, and descriptions in documents. The earliest attempt can date back to BM25F [19]. More recently, Liu et al. [13] explored the incorporation of multi-field information into the relevance models. Prior to the advent of PLMs, researchers have investi-

gated leveraging the document fields in neural ranking models [3,4,28]. For example, Zamani et al. [28] proposed to aggregate field-level representations using a matching network and trained the model with field-level dropout. There has been ongoing research on the utilization of multi-field information [25,20] after PLMs have become the dominant retriever backbone. For instance, Shan et al. [20] proposed to leverage field-level cross interactions between queries and items as an auxiliary fine-tuning objective to improve retrieval performance. Sun et al. [21] treated item aspects as text strings and proposed a pre-training method to enhance the retriever.

Although the aspects can be simply treated as fields, multi-field retrievers only focus on the document side and cannot handle the case that query aspects are also available. Moreover, aspects and fields have some essential differences: fields are comprised of unstructured text that has infinite semantic space, whereas an aspect is defined by a finite set of values, serving as the aspect annotations. Consequently, multi-aspect and multi-field retrieval face distinct challenges. MADRAL [10] has been experimented on the data having aspects for both queries and items, and it was not compared to any baselines that treat aspects as fields. Since we use the public MA-Amazon dataset that only has item aspects, we also include a straightforward baseline that uses aspects as fields and concatenates the aspect texts, i.e., BIBERT-CONCAT in Section 7.

3 Preliminaries of Multi-Aspect Dense Retrievers

Task Definition. In multi-aspect dense retrieval, queries and candidate items can have multiple aspects such as brand, color, and category. Given a query q or item i, each of its associated aspects a has a finite vocabulary of value set, denoted as V_a, and an embedding lookup table $T_a \in \mathbb{R}^{|V_a| \times H}$, where each value ID maps to an H-dimensional vector. Suppose that the aspect set is A containing k aspects, i.e., $A = a_1, a_2, \cdots, a_k$, their corresponding annotated value sets are $\mathcal{A}_{a_1}, \mathcal{A}_{a_2}, \cdots, \mathcal{A}_{a_k}$. The content tokens of q or i (that can include titles and descriptions) are denoted as $X = x_1, x_2, \cdots, x_n$. A multi-aspect dense retrieval model aims to learn effective representations of q and i by incorporating the aspect information and capturing the content semantics so that their similarities can reflect their relevance.

Multi-aspect Dense Retrievers with Aspect Learning. As shown in Figure 1 and 2, typical multi-aspect dense retrievers [10] usually have three major components: 1) *Aspect Representation*, that either declares extra aspect embeddings (e.g., in MADRAL) or reuses the "CLS" token (e.g., in MTBERT) to capture the aspect information; 2) *Aspect Learning*, that injects the aspect-value information into the aspect representation by predicting its associated value IDs during pre-training (may also be beneficial in fine-tuning); 3) *Aspect Fusion* that merges the learned aspect representations into the final query/item representation for relevance matching during fine-tuning. In MTBERT, all the aspect learning is conducted on the "CLS" token, so no additional fusion is needed.

In the next two sections, we elaborate on the component variants of aspect representation and aspect fusion we study. Since queries and items have the same learning process, we only use items in the illustration for brevity.

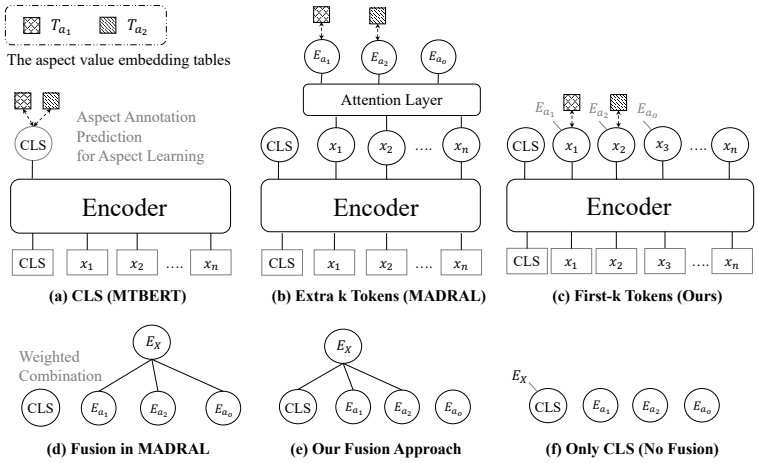

Fig. 2: The upper figures illustrate the aspect representation and learning of MTBERT, MADRAL, and our variant. The lower figures show aspect fusion methods to yield the final representation E_X. The weighted combination can be CLS gating or presence weighting. a_o denotes the special aspect "OTHER".

