Transferable and Efficient: Unifying Dynamic Multi-Domain Product Categorization

Shansan Gong[*1], **Zelin Zhou**[*2], **Shuo Wang**[3], **Fengjiao Chen**[4], **Xiujie Song**[5], **Xuezhi Cao**[6], **Yunsen Xian**[7], **Kenny Q. Zhu**[†8]
[1,2,5,8]Shanghai Jiao Tong University, Shanghai, China
[3,4,6,7]Meituan Inc. China
[1]gongshansan@sjtu.edu.cn, [2]ze-lin@sjtu.edu.cn,
[3]wangshuo81@meituan.com, [4]chenfengjiao02@meituan.com, [5]xiujiesong@sjtu.edu.cn,
[6]caoxuezhi@meituan.com, [7]xianyunsen@meituan.com, [8]kzhu@cs.sjtu.edu.cn

Abstract

As e-commerce platforms develop different business lines, a special but challenging product categorization scenario emerges, where there are multiple domain-specific category taxonomies and each of them evolves dynamically over time. In order to unify the categorization process and ensure efficiency, we propose a two-stage taxonomy-agnostic framework that relies solely on calculating the semantic relatedness between product titles and category names in the vector space. To further enhance domain transferability and better exploit cross-domain data, we design two plug-in modules: a heuristic mapping scorer and a pretrained contrastive ranking module with the help of "meta concepts", which represent keyword knowledge shared across domains. Comprehensive offline experiments show that our method outperforms strong baselines on three dynamic multi-domain product categorization (**DMPC**) tasks, and online experiments reconfirm its efficacy with a 5% increase on seasonal purchase revenue. Related datasets are released[1].

1 Introduction

Product categorization (Ding et al., 2002) is a specialized text classification task that classifies product titles or descriptions into a pre-defined taxonomy of categories. As businesses expand, major e-commerce platforms (*e.g.,* Amazon and Alibaba) are encountering increasingly complex scenarios, where there are multiple domain-specific category taxonomies and each of them evolves dynamically over time. We define it as **D***ynamic* **M***ulti-Domain* **P***roduct* **C***ategorization* (**DMPC**), which simultaneously considers the following **multi-domain taxonomies** and **taxonomy evolving** challenges.

In real-world businesses, e-commerce platforms usually maintain **multiple** business lines with relatively independent **taxonomies**. These business lines are catering for different customer demands or specific domain applications, for example, one provides express delivery while another specializes in low-price bargains. Multiple business domains correspond to different category taxonomy structures, with various depths and distinct literal expressions of category names. Conventional industry approaches train separate classifiers on each domain, which under-utilize the cross-domain data and their shared knowledge while raising the expenses of maintenance. Meanwhile, with the expansion and reorganization of businesses, each category **taxonomy** keeps **evolving** as well, where old categories might be deleted or integrated and new categories are possibly added. Conventional multi-class classifiers need to be re-trained every time taxonomy changes, which disrupts the operation and further diminishes the maintenance efficiency.

To mitigate **taxonomy evolving** issues, intuitively, we reformulate the canonical text classification problem as a text relevance matching problem. Moreover, to ensure both accuracy and online efficiency, we propose a two-stage *Taxonomy-agnostic Label Retrieval* (TaLR) framework (see Figure 1) capturing semantic similarity between a product title and its corresponding category names in the vector space, where candidate categories are first retrieved and then reranked for the final prediction. This reformulation is especially beneficial for evolving and newly added (zero-shot) categories as

[*] Equal contribution.
[†] Corresponding author.
[1]Datasets associated with this paper are released at https://github.com/ze-lin/TaLR.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 5: Industry Track, pages 476–486
July 10-12, 2023 ©2023 Association for Computational Linguistics

textual semantics are incorporated.

To leverage cross-domain data in **multi-domain taxonomies** challenge, we devise two plug-in modules in both stages to enhance TaLR's domain transferability. These modules are centralized with "meta concepts" that appear in the product titles, which represent fine-grained keyword knowledge shared across domains (Appendix B). As is shown in Figure 1, in the *retrieval* stage, besides the dense retrieval based on vector similarity (dense scorer), the statistical co-occurrence probability between meta concepts and category labels are exploited as well (mapping scorer) . In the *reranking* stage, meta concepts are incorporated with category labels as supervision signals for the contrastive pretraining of the scoring model (matching scorer). While the mapping scorer complements the superficial semantic dense retrieval with cross-domain commonsense knowledge, contrastive pretraining directly optimizes the vector space improving inter-domain alignment and uniformity. Details are given in Section 2.2 and Section 2.4.

In summary, our contributions are: (1) For the first time, we address the **DMPC** problem and release the corresponding multi-domain datasets in Chinese. (2) We propose a unified TaLR framework equipped with two well-designed plug-in modules empowered with meta concepts, which is robust and efficient against the two challenges in **DMPC** problem. (3) Offline experiments on our annotated real-world **DMPC** datasets show TaLR's ability to effectively transfer knowledge across domains and generalize to new domains. The unified TaLR outperforms three separately-trained SOTA classifiers by 1.65% on overall accuracy and maintains satisfactory accuracy in taxonomy evolving conditions. Online experiments reaffirm its efficacy with a 5% increase in seasonal purchase revenue.

