Hierarchical Multi-field Representations for Two-Stage E-commerce Retrieval

Anonymous ACL submission

Abstract

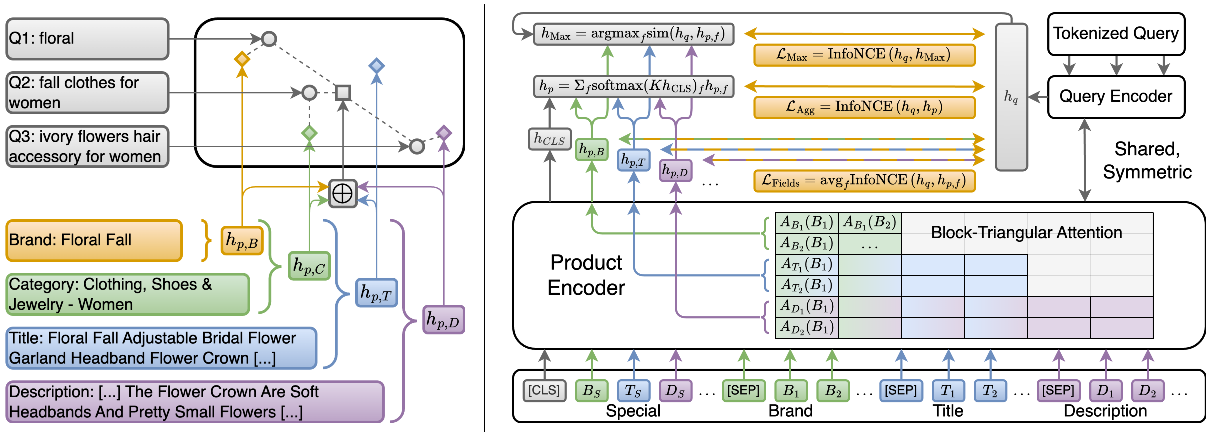
Dense retrieval methods typically target unstructured text data represented as flat strings. However, e-commerce catalogs often include structured information across multiple fields, such as brand, title, and description, which contain important information potential for retrieval systems. We present the Cascading Hierarchical Attention Retrieval Model (CHARM), a novel framework designed to encode structured product data into hierarchical field-level representations with progressively finer detail. Utilizing a novel block-triangular attention mechanism, our method captures the inter-dependencies between product fields in a specified hierarchy, yielding field-level representations and aggregated vectors suitable for fast and efficient retrieval. Combining both representations enables a two-stage retrieval pipeline, in which the aggregated vectors support initial candidate selection, while more expressive field-level representations facilitate precise fine-tuning for downstream ranking. Experiments on publicly available large-scale e-commerce datasets demonstrate that CHARM outperforms state-of-the-art baselines. Our analysis highlights the framework's ability to align different queries with appropriate product fields, enhancing retrieval accuracy and explainability.

1 Introduction

Online shopping has become an ubiquitous part of modern life, making it easier to explore product options and quickly find what we need. Product retrieval, i.e., the task of surfacing the right products for the right queries, is the backbone of this process and has been a focus of active research (Muhamed et al., 2023; Rossi et al., 2024; Li et al., 2024b; Kekuda et al., 2024). With increasing product diversity and user requirements, product retrieval has faced complex challenges such as diverse search intents (Luo et al., 2024), addressing keyword mismatches (Lakshman et al., 2021; Nigam et al., 2019) and scaling approaches to work on product corpora spanning millions of items (Li et al., 2024b). Unlike the extensively explored topic of free-form text retrieval, this work focuses on effectively retrieving items that are represented as e-commerce products consisting of structured data.

Most online stores define products using multiple fields such as brand, category, title, and description. Since customers vary in goals and search styles, finding a good product often involves different fields, requiring flexible and comprehensive retrieval strategies. Figure 1a shows an example. While keyword-based methods like TF-IDF (Salton and Buckley, 1988) and BM25 (Robertson et al., 2009) have been used for decades (Baeza-Yates et al., 1999), recent advances have shifted toward dense retrieval (Karpukhin et al., 2020; Li et al., 2021; Hofstätter et al., 2021; Nardini et al., 2024). In dense retrieval, the main challenge is to embed both queries and product information into a shared latent space where semantically similar pairs are close. However, most work focuses on unstructured input text, and handling structured product fields is often limited to auxiliary pre-training tasks rather than adapting the underlying retrieval (Sun et al., 2023, 2024; Kong et al., 2022).

We propose to leverage semi-structured product data by using field names and their corresponding text directly for dense e-commerce retrieval. We treat product fields as distinct views of the same product, each offering different levels of detail. This hierarchy is input to a transformer-based model that produces a cascade of field-level representations, where each layer incorporates information from the current and all previous fields. To this end, our Cascading Hierarchical Attention Retrieval Model (CHARM), introduces a novel block-triangular attention mechanism that allows each field to attend to its own tokens and all tokens from preceding fields. This attention pattern enables hierarchical accumulation of information, producing



(a) Two-stage retrieval with CHARM.

(b) Structured encoding via block-triangular attention.

Figure 1: CHARM overview. **a)** An aggregated product representation (⊕) is used for initial query (diamond) matching. Matches are re-evaluated based on the closest cascaded field representation (circle), where each field encodes its own and all preceding fields. **b)** Products are tokenized with special tokens per field, and encoded using a block-triangular attention mask that lets each field attend to itself and all previous fields. This structure enables hierarchical, cumulative field-wise representations to be computed in a single forward pass. Both the aggregated and individual field representations are trained to match queries, supporting retrieval at different levels of detail.

field-level representations that capture varying detail and allow the same product to match different queries. For example, shorter, ‘simpler queries tend to align with high-level fields, while longer, more ‘complex‘ queries match detailed representations. To reduce retrieval cost, we adopt a two-stage retrieval strategy. First, we aggregate the field-level vectors into a single representation used for initial retrieval to generate a shortlist of candidate products. Second, we compute full dot-product similarity between the query and the individual field-level vectors of the shortlisted products. Figure 1a illustrates how CHARM matches different queries to different fields of the same product.

We experimentally validate our approach on a public collection of large-scale e-commerce datasets (Reddy et al., 2022). CHARM outperforms common bi-encoder methods (Reimers and Gurevych, 2019; Lin et al., 2022), including approaches that utilize multiple representations for the same product (Zhang et al., 2022). Compared to the latter, it significantly reduces computational cost thanks to its two-stage retrieval process. Additional ablation studies show the effectiveness of the individual parts of CHARM. Finally, we explore how CHARM provides additional explainability through its field-specific matching. We find strong connections between different kinds of queries and product fields, and that more complex product fields yield increasingly diverse representations and query matches.

To summarize our contributions, we (i) propose a novel block-triangular attention mechanism that allows efficient multi-field processing in e-commerce product retrieval, enabling a cascading hierarchy of field-level product representations. (ii) integrate this mechanism with a two-stage retrieval process to combine fast initial shortlisting with powerful field-level matching. (iii) validate the effectiveness of our approach on several public datasets, matching or outperforming state-of-the-art baselines and providing a detailed analysis of our model’s behavior and its inherent explainability.

