Deep Metric Learning to Hierarchically Rank - An Application in Product Retrieval

Kee Kiat Koo, Ashutosh Joshi, Nishaanth Reddy, Ismail Tutar
Vaclav Petricek, Changhe Yuan, Karim Bouyarmane
Amazon
{kiatkoo,jashutos,nishaant,ismailt,petricek,ychanghe,bouykari}@amazon.com

Abstract

Most e-commerce search engines use customer behavior signals to augment lexical matching and improve search relevance. Many e-commerce companies like Amazon, Alibaba, Ebay etc. operate in multiple countries with country specific stores. However, customer behavior data is sparse in newer stores. To compensate for sparsity of behavioral data in low traffic stores, search engines often use cross-listed products in some form. However, cross-listing across stores is not uniform and in many cases itself sparse. In this paper, we develop a model to identify duplicate and near-duplicate products across stores. Such a model can be used to unify product catalogs worldwide, improve product meta-data or as in our case, use near-duplicate products across multiple to improve search relevance. To capture the product similarity hierarchy, we develop an approach that integrates retrieval and ranking tasks across multiple languages in a single step based on a novel Hierarchical Ranked Multi Similarity (HRMS) Loss that combines Multi-Similarity (MS) loss and Hierarchical Triplet Loss to learn a hierarchical metric space. Our method outperforms strong baselines in terms of catalog coverage and precision of the mappings. We also show via online A/B tests that the product mappings found by our method are successful at improving search quality in low traffic stores, measured in rate of searches with at least one click, significantly by 0.8% and improving cold start product engagement measured as new product clicks significantly by 1.72% in established stores.

1 Introduction

Modern Search Engines utilize two types of information associated with products in their matching and ranking stages: 1) semantic information in terms of product attributes (e.g. title, description, brand, color etc.) provided by the sellers, and 2) behavioral information derived from the customer's interaction with the product (e.g. clicks, adds-to-cart, purchases, reviews & ratings etc.). Behavioral features are critical for both matching and ranking stages and play an important role in improving the quality of search results.

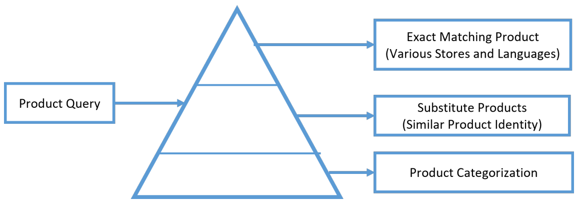
Large amount of customer behavior data is required for building high quality behavioral features. The difference in query volume across high-traffic established stores versus a new store is often staggering, with newer stores receiving orders of magnitude lower traffic. Lower traffic in newer stores lowers the quality of behavioral features and consequently the quality of search results. Also, many major e-commerce companies operate a multilingual search system where each country has a primary language and possibly multiple secondary languages. However, not all secondary languages have rich behavior data, resulting in poorer search quality compared to primary language queries.

One practical way to mitigate this sparsity in large scale commercial Search Engines is Integrative Knowledge Transfer (IKT) (Pan et al., 2008; Zhuo et al., 2008). As part of transfer learning, IKT is a technique used to infuse knowledge from one domain into a different domain similar to feature engineering and data integration methods (Pan, 2014). In e-commerce, IKT is used to synthetically associate behavioral data for a product from higher-resource source store with the cross-listed product in a lower-resource target store. Despite its success, IKT based methods suffer from selection bias: owing to more (transferred) behavioral data, cross-listed products dominate search results while products exclusive to a specific store get pushed down in relevance. This negative transfer can overwhelm local preferences and results in bad customer experience. In this paper, we reduce the selection bias by identifying products, across multiple stores, with near-identical shopping intents and apply IKT techniques to these product mappings. We refer to such products as substitutes.