2 Proposed Framework

$\forall i \in [1, n]$ domains, given a taxonomy G_i with depth of d_i and m leaf nodes, the path from root to leaf node forms the text which is regarded as hierarchical category label $y_i^{(j)} (j \in [1, m])$. For an input product title X_i along with its meta concept labels $\{\lambda_k\}$, our task is to output the correct category label it belongs to. Note that only one leaf category will be the correct answer. Detailed task formulation refers to Appendix A.

Our TaLR framework is structured into two stages: *Retrieval* and *Reranking*, as illustrated in Figure 1. We will zoom into each component of this framework.

2.1 Dense Scorer

We first train a dual-encoder to represent both categories and product titles in the vector space.

Negative sampling In the original text classification problem, each product title X_i has exactly one positive category label y_i. However in our reformulation, text relevance matching models need negative category labels during training, otherwise they would not succesfully converge. For each (X_i, y_i) pair, we prepare to construct the training examples $\mathcal{S}$ from multiple taxonomies by sampling $(N-1)$ negative categories. Instead of randomly chosen, "hard" negative examples are more informative for better convergence. Inspired by teacher-student paradigm (Hinton et al., 2015), we adopt a teacher classifier-based sampling strategy to sample strong negative categories for dual-encoder learning.

For each training dataset S_i of taxonomy G_i, we split it in k-fold manner, then take turns to train k BERT classifiers on every $\frac{k-1}{k}$ data , with the remain $\frac{1}{k}$ data as the development set. The m-class classifiers are optimized with the typical m-class cross-entropy loss. The k classifiers would inference $(N-1)$ most possible but not correct category labels concurrently in their corresponding development sets, and their results with ground truth positive labels constitutes the point-wise training set for the following dual-encoder training.

Dual-encoder training We adopt a siamese network architecture (Reimers and Gurevych, 2019) where the encoder respectively extracts the fixed-sized embeddings of product titles X_i and category names $\hat{y_i}$ which are denoted as $\mathbf{u}_x$ and $\mathbf{v}_y$. To better align the embedding of $\mathbf{u}_x$ and $\mathbf{v}_y$, we use Circle Loss (Sun et al., 2020) which allows each similarity score to optimize at its own pace. We simplify it as:

$$\mathcal{L} = \log \left(1 + \sum_{S} e^{\alpha(cos(\mathbf{u}_x^+, \mathbf{v}_y^+) - cos(\mathbf{u}_x^-, \mathbf{v}_y^-))} \right), \tag{1}$$

where α is the hyper-parameter, and $+, -$ denotes the positive and negative samples in $\mathcal{S}$ respectively. We also compare this loss function with other alternatives in Appendix D.1.

Candidates retrieval We can quickly derive relevant category label embeddings given an incoming

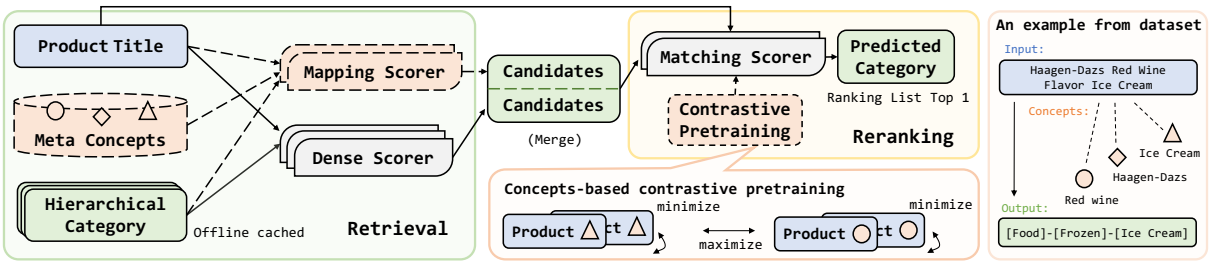


Figure 1: An overview of TaLR framework, containing *Retrieval* and *Reranking* stages. We show an example from our released dataset, in which the input is a product title with its meta concepts, and the output is its corresponding hierarchical category. In *Retrieval* stage, two lists of category candidates are sampled from mapping scorer and dense scorer. In the *Reranking* stage, merged category candidates are ranked by a matching scorer with contrastive information. Dark dashes refer to plug-in modules.

product title embedding, with one-vs-all similarity measurement like cosine-similarity implemented by Approximate Nearest Neighbor (ANN) techniques targeting time efficiency. Based on this, we can readily collect top-k candidate list C_{vec}.

2.2 Mapping Scorer

Dense scorer usually prioritizes semantic relatedness of literal expressions, neglecting the commonsense co-occurrence probability that lies within cross-domain training data. For example, *"Sunrise Roses 500g"* is often recognized as [Flower] by semantic matching algorithms, however, it is actually a variety of [Grape]. Therefore we introduce a mapping scorer in *Retrieval* stage capturing such commonsense knowledge to complement the above dense-retrieved candidates.

Mapping algorithm The shared meta concept set $\mathcal{M}$ is constructed by hybrid NER-related techniques. Details are in Appendix B. We can regard "meta concept" as a kind of keyword knowledge because they usually contain very concrete and accurate information. In our released datasets, one product title X is tagged with one or more meta concepts $\Lambda = \{\lambda_1, \lambda_2, ...\lambda_k\}$ from $\mathcal{M}$. For example, *"Haagen-Dazs Red Wine Flavor Ice Cream"* is tagged with ⟨RedWine⟩, ⟨Icecream⟩, ⟨HaagenDazs⟩ as meta concepts.

