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

Niklas Freymuth[1,2,†], Dong Liu[3], Thomas Ricatte[3] and Saab Mansour[3]

[1]Karlsruhe Institute of Technology, Karlsruhe, Germany
[2]Work done during internship at Amazon
[3]Amazon

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 hierarchical manner, 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. Experimentally, CHARM provides higher-quality retrieval compared to state-of-the-art dense retrieval methods on publicly available large-scale e-commerce datasets. Further, our analysis highlights the framework's ability to align different queries with appropriate product fields, enhancing retrieval accuracy and explainability.

Keywords

Information Retrieval, Multi-Field Retrieval, Block Attention

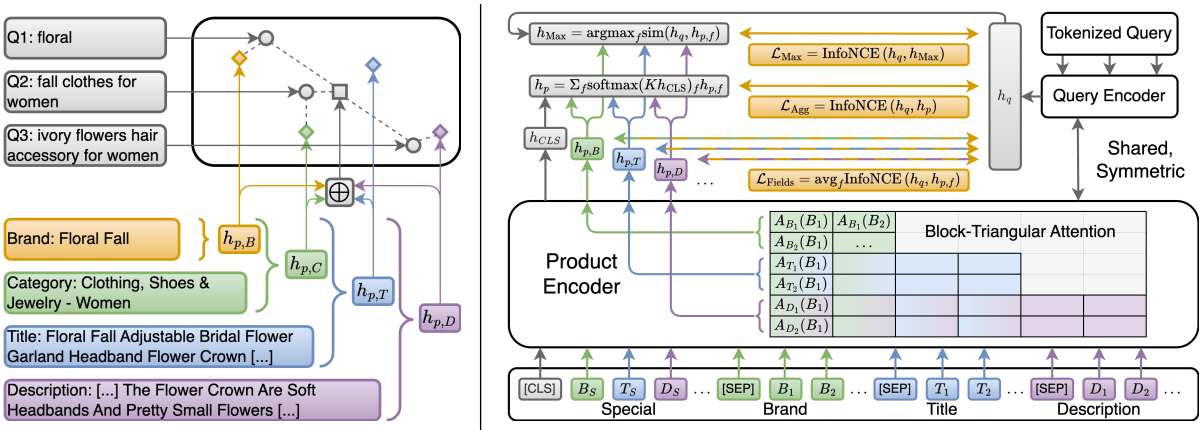
1. Introduction

Online shopping has become ubiquitous, offering customers quick access to a wide range of product options. 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 [1, 2, 3, 4]. With increasing product diversity and user requirements, product retrieval has faced complex challenges including diverse search intents [5], keyword mismatches [6, 7], and scaling to corpora with millions of items [3]. Unlike the extensively explored topic of free-form text retrieval, we focus on e-commerce products represented as semi-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 retrieval strategies that can flexibly leverage different fields. Figure 1a shows an example. While keyword-based methods like TF-IDF [8] and BM25 [9] have been used for decades [10], recent advances have shifted toward dense retrieval [11, 12, 13, 14]. 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, existing work typically treats product fields as unstructured text or uses them only in auxiliary objectives, rather than adapting retrieval to structured input [15, 16, 17].

In this work, we address three research questions. (i) How can structured product item fields be encoded so that retrieval flexibly captures varying levels of product detail? (ii) How can an approach that captures multiple fields balance efficiency with the ability to match queries at different granularities? (iii) Does such an approach outperform standard dense retrieval methods on large-scale e-commerce datasets?

Name of Conference, Month Day–Day, Year, City, Country
niklas.freymuth@kit.edu (N. Freymuth); liuadong@amazon.lu (D. Liu); tricatte@amazon.lu (T. Ricatte); saabm@amazon.com (S. Mansour)
0009-0001-7755-6811 (N. Freymuth)

(a) CHARM's two-stage retrieval.

(b) Structured encoding via block-triangular attention.

Figure 1: CHARM overview. **a)** First, an aggregated product representation (square) is used to perform initial shortlist matching against each query (circle). Matches are then re-evaluated based on the closest cascaded field representation (diamond), 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.

We propose to leverage semi-structured product item data by using field names and their corresponding text directly for dense e-commerce retrieval. We treat product item fields as distinct views of the same product, each offering different levels of detail. When sorted by, e.g., average length, these fields form a 'soft' hierarchy where each new field generally contains new information. A transformer-based model consumes these fields to produce a cascade of field-level representations, where each representation incorporates information from the current and all previous fields. To this end, our Cascading Hierarchical Attention Retrieval Model (CHARM) uses a novel block-triangular attention mechanism to accumulate information hierarchically across fields, producing representations that align with queries at different levels of detail. We find that, for example, short queries often match high-level fields, while longer queries align with detailed fields.

