Dynamic Filter Discovery and Ranking Framework for Search and Browse Experiences in E-Commerce

Ligaj Pradhan
The Home Depot
Atlanta, Georgia, USA
ligaj_pradhan@homedepot.com

Le Yu
The Home Depot
Atlanta, Georgia, USA
le_yu1@homedepot.com

Bingxin Li
The Home Depot
Framingham, Massachusetts, USA
bingxin_li@homedepot.com

Venkata Simhadri
The Home Depot
Austin, Texas, USA
venkata_g_simhadri1@homedepot.com

Jeyaprakash Singarayar
The Home Depot
Surprise, Arizona, USA
jeyaprakash_singarayar@homedepot.com

ABSTRACT

Searching and browsing events in an e-commerce platform can result in many products being retrieved and displayed to customers. In many cases, a sizable number of suggestions may not be exactly relevant to the customer's current intent e.g., one might see a lot of irrelevant products if he/she meant *'liquid kitchen cleaner'* but only searched for *'cleaner'*. As such, it can be very time-consuming to navigate multiple pages of results to find the desired product. Faceted Search with Search Filters can facilitate screening out numerous results that do not match the current user's search intent. However, the number of available product attributes can be extensive, and it would be impractical to present all as filters. A fixed set of manually curated filters would also be impractical as the result sets change with time. The main objective of the current paper is to explore and propose a two-stage framework that can dynamically generate and rank filters to help customers drill down to the product of interest. We first present an approach that dynamically generates and ranks filters for various categories. Then the second stage dynamically generates and ranks filters for search-terms by aggregating direct 'search-term level filter engagements' along with 'mappers mapping search-terms to top categories' to re-use the category-level filter suggestions from the first stage. Finally, we share the A/B test results from using this two-stage framework in a large e-commerce platform which clearly demonstrates statistically significant improvement in customer interactions with the filters.

CCS CONCEPTS

• **Computing methodologies → Information extraction**.

KEYWORDS

dynamic filter, faceted search, filter ranking, filter discovery

Conference SIGIReCom'23, July 27, 2023, Taipei, Taiwan

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/XXXXXXX.XXXXXXX

ACM Reference Format:
Ligaj Pradhan, Le Yu, Bingxin Li, Venkata Simhadri, and Jeyaprakash Singarayar. 2023. Dynamic Filter Discovery and Ranking Framework for Search and Browse Experiences in E-Commerce. In *Proceedings of SIGIR e-commerce workshop 2023 (Conference SIGIReCom'23)*. ACM, New York, NY, USA, 5 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Searching or browsing in an e-commerce platform can retrieve many products. Additionally, it is a common trend to include a diverse set of products with an increased recall to present more options to customers as their intentions may be unclear or indeterminate [10]. Diversification of the retrieved products may lead to more irrelevant products being displayed to customers. As such, it can be very time-consuming for customers to navigate multiple pages of results to find the desired product. Faceted Search using product attributes as filters can facilitate a guided navigation experience and help the customer to screen out numerous irrelevant results returned by the search/browse process[9]. However, these products would have many attributes that could be used as facets. It would certainly be impossible to present all attributes as filters. So, we need to select a set of highly effective attributes to present. A quick solution would be to provide a fixed set of manually curated filters, but this would also be less optimal as the result sets change with time and so do the product features that people are interested in[7]. Hence, we need to be able to generate and rank such filters dynamically. Such a dynamic filtering model automatically decides what attributes should be presented as filters and in what order. In this paper we propose such a dynamic filtering approach as a two-stage filtering framework. The first stage relies on category-level filter engagement signals and category-product-attribute mappings (available from the product catalogue) to dynamically generate and rank filters for various categories. The category-product-attribute mappings are available even when customers have not interacted with the filters and hence are critical to avoid cold start issues i.e., situations where we do not have any signal to capture the importance of the filters. Some examples of such captured inherent information can be how unique an attribute is to a category or how well an attribute can slice and dice the products within a category. Several papers have discussed how they use approaches like information theory to leverage such structures[7][3][8]. The output from this stage would directly serve filters for the browsing experience when the customer is navigating through various categories in the product taxonomy without typing any search-terms. The second stage predicts filters for the searching experience i.e., when the customer is typing a search-term to search for products. This stage uses customer engagement signals together with Query-to-Category mappers that map the search-terms into a product category for which we have already computed filters in the first stage. For mapping search-terms to categories in the experiments and A/B tests discussed in this paper, we used a rule-based model that used purchase order signals and selected the most specific sub-category that had at least 85 percent of all orders made with the search-term. Based on the data availability and the model complexity, several kinds of models are used to generate and rank filters in e-commerce today[9][5]. Most basic ones directly use a static list or popularity-based attributes for filtering. Some associate search-terms with categories and use category-level static or popular filters. More advanced approaches learn the ranking of the attributes by optimizing some business goals like conversion. However, the most advanced trend is to make faceting even more dynamic and personalized by using behavior-driven approaches considering customer and current query contexts[9][5]. Our proposed framework easily allows us to seamlessly incorporate any other Query-to-Filter and Category-to-Filter predictors as *'Additional Techniques'*. The framework proposes to aggregate scores given to each filter by various models to provide a final suggestion in both phases. A/B tests conducted with our filtering framework on our e-commerce platform for both Browse and Search Experiences showed a statistically significant lift in Filter Engagements and Add-To-Cart (ATC) rates across desktop and mobile channels.