4 Component Variants: Aspect Representation

To effectively incorporate the aspect-value information into an item, reasonable aspect representation is pivotal. Figure 2(a), 2(b), and 2(c) show the three variants of aspect representations we compare.

Reusing CLS Token (MTBERT). MTBERT [10] reuses the "CLS" token to conduct aspect learning (see Figure 2(a)). It naturally injects aspect annotations of an item into the "CLS" token that also captures content semantics. The aspect information can be learned on top of "CLS", which can be a decent starting point. However, it compresses the information from multiple aspects and the content into a single token without weighting mechanisms. So, it does not differentiate the importance of each information source to relevance ranking, which could yield suboptimal retrieval results.

Declaring k Extra Embeddings (MADRAL). MADRAL [10] represents aspects with extra embeddings (Figure 2(b)), that are computed based on the attention over the encoded content tokens $Encoder(X) = Encoder(x_1, x_2, \cdots, x_n)$, where $Encoder$ can be any transformer-based encoders like BERT. The aspect embeddings E_A, stacked from $E_{a_1}, E_{a_2}, \cdots, E_{a_k}$, is computed as follows:

$$\begin{aligned} E_A &= Attention(QW^Q, Encoder(X)W^K, Encoder(X)W^V), \\ Attention(Q,K,V) &= softmax(\frac{QK^T}{\sqrt{H}})V. \end{aligned} \tag{1}$$

Attention is the multi-head attention function involving Q, K, V in the standard transformer [26]. Q is the set of aspect embeddings in this case. In this way, each aspect has its own representation and can be dedicated to its own learning. The influence of each aspect on the final representation can be automatically learned.

However, in MADRAL, these new parameters are only learned from the aspect learning objectives introduced in Section 6, which could be challenging to learn them well from scratch, especially when there are not many aspect annotations.

Reusing First-k Content Tokens. Instead of training extra k aspect embeddings from scratch, we propose an alternative approach that reuses the encoder output of the first k content tokens, i.e., $x_1, x_2, \cdots, x_k$, to represent the k-associated aspects (shown in Figure 2(c)). In MADRAL, the embeddings of content tokens are loaded from a pre-trained BERT and also updated by the masked language model (MLM) loss when ingesting the local corpus. Hence, they are learned more sufficiently and can serve as better starting points than brand-new extra tokens. The tokens at the beginning of the content are usually important to represent the content semantics. Guiding these tokens with the aspect learning loss is a way that not only binds the aspect information to content tokens like MTBERT does but also differentiates the influence of each aspect on the final representation like MADRAL does. So, it can leverage the advantages from both perspectives.

5 Component Variants: Aspect Fusion

During relevance matching, the aspect embeddings are fused into a single item embedding E_X, i.e.,

$$E_X = \sum_{a \in A} w_a E_a. \tag{2}$$

MTBERT does not have the phase of aspect fusion and uses the CLS token directly for relevance matching. MADRAL [10] has three fusion networks: weighted sum, CLS-gating, and presence weighting. Since the first one does not perform well [10], we only study the latter two. We will elaborate from two perspectives: the weighting mechanism and the objects to fuse.

5.1 Weighting Mechanism

CLS Gating. The encoded embedding of "CLS" is projected to $|A|$ (i.e., the number of aspects) logits, with a linear layer: $Linear(E_{CLS}) \in \mathbb{R}^{|A|}$, and the softmax weight computed for each logit is used as the final weight. In other words, taking one logit $\gamma_a \in Linear(E_{CLS})$ for example, $w_a = \frac{e^{\gamma_a}}{\sum_{a' \in A} e^{\gamma_{a'}}}$.

Presence Weighting. This mechanism computes the weight of each aspect according to its presence probability in the item, i.e., $w_a = P(I_a)\gamma_a$, where I_a indicates that item I has annotated values for a, $P(I_a) = Sigmoid(E_a)$, and γ_a is a learned parameter. $P(I_a)$ is learned with a cross-entropy loss based on whether an aspect has associated values in an item, which will be described in Section 6.

5.2 Objects to Fuse

The objects to fuse into the final item representation have a huge impact on retrieval performance, which we will show in the experiments. For both CLS-Gating and presence weighting, besides the original objects MADRAL fuses, we

propose an alternative approach for fusion that slightly revises the fusion objects and can greatly enhance the performance. We introduce both ways as follows:

Aspect and Implicit Token ("OTHER"). In MADRAL, for both CLS-Gating and Presence Weighting, besides the standard aspects like brand and color, it adds a special aspect "OTHER" to capture the important information that may not be included in the explicit aspects. No aspect learning is conducted on this special aspect and it is supposed to learn implicit semantics automatically. If we denote the special aspect "OTHER" as a_o, the k elements in the set A in Equation (2) becomes:

$$A = \{a_1, a_2, \cdots, a_{k-1}, a_o\}. \tag{3}$$

The fusion weights and the embedding of a_o need to be learned during fine-tuning. When there is not sufficient training data with relevance labels to fine-tune the retriever, it could be difficult for the model to learn them well, especially a_o, which inevitably harms the retrieval performance.