To reduce retrieval cost, we combine this hierarchical approach with 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 [18]. CHARM outperforms common bi-encoder methods [19, 20], including approaches that utilize multiple representations for the same product [21]. Analyzing CHARM, 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) introduce block-triangular attention for efficient multi-field encoding, producing a cascading hierarchy of product item representations. (ii) combine this mechanism with a two-stage retrieval process that balances fast shortlisting with powerful field-level matching. (iii) validate our approach on large-scale e-commerce datasets, showing improvements over strong baselines and analyzing how field-specific representations improve explainability.

2. Related Work

Deep neural networks have significantly advanced information retrieval, beginning with character n-gram vector representations processed by multilayer perceptrons [22]. Transformer models [23], especially BERT [24], have enabled more effective retrieval via latent representations of queries and documents [11, 12, 13, 14]. Leveraging pre-trained Large Language Models (LLMs) [24, 25], these methods support holistic, semantic retrieval [26, 27], significantly outperforming classical techniques like TF-IDF [8] and BM25 [9] when fine-tuned [28], as highlighted in recent surveys [29, 20, 30].

Models such as BiBERT [19, 20] use contrastive training [31, 32] in a dual-encoder setup [33] to align texts by semantic similarity. A large corpus is encoded, and queries are matched to nearest neighbors. Extensions include multitask training [34], query expansion [35], multi-teacher distillation [36], and token-level embeddings [37]. Based on this line of work, dense retrieval has been effective in e-commerce [38, 1], enabling product search [39], click-through rate prediction [40], and ranking [41]. However, most of these works are based on single-vector matching, which comes with inherent theoretical limitations [42], and ignores the rich, multi-field structure of product data.

Recent work uses multi-field learning in retrieval to address these challenges. Here, *MADRAL* [17] 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* [16] 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. [15] address this issue by modeling inter-field dependencies using mutual prediction objectives during an additional Masked Language Modeling (MLM) pre-training phase [43], improving information aggregation across fields. This process boosts downstream contrastive learning [28, 43, 44, 45], 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*)[21] uses a diversity loss to produce distinct views from a single encoder. Multi-View Geometric Index (MVG)[46] 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 [47, 48] help, but require large candidate sets to ensure unique results after de-duplication. Two-stage retrieval [49] mitigates this issue by shortlisting candidates before re-ranking using field-level decompositions. Prior work [50, 51, 28] often treats both stages separately, and even joint training [52] typically uses separate models. Hybrid sparse-dense models like SPLADE [53, 54, 55] 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 [24]. BERT is a transformer-based [23] model that employs multi-head attention [56], 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

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

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 j and i, respectively, and $\mathbf{v}_i \in \mathbb{R}^d$ is the d-dimensional value vector of token i. The attention mask $M_{j,i}$ is set to $M_{j,ij} = 0$ if j is allowed to attend to i, and to $M_{j,i} = -\infty$ 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 [33, 19, 20] to map queries and products into a joint embedding space. Representations are aligned via the InfoNCE loss [57, 58]:

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

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 [59, 11]. 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 [18, 60]. These fields capture different levels of detail, and ordering them by, e.g., average length, naturally yields a soft hierarchy, where later fields generally add new information without being strict supersets.

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. The first vector encodes the first product field, the second combines the first and second fields, and so on. Given a 'soft' hierarchy where no field is an explicit superset of any previous field, this process allows for capturing increasingly expressive field-level representations by integrating the residual information introduced by each new field. Unlike prior work that enforces diversity via loss functions [21], our method, the Cascading Hierarchical Attention Retrieval Model (CHARM), fosters natural diversity by representing each hierarchy level with its own representation. This diverse set of representations offers a dense, structured alternative to shallow field-wise combinations [49].

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.,

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

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, placing a SEP token as the end of each field. For example, given two fields A and B, the input sequence would be $CLS\ SEP\ A_S\ A_1\ A_2\ \ldots\ SEP\ B_S\ B_1\ B_2\ \ldots$, using special tokens A_S and B_S. If a field is empty, its vector is naturally derived from earlier fields and its special token without requiring a change in representation .

From this input, we define the field-level representation for field $f \in \mathcal{F}$ as the output

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

of the field-specific special token f_S. Similar to [16], 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 $K \in \mathbb{R}^{d \times |\mathcal{F}|}$.