2 RELATED WORK

Currently, a relatively static facet list is still used in some commercial applications. The facet list selection itself requires a significant amount of time and cost. Furthermore, the fixed facets can hardly satisfy the needs of faceted search as the importance of specific facets may change during the search sessions. Recently, some researchers are making progress in dynamic facets generation.

Kim et al. provide dynamic categorization methods for facet selection[4]. The selection is based on ontological data from a Semantic Web environment. The study is an extension of the earlier work[11] of the authors, which was based on the idea of selecting more descriptive facets by applying entropy-based measures. The limitation is that the algorithm does not consider numeric facets and the use of disjunctive semantics for values, which are important components in the e-commerce area.

Feddoul et al. present an approach for automatic facet generation and selection over a knowledge graph[3]. The authors propose several intra-facets scorings like "predicate probability" and "value carnality", and inter-facets scoring like "semantic similarity between filters" to rank filters. We also incorporate some of such scoring to rank filters for categories but mostly rely on such signals only when there are very less direct engagement signals. In addition to this, we also extend our framework to utilize a multitude of other signals and incorporate *'Additional Techniques'* to rank filters for our categories.

Vandic et al. present a framework for dynamic facet ordering in e-commerce instead of a fixed-ordered list of facets[7]. The authors split the facet types into nominal, Boolean and numeric for better processing. Properties are ranked by their importance and impurity measures so that more specific facets and properties are ranked higher. The authors also proposed a measure of dispersion for numeric facets by employing Gini coefficient[1] to avoid losing their ordinal nature. Facet values are sorted descending on the number of corresponding products. The implementation is for the e-commerce browsing experience only. However, the searching experience in e-commerce is not considered, and search-terms and customer behaviors are not connected with facets in the paper.

Zhang et al. proposed a strategy for soft-faceted browsing in e-commerce[10], where the system not only shows items matching the customer's selected facet filter, but also a few possibly relevant items. The authors claim that this strategy could be beneficial when a customer does not have a very confident and strict preference for the selected facet values. Similar to the work presented by Vandic et al.[7], the implementation is only within the browsing experience but not the searching experience and search-terms.

We identify the potential gaps in current related work and propose a dynamic filter generating and ranking framework by linking categories and customer search-terms to form a global model, utilizing the category-filter model for search filtering.

3 PROPOSED FRAMEWORK

The proposed 'dynamic filter discovery and ranking framework' (DFDRF) is basically composed of two stages.

3.1 Category Level Filters

The first stage is the category-filter prediction stage, and it begins by mapping navigable attributes as potential filters for each category. This is indicated by the grayed box in Figure 1. Two different types of signals are used to compute scores that depict the suitability of each attribute to be used as filters for each category. The first type of signal is obtained from the category-product-attribute mapping structure like the percentage of products under a category having an attribute (predicate probability)[3], how many unique attribute-values are there among all products under a category (attribute cardinality)[3], how evenly are the attribute values distributed among the products under a category (coefficient of variance)[3], measure of uniqueness or specificity in terms of describing products under a category (Gini)[7], how unique is an attribute to a category as indicated by 'uniqueness' in equation 1,

$$uniqueness = \frac{\text{product count with the attribute in a category}}{\text{product count with the same attribute in all categories}} \tag{1}$$

The second type of signal is obtained from the customer interaction with the filters when browsing products from different categories (browsing experience). If customers engage very frequently with certain filters for a category, then those are deemed to be the most important ones to be presented as filters for that category and should be ranked higher in the list of filters. The framework also has the flexibility to accommodate continuous improvement to integrate alternate techniques that might be used in future to suggest filters for categories, as depicted by the *'Additional Techniques'* block in

