SHOAL: Large-scale Hierarchical Taxonomy via Graph-based Query Coalition in E-commerce

Zhao Li[1,*], Xia Chen[1,3], Xuming Pan[1], Pengcheng Zou[1], Yuchen Li[2], Guoxian Yu[3]
[1]Alibaba Group, Hangzhou, China
[2]School of Information Systems, Singapore Management University, Singapore
[3]College of Computer and Information Sciences, Southwest University, Chongqing, China
[1]{lizhao.lz, xia.cx, xuming.panxm, xuanwei.zpc}@alibaba-inc.com
[2]yuchenli@smu.edu.sg
[3]{xchen, gxyu}@swu.edu.cn

ABSTRACT

E-commerce taxonomy plays an essential role in online retail business. Existing taxonomy of e-commerce platforms organizes items into an ontology structure. However, the ontology-driven approach is subject to costly manual maintenance and often does not capture user's search intention, particularly when user searches by her personalized needs rather than a universal definition of the items. Observing that search queries can effectively express user's intention, we present a novel large-Scale Hierarchical taxOnomy via grAph based query coaLition (*SHOAL*) to bridge the gap between item taxonomy and user search intention. SHOAL organizes *hundreds of millions of items* into a *hierarchical topic structure*. Each topic that consists of a cluster of items denotes a conceptual shopping scenario, and is tagged with easy-to-interpret descriptions extracted from search queries. Furthermore, SHOAL establishes correlation between categories of ontology-driven taxonomy, and offers opportunities for explainable recommendation. The feedback from domain experts shows that SHOAL achieves a precision of 98% in terms of placing items into the right topics, and the result of an online A/B test demonstrates that SHOAL boosts the Click Through Rate (CTR) by 5%. SHOAL has been deployed in Alibaba and supports millions of searches for online shopping per day.

PVLDB Reference Format:
Zhao Li, Xia Chen, Xuming Pan, Pengcheng Zou, Yuchen Li and Guoxian Yu. SHOAL: Large-scale Hierarchical Taxonomy via Graph-based Query Coalition in E-commerce. *PVLDB*, 12(12): 1858-1861, 2019.
DOI: https://doi.org/10.14778/3352063.3352084

1. INTRODUCTION

*Zhao Li is the corresponding author.

Proceedings of the VLDB Endowment, Vol. 12, No. 12
ISSN 2150-8097.
DOI: https://doi.org/10.14778/3352063.3352084

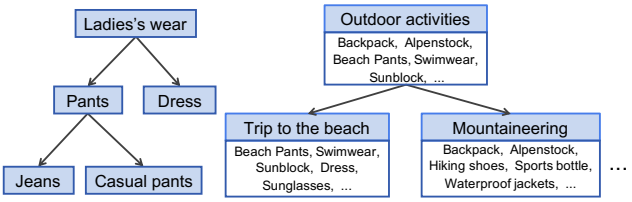


Figure 1: (a) an example of ontology-driven taxonomy with each node as a category; (b) an example of SHOAL with each node as a topic which is associated with a number of categories. For example, the topic "Trip to the beach" is associated with category "Beach pants", "Sunblock" and etc.

Item taxonomy is one of the most fundamental component of e-commerce platforms [8, 7]. Proper taxonomy facilitates efficient browsing navigation that enhances user search experiences. Existing item taxonomy organizes items into a categorical structure, driven by dictionary based ontology [5]. Take Figure 1(a) for an example, "Dress" is a leaf category and belongs to parent category "Ladies' wear". Although dictionary-based ontology seems to be effective in managing items, it fails to capture user's search intention in many scenarios due to the rigid categorization. For instance, when a user is searching items with query "Beach dress", the search engine then targets the items falling into category "Dress" and returns to the user. One could further move one level up and return items in category "Ladies' wear" that coarsely matches the user query. However, the user may also be keen to explore categories that are related to a shopping topic of "Trip to the beach", such as "Sunglasses" and "Sunblock". Existing taxonomy thus only provides users with accurate categorical information, while shopping topic based information further helps users in navigating to their desired items. From this perspective, it calls for a topic based taxonomy system on e-commerce platforms.

It is noted that search queries can effectively express user's intention, which motivates us to build a taxonomy system by leveraging massive number of search queries submitted to our e-commerce platform. Built on top of the query-item bipartite graph (Figure 2), we develop a novel large-Scale Hierarchical taxOnomy via grAph based query coaLi-

Figure 2: query-item bipartite graph.

tion (*SHOAL*) to organize items into a hierarchical topic structure. SHOAL places shopping topics into semantic hierarchies, which are natural ways to organize knowledge in a taxonomy system and render clear explainability. A toy example of SHOAL is showed in Figure 1(b). Query "Beach dress" will hit the topic "Trip to the beach" that includes multiple categories such as "Beach Pants" and "Swimwear". Compared with ontology-driven taxonomy showed in Figure 1(a), SHOAL has clear advantages in helping users identify more items that they are interested in.