Model Inference. 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. We share model weights and match the special tokens to help 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. 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

Method (Evaluation)	US (English)		ES (Spanish)		JP (Japanese)	
	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 [18]. * indicates results taken from [16], using different pre-training and training hyperparameters. + indicates MLM pre-training. Best and second best results are highlighted in **orange** and **teal**, respectively.

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 [21]. 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 [47, 48].

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 Additionally, we match the representations of the individual product fields, i.e., $\mathcal{L}_{\text{Max}} = \text{InfoNCE}\left(h_q, h_{\text{Max}}\right)$ 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

$$\mathcal{L} = \lambda_{\text{Agg}}\mathcal{L}_{\text{Agg}} + \lambda_{\text{Fields}}\mathcal{L}_{\text{Fields}} + \lambda_{\text{Max}}\mathcal{L}_{\text{Max}}. \tag{5}$$

The last two losses naturally lead to diverse solutions due to the block-triangular attention structure, allowing us to omit explicit diversity losses [21]. 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 [18], 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 [15, 16], 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 can be found in Appendix.

Each product includes multiple fields forming a hierarchy of increasingly detailed descriptions, namely "Color", "Brand", "Title", "Description", and "Bullet points". We use this field order unless mentioned otherwise, noting that, in general, each field in this order is likely to contain new information that is not present in any previous field. For the US set, we use an extended version [16] 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*). The Appendix provides detailed hyperparameters and dataset setups.

Baselines. We compare against BERT-based bi-encoder baselines (BERT remains widely used in production due to its favorable latency-accuracy tradeoff). MultiView document Representations (*MVR*) [21] 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 Section 3.2, and the diversity term

$$\mathcal{L}_{\text{Div}} = -\log \frac{e^{f(q,h_{p,\text{Max}})/\tau}}{\sum_f e^{f(q,h_{p,f})/\tau}} \tag{6}$$

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 [19, 20] 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* [16]-*CONCAT* and *MADRAL* [17], as reported in [15]. 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 [28] 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. 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. Further details on the setup and relevant training hyperparameters are available in Appendix.

Ablation Experiments. To isolate the contributions of CHARM, we ablate key components. We assess the impact of individual loss components from Equation 5, 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 [49]. *No MLM* omits the MLM pre-training stage entirely. *Asym. Encoders* replaces the query encoder's softmax-pooled special tokens with a standard CLS token, breaking symmetry with the product encoder. *Separate Encoders* does not

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.				
Asym. Encoders	-0.40	-0.16	-0.29	-0.18
Separate Encoders	-0.82	-0.84	-0.60	-1.20
Other Field Order	-0.25	-0.34	-0.34	-0.58

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.

share weights between product and query 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 [18, 16], 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 (*P*10), 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 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 shortlist, 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.

5.2. Ablation Results

Table 2 reports ablation results for CHARM (*Two-Stage)* on the US dataset. Each loss component in Equation 5 contributes meaningfully, while adding the diversity loss from Equation 6 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

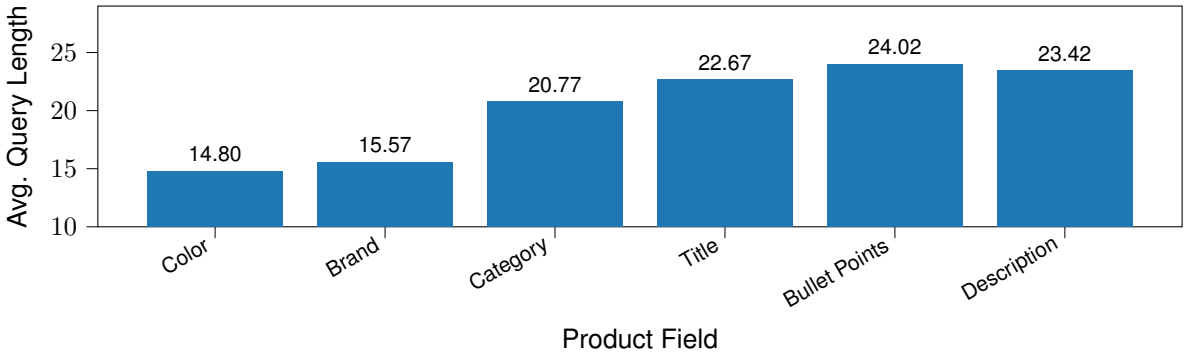


Figure 2: Average length of queries matching a product field by closest dot-product similarity. Product fields that are on a higher hierarchy level generally match longer queries.

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.