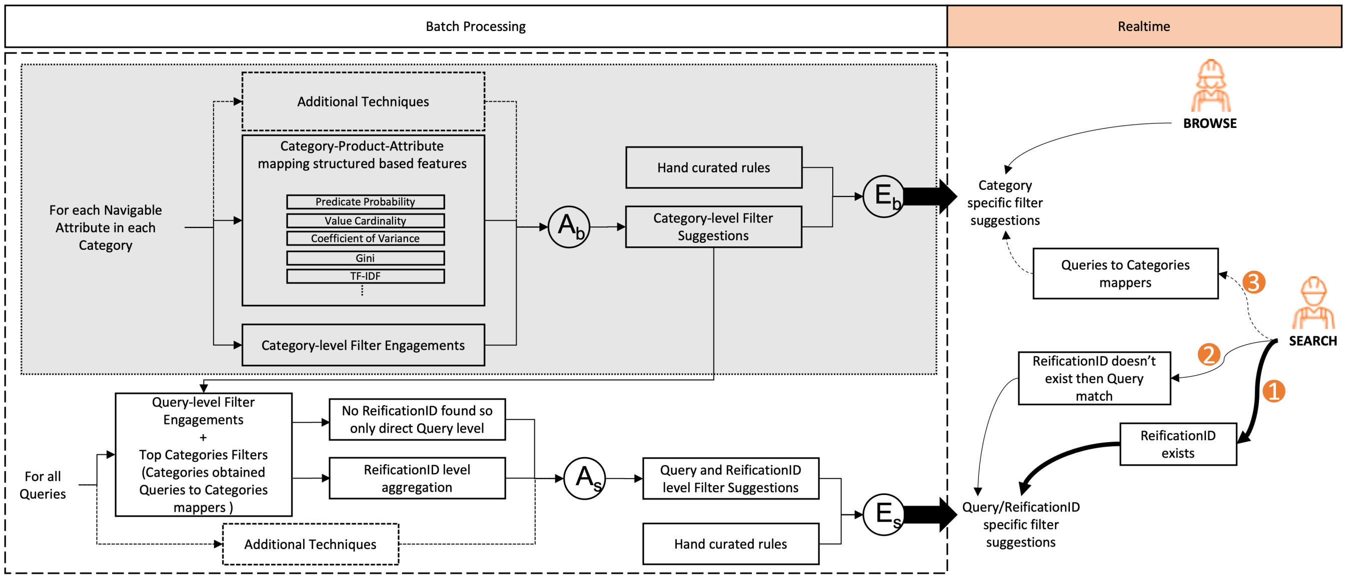

Figure 1: Filters Discovery and Ranking Framework(FDRF) for Browse and Search Experiences in E-Commerce

Figure 1. Finally, the framework aggregates the scores computed using these sources and generates final scores for each navigable attribute in each category as depicted by A_b step in Figure 1. The aggregation can be a simple weighted sum of scores or complex machine learning models that learn to rank different attributes using the input signals from these different sources. For the tests described in the paper, we took the weighted sum after learning suitable weights using linear regression to maximize NDCG score considering the filter order presented by customer engagement as the ground truth. Furthermore, there can be manually set rules and hand-crafted curations from different product experience teams to prioritize some and deprioritize other filters. The framework suggests aggregating the predicted category-filter suggestions with such rules in E_b step before making the ultimate category-filter suggestions.

3.2 Search-term Level Filters

The second stage of the proposed DFDRF framework is the search-term-filter predictor indicated by the non-grayed out region inside the outer most dashed block in Figure 1. Several signals are used to make search-term-level filter suggestions. The first signal is the output of the category-filter predictor stage itself. Search-terms are mapped to their suitable categories using various approaches and then filters are aggregated using the category-level predictions. Some search-terms are specific enough and can be effectively associated with only one category while others can be broad and would associate with multiple categories. Hence, there might be a situation where we need to aggregate filter scores from multiple categories to select filters for a certain search-term. Equation 2 shows how we map a search-term to categories based on customer purchase activities:

$$P(C_i|s) = \frac{\sum_{j=C_i} pp_{s_j}}{\sum_{j=C_1}^{C_N} pp_{s_j}} \tag{2}$$

where given search-term s, the probability of mapping it to a category C_i is calculated based on historical product purchase pp distribution across all categories (C_1) to (C_N). Usually, a threshold σ is set to get the final categories. We set σ to 85% and pick only one top category in our AB tests. E.g., for search-term *'top load washer'*, if 90% of the historical purchases are from category *'Washing Machine'*, then *'Washing Machine'* is the intented category for *'top load washer'* and the category level filters(explained in 3.1) could be reused for the search-term. A multi-category classification model could also be trained based on historical purchases to alleviate cold start situations where no purchase history could be found. The second source of information the framework utilizes is collected from the customer engagement with filters when they are searching products using different queries. Similar to the category-filter stage, if customers engage with certain filters more frequently for a certain search-term, then that filter is considered more important and should be ranked higher in the list of filters presented. Finally, the framework aggregates the scores computed from these sources as depicted by A_s step in Figure 1 and generates final scores for each navigable attribute for queries, similar to the first stage. Some search-terms are very similar based on the products ordered using them. We internally maintain an id called ReificationID to represent similar search-terms. We also aggregate the filter scores computed using category signal and engagement signal at such ReificationID-level. The framework also leaves flexibility for future extension by introducing other approaches and models to predict filters for search-terms as indicated by the *'Additional Techniques'* block. Furthermore, similar to the category-filter stage there can be manually set rules and hand-crafted curations for different queries to prioritize some and deprioritize other filters. The framework also aggregates this set of rules in E_s step before making the ultimate filter suggestions for queries.

In real time when a customer searches for a search-term (search experience), we first try to match it using a ReificationID. For example, if we get a query *'christmas lights mesh'* with a ReificationID as *'1234'*, then we try to pull filters suggestions for the ReificationID *'1234'*. This would have aggregated filter scores for similar search-terms like *'christmas lights mesh'*, *'christmas lights'*, *'christmas lights net'*, etc. If there is no ReificationID match, the framework then tries exact search-term match. In this case we will look for filter engagement scores for exact *'christmas lights mesh'* search-term. If the incoming search-term does not even get an exact match, then we rely on the Query-to-Category mappers to assign a category. We then show applicable filters from this category to represent filters for the search-term. If *'christmas lights mesh'* is mapped to *'Lights>Christmas Lights'* category by the Query-to-Category mappers, then we pull filters suggested in the first phase for category *'Lights>Christmas Lights'* and suggest the same as filters for *'christmas lights mesh'*. If any *'Additional Techniques'* are used to suggest filters for the same search-term, we then aggregate the scoring using a learned weighted sum as depicted by A_s step in Figure 1 to generate the candidate filter suggestions and their rankings. The final suggestions would then be generated after applying the hand curated rules to these candidates in E_s step.

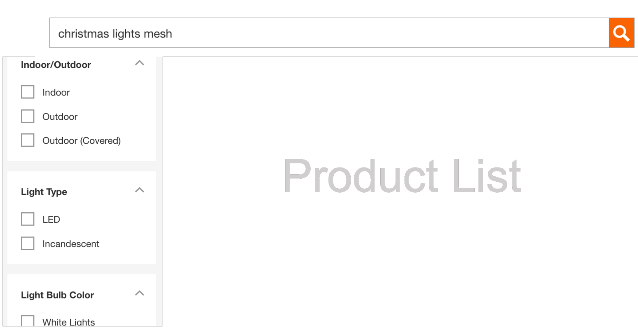

Figure 2: Filter suggestions for a search-term

4 RESULT

We conducted A/B tests for both Browse and Search Experiences on our e-commerce platform across desktop and mobile devices. The test experience used our proposed framework to discover and rank filters for Search Experience and only to rank the existing manually selected taxonomy-level filters for the Browse Experience (we were limited to not change the taxonomy-level curated filter list for the Browse Experience). The control experience used a manually curated set of filters for both Search and Browse. Hence, the major difference between the control and test group is the order of the filters. Only the Search Experience in the test group presented the dynamically generated facet list covering many relevant filters that were never shown to customers. The hypothesis is that re-ordering filters and surfacing new relevant filters driven by customer engagement and inherent knowledge from category-product-attribute mappings will lead to an increase in engagement with filters and improve revenue metrics. The key metrics used are click-through rate(CTR), add to cart rate(ATC), revenue per visit(RPV), average order value (AOV), revenue per visitor (RPV) and conversion. For the search experience, we saw a statistically significant increase in filter engagement by 6.5 percent and 3.2 percent in desktop and mobile channels, respectively. Similarly, we also observed an increase of 0.4% and 1% increase in AOV in desktop and mobile channels, respectively. For the browse experience, we again saw a statistically significant improvement in filter engagement rate by 6 percent and 3 percent in desktop and mobile, respectively. Similarly, we again observed an increase of 0.4% and 1% increase in AOV in desktop and mobile, respectively. We also saw a statistically significant improvement of 11.3 percent in ATC for the mobile group. Overall, the results indicated that our proposed framework enhanced the relevancy of filter suggestions which increased the filter engagement for both desktop and mobile. Especially for mobile channel, the engagement improvements in turn resulted in improved ATC.