Given product title X and a category label $\hat{y}$, our heuristic strategy establishes $X \rightarrow \hat{y}$ mapping as conditional co-occurrence probability $P(\hat{y}|X)$. First, we model this conditional probability for each category $\hat{y}$ as:

$$\begin{aligned} P(\hat{y}|X) &= P(\hat{y}|\lambda_1, \lambda_2, ...\lambda_k) \\ &= \max_{1 \le i \le k} P(\hat{y}|\lambda_i). \end{aligned} \tag{2}$$

Here we aggregate $P(\hat{y}|\lambda_1, \lambda_2, ...\lambda_k)$ with the maximum value among multiple λ_i referring to the same $\hat{y}$. Each $P(\hat{y}|\lambda_i)$ is collected from training data distributions:

$$P(\hat{y}|\lambda_i) = \frac{P(\hat{y}, \lambda_i)}{P(\lambda_i)} = \frac{\nu(\hat{y}, \lambda_i)}{\nu(\lambda_i)}, \tag{3}$$

where ν denotes the frequency in training data. Then, we collect candidate list C_{rule} by empirically setting a threshold of $P(\hat{y}|X) > 0.5$ to ensure both retrieval quantity and quality.

Candidates merging When retrieved candidates from the dense scorer and mapping scorer are prepared, we need to combine the two lists of candidates. Our concept-first strategy prioritizes candidates from C_{rule}. It puts at most 10 top candidates (usually less than 10) from C_{rule} into C_{union}, then keeps filling it with top candidates from C_{vec} until its size reaches 10.

2.3 Matching Scorer

To further measure the relatedness of product titles and category names with mutual interactions, we train a matching scorer in *Reranking* stage. During training, given a product title X and its retrieved candidates $C_{union} = \{c_1, c_2, ...c_l\}$, we concatenate tokenized sequences of X and each of these $c_i \in C_{union}$ with a [SEP] token as the input to BERT-based model. The ground truth label is 1 if c_i is the correct candidate otherwise 0. Optimization is followed with binary cross-entropy loss. During inference, the model gives similarity scores for each (X, c_i) pair, and the candidate with the highest similarity score would be our predicted category.

2.4 Contrastive Pretraining

For **multi-domain taxonomies**, category classes vary from one taxonomy to another. Despite the

assorted expressions of category classes among different domain taxonomies, we find their fine-grained concepts of products seldom shift. While previous retrieval stage pursues the recall of candidates and focuses less on class discrimination, the cross-encoder in *Reranking* stage possibly suffers from indistinguishable categories. Inspired by the supervised derivative of contrastive learning (Wang et al., 2021), we restrict the formation of positive pairs ensuring they not only share the same category class with X but also have at least one meta concept in common with X, otherwise they would be considered negative. This setting is tailored for the **multi-domain taxonomies** challenge pursuing cross-domain alignment and uniformity, where inter-concept semantics are tied closer and intra-concept ones are further distinguished.

Given a product title X with label y and tagged meta concept set Λ, we encode X as vector $\mathbf{u}$ and group encoded product titles as positive vector samples $\{\mathbf{v}_1^{y,\Lambda_1}, \mathbf{v}_2^{y,\Lambda_2}, ..., \mathbf{v}_D^{y,\Lambda_D}\}$, which are labeled with the same y and share an overlapped concept set Λ_d with Λ. We use BERT as the encoder backbone and tune its parameters with group contrast loss:

$$\mathcal{L}^{GC} = -\frac{1}{D}\sum_{d=1}^{D}\log\frac{\exp(\mathbf{u}\cdot\mathbf{v}_d^{y,\Lambda_d}/\tau)}{Pos+Neg}.$$
$$Pos = \sum_{d=1}^{D}\exp(\mathbf{u}\cdot\mathbf{v}_d^{y,\Lambda_d}/\tau), \quad (4)$$
$$Neg = \sum_{y',\Lambda'}^{D}\exp(\mathbf{u}\cdot\mathbf{v}^{y',\Lambda'}/\tau),$$

where y', Λ' denotes samples with either different label y' with y or non-overlapping meta concept set Λ' with Λ. The BERT model after contrastive pretraining can be used in matching scorer during *Reranking* stage in Section 2.3.

3 Dynamic Multi-Domain Datasets

3.1 Static Multi-domain Datasets

We select 3 business lines from our e-commerce platform: QuickDelivery (QD, targeting fast delivery), BargainHunters (BH, targeting low price), FreshGrocery (FG, targeting fresh vegetables). These data instances are collected from the real-world business, where the product titles are mostly assigned by sellers from the platform and the category labels stem from three pre-defined business taxonomies. We recruit experienced annotators to manually classify the products X_i into assorted categories y_i, with 1% sampling to guarantee annotation accuracy. Data groups with over 95% accuracy in quality checking are used in our final datasets. Meanwhile, X_i is tagged with concepts $\{\lambda_k\}$ following the Appendix B.

Table 1: Statistics for multi-domain datasets

Dataset	# training	# test	# classes	depth
QD	99k	11k	1987	3
BH	31k	5k	2632	4
FG	28k	3k	1065	4

[1] # classes: the total distinct leaf nodes.
[2] depth: the depth of categorical taxonomy tree.