2 Related Work

Deep neural networks have significantly advanced information retrieval, beginning with character n-gram vector representations processed by multi-layer perceptrons (Huang et al., 2013). Transformer models (Vaswani et al., 2017), especially BERT (Devlin et al., 2019), have enabled more effective retrieval via latent representations of queries and documents (Karpukhin et al., 2020; Li et al., 2021; Hofstätter et al., 2021; Nardini et al., 2024). Leveraging pre-trained Large Language Models (LLMs) (Devlin et al., 2019; Raffel et al., 2020), these methods support holistic, semantic retrieval (Hambarde and Proenca, 2023; Zhao et al., 2024), significantly outperforming classical techniques like TF-IDF (Salton and Buckley, 1988) and BM25 (Robertson et al., 2009) when fine-tuned (Fan et al., 2022), as highlighted in recent surveys (Guo et al., 2022a; Lin et al., 2022; Li and Xu, 2014).

Models such as BiBERT (Reimers and

Gurevych, 2019; Lin et al., 2022) use contrastive training (Hadsell et al., 2006; Jaiswal et al., 2020) in a dual-encoder setup (Bromley et al., 1993) to align texts by semantic similarity. A large corpus is encoded, and queries are matched to nearest neighbors. Extensions include multitask training (Abolghasemi et al., 2022), query expansion (Vishwakarma and Kumar, 2025), multi-teacher distillation (Lin et al., 2023), and token-level embeddings (Khattab and Zaharia, 2020). Based on this line of work, dense retrieval has been effective in e-commerce (He et al., 2023; Muhamed et al., 2023), enabling product search (Magnani et al., 2019), click-through rate prediction (Xiao et al., 2020), and ranking (Li et al., 2019), though often ignoring the rich, multi-field structure of product data. CHARM also uses a dual-encoder BiBERT setup, but without these orthogonal extensions.

Recent work uses multi-field learning in retrieval to address these challenges. *MADRAL* (Kong et al., 2022) incorporates field-specific modules into a dense encoder to produce joint representations for fields like color, brand, and category. However, it relies on pruned categorical labels, limiting generality, and uses auxiliary classification tasks rather than direct encoder inputs to incorporate field information. *MURAL* (Sun et al., 2024) extends *MADRAL* by aligning multi-granular field and token embeddings through self-supervised learning. Like our method, it uses softmax-weighted embedding aggregation and avoids explicit labels. Yet, it struggles with complex fields, such as long descriptions, where token-level signals fall short. Sun et al. (2023) address this issue by modeling inter-field dependencies using mutual prediction objectives during an additional Masked Language Modeling (MLM) pre-training phase (Gao and Callan, 2021), improving information aggregation across fields. This process boosts downstream contrastive learning (Fan et al., 2022; Gao and Callan, 2021; Ma et al., 2022; Li et al., 2023), further enhanced by product-specific reconstruction tasks. In contrast, CHARM modifies the encoder's attention via block-triangular masking, yielding multiple field-level representations..

Another line of work improves dense retrieval by using multiple representations per item. MultiView document Representations (*MVR*)(Zhang et al., 2022) uses a diversity loss to produce distinct views from a single encoder. Multi-View Geometric Index (MVG)(Jiang et al., 2022) applies this idea to e-commerce, augmenting product embeddings with historically matched queries. These methods increase retrieval cost proportionally to the number of representations per item. Efficient indexes using approximate nearest neighbor methods (Sivic and Zisserman, 2003; Malkov and Yashunin, 2018) help, but require large candidate sets to ensure unique results after de-duplication. Two-stage retrieval (Li et al., 2024a) mitigates this issue by shortlisting candidates before re-ranking using field-level decompositions. Prior work (Guo et al., 2022b; Yates et al., 2021; Fan et al., 2022) often treats both stages separately, and even joint training (Ren et al., 2021) typically uses separate models. Hybrid sparse-dense models like SPLADE (Formal et al., 2021b,a; Lassance and Clinchant, 2022) retain an index efficiency but rely on sparse term matching. In contrast, CHARM only performs dense matching, allowing it to model latent semantic relations more effectively while maintaining computational efficiency. While CHARM also uses shortlisting, it constructs hierarchical, context-aware representations in a single encoder pass.

3 Methodology

3.1 Preliminaries

Our retrieval pipeline is based on an encoder-only BERT (Devlin et al., 2019). BERT is a transformer-based (Vaswani et al., 2017) model that employs multi-head attention (Bahdanau et al., 2015), which allows each token of an input sequence to weigh the importance of other tokens to capture complex contextual relationships. For two tokens i, j, the attention of j towards i is

$$A_j(i) = \text{softmax}\left(\frac{\mathbf{q}_j \cdot \mathbf{k}_i^T + M_{j,i}}{\sqrt{d}}\right) \cdot \mathbf{v}_i, \quad (1)$$

where $\mathbf{q}_j \in \mathbb{R}^d$ and $\mathbf{k}_i \in \mathbb{R}^d$ represent the query and key vectors associated with tokens i and i, respectively, and $\mathbf{v}_j \in \mathbb{R}^d$ is the value vector of token j. The attention mask $M_{i,j}$ is set to $M_{i,j} = 0$ if i is allowed to attend to j, and to $M_{i,j} = -\infty$ otherwise. By default, BERT utilizes a full attention mask $\mathbf{M} = \mathbf{0}$, allowing each token to attend to all other tokens.

Given a BERT backbone, we adopt a dual encoder (Bromley et al., 1993; Reimers and Gurevych, 2019; Lin et al., 2022) to map queries and products into a joint embedding space. Representations are aligned via the InfoNCE loss (Sohn, 2016; Oord et al., 2018):

$$\text{InfoNCE}(h_q, h_p) = -\ln \frac{e^{s(h_q, h_{p^+})/\tau}}{\sum_{i=1}^{N} e^{s(h_q, h_{p_i})/\tau}}, \quad (2)$$

where τ is a temperature hyperparameter, h_q is the query embedding, h_{p^+} the positive product, and h_{p_i} includes h_{p^+}, in-batch, and hard negatives (Xiong et al., 2021; Karpukhin et al., 2020). We use the dot-product for the similarity function $s(\cdot, \cdot)$.

Product items typically consist of multiple fields, such as brand, title and description, each capturing different levels of detail (Reddy et al., 2022; Zhou et al., 2023). These fields form a natural hierarchy, where each adds progressively richer information. Ordering them by information content, for example by sorting by length, yields structured, increasingly detailed representations that can be used to generate multi-granular product embeddings.

3.2 Cascading Hierarchical Attention Retrieval Model (CHARM)

Block-triangular Attention. We propose to exploits the hierarchical structure of product information by generating multiple retrieval vectors, each corresponding to a different prefix of product fields. Unlike prior work that enforces diversity via loss functions (Zhang et al., 2022), our method, the Cascading Hierarchical Attention Retrieval Model (CHARM), fosters natural diversity by representing each hierarchy level with its own representation. The first vector encodes the top-level field, the second adds the next field, and so on. This process captures residual information introduced by each field, offering a dense, structured alternative to shallow field-wise combinations (Li et al., 2024a).

We implement CHARM using a modified attention mechanism. Specifically, we alter the attention mask $\mathbf{M}$ so that token i can only attend to tokens from its own and preceding fields, i.e.,

$$M_{i,j} = 0 \text{ if } F(i) \geq F(j), \quad -\infty \text{ otherwise} \quad (3)$$

Here, $F(i)$ is the index of the field containing token i, with fields ordered by their hierarchy level. This *block-triangular attention mask* lets token i attend only to tokens from its field or earlier ones, blocking access to later fields. This process yields a cascade of latent vectors with increasingly detailed field-level product representations in a single forward pass. To extract these representations, we insert field-wise special tokens into the input sequence X_p, placing a SEP token as the end of each field. If a field is empty, its vector is derived from earlier fields and its special token. Appendix A provides a schematic example and further details.

