HierCat: Hierarchical Query Categorization from Weakly Supervised Data at Facebook Marketplace

Yunzhong He[1], Cong Zhang[1], Ruoyan Kong[2], Chaitanya Kulkarni[1], Qing Liu[1], Ashish Gandhe[1], Amit Nithianandan[1], Arul Prakash[1]
[1]Meta, [2]University of Minnesota
United States
{yunzhong,conzhang,chaitanya2,qingl,ashigan,anithian,arulprakash}@meta.com
kong0135@umn.edu

ABSTRACT

Query categorization at customer-to-customer e-commerce platforms like Facebook Marketplace is challenging due to the vagueness of search intent, noise in real-world data, and imbalanced training data across languages. Its deployment also needs to consider challenges in scalability and downstream integration in order to translate modeling advances into better search result relevance. In this paper we present HierCat, the query categorization system at Facebook Marketplace. HierCat addresses these challenges by leveraging multi-task pre-training of dual-encoder architectures with a hierarchical inference step to effectively learn from weakly supervised training data mined from searcher engagement. We show that HierCat not only outperforms popular methods in offline experiments, but also leads to 1.4% improvement in NDCG and 4.3% increase in searcher engagement at Facebook Marketplace Search in two weeks of online A/B testing.

CCS CONCEPTS

• **Information systems** → **Query intent**; **Query representation**;
• **Applied computing** → **Online shopping**.

KEYWORDS

Query understanding, e-commerce, information retrieval

ACM Reference Format:
Yunzhong He[1], Cong Zhang[1], Ruoyan Kong[2], Chaitanya Kulkarni[1], Qing Liu[1], Ashish Gandhe[1], Amit Nithianandan[1], Arul Prakash[1]. 2023. HierCat: Hierarchical Query Categorization from Weakly Supervised Data at Facebook Marketplace. In *Companion Proceedings of the ACM Web Conference 2023 (WWW '23 Companion), April 30-May 4, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3543873.3584622

1 INTRODUCTION

Query categorization refers to the task of mapping a search query to a predefined product taxonomy, which usually consists of a large label space organized in a hierarchical structure. It helps e-commerce search engines to understand users' shopping intents in a structured way to provide high quality search experiences. On customer-to-customer (C2C) shopping platforms like Facebook Marketplace[1], query categorization can be particularly challenging due to the following observations:

Query	Category path
iPhone cases	Electronics//Cell Phones//Accessories//Cases
2019 brz black	Vehicles//Cars & Trucks//Coupes
1b1b for rent	Housing//Property Rentals

Table 1: Popular queries on Marketplace and their categories

Noisy training data: Training query categorization models based on <query, product> engagement is a common approach to avoid needing massive amount of labeled data [1, 10, 14, 15]. However, due to how product categories are labeled and the browsy behavior of user, training data mined from search engagement can be heterogeneous and noisy.

Vague search intent: Unlike a product listing, whose category is usually well-defined and precise, a search query's category can be vague. For example, query "used electronics" refers to a high-level category "Electronics" which is rarely predicated alone for a product listing.

Internationalization: Facebook Marketplace is a global platform and aims to offer consistent search experiences across languages. However, training data mined from search engagement are naturally skewed towards popular languages like English.

In this paper, we present HierCat, a query categorization system designed to learn from weakly supervised training data mined from search engagement. Unlike popular query categorization methods based on text classification approach [1, 3, 10, 13, 15], we introduce a dual-encoder architecture with a hierarchical inference step to leverage both the textual information of category labels and the hierarchical structure of product taxonomy. The encoder also incorporates state-of-the-art transformer architecture pre-trained on cross-language data as well as a downstream product retrieval task. We show that this simple and effective approach can substantially reduce the noise in training data, and generalizes to unseen queries better than many popular approaches. We also share our inference-time optimizations like category embedding caching and beam search to speed up inference, and our deployment story around

[1]http://www.facebook.com/marketplace

WWW '23 Companion, April 30-May 4, 2023, Austin, TX, USA

ACM ISBN 978-1-4503-9419-2/23/04…$15.00
https://doi.org/10.1145/3543873.3584622

translating modeling advances to better search experiences. HierCat is now powering hundreds of millions of search queries on Facebook Marketplace per day, helping users find relevant product listings of their interests.