Statistics of three datasets are listed in Table 1. Each sample in the three datasets has exactly one ground truth category. Varied class numbers and hierarchy depths of different taxonomies pose bigger challenges for multi-domain knowledge sharing.

3.2 Dynamic Test Set

To verify the generalizability of TaLR on zero-shot scenarios, we further construct two **taxonomy evolving** derivatives of the QD test set. (ii) QD-*integrate*: During a production business adjustment, 127 classes in the original taxonomy are integrated or replaced by similar categories, which affects 1371 samples in the original test set to form this subset. (i) QD-*divide*: 22 category nodes from the original QD taxonomy are divided into two or more nodes. 495 samples in the original test set suffer from this evolution.

3.3 Meta Concept Set

Beyond the category labels, each product title is associated with a list of meta concepts from a set $\mathcal{M}$ including over $30k$ entities covering the most fine-grained concepts in product titles. The tagging step $X \rightarrow \{\lambda_1, \lambda_2, ...\lambda_k\}$ is accomplished by an industrial Label Tagging System that exploits heterogeneous approaches. Details are in Appendix B.

4 Experiments

In this section, we discuss experimental results under static multi-domain settings and dynamic (taxonomy evolving & new taxonomy) conditions. A brief comparison of time efficiency between TaLR and simple *Reranking* is also included.

4.1 Baselines

We implement several baseline methods based on single-domain, multi-domain, and dynamic scenarios. To ensure fair comparisons, we also experiment concatenating product titles with meta concept text as input for some competitive baselines. Note that all the strong baselines are practicable in our online production environment, and those with unbearable space or time complexity are not considered. Works holding different assumptions (e.g. necessitate multi-label or not support Chinese) with us are not considered either. Finally, we deploy and benchmark the following common baselines:

Flat Classifier **TF-IDF&LR** represents product titles with TF-IDF weighted dense vectors, and executes classification with Logistic Regression. **FastText** (Bojanowski et al., 2017) is a common baseline adopted in online product categorization challenges. **BERT** classifier is used as the strong baseline in both single-domain and multi-domain (trained with multi-task learning) settings.

Hierarchical Classifier **HMCN** (Wehrmann et al., 2018) and **HiMatch** (Chen et al., 2021) leverage hierarchical information from taxonomy to guide the classification process, and we use BERT as a text encoder in both approaches. **XR-Linear** and **XR-Transformer** are two derivatives of PECOS (Yu et al., 2022) framework for extreme classification, which achieve competitive performance in most open product categorization datasets.

4.2 Experimental Setup

We mix up training data from three datasets to train the unified TaLR. We use **accuracy** score as the evaluation metric to meet real-world business demands. Accuracy mathematically equals to **Micro-F1** score in a single-label multi-class classification problem. More details can be found in Appendix C.

4.3 Overall Results

The overall accuracy score is shown in Table 2. Since traditional single-domain approaches cannot tackle **multi-domain taxonomies**, we train **separate** models on each business respectively. Among methods targeting one static taxonomy, hierarchical classifiers generally perform better than flat classifiers with the aid of taxonomy structure information. However, because these methods can only handle one static taxonomy, they not only suffer from efforts to maintain different models for each domain but also fail to leverage multi-domain data. While the multi-task BERT is able to train and infer on three domains within one model, it performs even worse than TF-IDF&LR on BH and FG. One possible reason is that the multi-task approach relies heavily on the weighting of losses, and if the task-specific training data distribution varies significantly, one task might dominate the joint distribution and constrain the optimization of other tasks. Simply concatenating meta concepts to titles does not always take effect, and this is expected since concatenated tokens implicitly contribute to the joint representation of one sentence (e.g. self-attention in transformer), which proves to be inferior to our explicit usage of statistical mapping and contrastive grouping.

For our proposed framework TaLR, variant (a) already outperforms other baselines in separate model training paradigm, while TaLR (b) further achieves even higher accuracy when jointly trained on the mixed multi-domain data where the multi-task BERT fails, verifying TaLR's efficacy on **multi-domain taxonomies**. We assume that the

Table 2: The accuracy of baselines and our TaLR framework with variants on static multi-domain datasets. The best results are **bolded**, and the best baseline results are starred. Overall accuracy is the weighted average w.r.t respective test set size. MS: mapping scorer, CL: contrastive learning.

Methods	Overall	QD	BH	FG
Separate models				
TF-IDF&LR	69.51	69.93	68.23	69.95
FastText	74.62	74.01	71.68	80.82
BERT	83.49	84.82	79.93*	84.23
BERT+♠	83.01	86.45	79.02	75.32
HMCN-F-BERT	82.14	83.72	77.09	84.25
HiMatch-BERT	84.08	86.12	77.38	84.19
HiMatch-BERT+♠	83.75	87.26*	77.26	78.53
XR-Linear	76.57	75.27	77.91	78.95
XR-Transformer	84.58*	79.74	79.23	84.58*
XR-Transformer+♠	81.45	85.34	74.59	78.53
(a): TaLR	85.90	87.88	81.92	85.09
Unified model				
BERT Multi-task	68.00	80.27	50.28	44.29
BERT Multi-task+♠	67.79	81.37	49.77	39.83
(b): TaLR	**86.23**	**88.16**	**82.48**	**85.25**
TaLR ablation test				
(c): (b) (-) CL	85.26	86.83	81.75	85.13
(d): (b) (-) MS	84.63	86.59	80.13	84.71
(e): (b) (-) CL&MS	82.82	83.85	79.15	84.71
(f): (b) (-) CL&MS +♠	84.38	87.43	80.64	79.77

♠ concatenate concept text after product title
(-) ablate cretain modules

measurement of semantic relatedness is transferable on either business domain, and their shared knowledge could be integrated via contrastive pretraining as well. Therefore, the unified training helps improving the performance on each respective domain instead of conflicting each other as BERT multi-task does.