We define the field-level representation as:

$$h_{p,f} = \text{BERT}(X_p, \mathbf{M})_f \quad (4)$$

where $h_{p,f}$ corresponds to the latent vector of the special token for field $f \in \mathcal{F}$. Similar to Sun et al. (2024), we compute an *aggregated representation* as $h_p = \sum_f w_f h_{p,f}$, with $w_f = \text{softmax}(K h_{\text{CLS}})_f$ and $K \in \mathbb{R}^{d \times |\mathcal{F}|}$.

Evaluation. We first encode all products into an index containing their *field-level representation* $h_{p,f}$ and *aggregated representation* h_p. The query is encoded analogously, using shared weights and matching special tokens, which helps align representations.

Retrieval then consists of two stages. We first shortlist the top-k products by comparing the query representation h_q to each h_p. Then, for each shortlisted product, we compute the maximum similarity between its field-level representations $h_{p,f}$ and h_q. This process requires only one model forward pass and supports efficient implementation via priority queues. Given N queries and M products, the overall complexity for this two-stage ranking is $O(N(M + k|\mathcal{F}|))$, compared to $O(NM|\mathcal{F}|)$ for full field-level retrieval (Zhang et al., 2022). Since typically $M \gg k|\mathcal{F}|$, our two-stage approach significantly reduces cost while maintaining retrieval quality by combining a fast initial retrieval stage with a more expressive second one. We use an exact k-Nearest Neighbor index for simplicity, but the method extends naturally to approximate nearest neighbor search (Sivic and Zisserman, 2003; Malkov and Yashunin, 2018).

Training. CHARM combines multiple InfoNCE losses, as described in Equation 2, to optimize both the aggregated and field-specific representations. We match the aggregated representation h_p with the query vector h_q via the loss

$$\mathcal{L}_{\text{Agg}} = \text{InfoNCE}\left(h_q, h_p\right),$$

ensuring an accurate first retrieval stage. Additionally, we match the representations of the individual product fields, i.e.,

$$\mathcal{L}_{\text{Fields}} = \text{avg}_f \text{InfoNCE}\left(h_q, h_{p,f}\right).$$

We finally add an additional loss

$$\mathcal{L}_{\text{Max}} = \text{InfoNCE}\left(h_q, h_{\text{Max}}\right) \quad (5)$$

	US (English)		ES (Spanish)		JP (Japanese)	
Method (Evaluation)	R@100	NDCG@50	R@100	NDCG@50	R@100	NDCG@50
MADRAL*	60.9	39.5				
MURAL-CONCAT*	63.9	42.8				
BIBERT	58.9 ± 0.4	38.4 ± 0.4	56.4 ± 0.6	39.0 ± 0.6	55.3 ± 0.8	40.6 ± 0.7
MVR (Avg.)	54.8 ± 0.5	34.1 ± 0.4	53.5 ± 0.7	35.8 ± 0.5	50.9 ± 0.8	36.4 ± 0.7
MVR (Best)	58.8 ± 0.4	37.3 ± 0.4	59.7 ± 0.7	40.8 ± 0.6	55.8 ± 0.7	39.8 ± 0.7
Our Models						
BIBERT+	63.8 ± 0.4	42.2 ± 0.4	64.4 ± 0.5	44.5 ± 0.6	59.7 ± 0.7	43.6 ± 0.6
BIBERT+-CONCAT	66.5 ± 0.4	44.3 ± 0.4	66.9 ± 0.6	46.0 ± 0.6	60.0 ± 0.7	43.2 ± 0.7
MVR+ (Avg.)	63.0 ± 0.4	41.2 ± 0.4	62.0 ± 0.7	41.7 ± 0.6	57.8 ± 0.8	40.9 ± 0.7
MVR+ (Best)	66.0 ± 0.5	43.8 ± 0.4	*67.8* ± 0.7	*47.0* ± 0.7	*61.3* ± 0.7	44.5 ± 0.7
CHARM (Agg.)	*66.8* ± 0.4	44.8 ± 0.4	66.7 ± 0.6	46.1 ± 0.5	60.3 ± 0.7	44.0 ± 0.7
CHARM (Best)	**67.0** ± 0.4	*45.2* ± 0.4	**68.1** ± 0.6	**47.4** ± 0.6	**61.9** ± 0.7	**45.2** ± 0.7
CHARM (Two-Stage)	66.8 ± 0.4	**45.3** ± 0.4	66.7 ± 0.6	47.0 ± 0.6	60.3 ± 0.7	*44.8* ± 0.7

Table 1: Comparison of means and bootstrapped confidence intervals of CHARM, MVR, MURAL and BiBERT Variants on the Multi-Aspect Amazon Shopping Queries Dataset (Reddy et al., 2022). * indicates results taken from Sun et al. (2024), using different pre-training and training hyperparameters. + indicates MLM pre-training. **Bold** indicates best performance, *italic* indicates second best.

favoring the product field vector $h_{\text{Max}} = \text{argmax}_f \text{sim}(h_q, h_{p,f})$ that most closely matches the query. Combining these losses, we get

$$\mathcal{L} = \lambda_{\text{Agg}}\mathcal{L}_{\text{Agg}} + \lambda_{\text{Fields}}\mathcal{L}_{\text{Fields}} + \lambda_{\text{Max}}\mathcal{L}_{\text{Max}}. \quad (6)$$

The last two losses naturally lead to diverse solutions due to the block-triangular attention structure, allowing us to omit explicit diversity losses (Zhang et al., 2022). This structure ensures that the field-level representations have access to different levels of the information hierarchy of the underlying product, resulting in changing ways to match the query as more product information becomes available. Each field's retrieval vector is optimized to match the query, with additional emphasis on the best-performing field throughout the optimization process. Combined with the loss on the aggregated representation, the total objective encourages the model to learn individually meaningful field-specific representations that can be efficiently combined for a fast first retrieval stage. Figure 1b provides a schematic overview of the CHARM architecture and its losses.

4 Experiments

4.1 Datasets

We evaluate on the English (US), Spanish (ES), and Japanese (JP) subsets of the Multi-Aspect Amazon Shopping Queries dataset (Reddy et al., 2022), which contains real-world e-commerce queries with annotated product matches. Each query is linked to an average of $20-29$ products, with labels indicating exact, substitute, complementary, or irrelevant matches. Following prior work (Sun et al., 2023, 2024), we train by sampling an exact match as a positive and a product from the other labels as a hard negative. Evaluation uses the full product corpus in the respective language. Dataset statistics are shown in Table 4.

Each product includes multiple fields forming a hierarchy of increasingly detailed descriptions, namely "Color", "Brand", "Title", "Description", and "Bullet points". We use this order unless noted otherwise. For the US set, we use an extended version (Sun et al., 2024) with an additional "Category" field inserted between "Brand" and "Title". Tokenization follows Section 3.2, with queries truncated to 64 tokens and products to 400.

4.2 Implementation Details and Baselines

During evaluation, we use a two-stage setup (CHARM *Two-Stage*), retrieving a shortlist of k=100 products per query from the aggregated representation, followed by fine-grained re-ranking using field-level representations. This evaluation setting balances efficiency and quality and is robust to the exact value of k. We also report performance for only the aggregated representation (CHARM *Agg.*) and the best-matching individual field using full search (CHARM *Best*).