2 METHODOLOGY

2.1 Problem formulation

Facebook Marketplace uses a product taxonomy system called Facebook Product Taxonomy (FPT) to categorize product items. The taxonomy consists of around six thousand categories organized in a tree structure, with maximum depth equals to six. The problem of query categorization is thus mapping a search query to the best category path on the taxonomy tree. Since a search query can be vague, a category path can end anywhere on the taxonomy tree and not necessarily on a leaf node. Formally, we model the conditional probability of a category path given a query $P(path|query)$, where $path$ is a sequence of category nodes on a taxonomy tree indexed by its level of the form $\{cate\_L_k, cate\_L_{k-1}, ..., cate\_L_1\}$, and $cate\_L_k$ is a category node at level k of the taxonomy tree. To avoid modeling combinatorial number of taxonomy paths, we leverage the structure of the taxonomy tree and only model the conditional probability of a node given its ancestors using the following factorization

$$\begin{aligned} &P(\{cate\_L_k, ..., cate\_L_1\} \mid query) = \\ &P(cate\_L_k \mid \{cate\_L_{k-1}, ..., cate\_L_1\}, query) \cdot \\ &P(\{cate\_L_{k-1}, ..., cate\_L_1\} \mid query) \end{aligned} \tag{1}$$

In the following sections, we will use $path$ and $category$ interchangeably, both referring to the final category prediction like "Home//Furniture//Sofa". Sometimes we will use $node$ to refer to a node on the taxonomy tree (e.g. "Sofa").

2.2 Weakly supervised training

We adopt a weakly supervised approach based on logged search engagement to train our classifiers. Specifically, we sample <query, product> pairs from 14 days of Facebook Marketplace's search engagement log on public content. All of the data are de-identified and aggregated, and are translated into 24 million <query, category> pairs based on a product item's category. <query, category> pairs with very low aggregation frequency are filtered out to reduce noise. Different predictions for the same search query are permitted because we hope to benefit from the richness of unfiltered information, especially since those labels may share some parent nodes (e.g. ,"Home//Furniture//Chair" v.s. "Home//Furniture//Sofa"). Note that at Facebook Marketplace, a product item's category label can come from the seller, model prediction, or both. Labels from seller selection have limited coverage and are not very granular, but are accurate. Model predicted labels have better coverage but limited precision, especially at deeper levels. Note that this poses the challenge of heterogeneous training data for query categorization models.

2.3 Pre-trained dual-encoder classification

Instead of predicting query category as conditional probabilities as in section 2.1, we first treat categorization as a classic query classification problem for simplicity. In other words, we train a neural classification model Φ that predicts a node on the taxonomy tree given a query without considering the taxonomy structure. Formally, for a query and category node pair (q, c), and the space of categories C, we minimize the cross-entropy loss defined as

$$L(q, c_i) = -\log \frac{\exp(\Phi(q, c_i))}{\sum_{c_j \in C} \exp(\Phi(q, c_j))} \tag{2}$$

For better generalization to different languages, we use a 2-layer XLM-encoder, a transformer-based language model pre-trained on large multilingual data [4] to process query text. We also represent query text using character trigrams and encode it using an EmbeddingBag encoder [12], as we find that multi-granular text representation with simple trigrams helps with short head queries. The two text representations are merged via a simple attention fusion layer as illustrated in figure 1.

An interesting property of the query categorization problem is that the input query often shares the same text as its desired label. For example, query "electric boat" and category "Vehicle//Boat" both contain the word "boat". Motivated by this observation, we borrow the two-tower architecture commonly used in retrieval models [11] to allow for an extra text encoder on the category labels and replace the final predication layer with the cosine similarities between query and category embeddings of dimension 128. Given that category text always come from a fixed set of six thousand categories, we only use EmbeddingBag encoder for the category tower to prevent over-fitting.