From the ablation tests, we can observe the effectiveness of the two plug-in modules in our TaLR framework from row (c) and (d), and the contribution of these two modules are orthogonal. Removing the mapping scorer in (d) drops the overall accuracy most, while removing contrastive pretraining in (c) results in its inferior performance than (a) as well. This indicates both modules are indispensable for the enhancement of exploiting multi-domain data. From (e)→(f), concatenating meta concepts somehow improves the overall performance, but (f) still loses to (b). This reaffirms our above assumption that our usage of meta concepts is superior to simple concatenation. To further analyze the effects of the two plug-in modules, we conduct Case Study in Appendix D.2.

4.4 Time Consumption

To meet online deployment requirement, the inference time consumption (seconds cost for each instance) needs to be considered. We compare TaLR with the vanilla model (single BERT cross-encoder) on the three datasets in Figure 2. On the one hand, the inference speed of TaLR is much faster (4 times faster for FG and 10 times faster for BH) than vanilla model owing to the *Retrieval* stage. On the other hand, the time consumption per item of TaLR increases almost linearly along with the number of classes, while for vanilla model the overhead grows more sharply, revealing the time efficiency of TaLR when the class number scales up.

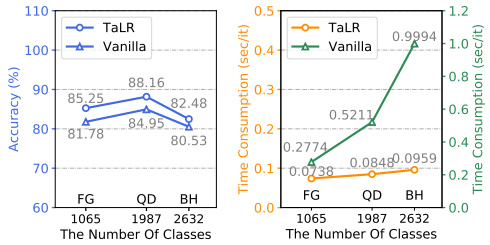


Figure 2: Accuracy results and inference time consumption when the number of classes grows.

4.5 Dynamic Test Set Experiment

In order to evaluate the ability of our framework on **taxonomy evolving** challenge, we use TaLR trained on the original multi-domain datasets to directly infer on two dynamic test sets. The vanilla BERT without any finetuning is a naive baseline **BERT-matching**. The BERT fine-tuned with few-shot new data (1%) is a strong baseline **BERT-few-shot**. Here "before" denotes the subset from the original test set and "after" denotes the subset with the same product titles but evolved categories. From the listed accuracy "before" and "after" taxonomy evolving in Table 3, we can conclude that TaLR sustains satisfactory accuracy compared with its strong counterpart trained with 1% extra data.

Table 3: The accuracy on two dynamic test sets. Δ is the change of accuracy after evolving. The best "after" scores and least drop Δ are bolded.

Methods	QD−*divide*			QD−*integrate*		
	Before	After	Δ	Before	After	Δ
BERT-matching	6.66	11.95	**+5.29**	13.39	2.23	-11.16
BERT-few-shot	90.51	43.54	-46.96	86.79	50.16	-36.53
TaLR	90.11	**69.71**	-20.40	85.20	**81.48**	**-3.72**

4.6 Extrapolating Results on New Taxonomy

Consider an extreme **taxonomy evolving** condition when a new business line emerges, a robust model is supposed to categorize incoming products based on the brand-new taxonomy.

Table 4: The accuracy of TaLR on the new taxonomy.

Methods	QD	BH	FG
BERT-matching	9.00	11.23	4.03
BERT-few-shot	43.29	35.19	29.80
TaLR	**60.57**	**65.45**	**62.69**
(-) contrastive	56.71	64.99	60.79
(-) mapping scorer	56.25	64.65	59.29

We deploy our experiments in a zero-shot manner, where we take turns to train TaLR on either two business data and test its performance on the remaining business. TaLR still outperforms BERT-few-shot. This shows TaLR's preeminent transferability with the reformulation of textual semantic matching, which helps improving user experience in this cold-start scenario. Each component in the ablation test verifies its effectiveness as well.

4.7 Online Experiment

We conduct online experiments on one downstream task where TaLR's domain-independent category recognition ability helps transfer user preferences from other domains and contributes to a more accurate recommendation. When TaLR is incorporated in the recommendation system, customer seasonal purchase revenue increases significantly over 5%.

5 Conclusion

To tackle **DMPC** problem, we propose a unified TaLR framework with two plug-in modules empowered with cross-domain meta concepts. With comprehensive experiments on real-world **DMPC** datasets, results under both multi-domain and taxonomy evolving conditions exhibit the transferability and maintenance efficiency of TaLR.

Acknowledgments

This work was generously supported by the Meituan-SJTU joint research grant.

References

Rohit Babbar and Bernhard Schölkopf. 2017. Dismec: Distributed sparse machines for extreme multi-label classification. In *Proceedings of the tenth ACM international conference on web search and data mining*, pages 721–729.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.

Haibin Chen, Qianli Ma, Zhenxi Lin, and Jiangyue Yan. 2021. Hierarchy-aware label semantics matching network for hierarchical text classification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4370–4379.

Hongshen Chen, Jiashu Zhao, and Dawei Yin. 2019. Fine-grained product categorization in e-commerce. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2349–2352.