The problem of identifying substitutes across stores with different languages can be formulated as cross-lingual information retrieval (CLIR). Existing works in CLIR focus on either query/document translation using machine translation (McCarley, 1999; Picchi and Peters, 1998) or expanding queries with translations (Ballesteros and Croft, 1998; Xu and Croft, 2017). Nguyen et al. (2008) uses Wikipedia as source for cross-lingual document sets, Tigrine et al. (2015) constructs cross-lingual ontologies from existing knowledge bases. All these works suffer from poor generalization to new domains such as product search. Most works in CLIR don't use behavioral features which are important for product search. Gupta et al. (2020) addresses some of these challenges for product search by estimating prior values of behavioral features for cold start products. However, their approach is neither cross-locale nor easily extendable to a cross-lingual setting. Ahuja et al. (2020) addresses the problem of product search in multiple languages with limited amount of data per language. They achieve this by learning language independent query and product representations with an end-to-end (query, product) relevance ranking model.

CLIR approaches, however, ignore hierarchical information associated with most product catalogs. Product catalogs of all major e-commerce businesses are organized in a product taxonomy (Karamanolakis et al., 2020). A typical example is 'Shoes->Mens Sports Shoes->Running Shoes'. It follows then that cross-listed products exist in a cross-lingual product hierarchy, even though the taxonomy mapping may not be one-to-one. Product taxonomy or other forms of hierarchy can still be leveraged when retrieving substitute products from catalogs of different stores. Hierarchical ranked loss functions can learn the characteristics of the product taxonomy and score exact cross-listed cross-lingual products higher than substitutes, and substitutes higher than dissimilar products in other categories (Figure 1).

In this paper, we propose an approach that combines retrieval and ranking across multiple languages in a single step. Our solution makes use of product hierarchy by combining Multi-Similarity (MS) loss (Wang et al., 2019b) and Hierarchical Triplet Loss (Ge, 2018) into **Hierarchical Ranked Multi Similarity (HRMS)** Loss. While there exist loss functions which optimize for output rank


Product Query
Exact Matching Product (Various Stores and Languages)
Substitute Products (Similar Product Identity)
Product Categorization
Figure 1: The diversity of a cross-listed cross-lingual multi-store product catalog means that for a given product there exist multiple possible exact and substitutes matches. A learned metric space should reflect the nature of the data set by ranking exact matches closer than substitutes and far from other levels of product categorization.

(Cakir et al., 2019; Wang et al., 2019a) or hierarchical objectives (Yang et al., 2021) independently, none learns both together. Using HRMS, we learn a ranked metric space by generating informative pairs based on the hierarchical nature of the data. By varying the margin based on the hierarchical label, we are able to improve the ranked query output compared to vanilla multi-similarity and ranking based loss functions. We demonstrate the use of HRMS by training a multi-modal cross-lingual model nicknamed ProductSIM based on a hierarchical cross-lingual product catalog. Finally we demonstrate the business use of ProductSIM by using its output to improve Search Results for new stores and products as measured by significant improvement in business metrics like Search-Rate-with-Clicks (percent of searches with at least one clicks) and New Product Impressions and Clicks. The main contribution of this paper is the Hierarchical Ranked Multi Similarity (HRMS) Loss, built by extending Multi Similarity loss for hierarchical ranking in metric space.

2 Method

In this section, we review the challenges of a hierarchical product data set and limitations of proxy based loss functions (Yang et al., 2022) applied to sparse data. We then introduce our Hierarchical Ranked Multi Similarity loss (HRMS), an adaption from Multi Similarly (MS) Loss.

2.1 Preliminaries

We assume a information retrieval setup where given a feature space X, there is a query $q \in X$ and a candidate set $R \subset X$. Our goal is to learn a non-linear mapping function $f(x_i)$ (using a deep neural net) that embeds an instance x_i onto a unit sphere of m-dimensional space. Formally we define the similarity of two items as S_{ij} given as the euclidean distance between $\{f(x_i), f(x_j)\}$ where $S_{ij} = -\|x_i - x_j\|_2$.

2.2 Hierarchically Labelled Data

To perform nearest neighbour retrieval, given q there exist a rank output in R according to their hierarchical labels In order to create this ranked list, each item x_i is assigned a label y_i^l at every level of the hierarchy L. Product data set contains an inherent hierarchy created in terms of an taxonomy (Karamanolakis et al., 2020). For example, using the product hierarchy 'Shoes->Mens Sports Shoes->Running Shoes', shoes classified as Running Shoes and with the same Brand would be considered substitutes. Similarly shoes classified as Mens Sports Shoes and with the same Brand could be through of as second level substitutes, but those classified merely as Shoes would not be considered substitutes. Similarly shoes classified as Mens Sports Shoes and with the same Brand could be through of as second level substitutes, but those classified merely as Shoes would not be considered substitutes.