Baselines. We compare against several bi-encoder baselines, each using a BERT backbone. MultiView document Representations (*MVR*) (Zhang et al., 2022) encodes multiple representations of a product and uses regular attention over them for matching. Each representation acts as a separate channel over shared product content. To prevent representation collapse, it employs a joint loss

$$\mathcal{L}_{\text{MVR}} = \mathcal{L}_{\text{Max}} + 0.01\mathcal{L}_{\text{Div}},$$

where $\mathcal{L}_{\text{Max}}$ is defined in Equation 5, and the diversity term

$$\mathcal{L}_{\text{Div}} = -\log \frac{e^{f(q,h_{p,\text{Max}})/\tau}}{\sum_f e^{f(q,h_{p,f})/\tau}} \quad (7)$$

encourages representation diversity by maximizing the score of the best-matching one while pushing others away. We align the number of *MVR* representations with the number of product fields for consistency. Since *MVR* lacks a native aggregated representation, we report both the best individual (*MVR (Best)* and mean-pooled (*MVR (Agg.)*) representations. Notably, *MVR* lacks a two-stage evaluation process, making it impractical to use in large-scale applications with too many representations. We also evaluate several BiBERT (Reimers and Gurevych, 2019; Lin et al., 2022) baselines, an InfoNCE loss (Equation 2) and training and evaluating on the CLS token embeddings. We consider three configurations. *BiBERT* uses only the "Title" field and no MLM, representing a naive baseline.*BiBERT**, adds MLM pretraining and corresponds to CHARM or *MVR* with a single field. *BiBERT***-CONCAT* concatenates all fields and applies MLM pretraining. Finally, we include results for *MURAL* (Sun et al., 2024)-*CONCAT* and *MADRAL* (Kong et al., 2022), as reported in Sun et al. (2023). Both use auxiliary pretraining objectives and differ slightly in training setup, making direct comparison difficult.

Pre-training. For CHARM and all models denoted with a $^+$, we first perform a simple MLM pre-training (Fan et al., 2022) on the product corpus of the respective dataset to adapt the initial BERT checkpoints to general product data. We use the same tokenization and data formatting as in the subsequent contrastive training. Appendix C.1 provides pre-training details. We then initialize the shared BERT backbone for the query and product encoders with the resulting pre-trained checkpoint. From this checkpoint, we train each method using its respective loss function. Appendix C.2 lists further details on the setup and relevant training hyperparameters.

Ablation Experiments. To isolate the contributions of CHARM, we ablate key components. We assess the impact of individual loss components from Equation 6, and additionally incorporate the *MVR* diversity loss. *Full Attention* removes the inductive bias of the hierarchical representations by allowing all representations to attend to the entire input. *Diagonal Attention* sets Equation 3 to an equality, enforcing independent field aggregation and eliminating interactions between fields (Li et al., 2024a). *No MLM* omits the MLM pre-training stage entirely. *Asymmetric Encoders* replaces the query encoder's softmax-pooled special tokens with a standard CLS token, breaking symmetry with the product encoder. Finally, *Other Field Order* tests an alternative field sequence based on relative retrieval importance, namely Title, Bullet Points, Category, Brand, Description, and Color.

4.3 Metrics

We compute Recall@$\{10, 100\}$ (R@$\{10, 100\}$) using query-product pairs labeled as "exact" as positive data and all others as negative data. We also report NDCG@50. Following Reddy et al. (2022); Sun et al. (2024), we weight exact pairs with 1.0, substitutes with 0.1, complementary matches with 0.01, and irrelevant matches with 0.0. Finally, we report Precision@10 ($P10$), evaluated by an oracle classifier model trained to predict if a query-product pair is "exact" or not. This metric allows us to also consider sensible query-product pairs that are not explicitly labeled in the training data.

5 Results

5.1 Retrieval Performance

Table 1 reports R@100 and NDCG@50 for CHARM, *MVR*, *MURAL*, and BiBERT variants. Appendix D provides results for R@10 and P@10. CHARM consistently outperforms baselines, including on the challenging JP dataset. Its aggregated representation matches or exceeds BiBERT$^+$-CONCAT, which outperforms BiBERT$^+$ trained only on titles, highlighting the value of additional fields and the effectiveness of our block-diagonal attention. In contrast, averaging *MVR* embeddings performs poorly, likely due to its diversity loss. Since we use $k = 100$ products for the short-list, the Recall@100 performance is the same between the aggregated and the two-stage evaluation. CHARM's two-stage evaluation boosts ranking metrics compared to the aggregated representation, outperforming other methods at comparable cost.

Method	R@10	R@100	NDCG@50	P@10
CHARM	**34.9**	**67.0**	**45.2**	**52.1**
Losses				
Added $\mathcal{L}_{\text{Div}}$	-0.03	$+0.02$	~ 0.00	~ 0.00
$\lambda_{\text{Max}} = 0$	-0.13	$+0.03$	-0.23	-0.12
$\lambda_{\text{Fields}} = 0$	-0.35	-0.52	-0.34	$+0.10$
$\lambda_{\text{Agg}} = 0$	-1.01	-6.46	-1.83	$+0.05$
Attention				
Diagonal Attention	-1.36	-1.73	-1.38	$+0.67$
Full Attention	-0.73	-0.16	-0.75	-1.13
(+Added $\mathcal{L}_{\text{Div}}$)	-0.68	-0.22	-0.74	-1.12
Pretraining				
No MLM	-3.18	-5.32	-4.52	-2.91
Misc.				
Other Field Order	-0.25	-0.34	-0.34	-0.58
Asym. Encoders	-0.40	-0.16	-0.29	-0.18

Table 2: Evaluation results for CHARM *(Two-Stage)* ablations on the US dataset. We report the performance for CHARM and the absolute difference to it for all ablations.

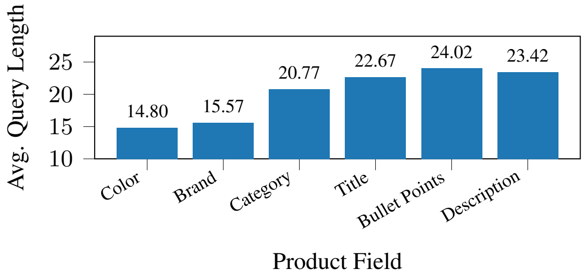


Figure 2: Average length of queries matching a product field by closest dot-product similarity. Product fields that are on a higher hierarchy level generally match longer queries.

5.2 Ablation Results

Table 2 reports ablation results for CHARM (*Two-Stage)* on the US dataset. Each loss component in Equation 6 contributes meaningfully, while adding the diversity loss from Equation 7 yields no improvement. Removing the loss on the aggregated representation (λ_{Agg}=0) leads to a poor shortlist, reducing R@100 performance despite minor impact on top matches, i.e., R@10.

Diagonal attention fails to capture the hierarchical and interleaved structure of product data. In contrast, full attention allows access to all fields but reduces representational diversity, even with an added diversity loss. MLM pre-training greatly improves performance, which is consistent with Table 1. Reordering fields by retrieval importance slightly harms results, suggesting that placing shorter, more compressed fields earlier in the hierarchy is beneficial. Replacing the softmax-pooled special tokens with a CLS token for queries degrades performance, likely due to broken encoder symmetry and less effective weight sharing.

6 Further Analysis

While CHARM shows modest performance gains compared to the considered baselines, its main advantage lies in the diversity and explainability induced by its block-triangular attention mechanism. We investigate these effects, as well as the matching capabilities of the resulting field-level product representations. For this analysis, we focus on the evaluation queries and product corpus of the US dataset. Unless mentioned otherwise, all evaluations use our two-stage retrieval process, and evaluate the top 10 products and their associated, most relevant product field for each query.