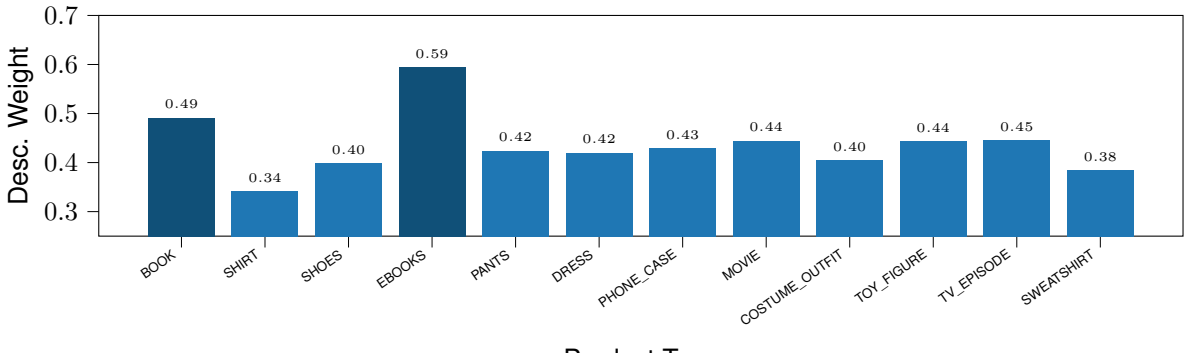
To assess the diversity of field-level representations across the corpus, we compute average pairwise Euclidean distance, dot-product similarity, and the log-determinant of the covariance matrix. As shown in Table 3, fields that appear later in the hierarchy produce more diverse representations, supporting 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 CHARM's robustness and lays the groundwork for explainable search systems that dynamically match important product fields.

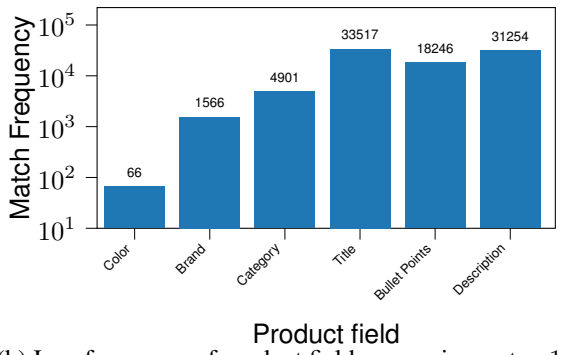
Query-Product Match Analysis. Figure 3b shows how often each product field appears among the top 10 matches for queries in the US dataset. More specific fields appear more frequently, with "Title" being the most common, likely due to its importance and low noise. The results suggest that CHARM often 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

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

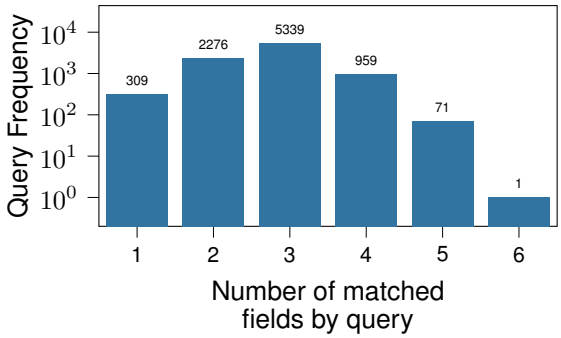
Table 3
Corpus diversity metrics by product field.



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

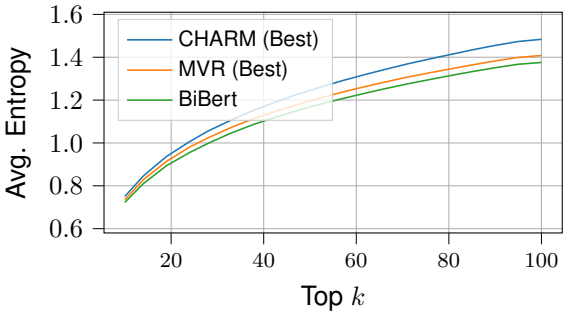


(b) Log-frequency of product fields appearing as top 10 matches for any query.

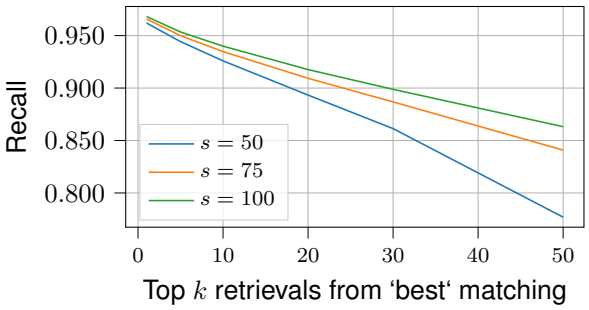


(c) Log-frequency of queries matching a number of product fields in their top 10 matches.