2.4 Product retrieval pre-training

For the query embedding tower, we introduce an optional pre-training task based on embedding-based retrieval (EBR) of Marketplace products given search queries, because empirically, we discover that EBR model training generates nicely clustered search queries [11]. For this work we adopt the Que2Search model [11] which shares a very similar text encoder architecture and is trained to minimize the cosine similarities between engaged query and product pairs. We retrain the EBR model to ensure that there is no architectural difference.

2.5 Hierarchical inference

So far we treat query categorization as a flat classification problem. We discover this approach to be problematic due to the noise in user engagement data. For example, both "Cell Phones//Accessories//Cases" and "Cell Phones" are popular labels mined for query "iPhone 12", because a user could be browsing iPhone cases while shopping for an iPhone. Intuitively, both labels are useful to infer the query being a "Electronics" and "Cell Phones", but in the eyes of a normal multi-class classifier, they are just inconsistent labels.

To address this problem, we introduce a hierarchical inference algorithm to re-normalize the probability mass over a taxonomy tree. As illustrated in algorithm 1, for all of the nodes on the taxonomy tree, we assign the cosine similarity generated from the dual-encoder model to $p\_cate[node]$. We then apply softmax over all of the leaf nodes to ensure they sum up to one. For any of the nodes at leaf-1 level, we propagate the probabilities of its children back to itself, and then apply softmax at the leaf-1 layer. We repeat this iteratively at each layer until all of the levels are

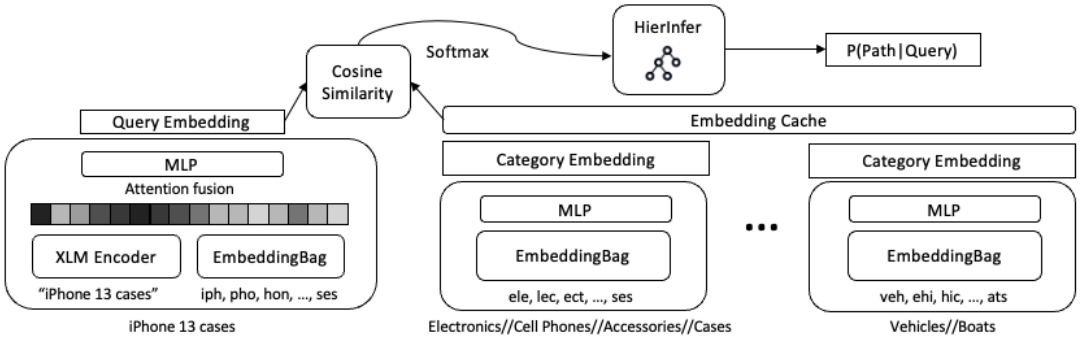

Figure 1: Model architecture

up-propagated and normalized. This ensures that the joint probability of all the nodes on a category path satisfies the conditional probability factorization we introduced in section 2.1. One caveat is that $p\_cate[node]$ still contains its original cosine similarity before probabilities from its children are added, and we believe that line 6 in algorithm 1 can be further generalized to be $\alpha * p\_cate[n] + SUM(p\_cate[CHILDREN(p\_cate[n])$ to control the impact of hierarchical inference vs. the original prediction. For simplicity of evaluation, we leave $\alpha = 1$ and did not tune the value in this paper.