The goal of HRMS hence for each item x_i is to create a ranked output list by placing items that are equivalent together resulting in $[x_0^0, x_1^0, x_0^1, x_2^1 ... x_i^L]$ where L denotes the number of hierarchical layers available in the data set and x^l denotes items which are hierarchically equivalent at l level. This is distinctive from ranking loss (Cakir et al., 2019) which assumes only a single layer or $L = 1$. In an hierarchical setting each l can have varying sizes and items in each l are to be ranked with an ordering. In this paper this ranked output list is created based on the hierarchical product data set but could be form generally from any data set with a hierarchical taxonomy.

Although R can aggregate to a root node, HRMS makes no assumption on the depth of L. In cases where $L = 1$ (single layer), HRMS will work similar to MS. The only assumption made in the data set is that the items aggregate hierarchically similar to a tree as illustrated in Figure 2. For example, items in the data set could be aggregated under different categories with increasing granularity. The items in the training data are considered weakly supervised as it capture only the relations of item to each other hierarchically.

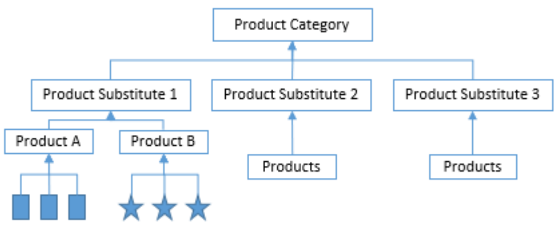


Figure 2: HRMS requires weakly supervised labels and assumes that items within the same layer are equivalent. In this paper we assume that each product can only belong to one product substitute grouping

2.3 Challenges of Hierarchical Product Data In Metric Learning

Challenge 1 - Sparsity of Data: State-of-the-art deep metric learning loss functions typically operate at the pair level or involve training a proxy at the class level. For example Additive Margin Softmax attempts to rank the true translation of each item against all alternatives discounted by a configurable margin (Wang et al., 2018). Proxy-based loss functions are not scalable in a massive cross-lingual hierarchical product data set as the number of proxies to be trained is akin to the number of products in the training data. To work around scalability issues one could down sample the training data, define the hierarchy at a higher level (i.e., product category level) to reduce the number of classes to a manageable level or use a high-powered cluster of machines. The alternatives may not be practically feasible or may not achieve optimal model performance.

There are also challenges using pair-based loss functions for hierarchical data sets. For a fixed number of training samples there is a prohibitively large number of tuples which could be selected. A number of these pairs are also likely to be non-informative which does not contribute to model training (Xuan et al., 2020). To work around issues with proxy and pair based methods, HRMS combines both training paradigms by grouping the data ahead by the lowest level in the hierarchy and using a pair miner to select informative pairs for training. We then shuffle the data each level of the hierarchy and generate a different set of classes per mini batch.

Challenge 2 - Multi Tiered Metric Space: Although proxy-based metric learning loss functions learn class-discriminative features the output may not necessarily be ranked since by design it pushes items which belong to different classes apart (Yang et al., 2022). To create a ranked output HRMS adapts ideas from learning to rank paradigm by creating a margin of separation between positive and negative examples using varying margin (Cakir et al., 2019).

2.4 HRMS - Pair Miner

Similar to MS, HRMS adopts a 2 step approach where informative pairs are first mined from each mini batch. HRMS assumes that the training data as input is grouped by the lowest level in the hierarchy. Through the implementation of an additional pair mining step, MS mines hard negatives and positives by discarding all other records that do not contribute to model training. As described in Wang et al. (2019b), an optimal pair mining method should optimize for self-similarity, negative relative similarity and positive relative similarity. Formally, given ϵ as a margin, y as labels and x_i as anchor, the miner selects pairs $\{x_i, x_j\}$ which meet the following criteria:

$$MSPairMiner(\epsilon, y) = \left\{ \{x_i, x_j\} \mid S_{ij}^{-} > (\min_{y_k = y_i} S_{ik} - \epsilon) \right\} \cup \left\{ \{x_i, x_j\} \mid S_{ij}^{+} < (\max_{y_k \neq y_i} S_{ik} + \epsilon) \right\} \quad (1)$$

where S_{ij}^{-} denotes hard negatives and S_{ij}^{+} hard positives that are identified through the miner.