Diversity of Field-level Representations. We analyze the average number of characters in a query that matches any given field, using this metric as a proxy for query complexity. Figure 2 shows that longer queries tend to align with later product fields, indicating that more complex queries benefit from more detailed representations.

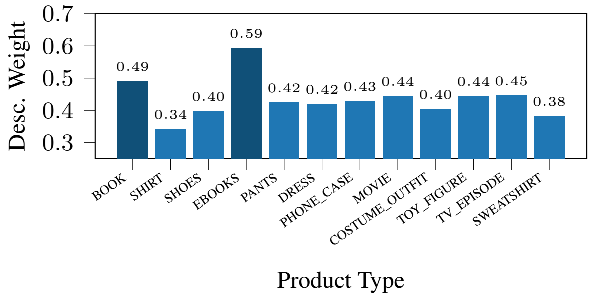
To assess the diversity of field-level representations across the corpus, we compute average pairwise Euclidean distance, dot-product similarity, and the log-determinant of the covariance matrix. As shown in Table 3, fields that appear later in the hierarchy produce more diverse representations, supporting the idea that CHARM learns a hierarchy of increasingly expressive embeddings matched to query complexity.

We also test whether the aggregated representation h_p meaningfully integrates field-level information. Using crawled product type metadata, we analyze the distribution of softmax weights w_f over fields by category. Figure 3a shows that media products like books assign more weight to the "Description" field compared to other product types such as clothing. This capability supports the robustness of our approach and lays the groundwork for explainable search systems that dynamically match important product fields.

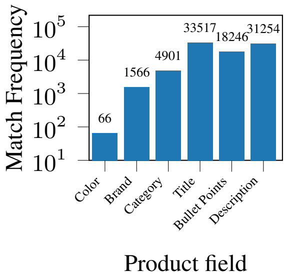
Query-Product Match Analysis. Figure 3b shows how often each product field appears among the top 10 matches for queries in the US dataset. More specific fields appear more frequently, with

Metric	Agg.	Color	Brand	Cat.	Title	Bullet P.	Desc.
$\uparrow$ Euclidean	2.618	1.126	1.985	2.906	4.014	4.067	4.054
$\downarrow$ Dot Product	19.35	19.75	19.60	19.38	19.24	19.40	19.44
$\uparrow$ Log-det	-5679	-7411	-6146	-5552	-4916	-4905	-4918

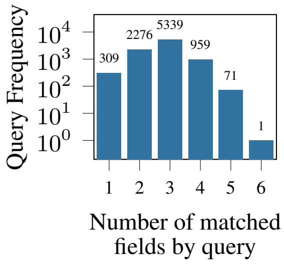
Table 3: Corpus diversity metrics by product field.



(a) "Description" field weight in the aggregated representation.



(b) Log-frequency of product fields appearing as top 10 matches for any query.



(c) Log-frequency of queries matching a number of product fields in their top 10 matches.

Figure 3: Field relevance and query matching.

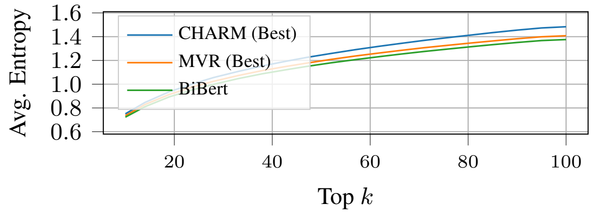


Figure 4: Average entropy of product type distributions across different methods and top-k values

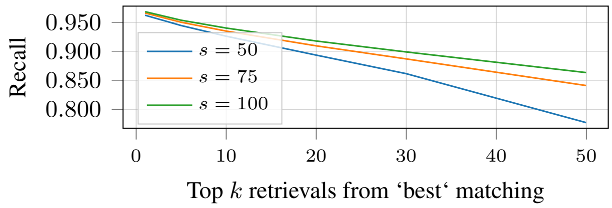


Figure 5: Preservation of ‘best‘ matches in two-stage retrieval for different initial shortlist sizes $s \in \{50, 75, 100\}$.

"Title" being the most common, likely due to its importance and low noise. The results suggest that CHARM often utilizes fields up to the "Title," while later fields like bullet points or descriptions may add little or even unnecessary information for many queries. Figure 3c shows that most queries match two to three different fields within their top 10. Thus, while queries often cover multiple types of product information, they usually do not span the full hierarchy. To analyze retrieval diversity, we compute the average entropy over product types in the top k results. Higher entropy reflects greater variety in the retrieved items. Figure 4 shows that CHARM consistently produces more diverse results than *MVR* and *BiBERT* across all values of k. Qualitatively, Figure 1a shows different queries matching the same product using different fields. Appendix E provides examples for the reverse direction, where the same query matches different products through different fields. In each case, the matched field adds useful information beyond the preceding ones in the hierarchy.

Two-stage retrieval. Figure 5 shows that our two-stage retrieval with shortlist size $k = 100$ effectively preserves high-quality matches. We measure how often the first retrieval stage includes the top matches identified by the best matching field, i.e., how many matches are shared between CHARM *(Agg.)* and CHARM *Best*. Recall curves across varying k and shortlist sizes s indicate strong similarities. For example, with a shortlist size of 50, over 90% of the ‘true‘ top 10 matches are successfully retained. This high preservation of relevant matches confirms that aggregated representations offer a good trade-off between efficiency and retrieval quality.

7 Conclusion

We present the Cascading Hierarchical Attention Retrieval Model (CHARM), an adaptive representation framework for efficient retrieval of multi-field e-commerce product data. CHARM introduces a novel block-triangular attention mechanism that allows each product field in a specified hierarchy to attend to itself and preceding fields, producing increasingly detailed field-level representations in a single forward pass. The representations are aggregated for shortlist retrieval, then re-ranked by matching queries to their best-aligned field. This two-stage process enables fast, accurate retrieval tailored to diverse query intents.

Our empirical results highlight the importance of leveraging multiple product fields and the effectiveness of the emerging diversity of CHARM compared to state-of-the-art baselines. We validate each component of our model through ablation studies and further show that CHARM fosters diverse, interpretable field representations. The model leverages diverse product fields, with deeper fields having more complex representations, and tends to align intricate queries with similarly complex product fields.

Limitations

CHARM currently requires a fixed, linear hierarchy of product field. While approach works well for the product types discussed in this work, many e-commerce stores curate more complex fields with less direct or hierarchical relationships. In future work, we will thus investigate extending the block-triangular attention matrix to more general attention graphs, allowing subsets of product fields to attend to arbitrary subsets for more effective and diverse communication between selected fields.

Further, our two-stage retrieval process requires a computational overhead that is constant regardless of the underlying query. Especially for simpler queries, that, e.g., just look for a certain brand, this process incurs unneccesary cost. To alleviate this issue, we want to assign different dimensions of the retrieval vector to the different product fields, matching the amount of retrieval dimensions to the information content of the field to allow for more effective retrieval.

Potential Risks. While our work is primarily methodological, efficient retrieval systems can influence downstream model behavior. In high-recall or user-facing scenarios, care should be taken to mitigate risks such as content bias or retrieval of low-quality information.

References

Amin Abolghasemi, Suzan Verberne, and Leif Azzopardi. 2022. Improving bert-based query-by-document retrieval with multi-task optimization. In *European Conference on Information Retrieval*, pages 3–12. Springer.