Aspect and Explicit Token ("CLS"). We believe that an effective item representation should capture both content semantics and the aspect information of the item. Rather than fusing with the embedding of "OTHER" that learns implicit information, we use the "CLS" token that captures the global item semantics explicitly as a pseudo aspect in the fusion. In other words, the set A in Equation (2) becomes:

$$A = \{a_1, a_2, \cdots, a_{k-1}, CLS\}. \tag{4}$$

Then, only fusion weights need to be learned during fine-tuning and the objective is clear: balancing the effects of content and aspects on the final representation for relevance matching.

Only CLS (No Aspect Fusion). Though we have pre-trained the aspect embeddings, during relevance matching (or fine-tuning), we do not fuse these aspect embeddings but instead use the embedding of "CLS" as the item embedding. In this way, the aspect learning process conducted on the extra k embeddings can be considered as purely multi-task learning that could guide the underlying parameters in the encoder to a better optimum. Then, the content tokens also carry some aspect information and the final "CLS" embedding could be a better representation for relevance matching.

6 Aspect Learning and Overall Training Objectives

Aspect Prediction (AP). The typical way of learning aspect embeddings is to predict the annotated value IDs of the aspects [1,2]. MADRAL[10] also adopts this method. Take an arbitrary aspect a for instance, given its ground-truth annotation set $\mathcal{A}_a$ and its global value set V_a, the loss function is:

$$\mathcal{L}^a_{AP} = -\sum_{v^+ \in \mathcal{A}_a} log \frac{exp(E_a \cdot E_{v^+})}{\sum_{v \in V_a} exp(E_a \cdot E_v)}, \tag{5}$$

where $E_v / E_{v^+} \in \mathbb{R}^H$ is the aspect value embedding from a's embedding lookup table T_a. This is the major loss function to learn aspect embeddings.

Aspect Presence Prediction (APP). MADRAL[10] also proposes the loss of predicting whether an item has a valid value for a certain aspect as a part

of presence weighting (introduced in Section 5.1). The APP loss is essentially a binary classification loss:

$$\mathcal{L}^a_{APP} = -y_a \log(P(I_a)) - (1 - y_a) \log(1 - P(I_a)), \tag{6}$$

where $y_a = 0$ when $\mathcal{A}_a = \varnothing$ and $y_a = 1$ otherwise. Since this objective can also guide aspect embedding training even if not used in the weighting function, we include it as an aspect learning objective, supplementary to the AP loss.

Pre-training Objective. The backbone model is pre-trained on the local corpus to adapt the model parameters and learn the aspect embeddings. The overall pre-training objectives consist of the masked language model (MLM) loss and the aspect learning objectives that can be scaled with λ_p, i.e.,

$$\mathcal{L}_{pretrain} = \mathcal{L}_{MLM} + \lambda_p \sum_{a \in A \setminus \{\text{OTHER}\}} (\mathcal{L}^a_{AP} + \mathcal{L}^a_{APP}). \tag{7}$$

Note that $\mathcal{L}^a_{APP}$ is optional and only takes effect when needed.

Fine-tuning Objective. Similarly, the loss for fine-tuning has relevance loss, and the aspect learning loss (with optional $\mathcal{L}^a_{APP}$) controlled by λ_f, i.e.,

$$\mathcal{L}_{finetune} = \mathcal{L}_{REL} + \lambda_f \sum_{a \in A \setminus \{\text{OTHER}\}} (\mathcal{L}^a_{AP} + \mathcal{L}^a_{APP}), \tag{8}$$

where $\mathcal{L}_{REL}$ is a standard softmax cross-entropy loss that uses the relevant items and in-batch negative samples for training as in [12].