Technically, we introduce a novel Parallel Hierarchical Agglomerative Clustering (*Parallel HAC*) to organize hundreds of millions of items into hierarchical topics based on the query-item graph. Each topic corresponds to a conceptual shopping topic and consists of a cluster of items. SHOAL tags the topics with easy-to-interpret descriptions extracted from the search queries. Furthermore, we establish relationships between query-driven topics and ontology-driven categories, which could be used to recommend semantically related categories to users. We have proved the effectiveness and high-efficiency of SHOAL in organizing large scale of items, and the effectiveness in improving user search experience on Alibaba e-commerce platform. The feedback from domain experts shows that SHOAL achieves a precision of 98% in terms of placing items into the right topics, and the result of an online A/B test demonstrates that SHOAL boosts the Click Through Rate (CTR) by 5%.

Related Studies. Clustering based methods are closely related to our problem, which learn the representation of terms and then organize them into a structure based on the representation similarity [1]. For example, Luu et al. [3] propose to use dynamic weighting neural network to identify relations via learning term embedding. Wang et al. [4] adopt a recursive way to construct topic hierarchies by clustering domain keyphrases. Recently, Zhang et al. [6] introduce TaxoGen that uses local term embedding and hierarchical clustering to construct a topic taxonomy in a recursive fashion. Compare with existing approaches, SHOAL considers both structural and textual similarities between the items, and constructs taxonomy over large scale of items by introducing a novel parallel clustering algorithm.

2. SHOAL FRAMEWORK

SHOAL consists of four components. First, we extract an item entity graph from the query-item bipartite graph (Sec. 2.1). Second, we introduce the parallel HAC on the item entity graph to construct the taxonomy that groups item entities with a hierarchical structure (Sec. 2.2). Third, we assign descriptions to the topics for better explain-ability (Sec. 2.3). Finally, in Sec. 2.4, we discuss how to establish correlations among ontology-driven categories to facilitate recommendation.

2.1 Item Entity Graph

Intuitively, a pair of items should have a stronger semantic relationship if the pair associates with similar sets of user queries and share similar product/content descriptions. Thus, we build an item entity graph to capture both query-driven similarity and content-driven similarity between the items. Formally, we denote $G(V, E, S)$ as an item entity graph with V as the vertex set representing the item entities, E as the edge set and each edge $e = (u, v) \in E$ is associated with a weight $S(e)$ to denote the degree of semantic similarity between item entity u and item entity v. Note that each item entity may contain a set of items with near-equivalent attribute labels and price. In the sequel, we introduce how to model the similarities between the item entities.

Query-driven Similarity. Each item entity is naturally associated with a set of queries in the query-item graph. Let u and v denote two item entities, $\mathcal{Q}_u$ and $\mathcal{Q}_v$ denote the associated set of queries for u and v respectively. We measure the query-driven similarity between u and v with the Jaccard similarity:

$$S_q(u, v) = \frac{|\mathcal{Q}_u \bigcap \mathcal{Q}_v|}{|\mathcal{Q}_u \bigcup \mathcal{Q}_v|} \quad (1)$$

Content-driven Similarity. We segment the title/s of an item entity into words and obtain a set of word vectors using the word2vec technique for each item entity. Let $\mathcal{V}_u$ and $\mathcal{V}_v$ denote the set of word vectors associated with entity u and v respectively. The content-driven similarity between u and v is modeled as:

$$S_c(u, v) = \frac{1}{|\mathcal{V}_u| \cdot |\mathcal{V}_v|} \sum_{\mathbf{w}_1 \in \mathcal{V}_u} \sum_{\mathbf{w}_2 \in \mathcal{V}_v} \frac{1}{2} + \frac{1}{2} \frac{\mathbf{w}_1 \mathbf{w}_2^T}{||\mathbf{w}_1|| \cdot ||\mathbf{w}_2||} \quad (2)$$

where $\mathbf{w}_1$ and $\mathbf{w}_2$ denote the embedding vectors of words. Finally, the overall similarity between u and v is defined as:

$$S(u, v) = \alpha S_q(u, v) + (1 - \alpha) S_c(u, v) \quad (3)$$

where the parameter α is set to 0.7 for the demonstration.