Algorithm 1: Hierarchical inference

```
1 Function hier_infer(nodes, p_cate):
2     Let p_cate[node] = cosine_similarity(query, node)
3     Let levels = L[MAX_LEVELS − 1, ..., 0]
4     for l ∈ levels do
5         for n ∈ nodes[l] do
6             p_cate[n] +=
                SUM(p_cate[CHILDREN(p_cate[n])])
7         end
8         p_cate[nodes[l]] = SOFTMAX(p_cate[[nodes[l]])
9     end
```

3 OFFLINE EXPERIMENTS

3.1 Experimental setup

Baseline models and ablation studies. We perform ablation studies to evaluate each of our modeling choices and compare against alternative query classification methods commonly used across the industry. We compare our results against FastText multi-label classifier [2, 8] that assigns one class per-level as an hierarchical classifier. FastText is commonly used at Meta for short text classification and the multi-label approach also achieves good performance in recent query classification work [1, 14]. We also test directly fine-tuning an XLM model [4], a state-of-the-art pre-trained transformer model for classification. Since the XLM-classification model is used as part of our dual-encoder architecture, it also serves as an ablation study on the new architecture we propose. In addition, we experiment with combining the XLM-classification model with the multi-label approach as an alternative way to encode label hierarchy.

Data and evaluation metrics. We train all of the models with the 24 million weakly supervised data described in section 2.2. For evaluation, we use a stratified sample of 173 thousand queries across different regions and languages (which were de-identified from any personal information of the users inputting the queries), and ask raters to map them to the right categories. We report micro-F1 and top five accuracy (true category matches with any one of the top five predictions) at different levels to access the model performances.

Product retrieval pre-training. We run separate experiments for adding product retrieval pre-training and report the relative improvement, because it is an optional step in our system that can work with any baseline architecture. For this ablation study we use the XLM + trigram query encoder.

3.2 Ablation results

As illustrated in table 2, dual-encoder architecture with hierarchical inference achieves the best performance across all levels, while hierarchical inference itself improves the performance on higher levels independent of the classification algorithm, because of its consolidation effect on higher categories by leveraging probabilities from their children. XLM classification model also outperforms FastText classifier possibly because of better generalization across languages [4]. In addition, pre-training on product retrieval task, as shown in table 3, significantly improves leaf-level performance but does not help with L1. Our hypothesis is that leaf-level classification requires more granular embedding representations, and thus benefits from pre-trained embeddings more.

4 ONLINE EXPERIMENTS

4.1 Serving-time optimizations

We deployed HierCat on Meta's dedicated inference cloud [7] and implemented several additional optimizations in the model's production path to improve latency and quality.

Embedding caching: to speed up model inference, we only deployed the query encoder, and the category embeddings are precomputed and cached. During inference, we first compute the query embedding, and then loop through the cached category embeddings to obtain the logits. Finally, hierarchical inference is performed to obtain the correct probability distribution over a taxonomy tree. A query level cache is also implemented to avoid duplicate inference calls for popular queries.

Technique	L1 F1	L1 acc@5	L3 F1	L3 acc@5	L6 F1	L6 acc@5
FastText multi-label	0.582	0.866	0.176	0.389	0.165	0.357
XLM classification	0.762	0.869	0.339	0.416	0.233	0.315
XLM multi-label	0.661	0.874	0.241	0.376	0.195	0.303
XLM classification + hierInfer	0.765	0.907	0.344	0.426	0.233	0.315
Dual-encoder XLM + hierInfer	0.767	0.907	**0.388**	**0.516**	**0.259**	**0.369**
Dual-encoder XLM + hierInfer + trigram	**0.774**	**0.913**	0.363	0.514	0.237	0.366

Table 2: Results for baseline comparisons and ablation studies

L1 F1	L1 acc@5	L3 F1	L3 acc@5	L6 F1	L6 acc@5
neutral	neutral	+3%	neutral	+16%	+4%

Table 3: Improvements from product retrieval pre-training

Technique	Engagement	NDCG
L1 & L2 category boost	neutral	+1.4%
L3 category boost	+4.3%	neutral

Table 4: Online A/B testing results for structured retrieval

Beam search: to further speed up inference and obtain a consistent taxonomy path, a beam search is performed by iteratively selecting the best children for each parent nodes. In production we simply use a beam size of one for simplicity. Beam search is terminated early if the score is lower than a threshold tuned from percentile method.