7 EXPERIMENTAL SETTINGS

Multi-aspect Amazon ESCI Dataset (MA-Amazon). The MA-Amazon dataset [21] has 482k products with the aspects of "brand", "color", and "category" besides their titles and descriptions. Only items have aspect information. The coverage of brand, color, and category of levels 1-2-3-4 on the items are 94%, 67%, and 87%-87%-85%-71%, respectively. The relevance dataset for fine-tuning has 17k, 3.5k, and 8.9k real-world queries for training, validation, and testing, respectively. For each query, the relevance dataset provides an average of 20.1 items, each accompanied by relevance judgments - "Exact", "Substitute", "Complement", and "Irrelevant". As in [18,21], we treat *Exact* as positive instances and the other judgments as negative during training and for recall calculation. Although MA-Amazon does not have query aspects as the private Google shopping data does, it is public and has large-scale real-world queries with relevance judgments. We are unaware of other such public datasets, so we only conduct experiments on MA-Amazon.

Methods for Comparison. We compare the MADRAL variants with various dense retrieval baselines, some incorporating aspect information and others not. Besides the baselines the original MADRAL compares, we also include a baseline that uses the aspects as text strings rather than conducting aspect classification for its learning. The baselines are: *BIBERT*: A typical bi-encoder retriever and the backbone of MADRAL, using BERT's `CLS` token for query and item encoding. It is pre-trained with $\mathcal{L}_{MLM}$ in Eq. (7) and fine-tuned with $\mathcal{L}_{REL}$ in Eq. (8); *Condenser*: An advanced pre-trained model for textual dense retrieval. It enhances the `CLS` embedding during pre-training by connecting middle-layer tokens

to the top layers; *BIBERT-CONCAT*: A straightforward method that concatenates the text strings of aspect annotations with item content, adds an indicator token before each aspect, and also uses $\mathcal{L}_{MLM}$ for pre-training and $\mathcal{L}_{REL}$ for fine-tuning; *MTBERT*: A multi-task model based on BIBERT proposed in [10], reusing CLS for aspect prediction alongside MLM during pre-training. The MADRAL variants are: *MADRAL-ori*: The best original MADRAL model [10] introduced in Section 4 & 5.1; *MADRAL-en-v1*: Our first enhanced version of MADRAL-ori that only refines it with the best aspect fusion method in Section 5. They only differ during fine-tuning; *MADRAL-en-v2*: The second enhanced version of MADRAL-ori that incorporates the change in version 1 and the best aspect representation in Section 4.

Evaluation Metrics. We use recall (R) and normalized discounted cumulative gain (NDCG) as evaluation metrics. Specifically, we report R@100, R@500, NDCG@10, and NDCG@50. As in [18,21], the gains of E, S, C, and I judgments are set to 1.0, 0.1, 0.01, and 0.0, respectively. We conducted two-tailed paired t-tests ($p < 0.05$) to check statistically significant differences.

Implementation Details. Since MADRAL does not release code, we have implemented all the MADRAL variants and the baseline methods on our own to ensure consistent experimentation details and fair comparisons. **Pre-training.** For all methods, we share the encoder for both queries and items to promote knowledge sharing. In particular, we pre-train the models on the products in MA-Amazon to acquire the shared encoder for subsequent fine-tuning. In line with prior research [16,15], we initialize all BERT components using Google's public checkpoint and employ the Adam optimizer with the linear warm-up technique. The learning rate and pre-training epoch are set to 1e-4 and 20 respectively. We accommodate a maximum token length of 156 and employ MLM mask ratios of 0.15. For the scaling coefficient of AP and APP objectives, i.e., λ_p in Eq.7, we slightly tune it from 0 to 0.5. Based on the validation results, it is set to 0.1. **Fine-tuning.** For both datasets, we fine-tune all the models for 20 epochs. Following the previous work [8], we include a hard negative sample for each query besides in-batch negatives. We use a learning rate of 5e-6 and a batch size of 64. The maximum token lengths are set to 32 for queries and 156 for items. λ_f in Eq.8 is scanned from {0,0.05,0.1,0.2,0.3}, and the best value is 0 according to evaluation results. We conduct fine-tuning on the pre-trained model checkpoints every two epochs and select the best-performing one on the validation set.

8 Experimental Results

8.1 Comparisons between MADRAL Variants and Baselines

The results of the baselines, the original MADRAL, and our two variants of enhanced MADRAL are shown in Table 1. Among the baselines, we find that better pre-trained models (i.e., Condenser) have better performance, and incorporating the aspect information (MTBERT and BIBERT-CONCAT) can boost the retrieval performance. Note that the method that considers aspects as text strings, i.e., BIBERT-CONCAT, achieves competitive performance compared to methods that use aspects as auxiliary training objectives. For instance, the

Table 1: Comparisons between various retrievers. †, ‡, and ∗ indicate significant improvements over BIBERT, BIBERT-CONCAT, and MTBERT, respectively. The best overall and baseline results are underlined and bold.