Figure 3: Field relevance and query matching.



(a) Average entropy of product type distributions across different methods and top-k values



(b) Preservation of 'best' matches in two-stage retrieval for different initial shortlist sizes $s \in \{50, 75, 100\}$.

Figure 4: Product type entropy and two-stage retrieval preservation.

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 4a 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 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 4b 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. CHARM's two-stage retrieval process requires a computational overhead that is constant regardless of the underlying query. For simple queries, that, e.g., just look for a certain brand, this process incurs unneccesary cost. Here, 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.

Broader Impact. More effective structured retrieval can shift how products are surfaced in online marketplaces, potentially influencing seller visibility and consumer choice. Such systems can improve discovery and access for consumers, but may also affect market dynamics and thus require increased attention to fairness.

References

[1] A. Muhamed, S. Srinivasan, C.-H. Teo, Q. Cui, B. Zeng, T. Chilimbi, S. Vishwanathan, Web-scale semantic product search with large language models, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2023, pp. 73–85.

[2] N. Rossi, J. Lin, F. Liu, Z. Yang, T. Lee, A. Magnani, C. Liao, Relevance filtering for embedding-based retrieval, in: Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, CIKM '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 4828–4835. URL: https://doi.org/10.1145/3627673.3680095. doi:10.1145/3627673.3680095.

[3] S. Li, F. Lv, R. Zhang, D. Ou, Z. Zhang, M. de Rijke, Text matching indexers in taobao search, in: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 5339–5350. URL: https://doi.org/10.1145/3637528.3671654. doi:10.1145/3637528.3671654.

[4] A. Kekuda, Y. Zhang, A. Udayashankar, Embedding based retrieval for long tail search queries in ecommerce, in: Proceedings of the 18th ACM Conference on Recommender Systems, RecSys '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 771–774. URL: https://doi.org/10.1145/3640457.3688039. doi:10.1145/3640457.3688039.

[5] C. Luo, X. Tang, H. Lu, Y. Xie, H. Liu, Z. Dai, L. Cui, A. Joshi, S. Nag, Y. Li, et al., Exploring query understanding for amazon product search, in: 2024 IEEE International Conference on Big Data (BigData), IEEE, 2024, pp. 2343–2348.

[6] V. Lakshman, C. H. Teo, X. Chu, P. Nigam, A. Patni, P. Maknikar, S. Vishwanathan, Embracing structure in data for billion-scale semantic product search, arXiv preprint arXiv:2110.06125 (2021).

[7] P. Nigam, Y. Song, V. Mohan, V. Lakshman, W. A. Ding, A. Shingavi, C. H. Teo, H. Gu, B. Yin, Semantic product search, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 2876–2885. URL: https://doi.org/10.1145/3292500.3330759. doi:10.1145/3292500.3330759.

[8] G. Salton, C. Buckley, Term-weighting approaches in automatic text retrieval, Information processing & management 24 (1988) 513–523.

[9] S. Robertson, H. Zaragoza, et al., The probabilistic relevance framework: Bm25 and beyond, Foundations and Trends® in Information Retrieval 3 (2009) 333–389.

[10] R. Baeza-Yates, B. Ribeiro-Neto, et al., Modern information retrieval, volume 463, 1999.

[11] V. Karpukhin, B. Oğuz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, W.-t. Yih, Dense passage retrieval for open-domain question answering, arXiv preprint arXiv:2004.04906 (2020).

[12] Y. Li, Z. Liu, C. Xiong, Z. Liu, More robust dense retrieval with contrastive dual learning, in: Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval, 2021, pp. 287–296.

[13] S. Hofstätter, S.-C. Lin, J.-H. Yang, J. Lin, A. Hanbury, 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, 2021, pp. 113–122.

[14] F. M. Nardini, C. Rulli, R. Venturini, Efficient multi-vector dense retrieval with bit vectors, in: European Conference on Information Retrieval, Springer, 2024, pp. 3–17.

[15] X. Sun, K. Bi, J. Guo, X. Ma, Y. Fan, H. Shan, Q. Zhang, Z. Liu, 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, Association for Computing Machinery, New York, NY, USA, 2023, p. 4300–4304. URL: https://doi.org/10.1145/3583780.3615157. doi:10.1145/3583780.3615157.

[16] X. Sun, K. Bi, J. Guo, S. Yang, Q. Zhang, Z. Liu, G. Zhang, X. Cheng, 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, 2024, pp. 674–682.