HRMS adapted the pair mining step in MS at the hierarchical level with varying ϵ_l with y_l reflecting the hierarchy depending on the granularity level. In the product data set example, catalog items which are exact match of each other but in different languages will be considered to be at higher granularity and hence assigned a lower ϵ_l (closer to the leaf node in a tree structure) compared to products at a lower granularity level which are functionality similar but are not substitutable, say of different brand.

$$HRMSPairMiner(\epsilon^{1..L}, y) = \bigcup_{\forall (l \in L)} \left\{ \left\{ \{x_i, x_j\} \mid S_{ij}^{-} > (\min_{y_k^l = y_i^l} S_{ik} - \epsilon^l) \right\} \cup \left\{ \{x_i, x_j\} \mid S_{ij}^{+} < (\max_{y_k^l \neq y_i^l} S_{ik} + \epsilon^l) \right\} \right\} \quad (2)$$

where ϵ^l is a hyper parameter which is inversely proportion to the granularity in the hierarchical tree. Note that y^l varies hierarchically, items which are considered negative at one level may be positives at another as per a tree structure. Since it is possible for pairs to be selected as positive and negative across different levels in L, a deconflicting step is added discarding pairs in the negative set if they are added as positive pairs in other hierarchical levels.

2.5 HRMS - Loss Function

Together with the miner, MS also proposes a pair weighting scheme based on binomial deviance (Yi et al., 2014; Lazic) and lifted structure loss (Song et al., 2015). Binomial deviance is suitable for data with large variance as it is robust to outliers. Specifically the penalty term increases linearly for increasingly negative margin, controlling the impact of outliers on the overall loss. Lifted structure loss attempts to optimize across all pairs in the mini-batch, $O(m^2)$ rather than optimizing across $O(m)$ pairs. It does so by proposing a smooth upper bound on the loss function in a way that does not require mining all pairs within the mini-batch repeatedly. By combining both losses together, MS attempts to weight the pairs more accurately by considering both self similarity (binomial deviance loss) and negative relative similarity (lifted structured loss) in its loss function. MS loss is formulated as:

$$MSLoss(\alpha, \beta, \lambda, P_i, N_i) = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{\alpha} \log[1 + \sum_{k \in P_i} e^{-\alpha(S_{ik} - \lambda)}] + \frac{1}{\beta} \log[1 + \sum_{k \in N_i} e^{\beta(S_{ik} - \lambda)}] \quad (3)$$

where m denotes all training samples filtered by the pair miner, P_i the selected positive pairs and N_i the selected negative pairs. α, β and λ are hyper parameters as in Binomial deviance loss.

Similar to the pair miner, HRMS adapts MS hierarchically varying α and β proportionally to the granularity in the tree. λ is kept constant across different hierarchical levels as it is an offset value that is applied equally to both positive and negative terms in binomial deviance loss. HRMS is aggregated hierarchically and back propagates through all levels in L per mini-batch.

$$HRMSLoss(\alpha^{1..L}, \beta^{1..L}, \lambda, P_i^{1..L}, N_i^{1..L}) =$$

$$\sum_{l \in L} \frac{1}{m^l} \sum_{i=1}^{m^l} \left\{ \frac{1}{\alpha^l} \log[1 + \sum_{k \in P_i^l} e^{-\alpha^l (S_{ik} - \lambda)}] \right\} +$$

$$\left\{ \frac{1}{\beta^l} \log[1 + \sum_{k \in N_i^l} e^{\beta^l (S_{ik} - \lambda)}] \right\} \quad (4)$$

where m^l denotes the pairs selected by the pair miner, P_i^l the selected positive pairs and N_i^l the selected negative pairs for the specific hierarchical level l

3 Data Corpus

The model is trained using a sample of products from the worldwide Amazon catalog. Since a product can be cross-listed across multiple stores and in different languages, we use the term *catalog item* to reference a product which is represented in a specific language (e.g., English, France), *product* referring to a item which consist of multiple catalog items (same product across multiple stores and languages) and *product family* which consist of a set of items that are of the same product identity, but differ according to a consistent dimension, for example the same t-shirt in different sizes and colors. Since it is not possible to annotate substitute products at Amazon scale, to train our network, we consider products within the same product family as *substitutes*.