Method	R@100	R@500	NDCG@10	NDCG@50
BIBERT	0.6075	0.7795	0.3148	0.3929
Condenser	0.6091	0.7801	0.3191	0.3960
BIBERT-CONCAT	0.6137	0.7814	<u>0.3223</u>	<u>0.4005</u>
MTBERT	<u>0.6139</u>†	<u>0.7849</u>†‡	0.3183†	0.3969†
MADRAL-ori	0.5016	0.7121	0.2086	0.2823
MADRAL-en-v1	0.6159†	0.7892†‡*	0.3220†*	0.4003†*
MADRAL-en-v2	**0.6219**†‡*	**0.7922**†‡*	**0.3291**†‡*	**0.4076**†‡*

performance of MTBERT is better than BIBERT-CONCAT at lower positions (R@500) but worse at higher positions (NDCG@10,50). This indicates that treating aspects as text strings is also an effective approach to leverage these aspects, as observed in [21,20]. Yet, careful learning strategies are required when concatenating the aspect strings with the original content, especially on the query side [21]. It is also promising to study how to combine the two ways of using aspects (i.e., as text strings and by conducting associated value ID prediction) [22]. Further discussions on leveraging aspects as strings are beyond the scope of this paper and interested users can refer to [21,20,22] for more information.

When we compare the performance of the MADRAL variants, it is surprising that the best results the original MADRAL can achieve (i.e., with presence weighting) are still worse than the baselines by a large margin. We attribute this to the insufficient learning of the special aspect - "OTHER" during fine-tuning. In contrast, the best variants we propose to enhance MADRAL in terms of the objects to fuse and aspect representation (denoted as "-en-v1" and "-en-v2" respectively) can significantly boost the retrieval performance. MADRAL-en-v1 uses "CLS" instead of "OTHER" and performs significantly better than MTBERT regarding almost all the metrics. It is also better than BIBERT-CONCAT at lower positions and similar at higher positions. Besides replacing "OTHER" with "CLS", MADRAL-en-v2 uses the first k content tokens to represent the aspects instead of declaring extra k aspect tokens. These two changes together lead to significantly better performance than all the baselines, showing that the proposed variants are effective ways of enhancements.

8.2 Comparisons of Fusion Methods

If we only modify the fusion methods of MADRAL, only fine-tuning will be affected and the changes are relatively small. We introduce our studies on this part before aspect representation variants that incur more changes. Table 2 shows how the fusion objects affect the retrieval performance when equipped with different aspect learning objectives (AP only or AP plus APP) during pretraining and multiple weighting mechanisms during fine-tuning. We can see that for both CLS-gating and presence weighting, using the token "CLS" to explicitly

capture content semantics can boost the performance over learning implicit semantics with the token "OTHER". This confirms our speculation that learning the embedding of "OTHER" sufficiently from scratch during fine-tuning can be challenging, which may be a less severe issue on the Google Shopping data in [10] than on the MA-Amazon data as it has more training data for fine-tuning.

Table 2: Comparisons of various aspect fusion methods. The best results are in bold. All the methods that use either explicit token "CLS" in fusion (see Section 5.2) or do not conduct fusion (see Section 5.1) are significantly better than fusion with implicit token "OTHER". † indicates significant improvements over the BIBERT.

Method	R@100	R@500	NDCG@10	R@100	R@500	NDCG@10
Aspect Fusion	Explicit Token("CLS")			Implicit Token("OTHER")		
CLS-Gating	0.6090	0.7832†	0.3158	0.4743	0.7002	0.1809
APP+CLS-Gating	0.6118†	0.7836†	**0.3194**†	0.4717	0.6973	0.1744
PresenceWeighting	**0.6148**†	**0.7878**†	0.3184†	0.5016	0.7121	0.2086
NoAspectFusion	0.6120†	0.7835†	0.3172†	-	-	-
APP+NoAspectFusion	0.6117†	0.7832†	0.3188†	-	-	-

When we do not conduct aspect fusion and only use CLS as the final representation during fine-turning, the performance is also significantly better than BIBERT and competitive with the aspect fusion methods with "CLS" in it. It indicates that the aspect learning objectives (AP and APP) during pre-training are beneficial for the backbone model's parameters. It again confirms that introducing the special token "OTHER" during fine-tuning and training it from scratch is the reason that harms model performance. A side observation is that using the aspect presence prediction objective as an auxiliary pre-training task can improve some metrics a little, e.g., NDCG@10, showing that it does not have to be paired with the presence weighting mechanism when training MADRAL.