Pradipto Das, Yandi Xia, Aaron Levine, Giuseppe Di Fabbrizio, and Ankur Datta. 2016. Large-scale taxonomy categorization for noisy product listings. In *2016 IEEE international conference on big data (big data)*, pages 3885–3894. IEEE.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ying Ding, M Korotkiy, Borys Omelayenko, V Kartseva, V Zykov, Michel Klein, Ellen Schulten, and Dieter Fensel. 2002. Goldenbullet: Automated classification of product data in e-commerce. In *Proceedings of the 5th international conference on business information systems*. Citeseer.

Xin Luna Dong, Xiang He, Andrey Kan, Xian Li, Yan Liang, Jun Ma, Yifan Ethan Xu, Chenwei Zhang, Tong Zhao, Gabriel Blanco Saldana, et al. 2020. Autoknow: Self-driving knowledge collection for products of thousands of types. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2724–2734.

Jung-Woo Ha, Hyuna Pyo, and Jeonghee Kim. 2016. Large-scale item categorization in e-commerce using multiple recurrent neural networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 107–115.

Idan Hasson, Slava Novgorodov, Gilad Fuchs, and Yoni Acriche. 2021. Category recognition in e-commerce using sequence-to-sequence hierarchical classification. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 902–905.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Abhinandan Krishnan and Abilash Amarthaluri. 2019. Large scale product categorization using structured and unstructured attributes. *arXiv preprint arXiv:1903.04254*.

Younghun Lee, Lei Chen, Yandi Xia, and Wei-Te Chen. 2020. Cbb-fe, camembert and bit feature extraction for multimodal product classification and retrieval.

Maggie Yundi Li, Stanley Kok, and Liling Tan. 2018. Don't classify, translate: Multi-level e-commerce product categorization via machine translation. *arXiv preprint arXiv:1812.05774*.

Xusheng Luo, Luxin Liu, Yonghua Yang, Le Bo, Yuanpeng Cao, Jinghang Wu, Qiang Li, Keping Yang, and Kenny Q Zhu. 2020. Alicoco: Alibaba e-commerce cognitive concept net. In *Proceedings of the 2020 ACM SIGMOD international conference on management of data*, pages 313–327.

Ioannis Partalas, Aris Kosmopoulos, Nicolas Baskiotis, Thierry Artieres, George Paliouras, Eric Gaussier, Ion Androutsopoulos, Massih-Reza Amini, and Patrick Galinari. 2015. Lshtc: A benchmark for large-scale text classification. *arXiv preprint arXiv:1503.08581*.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. 2020. Circle loss: A unified perspective of pair similarity optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6398–6407.

Ximei Wang, Jinghan Gao, Mingsheng Long, and Jianmin Wang. 2021. Self-tuning for data-efficient deep learning. In *International Conference on Machine Learning*, pages 10738–10748. PMLR.

Jonatas Wehrmann, Ricardo Cerri, and Rodrigo Barros. 2018. Hierarchical multi-label classification networks. In *International conference on machine learning*, pages 5075–5084. PMLR.

Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. 2020. Product knowledge graph embedding for e-commerce. In *Proceedings of the 13th international conference on web search and data mining*, pages 672–680.

Hu Xu, Bing Liu, Lei Shu, and P Yu. 2019. Open-world learning and application to product classification. In *The World Wide Web Conference*, pages 3413–3419.

Li Yang, E Shijia, Shiyao Xu, and Yang Xiang. 2020. Bert with dynamic masked softmax and pseudo labeling for hierarchical product classification. In *MWPD@ ISWC*.

Hsiang-Fu Yu, Chia-Hua Ho, Prakash Arunachalam, Manas Somaiya, and Chih-Jen Lin. 2012. Product title classification versus text classification. *Csie. Ntu. Edu. Tw*, pages 1–25.

Hsiang-Fu Yu, Kai Zhong, Jiong Zhang, Wei-Cheng Chang, and Inderjit S Dhillon. 2022. Pecos: Prediction for enormous and correlated output spaces. *Journal of Machine Learning Research*.

A Dynamic Multi-Domain Problem

We clarify the **DMPC** problem as follows. Given a set $\mathcal{G}$ of n relatively independent label taxonomies at initial time t_0

$$\{G_1, G_2, G_3, ..., G_n\},$$

each of which correlates with a domain-specific product categorization task. The taxonomy of product categories G_i is tree-structured with depth d_i, and it contains m_i category leaf nodes:

$$\{y_i^{(1)}, y_i^{(2)}, y_i^{(3)}, ..., y_i^{(m_i)}\} \subseteq G_i.$$

Part of the nodes is enrolling in a dynamic trending. As time goes $t_{>0}$, the category node $y_i^{(a)}$ of a certain product might be **divided** into two categories $y_i^{(a1)}$ and $y_i^{(a2)}$ or **integrated** with another category $y_i^{(b)}$ to form $y_i^{(ab)}$. The **emergence** of a new category node $y_i^{(m+1)}$ with corresponding product titles is also possible. In addition, an emerging taxonomy G_{n+1} may sprout when a new business is cultivated.

A single product categorization task on taxonomy G_i ($i = 1$) is a traditional classification task, in which the training data and test data are organized in tuples

$$\mathcal{S} = \{(X_i^{(1)}, y_i^{(1)}), ..., (X_i^{(m_i)}, y_i^{(m_i)}), ...\}.$$

Each X_i in $\mathcal{S}$ represents the title of one product and y_i is the corresponding class node in the categorical taxonomy tree.