4.2 Structured retrieval

We leveraged HierCat in Facebook Marketplace Search to reduce the category mismatched results by emitting the per-level categories as optional retrieval terms, and boosting retrieval scores upon category matches. This is done with Meta's Unicorn system [5] but can also be achieved with open-source solutions like Elasticsearch [6]. The term weights are tuned by applying a linear regression to predict downstream ranking scores, and boosts are given to category matches up to level three. Two weeks of online A/B testing shows that such retrieval boost significantly improves both NDCG and online user engagement, as illustrated in table 4 (metric improvements in table 4 are all relative and incremental, and L3 category boost was launched after L1 & L2 boost). Interestingly, we found that NDCG improved when L1 & L2 boosts were added, while engagement stayed neutral until L3 boost was added. We discovered that this is due to discrepancies between the NDCG rating guideline and user behavior - while former focused more on top level category matches, it is the L3 match, which is the median of level distributions that is driving user engagement.

4.3 Level distributions

To understand the level of noise in the weakly supervised data mined from search engagement and how well our method is able to counter such noise, we examine the level distribution of query categories from logged online predictions, ground truth labels and training data as shown in figure 2. We can indeed observe a better overlay with the ground truth distribution from our production model, indicating that our method is able to de-bias its training data to be closer to ground truth.

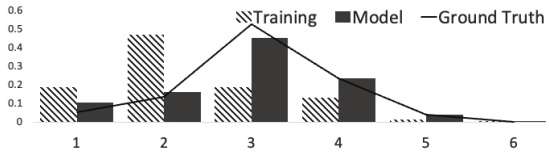

Figure 2: Level distribution comparisons

5 RELATED WORK

Query classification aims to classify search queries into a set of topical space to better understand users' intents. This is generally treated as a text classification problem where the input texts are short and the class space is large. For example, Liu et al. [10] proposed an extreme classification approach using a mixture of CNN and Naive Bayes classifier based on such characteristics. In e-commerce, the label space is often a product taxonomy that possesses a tree structure, yet regular multi-class classification approach would either treat each node as a class, or only consider the the leaf categories [1, 10, 13, 15]. Liu et al. [14] explored using multi-label FastText as a hierarchical classifier, and a back-off strategy to optimize the granularity-precision trade-off. Techniques to incorporate different dimensions of signals like contextual information [3, 15] and search engine feedback [1] are also explored to disambiguate search intent and augment training data. In terms of the classification algorithm, popular text classification methods like XML-CNN [9] or FastText [2, 8] are often used as a component in a query classification system [10, 14]. Weakly supervised labels mined from search logs are also explored as a data augmentation technique [1, 10, 15], while the characteristics of such data especially its implication to hierarchical classification is rarely discussed to the best of our knowledge.

6 CONCLUSION

In this paper we demonstrate HierCat, a query categorization system that learns from weakly supervised data by leveraging transformer encoders pre-trained on downstream product retrieval tasks, and the hierarchical structure of an e-commerce product catalog. We show the effectiveness of each modeling choice we made through ablation studies and comparing against popular baseline methods. To the best of our knowledge, HierCat is also the first to demonstrate the effectiveness of transformer-based dual-encoder architecture in e-commerce query categorization despite that it is a rather popular architecture in information retrieval tasks [11]. We deploy HierCat on Facebook Marketplace Search, share our inference-time optimizations, and show through online A/B testing that it significantly improves NDCG and searcher engagement.