The training data contains 14m product families with an average of 6 unique languages each and 8.5 catalog items per product (A product may contain more than 1 catalog item that is in the same language). Each product family contain an average of 8.7 products. In all, the data set contains over 1 billion catalog items across 22 text and image attributes. These fields are displayed to customers when they view the product on the website.

4 Product DML Model - ProductSIM

ProductSIM is a light weight multi-modal DML model whose goal is to output a trained set of embeddings such that similar products are positioned close together in Euclidean space. ProductSIM is built using Language-Agnostic BERT Sentence Enbedding (LaBSE), Shift Vision Transformers (ShiftViT) as language and image encoders respectively. The embeddings from respective encoders

	Lurrose Criss Cross Headbands Dots Headbands Wide Headwraps Bohemian Headbands Yoga Cotton Hair Bands Turban Head Wraps for Women Girls (Black)	Product 1
	Lurrose Criss Cross Stirnbänder Punkte Stirnbänder Wide Headwraps Bohemia Stirnbänder Yoga Baumwolle Haarbänder Turban Head Wraps für Damen Mädchen (schwarz)	
	Lurrose Criss Cross diademas lunares diademas anchas para la cabeza bohemia diademas de algodón para yoga bandas para el pelo turbante para mujer niña (negro)	
	Lurrose Criss Cross Headbands Dots Headbands Wide Headwraps Bohemia Headbands Yoga Cotton Hair Bands Turban Head Wraps for Woman Girl (White)	Product 2
	Lurrose Criss Serre-tête croisé à pois larges bandeaux bohème en coton pour femme fille (blanc)	
	Lurrose Criss Cross Pannband Prickar Pannband Breda Huvuddukar Bohemia Pannband Yoga Bomull Hårband Turban Huvudsjal för Kvinna Flicka (Vit)	
	Lurrose Criss Cross Headbands Dots Headbands Wide Headbands Bohemia Headbands Yoga Cotton Hairbands Turban Headwraps for Woman Girl (Purple)	Product 3
	Lurrose Criss Cross Hoofdbanden Dots Hoofdbanden Brede Hoofdbanden Bohemen Hoofdbanden Yoga Katoen Haarbanden Tulband Hoofdwraps voor Vrouw Meisje (Paars)	
	Lurrose Criss Cross Stirnbänder Punkte Stirnbänder Wide Headwraps Bohemia Stirnbänder Yoga Baumwolle Haarbänder Turban Head Wraps für Damen Mädchen (lila)	

Figure 3: In this illustration, each product exists in 3 different languages. All 3 products collectively belong to the same product family as they are identical and differ only in a single attribute (color).

are then fed through a fusion encoder to capture inter-modality interaction in a late fusion manner (Baltrusaitis et al., 2019). Details on ProductSIM can be found in Appendix A.

5 Offline Evaluations

We validate ProductSIM on a holdout set from the training data. We select the holdout set such that each product and its substitutes exist in at least two languages across multiple stores. The catalog items from the same products are exact matches and should be ranked highest in the retrieved ranked list. Different products within a single product family are considered substitutes and should be ranked lower than exact matches. All other products are irrelevant and should not be retrieved. We selected an evaluation test set of 33k product families (substitutes) containing an average of 6.7 products in 11.2 languages on average per family. Each product contains an average of 3.7 catalog items.

For a given catalog item of a product, we generate the ProductSIM embedding and retrieve the top-k neighbors by Euclidean distance. Our goal is to retrieve all the other catalog items of the input product in the top-k neighbors ranked such that exact matches are ranked ahead of substitutes. We use the embeddings from the test set and built a flat FAISS index (Johnson et al., 2017) for the purpose of performing exhaustive search for each query.