5 DISCUSSION AND CONCLUSION

Query-level facet suggestion making use of category-level suggestions allows us to suggest filters even if there is no direct filter engagement signal for the query or similar queries[2]. The framework proposes Query-to-Category mappers to enable this. Currently, we use the product purchase signal as it provides a very strong signal to link product categories with queries. This can be easily extended by training ML models using several signals like product impressions, clicks, ATCs and orders. After associating suitable categories to queries, we use the filter suggestions for those categories to enhance the filter suggestions for the corresponding queries.

The framework also provides the ability to seamlessly plug in other Category-to-Filters and Search-terms-to-Filters prediction models. This flexibility of adding new techniques and models is indicated by the *'Additional Techniques'* box in Figure 1.

The framework heavily relies on filter engagement i.e., filters clicked by customers when searching or browsing. If we only rely on such direct engagement signals, we will not be able to suggest any filters when there is no engagement e.g.[6]. Hence, we also use other features exploiting the category-product-attribute to discover what attributes can be used as effective filters. Predicate probability, value cardinality, coefficient of variance, Gini and uniqueness to category are some of such features we use while predicting filters for categories. We also utilize ReificationIDs to aggregate filter scores from similar search-terms.

Searching and Browsing Categories in today's e-commerce setup requires a low-latency response. One of the key aspects of the proposed framework is the offline precomputation of desirable filters and their ranking for existing search-terms and categories. During serving time we can directly use these selected filters and save some precious time. All the activities inside the outermost dashed box in Figure 1 happen offline (in daily batch mode) and computes category specific filter suggestions and query/ReificationID specific filter suggestions.

A/B tests conducted with our filtering framework on our e-commerce platform for both Browse and Search Experiences showed statistically significant lifts in filter engagements and ATCs across desktop and mobile channels.

REFERENCES

[1] Lidia Ceriani and Paolo Verme. 2012. The origins of the Gini index: extracts from Variabilità e Mutabilità (1912) by Corrado Gini. *The Journal of Economic*

Inequality 10 (2012), 421–443.

[2] Patrick John Chia, Bingqing Yu, and Jacopo Tagliabue. 2021. " Are you sure?": Preliminary Insights from Scaling Product Comparisons to Multiple Shops. *arXiv preprint arXiv:2107.03256* (2021).

[3] Leila Feddoul, Sirko Schindler, and Frank Löffler. 2019. Automatic facet generation and selection over knowledge graphs. In *Semantic Systems. The Power of AI and Knowledge Graphs: 15th International Conference, SEMANTiCS 2019, Karlsruhe, Germany, September 9–12, 2019, Proceedings 15*. Springer International Publishing, 310–325.

[4] Hak-Jin Kim, Yongjun Zhu, Wooju Kim, and Taimao Sun. 2014. Dynamic faceted navigation in decision making using Semantic Web technology. *Decision Support Systems* 61 (2014), 59–68.

[5] Evgeniia Mikhailova and Eugene Steinberg. 2020 [Online]. Relevant facets: how to select and promote facets with deep learning. https://blog.griddynamics.com/relevant-facets-how-to-select-and-promote-facets-with-deep-learning/

[6] Jacopo Tagliabue, Bingqing Yu, and Marie Beaulieu. 2020. How to grow a (product) tree: personalized category suggestions for eCommerce type-ahead. *arXiv preprint arXiv:2005.12781* (2020).

[7] Damir Vandic, Steven Aanen, Flavius Frasincar, and Uzay Kaymak. 2017. Dynamic facet ordering for faceted product search engines. *IEEE Transactions on Knowledge and Data Engineering* 29, 5 (2017), 1004–1016.

[8] Damir Vandic, Flavius Frasincar, and Uzay Kaymak. 2013. Facet selection algorithms for web product search. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. 2327–2332.

[9] Bifan Wei, Jun Liu, Qinghua Zheng, Wei Zhang, Xiaoyu Fu, and Boqin Feng. 2013. A survey of faceted search. *Journal of Web engineering* (2013), 041–064.

[10] Yinan Zhang, Parikshit Sondhi, Anjan Goswami, and ChengXiang Zhai. 2020. Towards a Soft Faceted Browsing Scheme for Information Access. *arXiv preprint arXiv:2002.08577* (2020).

[11] Yongjun Zhu, Dongkyu Jeon, Wooju Kim, June S Hong, Myungjin Lee, Zhuguang Wen, and Yanhua Cai. 2013. The dynamic generation of refining categories in ontology-based search. In *Semantic Technology: Second Joint International Conference, JIST 2012, Nara, Japan, December 2-4, 2012. Proceedings 2*. Springer, 146–158.