2.2 Parallel HAC

Hierarchical Agglomerative Clustering (HAC) is a natural choice to cluster item entities with the hierarchical structure in the item entity graph. It works by iteratively merging two nodes with the largest similarity in the graph until the all similarity scores are less than a threshold. However, there are two major challenges to apply HAC directly: 1) HAC requires a fully connected similarity matrix to update the similarity between nodes, while the similarity matrix in our scenario is sparse. The reason of sparsity is that we need to filter out the values in S that are too low, in view of the actual situation where one item entity should have only a few neighbor entities (Challenge 1). 2) HAC does not scale to large graphs as each of its iterations requires to scan the entire graph and there could be $O(V)$ iterations in the worst case (Challenge 2). To overcome the challenges, we introduce a novel parallel hierarchical agglomerative clustering (Parallel HAC) algorithm. Parallel HAC employs a two-dimensional embedding for similarity calculation in sparse graphs and distributed merge for parallel acceleration.

Two-Dimensional Embedding for Sparse Similarity Calculation. Take Figure 3 as an illustrative example. After merging node A and node B as AB, HAC needs to update the similarity score between AB and the neighbors of A and B such as C. HAC requires the similarity is available between A, B and C, which does not hold for sparse graphs.

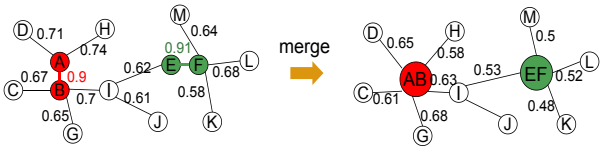


Figure 3: An illustration of Parallel HAC. Each edge is associated with a similarity score.

To resolve this issue, we update the similarity between AB and C as follows:

$$S(AB,C) = \frac{\sqrt{n_A}}{\sqrt{n_A+n_B}}S(A,C) + \frac{\sqrt{n_B}}{\sqrt{n_A+n_B}}S(B,C) \quad (4)$$

where $S(A,C)$ and $S(B,C)$ denote the similarity between A and C, and the similarity between B and C, respectively. n_A and n_B denote the number of item entities in A and B, respectively. $S(A,C)=0$ if the similarity between A and C is unavailable. From Eq.(4), SHOAL measures the similarity between AB and C by using a sqrt normalization. Theoretically, the normalization calculates the similarity between AB and C by embedding nodes into a two-dimensional space, in which the similarity between two nodes is the sqrt root of the projected region.

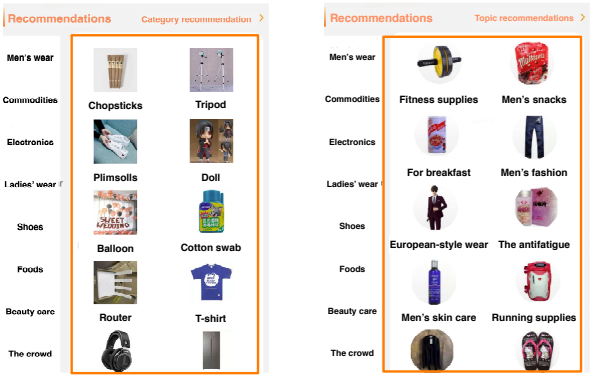
Distributed Merging. In each iteration of finding the edge with the largest similarity, HAC needs to scan all the edges and merges only one pair of nodes, which greatly limits the computational efficiency for large scale graph processing. To efficiently extract the taxonomy, we develop a distributed solution to discover local maximal edges via graph diffusion, and to then merge the corresponding nodes in parallel. For each iteration of the graph diffusion process, every node receives the maximal that its neighbors discover from its neighbors and "diffuses" the maximal edge to its neighbors. After a number of iterations, the local maximal edges are obtained as the maximal edges that each node receives. As shown in Figure 3, the edge between A and B, and the edge between E and F are two local maximal edges obtained after two diffusion iterations. Obviously, the smaller the number of iterations of graph diffusion is, the larger the number of local maximal edges is, and the higher the degree of parallelization. The maximum number of iterations of graph diffusion is set to 2 for SHOAL.

We consider the graph modularity [2] as a benchmarking metric to evaluate the effectiveness of parallel HAC. The results have shown that Parallel HAC consistently produces clusters with modularity > 0.3. Besides, we deploy SHOAL on the Alibaba distributed graph platform (ODPS) and the proposed Parallel HAC is able to efficiently generate taxonomy for 200 millions of item entities within 4 hours.

2.3 Topic Description Matching

SHOAL augments meaningful descriptions to each topic extracted for building an interpretable taxonomy. We leverage the query-item graph to produce query-centric descriptions. It is noted that one topic is associated with a number of items and each item is linked to a number of queries in the query-item graph. The key question is how to choose the most representative query for describing a topic.