REFERENCES

[1] Ali Ahmadvand, Sayyed M. Zahiri, Simon Hughes, Khalifa Al Jadda, Surya Kallumadi, and Eugene Agichtein. 2021. APRF-Net: Attentive Pseudo-Relevance Feedback Network for Query Categorization. *CoRR* abs/2104.11384 (2021). arXiv:2104.11384 https://arxiv.org/abs/2104.11384

[2] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomás Mikolov. 2016. Enriching Word Vectors with Subword Information. *CoRR* abs/1607.04606 (2016). arXiv:1607.04606 http://arxiv.org/abs/1607.04606

[3] Huanhuan Cao, Derek Hao Hu, Dou Shen, Daxin Jiang, Jian-Tao Sun, Enhong Chen, and Qiang Yang. 2009. Context-Aware Query Classification. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Boston, MA, USA) *(SIGIR '09)*. Association for Computing Machinery, New York, NY, USA, 3–10. https://doi.org/10.1145/1571941.1571945

[4] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised Cross-lingual Representation Learning at Scale. *CoRR* abs/1911.02116 (2019). arXiv:1911.02116 http://arxiv.org/abs/1911.02116

[5] Michael Curtiss, Iain Becker, Tudor Bosman, Sergey Doroshenko, Lucian Grijincu, Tom Jackson, Sandhya Kunnatur, Soren Lassen, Philip Pronin, Sriram Sankar, Guanghao Shen, Gintaras Woss, Chao Yang, and Ning Zhang. 2013. Unicorn: A System for Searching the Social Graph. *Proc. VLDB Endow.* 6, 11 (Aug. 2013), 1150–1161. https://doi.org/10.14778/2536222.2536239

[6] Clinton Gormley and Zachary Tong. 2015. *Elasticsearch: The Definitive Guide* (1st ed.). O'Reilly Media, Inc.

[7] Kim Hazelwood, Sarah Bird, David Brooks, Soumith Chintala, Utku Diril, Dmytro Dzhulgakov, Mohamed Fawzy, Bill Jia, Yangqing Jia, Aditya Kalro, James Law, Kevin Lee, Jason Lu, Pieter Noordhuis, Misha Smelyanskiy, Liang Xiong, and Xiaodong Wang. 2018. Applied Machine Learning at Facebook: A Datacenter Infrastructure Perspective. In *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. 620–629. https://doi.org/10.1109/HPCA.2018.00059

[8] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of Tricks for Efficient Text Classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Association for Computational Linguistics, Valencia, Spain, 427–431. https://www.aclweb.org/anthology/E17-2068

[9] Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. 2017. Deep Learning for Extreme Multi-Label Text Classification. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Shinjuku, Tokyo, Japan) *(SIGIR '17)*. Association for Computing Machinery, New York, NY, USA, 115–124. https://doi.org/10.1145/3077136.3080834

[10] Xianjing Liu, H. Zhang, Mingkuan Liu, and Alan Lu. 2019. System Design of Extreme Multi-label Query Classification using a Hybrid Model. In *eCOM@SIGIR*.

[11] Yiqun Liu, Kaushik Rangadurai, Yunzhong He, Siddarth Malreddy, Xunlong Gui, Xiaoyi Liu, and Fedor Borisyuk. 2021. Que2Search: Fast and Accurate Query and Document Understanding for Search at Facebook. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Virtual Event, Singapore) *(KDD '21)*. Association for Computing Machinery, New York, NY, USA, 3376–3384. https://doi.org/10.1145/3447548.3467127

[12] Adam Paszke, S. Gross, Soumith Chintala, Gregory Chanan, E. Yang, Zach DeVito, Zeming Lin, Alban Desmaison, L. Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch.

[13] Michael Skinner and Surya Kallumadi. 2019. E-commerce Query Classification Using Product Taxonomy Mapping: A Transfer Learning Approach. In *eCOM@SIGIR*.

[14] Hang Yu and Lester Litchfield. 2020. Query Classification with Multi-Objective Backoff Optimization. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Virtual Event, China) *(SIGIR '20)*. Association for Computing Machinery, New York, NY, USA, 1925–1928. https://doi.org/10.1145/3397271.3401320

[15] Jiashu Zhao, Hongshen Chen, and Dawei Yin. 2019. A Dynamic Product-Aware Learning Model for E-Commerce Query Intent Understanding. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (Beijing, China) *(CIKM '19)*. Association for Computing Machinery, New York, NY, USA, 1843–1852. https://doi.org/10.1145/3357384.3358055