Ricardo Baeza-Yates, Berthier Ribeiro-Neto, and 1 others. 1999. *Modern information retrieval*, volume 463.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations (ICLR)*.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1993. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, 6.

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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library.

Yixing Fan, Xiaohui Xie, Yinqiong Cai, Jia Chen, Xinyu Ma, Xiangsheng Li, Ruqing Zhang, Jiafeng Guo, and 1 others. 2022. Pre-training methods in information retrieval. *Foundations and Trends® in Information Retrieval*, 16(3):178–317.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021a. Splade v2: Sparse lexical and expansion model for information retrieval. *arXiv preprint arXiv:2109.10086*.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021b. Splade: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 2288–2292, New York, NY, USA. Association for Computing Machinery.

Luyu Gao and Jamie Callan. 2021. Condenser: a pre-training architecture for dense retrieval. *arXiv preprint arXiv:2104.08253*.

Luyu Gao, Xueguang Ma, Jimmy J. Lin, and Jamie Callan. 2022. Tevatron: An efficient and flexible toolkit for dense retrieval. *ArXiv*, abs/2203.05765.

Luyu Gao, Yunyi Zhang, Jiawei Han, and Jamie Callan. 2021. Scaling deep contrastive learning batch size under memory limited setup. In *6th Workshop on Representation Learning for NLP, RepL4NLP 2021*, pages 316–321. Association for Computational Linguistics (ACL).

Jiafeng Guo, Yinqiong Cai, Yixing Fan, Fei Sun, Ruqing Zhang, and Xueqi Cheng. 2022a. Semantic models for the first-stage retrieval: A comprehensive review. *ACM Trans. Inf. Syst.*, 40(4).

Jiafeng Guo, Yinqiong Cai, Yixing Fan, Fei Sun, Ruqing Zhang, and Xueqi Cheng. 2022b. Semantic models for the first-stage retrieval: A comprehensive review. *ACM Transactions on Information Systems (TOIS)*, 40(4):1–42.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE.

Kailash A Hambarde and Hugo Proenca. 2023. Information retrieval: recent advances and beyond. *IEEE Access*.

Yunzhong He, Yuxin Tian, Mengjiao Wang, Feier Chen, Licheng Yu, Maolong Tang, Congcong Chen, Ning Zhang, Bin Kuang, and Arul Prakash. 2023. Que2engage: Embedding-based retrieval for relevant and engaging products at facebook marketplace. In *Companion Proceedings of the ACM Web Conference 2023*, pages 386–390.

Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 113–122.

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 *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2333–2338.

Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. 2020. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2.

Nan Jiang, Dhivya Eswaran, Choon Hui Teo, Yexiang Xue, Yesh Dattatreya, Sujay Sanghavi, and Vishy Vishwanathan. 2022. On the value of behavioral representations for dense retrieval. *arXiv preprint arXiv:2208.05663*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Akshay Kekuda, Yuyang Zhang, and Arun Udayashankar. 2024. Embedding based retrieval for long tail search queries in ecommerce. In *Proceedings of the 18th ACM Conference on Recommender Systems*, RecSys '24, page 771–774, New York, NY, USA. Association for Computing Machinery.

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*, pages 39–48.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*.

Weize Kong, Swaraj Khadanga, Cheng Li, Shaleen Kumar Gupta, Mingyang Zhang, Wensong Xu, and Michael Bendersky. 2022. Multi-aspect dense retrieval. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3178–3186.

Vihan Lakshman, Choon Hui Teo, Xiaowen Chu, Priyanka Nigam, Abhinandan Patni, Pooja Maknikar, and SVN Vishwanathan. 2021. Embracing structure in data for billion-scale semantic product search. *arXiv preprint arXiv:2110.06125*.

Carlos Lassance and Stéphane Clinchant. 2022. An efficiency study for splade models. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2220–2226.

Hang Li and Jun Xu. 2014. Semantic matching in search. *Foundations and Trends® in Information Retrieval*, 7(5):343–469.

Millicent Li, Tongfei Chen, Benjamin Van Durme, and Patrick Xia. 2024a. Multi-field adaptive retrieval. *arXiv preprint arXiv:2410.20056*.

Rui Li, Yunjiang Jiang, Wenyun Yang, Guoyu Tang, Songlin Wang, Chaoyi Ma, Wei He, Xi Xiong, Yun Xiao, and Eric Yihong Zhao. 2019. From semantic retrieval to pairwise ranking: Applying deep learning in e-commerce search. SIGIR'19, page 1383–1384, New York, NY, USA. Association for Computing Machinery.

Sen Li, Fuyu Lv, Ruqing Zhang, Dan Ou, Zhixuan Zhang, and Maarten de Rijke. 2024b. Text matching indexers in taobao search. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 5339–5350, New York, NY, USA. Association for Computing Machinery.

Xinze Li, Zhenghao Liu, Chenyan Xiong, Shi Yu, Yu Gu, Zhiyuan Liu, and Ge Yu. 2023. Structure-aware language model pretraining improves dense retrieval on structured data. *Preprint*, arXiv:2305.19912.

Yizhi Li, Zhenghao Liu, Chenyan Xiong, and Zhiyuan Liu. 2021. More robust dense retrieval with contrastive dual learning. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*, pages 287–296.

Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. 2022. *Pretrained transformers for text ranking: Bert and beyond*. Springer Nature.

Sheng-Chieh Lin, Akari Asai, Minghan Li, Barlas Oguz, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. 2023. How to train your dragon: Diverse augmentation towards generalizable dense retrieval. *arXiv preprint arXiv:2302.07452*.

Chen Luo, Xianfeng Tang, Hanqing Lu, Yaochen Xie, Hui Liu, Zhenwei Dai, Limeng Cui, Ashutosh Joshi, Sreyashi Nag, Yang Li, and 1 others. 2024. Exploring query understanding for amazon product search. In *2024 IEEE International Conference on Big Data (BigData)*, pages 2343–2348. IEEE.

Xinyu Ma, Jiafeng Guo, Ruqing Zhang, Yixing Fan, and Xueqi Cheng. 2022. Pre-train a discriminative text encoder for dense retrieval via contrastive span prediction. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 848–858.

Alessandro Magnani, Feng Liu, Min Xie, and Somnath Banerjee. 2019. Neural product retrieval at walmart. com. In *Companion Proceedings of The 2019 World Wide Web Conference*, pages 367–372.

YA Malkov and DA Yashunin. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):824–836.

Aashiq Muhamed, Sriram Srinivasan, Choon-Hui Teo, Qingjun Cui, Belinda Zeng, Trishul Chilimbi, and SVN Vishwanathan. 2023. Web-scale semantic product search with large language models. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 73–85. Springer.

Franco Maria Nardini, Cosimo Rulli, and Rossano Venturini. 2024. Efficient multi-vector dense retrieval with bit vectors. In *European Conference on Information Retrieval*, pages 3–17. Springer.

Priyanka Nigam, Yiwei Song, Vijai Mohan, Vihan Lakshman, Weitian (Allen) Ding, Ankit Shingavi, Choon Hui Teo, Hao Gu, and Bing Yin. 2019. Semantic product search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 2876–2885, New York, NY, USA. Association for Computing Machinery.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and 1 others. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Chandan K Reddy, Lluís Màrquez, Fran Valero, Nikhil Rao, Hugo Zaragoza, Sambaran Bandyopadhyay, Arnab Biswas, Anlu Xing, and Karthik Subbian. 2022. Shopping queries dataset: A large-scale esci benchmark for improving product search. *arXiv preprint arXiv:2206.06588*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Conference on Empirical Methods in Natural Language Processing*.

Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qiaoqiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021. Rocketqav2: A joint training method for dense passage retrieval and passage re-ranking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2825–2835.

Stephen Robertson, Hugo Zaragoza, and 1 others. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Nicholas Rossi, Juexin Lin, Feng Liu, Zhen Yang, Tony Lee, Alessandro Magnani, and Ciya Liao. 2024. Relevance filtering for embedding-based retrieval. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, CIKM '24, page 4828–4835, New York, NY, USA. Association for Computing Machinery.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

Sivic and Zisserman. 2003. Video google: A text retrieval approach to object matching in videos. In *Proceedings ninth IEEE international conference on computer vision*, pages 1470–1477. IEEE.

Kihyuk Sohn. 2016. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29.

Xiaojie Sun, Keping Bi, Jiafeng Guo, Xinyu Ma, Yixing Fan, Hongyu Shan, Qishen Zhang, and Zhongyi Liu. 2023. Pre-training with aspect-content text mutual prediction for multi-aspect dense retrieval. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, page 4300–4304, New York, NY, USA. Association for Computing Machinery.

Xiaojie Sun, Keping Bi, Jiafeng Guo, Sihui Yang, Qishen Zhang, Zhongyi Liu, Guannan Zhang, and Xueqi Cheng. 2024. A multi-granularity-aware aspect learning model for multi-aspect dense retrieval. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 674–682.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*.

Deepak Vishwakarma and Suresh Kumar. 2025. Fine-tuned bert algorithm-based automatic query expansion for enhancing document retrieval system. *Cognitive Computation*, 17(1):1–16.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and 1 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Zhibo Xiao, Luwei Yang, Wen Jiang, Yi Wei, Yi Hu, and Hao Wang. 2020. Deep multi-interest network for click-through rate prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 2265–2268, New York, NY, USA. Association for Computing Machinery.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

Andrew Yates, Rodrigo Nogueira, and Jimmy Lin. 2021. Pretrained transformers for text ranking: BERT and beyond. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorials*, pages 1–4, Online. Association for Computational Linguistics.

Shunyu Zhang, Yaobo Liang, Ming Gong, Daxin Jiang, and Nan Duan. 2022. Multi-view document representation learning for open-domain dense retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5990–6000, Dublin, Ireland. Association for Computational Linguistics.

Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2024. Dense text retrieval based on pretrained language models: A survey. *ACM Transactions on Information Systems*, 42(4):1–60.

Jianghong Zhou, Bo Liu, Jhalak Nilesh Acharya, Yao Hong, Kuang-chih Lee, and Musen Wen. 2023. Leveraging large language models for enhanced product descriptions in ecommerce. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, page 88.

A Block-triangular Attention

Figure 6 visualizes a block-diagonal attention matrix for exemplary "(B)rand", "(T)itle" and "(D)escription" fields. In practice, we move all special tokens directly behind the CLS token while maintaining their attention structure to ensure a consistent positional encoding.

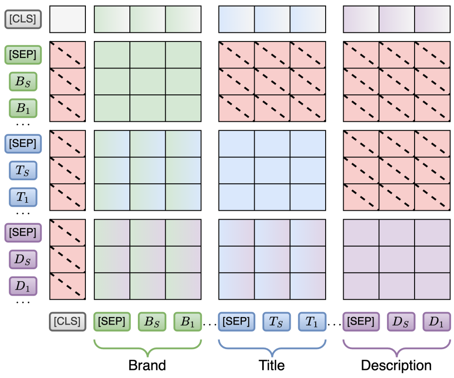


Figure 6: Exemplary block-diagonal attention matrix. Each row (i) represents the attention of one token to all tokens in the sequence, while each column (j) shows which other tokens a token is attended by. The two-colored cells indicate that tokens of one field attend to another field ($M_{i,j} = 0$ in Equation 1). The red dotted cells indicate masking ($M_{i,j} = -\infty$), which ensures that the tokens of a given field can only attend to tokens of this or previous fields. Combined with increasingly detailed fields, this structure yields an information cascade, where the latent vectors of each product field's tokens include increasingly detailed representations.

B Datasets

We provide statistics for the number of train and evaluation queries, their average number of positive and negative product pairs, and size of the full product corpus in Table 4.

Dataset	Type	Amount	Pos.	Neg.
US	Train Queries	17,388	8.70	11.41
	Test Queries	8,955	8.90	11.38
	Corpus	482,105	–	–
ES	Train Queries	11,336	13.44	9.77
	Test Queries	3,844	12.91	11.37
	Corpus	259,973	–	–
JP	Train Queries	7,284	13.20	15.51
	Test Queries	3,123	13.32	15.11
	Corpus	233,850	–	–

Table 4: Dataset statistics for US, ES, and JP subsets of the Multi-Aspect Amazon Shopping Queries dataset (Reddy et al., 2022). "Pos." and "Neg." denote the average number of positive and negative pairs in the dataset, respectively.

C Hyperparameters

All model trainings and pre-trainings are conducted using the ADAM (Kingma and Ba, 2015) optimizer with a linear learning rate scheduling and a warm-up ratio of 0.1. We further train and evaluate using 16-bit floating point operations, and clip the maximum gradient norm to 1.0 for all trainings. Each experiment uses 4 Nvidia V100 GPUs.

C.1 MLM Pre-training.

Table 5 provides hyperparameters for the MLM pre-training stage. We use the resulting model checkpoints as the initial weights for all experiments unless mentioned otherwise. We use the same general pre-training parameters across datasets, except that we employ a multilingual BERT (mBERT) (Devlin et al., 2019) model for the non-english ES and JP datasets. Since this model is more expensive to run due to an increased token vocabulary, we only train these datasets for $30,000$ steps instead of the $40,000$ for the US one.

	Dataset		
Parameter	**US**	**JP**	**ES**
Pretrained checkpoint	BERT (uncased)[2]	mBERT (cased)[3]	
Training steps	40,000	30,000	
MLM masking rate	0.15		
Learning rate	1.0×10^{-4}		
Batch size	512		

Table 5: Parameters for the MLM pre-training. Parameters that are only listed once are shared between datasets.

C.2 Training Setup and Hyperparameters.

We implement all experiments in pytorch (Paszke et al., 2019), using the huggingface transformer package (Wolf et al., 2020) and Tevatron (Gao et al., 2022) for the contrastive training. We perform the retrieval using FAISS-GPU (Johnson et al., 2019; Douze et al., 2024) with a full similarity search and a dot-product similarity metric.

All training runs denoted with an $^+$ use the final checkpoints from the MLM pre-training stage of the respective dataset as initial model weights. Runs without $^+$ use the official BERT checkpoints, as mentioned in Table 5. The pre-training allows each model to benefit from task-relevant

[2] https://huggingface.co/google-bert/bert-base-uncased

[3] https://huggingface.co/google-bert/bert-base-multilingual-cased

language representations prior to contrastive fine-tuning. Additional training hyperparameters used for CHARM across datasets are listed in Table 6. For baseline methods, we adopt the same configuration, except for the number of training epochs, which is set to 200, and the temperature parameters, where we use τ=0.1 for US and τ=0.1 for ES and JP. All other hyperparameters remain unchanged unless specified otherwise. Since the batch size of 1024 does not fit into memory for regular hardware, we use gradient caching for contrastive training (Gao et al., 2021) to allow for all batch samples to act as in-batch negatives for all other samples.