8.3 Comparisons of Aspect Representation

We compare different ways of aspect representation in the model architecture in Table 3. The reported numbers are based on their best fusion methods, i.e., presence weighting for extra k and CLS gating for the rest. Similar to Table 2, we show the performance of both using "OTHER" and "CLS" in aspect fusion. It is obvious that reusing the first k content tokens as aspect representations outperforms declaring extra k embeddings in the original MADRAL. Since reusing existing tokens will not face the issue of insufficient learning of brand-new parameters, it is not surprising that "first k" can be a better aspect representation option. Another interesting finding is that the gap between using "CLS" and "OTHER" for reusing content tokens is much smaller than "Extra k". This means that based on the pre-trained model that attaches the aspect learning with existing content tokens, it reduces the difficulty for the model to act effectively by learning an implicit embedding "OTHER" during fine-tuning.

Table 3: Study of aspect representation approach. "br", "co", and "ca" are short for "brand", "color", and "category". The best results are in bold. † and ‡ indicates significant improvements over BIBERT and the "Extra k" method, respectively.

Method	"CLS" in Fusion			"OTHER" in Fusion		
	R@100	R@500	NDCG@10	R@100	R@500	NDCG@10
Extra k	0.6148†	0.7878†	0.3184†	0.5016	0.7121	0.2086
First k (br,co,ca)	0.6216†‡	0.7919†‡	0.3285†‡	0.6087‡	0.7839†‡	0.3157‡
First k (ca,co,br)	**0.6219**†‡	**0.7922**†‡	**0.3291**†‡	0.6132†‡	0.7859†‡	0.3179†‡
Random k	0.6188†‡	0.7874†	0.3230†‡	0.6081‡	0.7836†‡	0.3113‡

We also study whether the positions of the content tokens that the aspects are mapped to would affect model performance. By using the first 3 tokens to represent different aspects, i.e., (brand, color, category) and (category, color, brand), we do not see significant differences when CLS is in the fusion while the latter is better when "OTHER" is in the fusion. Since "category" is the most important aspect [10,21], it seems that mapping it to a higher position can help the model be stable when learning a new representation during fine-tuning. We also mapped the aspects to random k positions and conducted aspect learning, denoted as "Random k". As smaller positions are often more important in representing an item, we limited the random selection to within the first twenty positions. The performance of "Random k" (i.e., positions 3, 9, 15, and 7 as "brand", "color", "category", and "OTHER" when needed) is lower than "First k" but still beats "Extra k". It implies that reusing positions at higher positions for aspect representation would be better, which is not surprising since the beginning tokens are usually more important than the others.

Table 4: Accuracy@3 of the predicted values using the aspect embeddings learned after pre-training and fine-tuning.

Method	**Category**		**Brand**		**Color**	
	pre-train	**fine-tune**	**pre-train**	**fine-tune**	**pre-train**	**fine-tune**
MTBERT(al=0)	0.9728	0.1118	0.9727	0.0004	0.7843	0.0060
MTBERT(al=0.05)	0.9728	0.4758	0.9727	0.0066	0.7843	0.2367
MADRAL-en-v1(al=0)	0.9712	0.8786	0.9811	0.8251	0.7725	0.2008
MADRAL-en-v1(al=0.05)	0.9712	0.9664	0.9811	0.9649	0.7725	0.7333

8.4 Effect of AL Coefficient λ_f in Fine-Tuning

In the original paper [10], the aspect learning objectives during fine-tuning are helpful for the retrieval performance but we find that they would harm both MTBERT and MADRAL. Figure 3 and Table 4 show the retrieval performance and the accuracy of aspect prediction when using different coefficients of aspect learning during fine-turning. From Table 4, we observe that a small amount of aspect prediction (AP) loss during fine-tuning will boost the AP accuracy. However, the one with higher AP accuracy has worse retrieval performance, as shown in Figure 3. The more AP is used, the more retrieval performance will drop. This implies that the learning objectives between relevance matching and aspect prediction guide the model in different directions.