In **DMPC** problem, when $i \geq 2$, to unify the training data and the inference procedure cross G_i, we reformulate classification as the matching between X_i and y_i. While traditional classifiers regard y_i as meaningless label ordinals, we instead treat them along the path of top-bottom taxonomy nodes equivalently with the product title as free text. In this reformulated text semantic similarity matching task, the data samples are:

$$\mathcal{S}_i = \{(X_i^{(1)}, y_i^{(1)}, Y_{\mathbb{1}}^{(1)}), ..., (X_i^{(m_i)}, y_i^{(m_i)}, Y_{\mathbb{1}}^{(m_i)}), ...\},$$

$$\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_i\},$$

where $Y_{\mathbb{1}} \in \{0, 1\}$ is an indicator denoting whether the text pair X_i and y_i is matched ($Y_{\mathbb{1}} = 1$) or not ($Y_{\mathbb{1}} = 0$).

B Details of Meta Concept Set Construction and Tagging

Meta concepts are fine-grained tags that have been widely used in industrial knowledge graphs (e.g. Amazon (Dong et al., 2020), Walmart (Xu et al., 2020), Alibaba (Luo et al., 2020)). Details of meta concept construction and tagging are listed below. We will use "concept" instead of "meta concept" for brevity.

B.1 Concept Set Construction

Concept set construction is conducted in a semi-supervised manner. First, we use a domain-specific named entity recognition (NER) model to mine fine-grained entities from product titles. These entities are complemented with queries from search engine and cumulated knowledge from experts to form the initial pool of concepts. Based on that, we use a naive classifier to pick-up high-quality concepts with high search frequency or broad product coverage. Then, manual annotation is performed on the remaining 20k entities, achieving 95% accuracy in quality checking. Finally, we collect over 30k concepts covering the most fine-grained knowledge in product titles.

B.2 Concept Tagging

Concept tagging is comprised of two stages.

The first stage is concept recall. In order to find candidate concepts for each product, we adopt three approaches: NER, knowledge reduction and semantic recall. First, seed candidates are found by NER on product titles. Second, we extend seed candidates with their neighbors in commonsense knowledge graphs, such as synonyms and brand-concept relations (some brands sell specific products). Third, for those products without seed candidates, we use Sentence-BERT to retrieve concepts by textual semantics. The low-quality concepts recalled will be filtered in the next stage, i.e. concept classification.

The second stage is concept classification. Based on the candidates collected in the previos stage, we train a binary classifier to filter out concepts which attain low relevance score with product titles. The classifier is fine-tuned with knowledge integration which will be introduced in our successive work.

C Implementation Details

For fair comparisons, all the "BERT" abbreviations mentioned in this work are Google BERT-base pre-trained on Chinese corpus. For TF-IDF and FastText baselines, We use jieba [2] toolkit to generate Chinese word segments and tune hyper-parameters on each dataset respectively. BERT-related models are initialized from the pretrained Google BERT-base (Chinese) and tuned with 2e-5 learning rate, 512 batch size, 32 sequence length, except that the cross-encoder BERT in *Reranking* stage extends the sequence length to 64. All BERT related appoaches are trained 40 epochs while multi-task baseline trained at most 120 epochs.

Table 5: Examples from the three Datasets

Product title	Taxonomy path
QD	
Towel gourd 1 pcs & soy bean 150g	Vegetable → Mixed Product → Vegetables mixture
Concepts:{⟨soy bean⟩, ⟨towel gourd⟩}	
BH	
Fresh bamboo shoots (dig from mountains)	Vegetable/Fruit → Vegetable → Tubers → Bamboo
Concepts:{⟨bamboo shoot⟩, ⟨native product⟩}	
FG	
Butter leaf lettuce 100g	Fresh → Vegetable → Leaf → Lettuce
Concepts:{⟨lettuce⟩, ⟨butter lettuce⟩}	

D Experiment Analysis

D.1 Details of Dense Scorer

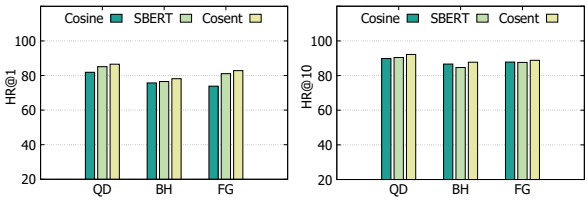
In *Retrieval* stage, it is encouraged to exploit the potential candidates as accurately as possible, otherwise the latter *Reranking* stage would never make right predictions if the true label is not covered by the retrieved candidates. Hence we use HR@k to measure the retrieval performance.