[17] W. Kong, S. Khadanga, C. Li, S. K. Gupta, M. Zhang, W. Xu, M. Bendersky, Multi-aspect dense

retrieval, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 3178–3186.
[18] C. K. Reddy, L. Màrquez, F. Valero, N. Rao, H. Zaragoza, S. Bandyopadhyay, A. Biswas, A. Xing, K. Subbian, Shopping queries dataset: A large-scale esci benchmark for improving product search, arXiv preprint arXiv:2206.06588 (2022).
[19] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in: Conference on Empirical Methods in Natural Language Processing, 2019. URL: https://api.semanticscholar.org/CorpusID:201646309.
[20] J. Lin, R. Nogueira, A. Yates, Pretrained transformers for text ranking: Bert and beyond, Springer Nature, 2022.
[21] S. Zhang, Y. Liang, M. Gong, D. Jiang, N. Duan, Multi-view document representation learning for open-domain dense retrieval, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 5990–6000. URL: https://aclanthology.org/2022.acl-long.414. doi:10.18653/v1/2022.acl-long.414.
[22] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, L. Heck, Learning deep structured semantic models for web search using clickthrough data, in: Proceedings of the 22nd ACM international conference on Information & Knowledge Management, 2013, pp. 2333–2338.
[23] A. Vaswani, N. M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017. URL: https://arxiv.org/abs/1706.03762.
[24] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), 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), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://aclanthology.org/N19-1423. doi:10.18653/v1/N19-1423.
[25] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, Journal of machine learning research 21 (2020) 1–67.
[26] K. A. Hambarde, H. Proenca, Information retrieval: recent advances and beyond, IEEE Access (2023).
[27] W. X. Zhao, J. Liu, R. Ren, J.-R. Wen, Dense text retrieval based on pretrained language models: A survey, ACM Transactions on Information Systems 42 (2024) 1–60.
[28] Y. Fan, X. Xie, Y. Cai, J. Chen, X. Ma, X. Li, R. Zhang, J. Guo, et al., Pre-training methods in information retrieval, Foundations and Trends® in Information Retrieval 16 (2022) 178–317.
[29] J. Guo, Y. Cai, Y. Fan, F. Sun, R. Zhang, X. Cheng, Semantic models for the first-stage retrieval: A comprehensive review, ACM Trans. Inf. Syst. 40 (2022). URL: https://doi.org/10.1145/3486250. doi:10.1145/3486250.
[30] H. Li, J. Xu, Semantic matching in search, Foundations and Trends® in Information Retrieval 7 (2014) 343–469. URL: http://dx.doi.org/10.1561/1500000035. doi:10.1561/1500000035.
[31] R. Hadsell, S. Chopra, Y. LeCun, Dimensionality reduction by learning an invariant mapping, in: 2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06), volume 2, IEEE, 2006, pp. 1735–1742.
[32] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, F. Makedon, A survey on contrastive self-supervised learning, Technologies 9 (2020) 2.
[33] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, R. Shah, Signature verification using a" siamese" time delay neural network, Advances in neural information processing systems 6 (1993).
[34] A. Abolghasemi, S. Verberne, L. Azzopardi, Improving bert-based query-by-document retrieval with multi-task optimization, in: European Conference on Information Retrieval, Springer, 2022, pp. 3–12.
[35] D. Vishwakarma, S. Kumar, Fine-tuned bert algorithm-based automatic query expansion for