In order to measure the ranked nature of the results, we adopted the implementation of weighted nDCG similar to Reddy et al. (2022). In our eval-

Loss Function	**Recall**	**nDCG**
Multi Similarity Loss (Wang et al., 2019b)	<u>77.38</u>	<u>84.33</u>
Triplet Loss with Easy Positive Mining (Xuan et al., 2020)	68.71	83.77
Supervised Constrastive Learning (Khosla et al., 2020)	76.03	83.88
Proxy-NCA (Movshovitz-Attias et al., 2017)	77.2	83.47
HRMS (Ours)	**79.68**	**86.27**

Table 1: Loss Function Ablation: We evaluate the efficacy of HRMS on an holdout set from the data corpus. All results are reported using ProductSIM as backbone. Bold denotes best results and underline second best

uation data set we have 2 degrees of relevance for each query: Exact (catalog items) and Substitute (product family), and we set gain values of 1.0, 0.25 respectively. We report results for both Recall (where both exact and substitutes are treated as positives) and nDCG. As illustrated in Table 1, HRMS loss achieves best performance against other loss functions in both nDCG and recall.

6 Impact on Search Results

All search engines use behavioral data to match and rank products to queries. Low traffic stores and new products lack such behavioral data and thus suffer from less relevant search results. We used Integrative Knowledge Transfer (IKT) (Pan et al., 2008) to boost the behavioral data associated with the new stores and products by synthetically transferring query-product associations from high-traffic stores to low-traffic stores. In addition to cross-listed products we used product substitutes identified by ProductSIM to identify *like* products to transfer behavior. The following experiments show the impact of expanded IKT coverage on search results.

Low-Traffic Store: We used ProductSIM to map products from a low-traffic store to those in multiple high-traffic stores. We then boosted the behavior associated with the products in the low-traffic store by using the behavior associated with the mapped products from the high-traffic store(s). We did this by treating the traffic associated with a product in the established store as if it occurred in the low-traffic store albeit in the context of the product identified by ProductSIM. In our experiment, we used a store where roughly 50% of the query traffic was in English and 50% in a single non-English language. Using ProductSIM, we boosted the behavior associated with 21% of the active catalog products, thereby boosting their rank in the search results. This improved customer engagement, in an online A/B test using as measured by Search click rate defined as number of searches with at least one click over total number of searches by 0.8% (p-value = 0.026)

New Products in a High-Traffic Store: Cold start is a known problem in product search (Han et al., 2022). This is specially true in high-traffic stores where established products dominate search impressions. We used ProductSIM to map new products in a high-traffic store to established products in the same store when possible, and a different store otherwise. Boosting behavior associated with the new products using the mapping, we were able to increase customer engagement with these products. In particular in an online A/B test, we boosted New Product Impressions by 1.76% (p-value= 0.0) and New Product Clicks by 1.72% (p-value = 0.0). The strong correlation between impressions and clicks also tells us that we boosted products that customers desired.

7 Related Work

Hierarchical vs Ranking Metric Loss Functions: While hierarchical and ranking loss functions are similar in pulling similar items together and dissimilar items apart, there are important differences in its mechanism:

1. Ranking loss optimise the total ordering of objects as induced by the learned metric. They typically requires a ranked list of example where given a query item there exist an inherit position in its output (Cakir et al., 2019).

2. The goal of hierarchical loss functions is to learn an adaptive class structure such that it encodes global context on a manifold sphere. Hierarchical loss functions are used to guide triplet samples generation for each mini-batch such that they are informative for learning. (Ge, 2018)

More recently, attempts are made to learn hierarchical metric learning representation by adapting proxy-based methods (Yang et al., 2022). While such methods work well for dense data sets (small number of classes and large number of samples per class), it is challenging to scale to granular settings during training (e.g., E-Commerce Product Data Sets), where classes could be defined at the product level and a small number of examples are available per class. Further existing methods do not optimize for the ranked output hierarchically instead focus either on recall or mean average precision (Musgrave et al., 2020)

In this paper we combine ideas from both paradigm to propose HRMS. By mirroring the characteristics of a real life hierarchical product data set and without the complexity of a Graph Neural Network, we show that it is possible to induce a model to learn both retrieval and ranking simultaneously.