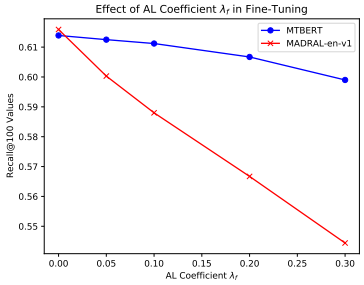

Fig. 3: Effect of AL Coefficient λ_f

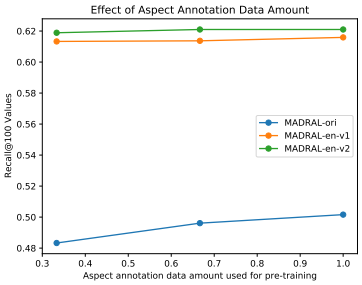

Fig. 4: Effect of Annotation Amount

8.5 Effect of Aspect Annotation Amount

We divide the aspect annotations of the item into three partitions and gradually include more aspect annotations for aspect learning during pre-training. The MLM loss on the entire corpus is always used for pre-training. Figure 4 illustrates the effect of aspect annotation amount on the original MADRAL and our two versions of enhancements. For MADRAL-ori, more annotations help the model perform better (i.e., from 0.4833 to 0.5016), which is consistent with our speculation that the extra k aspect embeddings require more aspect annotations for sufficient learning. When "CLS" replaces "OTHER" during fusion (in MADRAL-en-v1), more annotations bring fewer benefits (i.e., from 0.6133 to 0.6159), indicating that this fusion manner requires fewer aspect annotations to act effectively. When the first k content tokens are adjusted by aspect learning (in MADRAL-en-v2), the recall at 100 saturates with 2/3 aspect annotations. It shows that when aspect learning is used for refining existing important content tokens (first k), even fewer aspect annotations are needed to act effectively.

9 Conclusion

In conclusion, this paper presents a critical examination of the first multi-aspect dense retrieval model, MADRAL. Observing its failure on the public MA-Amazon data, we conduct a thorough investigation into MADRAL's components of aspect representation and fusion. We propose several alternative approaches for each component and compare them with their original counterparts. We find that it has a detrimental effect on retrieval performance to learn implicit semantics with the special aspect "OTHER". In contrast, the proposed variants, including replacing "OTHER" with "CLS" (that represents the overall content semantics explicitly) and representing aspects with the first few content tokens, have demonstrated significant improvements in retrieval performance.

Acknowledgments

This work was funded by the National Natural Science Foundation of China (NSFC) under Grants No. 62302486, the CAS Special Research Assistant Funding Project, the Lenovo-CAS Joint Lab Youth Scientist Project, and the project under Grants No. JCKY2022130C039.

References

1. Ai, Q., Azizi, V., Chen, X., Zhang, Y.: Learning heterogeneous knowledge base embeddings for explainable recommendation. Algorithms **11**(9), 137 (2018)
2. Ai, Q., Zhang, Y., Bi, K., Croft, W.B.: Explainable product search with a dynamic relation embedding model. ACM Trans. Inf. Syst. **38**(1), 4:1–4:29 (2020)
3. Balaneshinkordan, S., Kotov, A., Nikolaev, F.: Attentive neural architecture for ad-hoc structured document retrieval. In: Cuzzocrea, A., Allan, J., Paton, N.W., Srivastava, D., Agrawal, R., Broder, A.Z., Zaki, M.J., Candan, K.S., Labrinidis, A., Schuster, A., Wang, H. (eds.) Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018. pp. 1173–1182. ACM (2018)
4. Choi, J.I., Kallumadi, S., Mitra, B., Agichtein, E., Javed, F.: Semantic product search for matching structured product catalogs in e-commerce. CoRR **abs/2008.08180** (2020)
5. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR **abs/1810.04805** (2018)
6. Gao, L., Callan, J.: Condenser: a pre-training architecture for dense retrieval. In: Moens, M., Huang, X., Specia, L., Yih, S.W. (eds.) Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021. pp. 981–993. Association for Computational Linguistics (2021)
7. Gao, L., Callan, J.: Unsupervised corpus aware language model pre-training for dense passage retrieval. In: Muresan, S., Nakov, P., Villavicencio, A. (eds.) Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022. pp. 2843–2853. Association for Computational Linguistics (2022)
8. Karpukhin, V., Oguz, B., Min, S., Wu, L., Edunov, S., Chen, D., Yih, W.: Dense passage retrieval for open-domain question answering. CoRR **abs/2004.04906** (2020)
9. Khattab, O., Zaharia, M.: Colbert: Efficient and effective passage search via contextualized late interaction over BERT. In: Huang, J.X., Chang, Y., Cheng, X., Kamps, J., Murdock, V., Wen, J., Liu, Y. (eds.) Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020. pp. 39–48. ACM (2020)
10. Kong, W., Khadanga, S., Li, C., Gupta, S.K., Zhang, M., Xu, W., Bendersky, M.: Multi-aspect dense retrieval. In: Zhang, A., Rangwala, H. (eds.) KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022. pp. 3178–3186. ACM (2022)
11. Lin, J., Nogueira, R., Yates, A.: Pretrained Transformers for Text Ranking: BERT and Beyond. Synthesis Lectures on Human Language Technologies, Morgan & Claypool Publishers (2021)
12. Lin, J., Nogueira, R.F., Yates, A.: Pretrained Transformers for Text Ranking: BERT and Beyond. Synthesis Lectures on Human Language Technologies, Morgan & Claypool Publishers (2021)
13. Liu, B., Lu, X., Kurland, O., Culpepper, J.S.: Improving search effectiveness with field-based relevance modeling. In: Proceedings of the 23rd Australasian Document Computing Symposium, ADCS 2018, Dunedin, New Zealand, December 11-12, 2018. pp. 11:1–11:4. ACM (2018)