enhancing document retrieval system, Cognitive Computation 17 (2025) 1–16.
[36] S.-C. Lin, A. Asai, M. Li, B. Oguz, J. Lin, Y. Mehdad, W.-t. Yih, X. Chen, How to train your dragon: Diverse augmentation towards generalizable dense retrieval, arXiv preprint arXiv:2302.07452 (2023).
[37] O. Khattab, M. Zaharia, 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, 2020, pp. 39–48.
[38] Y. He, Y. Tian, M. Wang, F. Chen, L. Yu, M. Tang, C. Chen, N. Zhang, B. Kuang, A. Prakash, Que2engage: Embedding-based retrieval for relevant and engaging products at facebook marketplace, in: Companion Proceedings of the ACM Web Conference 2023, 2023, pp. 386–390.
[39] A. Magnani, F. Liu, M. Xie, S. Banerjee, Neural product retrieval at walmart. com, in: Companion Proceedings of The 2019 World Wide Web Conference, 2019, pp. 367–372.
[40] Z. Xiao, L. Yang, W. Jiang, Y. Wei, Y. Hu, H. Wang, Deep multi-interest network for click-through rate prediction, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, CIKM '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 2265–2268. URL: https://doi.org/10.1145/3340531.3412092. doi:10.1145/3340531.3412092.
[41] R. Li, Y. Jiang, W. Yang, G. Tang, S. Wang, C. Ma, W. He, X. Xiong, Y. Xiao, E. Y. Zhao, From semantic retrieval to pairwise ranking: Applying deep learning in e-commerce search, SIGIR'19, Association for Computing Machinery, New York, NY, USA, 2019, p. 1383–1384. URL: https://doi.org/10.1145/3331184.3331434. doi:10.1145/3331184.3331434.
[42] O. Weller, M. Boratko, I. Naim, J. Lee, On the theoretical limitations of embedding-based retrieval, 2025. URL: https://arxiv.org/abs/2508.21038. arXiv:2508.21038.
[43] L. Gao, J. Callan, Condenser: a pre-training architecture for dense retrieval, arXiv preprint arXiv:2104.08253 (2021).
[44] X. Ma, J. Guo, R. Zhang, Y. Fan, X. Cheng, 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, 2022, pp. 848–858.
[45] X. Li, Z. Liu, C. Xiong, S. Yu, Y. Gu, Z. Liu, G. Yu, Structure-aware language model pretraining improves dense retrieval on structured data, 2023. URL: https://arxiv.org/abs/2305.19912. arXiv:2305.19912.
[46] N. Jiang, D. Eswaran, C. H. Teo, Y. Xue, Y. Dattatreya, S. Sanghavi, V. Vishwanathan, On the value of behavioral representations for dense retrieval, arXiv preprint arXiv:2208.05663 (2022).
[47] Sivic, Zisserman, Video google: A text retrieval approach to object matching in videos, in: Proceedings ninth IEEE international conference on computer vision, IEEE, 2003, pp. 1470–1477.
[48] Y. Malkov, D. Yashunin, Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs., IEEE Transactions on Pattern Analysis and Machine Intelligence 42 (2018) 824–836.
[49] M. Li, T. Chen, B. Van Durme, P. Xia, Multi-field adaptive retrieval, arXiv preprint arXiv:2410.20056 (2024).
[50] J. Guo, Y. Cai, Y. Fan, F. Sun, R. Zhang, X. Cheng, Semantic models for the first-stage retrieval: A comprehensive review, ACM Transactions on Information Systems (TOIS) 40 (2022) 1–42.
[51] A. Yates, R. Nogueira, J. Lin, Pretrained transformers for text ranking: BERT and beyond, in: G. Kondrak, K. Bontcheva, D. Gillick (Eds.), Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorials, Association for Computational Linguistics, Online, 2021, pp. 1–4. URL: https://aclanthology.org/2021.naacl-tutorials.1/. doi:10.18653/v1/2021.naacl-tutorials.1.
[52] R. Ren, Y. Qu, J. Liu, W. X. Zhao, Q. She, H. Wu, H. Wang, J.-R. Wen, 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, 2021, pp. 2825–2835.
[53] T. Formal, B. Piwowarski, S. Clinchant, 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, Association for Computing Machinery, New York, NY, USA, 2021, p. 2288–2292. URL: https://doi.org/10.1145/3404835.3463098. doi:10.1145/3404835.3463098.
[54] T. Formal, C. Lassance, B. Piwowarski, S. Clinchant, Splade v2: Sparse lexical and expansion model for information retrieval, arXiv preprint arXiv:2109.10086 (2021).
[55] C. Lassance, S. Clinchant, An efficiency study for splade models, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022, pp. 2220–2226.
[56] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in: International Conference on Learning Representations (ICLR), 2015.
[57] K. Sohn, Improved deep metric learning with multi-class n-pair loss objective, Advances in neural information processing systems 29 (2016).
[58] A. v. d. Oord, Y. Li, O. Vinyals, Representation learning with contrastive predictive coding, arXiv preprint arXiv:1807.03748 (2018).
[59] L. Xiong, C. Xiong, Y. Li, K.-F. Tang, J. Liu, P. N. Bennett, J. Ahmed, A. Overwijk, Approximate nearest neighbor negative contrastive learning for dense text retrieval, in: International Conference on Learning Representations, 2021.
[60] J. Zhou, B. Liu, J. N. Acharya, Y. Hong, K.-c. Lee, M. Wen, Leveraging large language models for enhanced product descriptions in ecommerce, in: The 2023 Conference on Empirical Methods in Natural Language Processing, 2023, p. 88.
[61] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, International Conference on Learning Representations (ICLR) (2015).
[62] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, Advances in neural information processing systems 32 (2019).
[63] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al., Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2020, pp. 38–45.
[64] L. Gao, X. Ma, J. J. Lin, J. Callan, Tevatron: An efficient and flexible toolkit for dense retrieval, ArXiv abs/2203.05765 (2022).
[65] J. Johnson, M. Douze, H. Jégou, Billion-scale similarity search with GPUs, IEEE Transactions on Big Data 7 (2019) 535–547.
[66] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P.-E. Mazaré, M. Lomeli, L. Hosseini, H. Jégou, The faiss library (2024). arXiv:2401.08281.
[67] L. Gao, Y. Zhang, J. Han, J. Callan, Scaling deep contrastive learning batch size under memory limited setup, in: 6th Workshop on Representation Learning for NLP, RepL4NLP 2021, Association for Computational Linguistics (ACL), 2021, pp. 316–321.