8 Conclusion and Future Work

In this paper we propose a metric learning loss function which is suitable for use on hierarchical data sets similar to Amazon catalog. We extend the existing multi similarity loss with adaptive margin for a hierarchical data set. Hierarchical Ranked Multi Similarity Loss (HRMS) works by optimizing for ranked retrieval instead of multi-similarity or ranked loss individually. Future work could consider the multi-aspect dimension of similarity (Kong et al., 2022). Similarity is context specific, for example one could look for products with similar brand.

References

Aman Ahuja, Nikhil Rao, Sumeet Katariya, Karthik Subbian, and Chandan K Reddy. 2020. Language-agnostic representation learning for product search on e-commerce platforms. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 7–15.

Lisa Ballesteros and W Bruce Croft. 1998. Resolving ambiguity for cross-language retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 64–71.

Tadas Baltrusaitis, Chaitanya Ahuja, and Louis-Philippe Morency. 2019. Multimodal machine learning: A survey and taxonomy. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(2):423–443.

Fatih Cakir, Kun He, Xide Xia, Brian Kulis, and Stan Sclaroff. 2019. Deep metric learning to rank. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1861–1870.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding.

Weifeng Ge. 2018. Deep metric learning with hierarchical triplet loss. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 269–285.

Parth Gupta, Tommaso Dreossi, Jan Bakus, Yu-Hsiang Lin, and Vamsi Salaka. 2020. Treating cold start in product search by priors. In *Companion Proceedings of the Web Conference 2020*, pages 77–78.

Cuize Han, Pablo Castells, Parth Gupta, Xu Xu, and Vamsi Salaka. 2022. Addressing cold start in product search via empirical bayes. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 3141–3151.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*.

Giannis Karamanolakis, Jun Ma, and Xin Luna Dong. 2020. Txtract: Taxonomy-aware knowledge extraction for thousands of product categories. *arXiv preprint arXiv:2004.13852*.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc.

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*, KDD '22, page 3178–3186, New York, NY, USA. Association for Computing Machinery.

SE Lazic. The elements of statistical learning: Data mining, inference, and prediction, 2nd edn.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9).

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. *CoRR*, abs/2103.14030.

J Scott McCarley. 1999. Should we translate the documents or the queries in cross-language information retrieval? In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 208–214.

Yair Movshovitz-Attias, Alexander Toshev, Thomas K. Leung, Sergey Ioffe, and Saurabh Singh. 2017. No fuss distance metric learning using proxies.

Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. 2020. A metric learning reality check.

Dong Nguyen, Arnold Overwijk, Claudia Hauff, Dolf RB Trieschnigg, Djoerd Hiemstra, and Franciska De Jong. 2008. Wikitranslate: query translation for cross-lingual information retrieval using only wikipedia. In *Workshop of the Cross-Language Evaluation Forum for European Languages*, pages 58–65. Springer.

Sinno Jialin Pan, Dou Shen, Qiang Yang, and James T Kwok. 2008. Transferring localization models across space. In *AAAI*, pages 1383–1388.

Weike Pan. 2014. Collaborative recommendation with auxiliary data: A transfer learning view. *CoRR*, abs/1407.2919.

Eugenio Picchi and Carol Peters. 1998. Cross-language information retrieval: A system for comparable corpus querying. In *Cross-language information retrieval*, pages 81–92. Springer.

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.

Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. 2015. Deep metric learning via lifted structured feature embedding. *CoRR*, abs/1511.06452.

Abdel Nasser Tigrine, Zohra Bellahsene, and Konstantin Todorov. 2015. Light-weight cross-lingual ontology matching with lyam++. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 527–544. Springer.

Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, et al. 2021. Mlp-mixer: An all-mlp architecture for vision. *arXiv preprint arXiv:2105.01601*.

Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. 2018. Additive margin softmax for face verification. *IEEE Signal Processing Letters*, 25(7):926–930.

Guangting Wang, Yucheng Zhao, Chuanxin Tang, Chong Luo, and Wenjun Zeng. 2022. When shift operation meets vision transformer: An extremely simple alternative to attention mechanism. *CoRR*, abs/2201.10801.