We address this problem by devising a similar strategy described in [6]. A representative query for a specific topic t_k should appear frequently in t_k but not in the other topics. Hence, we consider the following two factors for calculating the representativeness of a query q for topic t_k: (a) **Popularity.** a representative query should appear frequently with t_k; (b) **Concentration.** a representative query should be more relevant to query compared with the other topics.



(a) Control Group (b) Experiment Group

Figure 4: Recommendation Evaluation for SHOAL.

To combine the above two factors, we define the representativeness of query q for topic t_k as:

$$r(q,t_k) = \sqrt{pop(q,t_k)\cdot con(q,t_k)}$$

where $pop(q,t_k)$ and $con(q,t_k)$ are the popularity and concentration scores of q for t_k. Let $\mathcal{I}_k$ denotes the items belonging to t_k, $pop(q,t_k)$ is defined as the normalized frequency of q in t_k:

$$pop(q,t_k) = \frac{\log tf(q,\mathcal{I}_k)+1}{\log tf(\mathcal{I}_k)}$$

where $tf(q,\mathcal{I}_k)$ is number of occurrences of query q with $\mathcal{I}_k$ and $tf(\mathcal{I}_k)$ is the total number of tokens in $\mathcal{I}_k$.

The concentration score is calculated as:

$$con(q,t_k) = \frac{\exp\left(rel(q,D_k)\right)}{1+\sum_{1\le j\le K}\exp\left(rel(q,D_j)\right)}$$

where D_k is a pseudo document by concatenating all the title of items in t_k, and $rel(q,D_k)$ is the BM25 relevance of query q to D_k.

2.4 Category Correlation

SHOAL can mine correlations among ontology-driven categories from the extracted query-driven taxonomy. The correlation serves as an important subsystem in Alibaba search engine which recommends similar categories to users. To evaluate the correlation, we utilize the root topics (i.e., root nodes in the hierarchical structure) as pivots to link similar categories. We calculate the correlation strength between two categories as the number of co-occurrences in the same root topic. Let C_i and C_j denote the i-th category and the j-th category, respectively. The correlation strength between C_i and C_j is defined as follows:

$$S_c(C_i,C_j) = \sum_{t_k\in\mathcal{T}}[C_i\in\mathcal{C}_k \text{ and } C_j\in\mathcal{C}_k] \quad (5)$$

where $\mathcal{T}$ denotes the full set of topics and t_k is the k-th topic in $\mathcal{T}$. $\mathcal{C}_k$ is the set of categories associated with topic t_k. $[X]=1$ if X is true, and $[X]=0$ otherwise. There exists a correlation between C_i and C_j only if $S_c(C_i,C_j)>10$.

3. EVALUATIONS

SHOAL is constructed from *hundreds of millions of* items and a sliding window containing search queries in the last seven days on the Taobao platform of Alibaba. We report the results of item taxonomy of SHOAL to domain experts,