14. Luan, Y., Eisenstein, J., Toutanova, K., Collins, M.: Sparse, dense, and attentional representations for text retrieval. Trans. Assoc. Comput. Linguistics **9**, 329–345 (2021)
15. Ma, X., Guo, J., Zhang, R., Fan, Y., Cheng, X.: Pre-train a discriminative text encoder for dense retrieval via contrastive span prediction. In: Amigó, E., Castells, P., Gonzalo, J., Carterette, B., Culpepper, J.S., Kazai, G. (eds.) SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022. pp. 848–858. ACM (2022)
16. Ma, X., Guo, J., Zhang, R., Fan, Y., Ji, X., Cheng, X.: Prop: Pre-training with representative words prediction for ad-hoc retrieval. Proceedings of the 14th ACM International Conference on Web Search and Data Mining (2021)
17. Qu, Y., Ding, Y., Liu, J., Liu, K., Ren, R., Zhao, W.X., Dong, D., Wu, H., Wang, H.: Rocketqa: An optimized training approach to dense passage retrieval for open-domain question answering. In: Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tür, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., Zhou, Y. (eds.) Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021. pp. 5835–5847. Association for Computational Linguistics (2021)
18. Reddy, C.K., Màrquez, L., Valero, F., Rao, N., Zaragoza, H., Bandyopadhyay, S., Biswas, A., Xing, A., Subbian, K.: Shopping queries dataset: A large-scale ESCI benchmark for improving product search. CoRR **abs/2206.06588** (2022)
19. Robertson, S., Zaragoza, H., Taylor, M.: Simple bm25 extension to multiple weighted fields. In: Proceedings of the thirteenth ACM international conference on Information and knowledge management. pp. 42–49 (2004)
20. Shan, H., Zhang, Q., Liu, Z., Zhang, G., Li, C.: Beyond two-tower: Attribute guided representation learning for candidate retrieval. In: Proceedings of the ACM Web Conference 2023. pp. 3173–3181 (2023)
21. Sun, X., Bi, K., Guo, J., Ma, X., Yixing, F., Shan, H., Zhang, Q., Liu, Z.: Pre-training with aspect-content text mutual prediction for multi-aspect dense retrieval. arXiv preprint arXiv:2308.11474 (2023)
22. Sun, X., Bi, K., Guo, J., Yang, S., Zhang, Q., Liu, Z., Zhang, G., Cheng, X.: A multi-granularity-aware aspect learning model for multi-aspect dense retrieval (2023)
23. Sun, Y., Wang, S., Li, Y., Feng, S., Chen, X., Zhang, H., Tian, X., Zhu, D., Tian, H., Wu, H.: ERNIE: enhanced representation through knowledge integration. CoRR **abs/1904.09223** (2019), `http://arxiv.org/abs/1904.09223`
24. Tahami, A.V., Ghajar, K., Shakery, A.: Distilling knowledge for fast retrieval-based chat-bots. In: Huang, J.X., Chang, Y., Cheng, X., Kamps, J., Murdock, V., Wen, J., Liu, Y. (eds.) Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020. pp. 2081–2084. ACM (2020)
25. Ueda, A., Santos, R.L.T., Macdonald, C., Ounis, I.: Structured fine-tuning of contextual embeddings for effective biomedical retrieval. In: Diaz, F., Shah, C., Suel, T., Castells, P., Jones, R., Sakai, T. (eds.) SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021. pp. 2031–2035. ACM (2021)
26. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)

27. Xiong, L., Xiong, C., Li, Y., Tang, K., Liu, J., Bennett, P.N., Ahmed, J., Overwijk, A.: Approximate nearest neighbor negative contrastive learning for dense text retrieval. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net (2021)
28. Zamani, H., Mitra, B., Song, X., Craswell, N., Tiwary, S.: Neural ranking models with multiple document fields. In: Chang, Y., Zhai, C., Liu, Y., Maarek, Y. (eds.) Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018. pp. 700–708. ACM (2018)
29. Zhang, S., Liang, Y., Gong, M., Jiang, D., Duan, N.: Multi-view document representation learning for open-domain dense retrieval. In: Muresan, S., Nakov, P., Villavicencio, A. (eds.) Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022. pp. 5990–6000. Association for Computational Linguistics (2022)