Xinshao Wang, Yang Hua, Elyor Kodirov, Guosheng Hu, Romain Garnier, and Neil M. Robertson. 2019a. Ranked list loss for deep metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R. Scott. 2019b. Multi-similarity loss with general pair weighting for deep metric learning. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5017–5025.

Jinxi Xu and W Bruce Croft. 2017. Quary expansion using local and global document analysis. In *Acm sigir forum*, volume 51, pages 168–175. ACM New York, NY, USA.

Hong Xuan, Abby Stylianou, and Robert Pless. 2020. Improved embeddings with easy positive triplet mining. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2474–2482.

Zhibo Yang, Muhammet Bastan, Xinliang Zhu, Douglas Gray, and Dimitris Samaras. 2021. Hierarchical proxy-based loss for deep metric learning. *CoRR*, abs/2103.13538.

Zhibo Yang, Muhammet Bastan, Xinliang Zhu, Douglas Gray, and Dimitris Samaras. 2022. Hierarchical proxy-based loss for deep metric learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1859–1868.

Dong Yi, Zhen Lei, and Stan Z. Li. 2014. Deep metric learning for practical person re-identification. *CoRR*, abs/1407.4979.

Hankz Hankui Zhuo, Qiang Yang, Derek Hu, and Lei Li. 2008. Transferring knowledge from another domain for learning action models. volume 5351, pages 1110–1115.

A ProductSIM Model Architecture

ProductSIM is a light weight multi-modal DML model whose goal is to output a trained set of embeddings such that similar products are positioned close together in Euclidean space. ProductSIM is built using Language-Agnostic BERT Sentence Enbedding (LaBSE), Shift Vision Transformers (ShiftViT) as language and image encoders respectively. The embeddings from respective encoders are then fed through a fusion encoder to capture inter-modality interaction in a late fusion manner (Baltrusaitis et al., 2019).

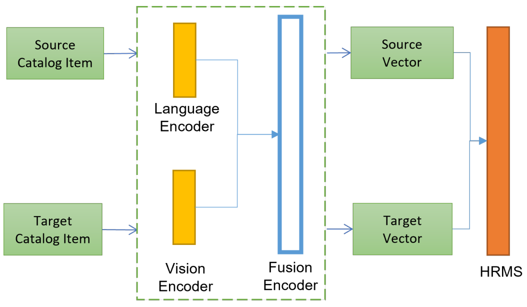


Figure 4: ProductSIM is built using LaBSE, ShiftViT as language and image encoders respectively. The output of both encoders are than fed through a series of fully connected layers (as shown using unfilled blue rectangular boxes).

Language Encoder: We used the pretrained LaBSE (Feng et al., 2020) that is trained on data from CommonCrawl and Wikipedia as base model. Language-Agnostic BERT Sentence Enbedding (LaBSE) follows the setup of a Bidirectional Encoder Representations from Transformers (BERT) model which uses 12 layers transformer with 12 heads and 768 hidden size. Similar to prompt based learning methods (Liu et al., 2023), we feed both structured and unstructured attributes to the language encoder by suffixing the attribute value with attribute name.

Vision Encoder: Multi layer perceptron (MLP) based vision models recently gained popularity in being able to achieve competitive results with higher throughput then compared to vision transformers (Tolstikhin et al., 2021). Considering the need for efficiency and frugality at scale, ProductSIM utilizes the smallest available (ShiftViT-Tiny) ImageNet pretrained ShiftViT as the vision embedding model (Wang et al., 2022). ShiftViT follows Swin Transformers to build hierarchical representation but replaces the attention mechanism with a shift operation (Liu et al., 2021). The zero-parameter shift operation moves a portion of input channels along four directions to model spatial relationships in images while keeping other layers untouched. The shifted output are then parsed through a series of feed forward layers to fuse the channels together.

Fusion Encoder: Since ShiftViT forms hierarchical and not token representations of an image, we did not opt for an transformer-based fusion encoder approach. Instead, ProductSIM concatenates both normalized image and language representations by feeding them into a series of fully-connected layers.