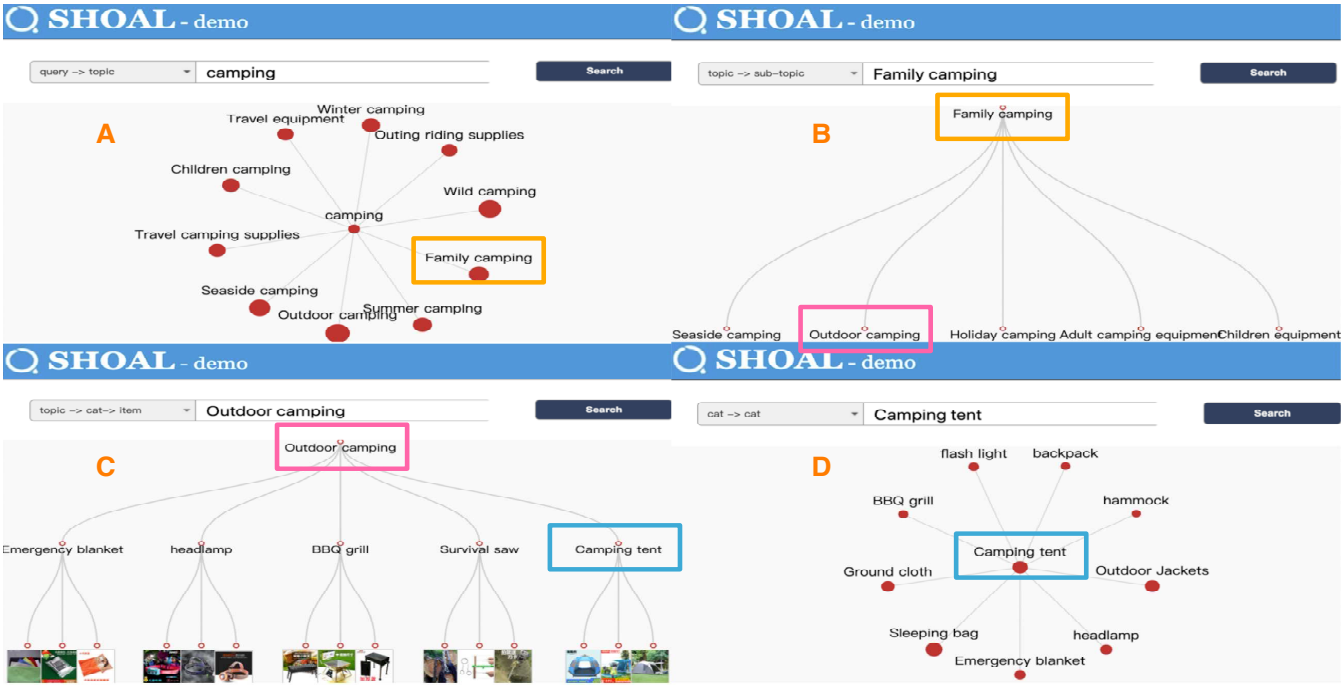


Figure 5: SHOAL GUI.

and the feedback shows that the precision of item taxonomy is more than 98% according to sampling evaluation, where experts pick 1000 topics and randomly select 100 items placed under each topic to evaluate the precision. We also conduct an online A/B test to verify the effectiveness of SHOAL in a real recommendation scenario with 3 million users. In the control group, the recommendations are generated by matching the ontology-driven categories (Figure 4 (a)), while the recommendations are made by matching the topics of SHOAL in the experiment group (Figure 4 (b)). The result of the A/B test demonstrates that SHOAL boosts the Click Through Rate (CTR) by 5%.

3.1 Demonstration Scenarios

We build a GUI system which enables easy exploration among items, categories and topics to showcase SHOAL. Figure 5 shows the interface of SHOAL, which includes four demonstration scenarios (A, B, C and D).

Query→Topic (A). Audiences of the demo can enter keyword queries, e.g., "camping", to search for relevant topics. The query processor finds related topics for the input query and displays the results as a visual star graph, where the center node denotes the query "camping" that an audience inputs, and the surrounding nodes which are clickable represent the related topics.

Topic→Sub-topic (B). Audiences can further click on any topic from scenario (A), e.g., "family camping", to show its sub-topics and explore the topic hierarchy extracted by SHOAL. Audiences can also enter a keyword query that finds a matching topic and its sub-topics.

Topic→Category→Item (C). Audiences can discover categories associated with a matching topic (via either enter keyword queries or navigate from scenario (A) or (B)). Furthermore, we visualize the relevant items that fall under a category.

Category→Category (D). Audiences can further explore correlation between categories. Upon receiving a query on a category, SHOAL displays the matching category as the center and a number of surrounding nodes representing the related categories which are obtained following the techniques described in Sec. 2.4.

4. REFERENCES

[1] D. Mimno, W. Li, and A. McCallum. Mixtures of hierarchical topics with pachinko allocation. In *ICML*, pages 633–640, 2007.

[2] M. E. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113, 2004.

[3] A. Tuan Luu, Y. Tay, S. C. Hui, and S. K. Ng. Learning term embeddings for taxonomic relation identification using dynamic weighting neural network. In *EMNLP*, pages 403–413, 2016.

[4] C. Wang, M. Danilevsky, N. Desai, Y. Zhang, P. Nguyen, T. Taula, and J. Han. A phrase mining framework for recursive construction of a topical hierarchy. In *KDD*, pages 437–445, 2013.

[5] S.-S. Weng, H.-J. Tsai, S.-C. Liu, and C.-H. Hsu. Ontology construction for information classification. *Expert Systems with Applications*, 31(1):1–12, 2006.

[6] C. Zhang, F. Tao, X. Chen, J. Shen, M. Jiang, B. Sadler, M. Vanni, and J. Han. Taxogen: Unsupervised topic taxonomy construction by adaptive term embedding and clustering. In *KDD*, 2018.

[7] Y. Zhang, A. Ahmed, V. Josifovski, and A. Smola. Taxonomy discovery for personalized recommendation. In *WSDM*, pages 243–252, 2014.

[8] C.-N. Ziegler, G. Lausen, and L. Schmidt-Thieme. Taxonomy-driven computation of product recommendations. In *CIKM*, pages 406–415, 2004.