	Dataset		
Parameter	**US**	**JP**	**ES**
Learning rate		5.0e−6	
Batch size		1024	
τ (Eq. 2)	0.1	0.5	0.5
Training epochs	200	300	200
λ_{Fields} (Eq. 6)	1	0.05	0.05
λ_{Agg} (Eq. 6)		1	
λ_{Max} (Eq. 6)		1	

Table 6: Parameters for the contrastive training. Parameters that are only listed once are shared between datasets.

C.3 Computational Resources.

We run all experiments in the cloud, using NVIDIA V100 instances. Each training is parallelized across 4 GPUs, and takes between 6 and 12 hours, depending on the dataset.

D Extended Results

To complement the aggregate results in Table 1, we report detailed performance on each language subset in Tables 7, 8, and 9. These tables report R@10, R@100, NDCG@50, and P@10 for English (US), Spanish (ES), and Japanese (JP), respectively. We find that the results for R@10 and P@10 are overall consistent with the metrics reported in the main paper. Across datasets, CHARM (Best) slightly outperforms CHARM (Two-Stage) on R@10, reflecting the benefit of full-field retrieval for optimizing top-ranked results. In contrast, the two-stage setup trades some top-k precision for faster inference via its shortlist based on the aggregated representation. This result highlights the typical trade-off between retrieval quality and efficiency in multi-stage retrieval settings.

E Example Matches

Table 10 provides examples where the same query retrieves different products by matching on different fields. The matched field contribute new and more specific information compared to the previous field, such as highlighting specific features in bullet points versus generic category labels. For example, in the last row, the query *pink womans toolbag* is matched through a bullet point emphasizing "pink" and a title mentioning "Ladies Tool Bag," combining to capture the full query intent. These examples show how different fields can contain complementary information, and how capturing this information hierarchically leads to more accurate matching.

Method (Evaluation)	US (English) R@10	R@100	NDCG@50	P@10
MADRAL*		60.9	39.5	
MURAL-CONCAT*		63.9	42.8	
BIBERT	28.7 ± 0.4	58.9 ± 0.4	38.4 ± 0.4	47.3
MVR (Avg.)	25.2 ± 0.4	54.8 ± 0.5	34.1 ± 0.4	44.2
MVR (Best)	28.2 ± 0.4	58.8 ± 0.4	37.3 ± 0.4	46.2
Our Models				
$BIBERT^+$	31.8 ± 0.4	63.8 ± 0.4	42.2 ± 0.4	50.0
$BIBERT^+$-CONCAT	33.7 ± 0.4	66.5 ± 0.4	44.3 ± 0.4	50.7
MVR^+ (Avg.)	31.4 ± 0.4	63.0 ± 0.4	41.2 ± 0.4	48.8
MVR^+ (Best)	33.7 ± 0.4	66.0 ± 0.5	43.8 ± 0.4	50.8
CHARM (Agg.)	34.2 ± 0.4	*66.8* ± 0.4	44.8 ± 0.4	51.2
CHARM (Best)	**34.9** ± 0.4	**67.0** ± 0.4	*45.2* ± 0.4	52.1
CHARM (Two-Stage)	*34.8* ± 0.4	66.8 ± 0.4	**45.3** ± 0.4	51.9

Table 7: Results on the US (English) subset. *: from Sun et al. (2024), +: MLM pre-trained.

Method (Evaluation)	ES (Spanish) R@10	R@100	NDCG@50	P@10
BIBERT	24.9 ± 0.6	56.4 ± 0.6	39.0 ± 0.6	56.5
MVR (Avg.)	22.4 ± 0.6	53.5 ± 0.7	35.8 ± 0.5	54.3
MVR (Best)	26.3 ± 0.5	59.7 ± 0.7	40.8 ± 0.6	57.3
Our Models				
$BIBERT^+$	28.5 ± 0.5	64.4 ± 0.5	44.5 ± 0.6	62.1
$BIBERT^+$-CONCAT	29.1 ± 0.5	66.9 ± 0.6	46.0 ± 0.6	62.6
MVR^+ (Avg.)	26.1 ± 0.6	62.0 ± 0.7	41.7 ± 0.6	60.0
MVR^+ (Best)	30.4 ± 0.5	*67.8* ± 0.7	*47.0* ± 0.7	63.4
CHARM (Agg.)	29.4 ± 0.5	66.7 ± 0.6	46.1 ± 0.5	62.6
CHARM (Best)	**30.5** ± 0.6	**68.1** ± 0.6	**47.4** ± 0.6	63.8
CHARM (Two-Stage)	*30.4* ± 0.6	66.7 ± 0.6	47.0 ± 0.6	63.6

Table 8: Results on the ES (Spanish) subset. + indicates MLM pre-trained models.

Method (Evaluation)	JP (Japanese) R@10	R@100	NDCG@50	P@10
BIBERT	27.4 ± 0.6	55.3 ± 0.8	40.6 ± 0.7	56.5
MVR (Avg.)	24.3 ± 0.6	50.9 ± 0.8	36.4 ± 0.7	44.0
MVR (Best)	26.7 ± 0.6	55.8 ± 0.7	39.8 ± 0.7	46.1
Our Models				
$BIBERT^+$	29.1 ± 0.7	59.7 ± 0.7	43.6 ± 0.6	62.1
$BIBERT^+$-CONCAT	28.9 ± 0.6	60.0 ± 0.7	43.2 ± 0.7	62.6
MVR^+ (Avg.)	27.4 ± 0.7	57.8 ± 0.8	40.9 ± 0.7	48.6
MVR^+ (Best)	30.1 ± 0.7	*61.3* ± 0.7	44.5 ± 0.7	50.7
CHARM (Agg.)	29.5 ± 0.7	60.3 ± 0.7	44.0 ± 0.7	50.2
CHARM (Best)	**30.5** ± 0.7	**61.9** ± 0.7	**45.2** ± 0.7	51.9
CHARM (Two-Stage)	*30.3* ± 0.6	60.3 ± 0.7	*44.8* ± 0.7	51.2

Table 9: Results on the JP (Japanese) subset. + indicates MLM pre-trained models.

Query	Matched Field	Previous Field
ergonomic desk	Category: Home & Kitchen - Furniture - Home Office Furniture - Home Office Desks	Brand: EUREKA ERGONOMIC
	Title: RESPAWN RSP-3000 Computer Ergonomic Height Adjustable Gaming Desk [...]	Category: Home & Kitchen - Furniture - Home Office Furniture - Home Office Desks
	Bullet Points: Go from sitting to standing in one smooth motion with this complete active workstation providing comfortable viewing angles and customized user heights [...]	Title: VIVO Electric Height Adjustable 43 x 24 inch Stand Up Desk
pink womans toolbag	Category: Tools & Home Improvement - Power & Hand Tools - Tool Organizers - Tool Bags	Brand: The Original Pink Box
	Title: Pretty Pink Tool Carry-All With Red Trim-12-1/2 X 9-1/2 X 8 Inches With Multiple Pockets And Metal Handle	Category: Tools & Home Improvement - Power & Hand Tools - Tool Organizers - Tool Bags
	Bullet Points: Perfect basic set all the essentials are here. Tools and bag are lovely pink with rubbery grips. Great quality tools.	Title: IIT 89808 Ladies Tool Bag 9 Piece

Table 10: Qualitative examples of a query matching different products on different fields.