A. Block-triangular Attention

Figure 5 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. Each block of this matrix can be seen as a variable-length 'meta-token' that encodes information about its field conditioned on all previous fields.

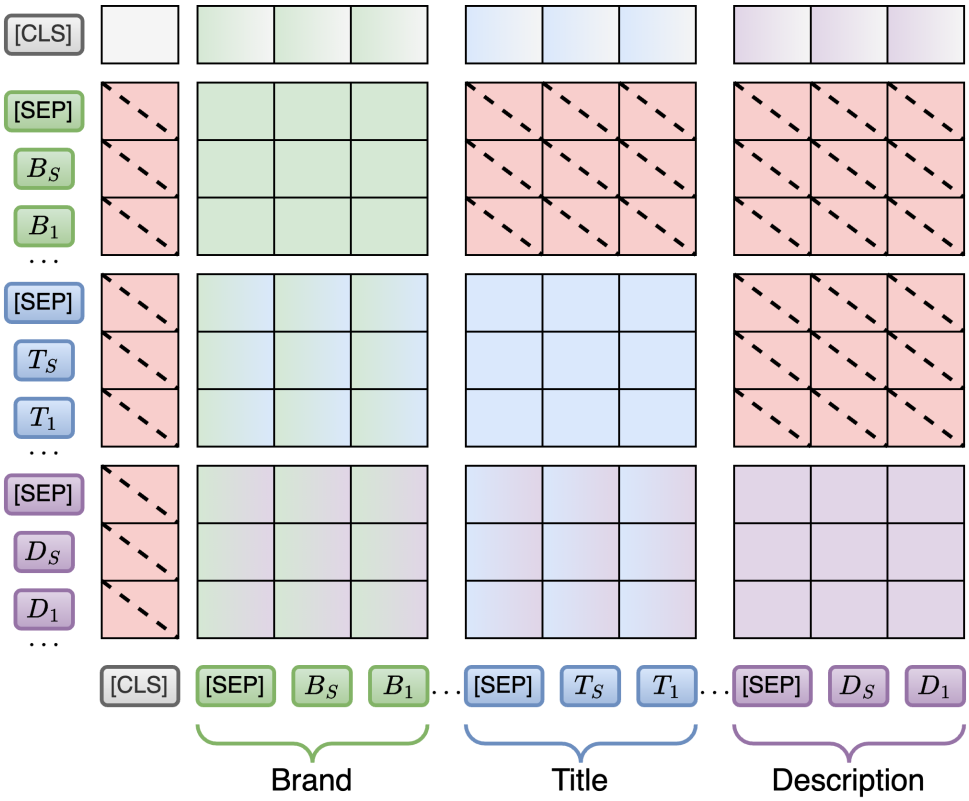


Figure 5: Exemplary block-diagonal attention matrix. Each row (i) represents the attention of one token to all tokens in the sequence, while each column (j) shows which other tokens a token is attended by. The two-colored cells indicate that tokens of one field attend to another field ($M_{i,j} = 0$ in Equation 1 of paper main content). 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 [18]. "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 [61] 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) [24] 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 [62], using the huggingface transformer package [63] and Tevatron [64] for the contrastive training. We perform the retrieval using FAISS-GPU [65, 66] 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 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 [67] to allow for all batch samples to act as in-batch negatives for all other samples.

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 of paper main content, 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

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

	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. 5)	1	0.05	0.05
λ_{Agg} (Eq. 5)		1	
λ_{Max} (Eq. 5)		1	

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

	US (English)			
Method (Evaluation)	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 [16], +: MLM pre-trained.

	ES (Spanish)			
Method (Evaluation)	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.

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.

	JP (Japanese)			
Method (Evaluation)	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.

E. Example Matches

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.

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.