We compare several alternatives of the loss function for Dense Scorer, specifically, different approaches for $(\mathbf{u}_x, \mathbf{v}_y)$ similarity measurement. The loss used in Eq. (1) is termed as Cosent loss[3]. Besides this, one straightforward method is to compute the cosine similarity between vector $\mathbf{u}_x$ and $\mathbf{v}_y$ and optimize the model using vanilla binary cross entropy loss.

$$\delta = cos(\mathbf{u}_x, \mathbf{v}_y) = \frac{< \mathbf{u}_x, \mathbf{v}_y >}{||\mathbf{u}_x||\,||\mathbf{v}_y||}, \qquad (5)$$

[2] https://github.com/fxsjy/jieba

[3] This name is after https://kexue.fm/archives/8847



(a) HR@1 (b) HR@10

Figure 3: The retrieval results of the vector-based unit over different loss functions.

$$\mathcal{L}^{bce} = -\sum_{S} Y_{\mathbb{1}} \log(\delta) + (1 - Y_{\mathbb{1}}) \log(1 - \delta), \quad (6)$$

where $Y_{\mathbb{1}}$ is the binary class. For the sake of the alignment between embedding $\mathbf{u}_x$ and $\mathbf{v}_y$, we also refer to the classification objective function in SBERT (Reimers and Gurevych, 2019).

$$o = softmax(W_o(\mathbf{u}_x, \mathbf{v}_y, |\mathbf{u}_x - \mathbf{v}_y|)), \quad (7)$$

where $W_o \in \mathbb{R}^{3l \times 2}$ is the weighting parameter to project the concatenation of $\mathbf{u}_x$, $\mathbf{v}_y$ and the element-wise difference $|\mathbf{u}_x - \mathbf{v}_y|$ to binary classes. l is the dimension of embeddings. The second element in vector o can be regarded as the probability whether $\mathbf{u}_x$ and $\mathbf{v}_y$ are matched or not, hence we can adopt the same binary cross entropy loss function in Eq. (6) to optimize the model.

In Figure 3, as k goes on, the HR score increases, and the model trained with Cosent loss is consistently better than others, while the model trained with SBERT loss performs unstably, sometimes worse than Cosine loss. One explanation is that comparing with Cosine loss and SBERT loss, the Cosent loss focuses on the positive-versus-negative pairwise optimization, which means the model only cares for the relative order of the prediction results instead of the specific value. And this setting brings consistent recall of candidates.

D.2 Case Study

For product *"New Farmer® walnut flavored sunflower seed 160g"* which should be categorized into [`Sunflower Seed`], TaLR without contrastive learning wrongly assign it to [`Walnuts`]; When concept "sunflower seed" is incorporated in contrastive pretraining, TaLR is capable of distinguishing the right answer. For product *"CELSIUS® cola flavored 300ml"* which should belong to [`Sports Drink`], TaLR without mapping scorer wrongly label it as [`Cola`]; When concept "CELSIUS®" is engaged in retrieval, TaLR could finally sort out the answer.

E Related Work

E.1 Large-Scale Taxonomy Classification

Text classification with a large hierarchy of classes attracts attention and has been studied with the evolving of LSHTC (Partalas et al., 2015) Challenge, which includes over 12000 categories. DiSMEC (Babbar and Schölkopf, 2017) devises one-vs-all linear classifiers with explicit control of model size. HMCN (Wehrmann et al., 2018) discovers hierarchical information by jointly optimizing local and global loss functions. HiMatch (Chen et al., 2021) encodes the complex structure of the label hierarchy as well as the input text, to capture the text-label semantics relationship. PECOS (Yu et al., 2022) ranks output classes with hierarchical clustering, and the semantics of categories are incorporated as well. These methods assume that label taxonomies are stable, neglecting that taxonomy evolves gradually.

E.2 Product Categorization

Product categorization is a hierarchical text classification task assigning categories to product instances.

Approaches in early times are centralized with text features and basic machine learning algorithms. (Ding et al., 2002) introduces KNN and Naive Bayes to the field of product categorization, while (Yu et al., 2012) conducts experiments using TF-IDF with an SVM classifier. Restricted by the bag-of-words paradigm, these methods lack the ability to represent text with contextual semantics.

Neural network based methods prevail since 2013. (Ha et al., 2016) proposes an end-to-end deep learning model composed of multiple RNNs and fully-connected layers, which exhibits a significant advantage over traditional bag-of-words approaches. (Das et al., 2016) conducts a comparison between linear, CNN, and gradient boosting models. Multi-CNN and multi-LSTM are applied in (Krishnan and Amarthaluri, 2019) combining structured and unstructured attributes of products. (Chen et al., 2019) utilizes several convolutional approaches for a better representation of words, and they further adopt literal matching between product content and category label texts to deal with new categories. However, they do not consider the more complicated category *divide* situation.

Recent studies follow the pretrain-finetune paradigm since the great success of BERT (Devlin et al., 2019). (Lee et al., 2020) uses the Camem-

BERT pretrained on French corpus as text encoder in SIGIR 2020 Challenge. (Yang et al., 2020) exploits BERT with a dynamic masking strategy and achieves first place on the 2020 Semantic Web Challenge.

Apart from end-to-end classification approaches, (Hasson et al., 2021; Li et al., 2018) adopts hierarchical Seq2seq models for product categorization. Nonetheless, their models need to be re-trained whenever category taxonomy vocabulary changes.

E.3 Incremental Learning

Class incremental learning resolves the problem that the classes increase progressively in a stream, and the classifier should continuously learn the incoming classes while sustaining accuracy on the seen classes as well. iCaRL (Rebuffi et al., 2017) is proposed to circumvent the catastrophic forgetting problem by storing the information of previous classes. (Xu et al., 2019) extends incremental learning as an open-world learning problem, where a model rejects unseen classes instead of assigning them into the seen class vocabulary. However, in their open-world learning setting, other taxonomy evolving situations (like split and merge) and multi-domain taxonomies are not taken into consideration.