Featured Keywords: Enabling Product Discovery in E-Commerce Through Unstructured Product Attributes

Janani Balaji
The Home Depot
Greater Atlanta Area, Georgia, USA
janani_balaji@homedepot.com

Goutham Simhadri
The Home Depot
Greater Atlanta Area, Georgia, USA
venkata_g_simhadri1@homedepot.com

Suhail Shaikh
The Home Depot
Greater Atlanta Area, Georgia, USA
suhail_shaikh@homedepot.com

Jeyaprakash Singarayar
The Home Depot
Greater Atlanta Area, Georgia, USA
jeyaprakash_singarayar@homedepot.com

ABSTRACT

Faceted browsing is now an integral component of most large scale e-commerce websites. Facets help customers quickly sift through the initial search results and identify a more targeted subset based on their preferred product features. However, the current faceting systems only enable filtering based on structured product attributes, thereby limiting the scope for product discovery. In this paper, we introduce a novel facet that leverages customer behavior and product features extracted from unstructured product metadata to provide a centralized gateway for filtering by both structured and unstructured product attributes. We implemented this feature on the entire catalog of products at The Home Depot and our A/B tests showed a significant increase in customer engagement.

CCS CONCEPTS

• **Information systems → Query suggestion**; **Query log analysis**; **Information extraction**.

KEYWORDS

facets, product discovery, keyword extraction, keyword ranking

ACM Reference Format:
Janani Balaji, Goutham Simhadri, Suhail Shaikh, and Jeyaprakash Singarayar. 2021. Featured Keywords: Enabling Product Discovery in E-Commerce Through Unstructured Product Attributes. In *Proceedings of ACM SIGIR Workshop on eCommerce (SIGIR eCom'21)*. ACM, New York, NY, USA, 8 pages.

1 INTRODUCTION

Facets are the quintessential e-commerce tools that provide customers greater flexibility in product search, by enabling simultaneous query refinements by several dimensions. While the principal intent for a search is set through the initial query, most of the refinement happens post-hoc, through an optimal combination of facets. In addition to helping users filter their choices, facets can also help educate users about previously unknown product attributes, thereby propelling parallel product engagement. Accordingly, E-Commerce retailers have responded by providing more intuitive, exhaustive and seamless faceting experiences. We now see several intuitive input modalities, like slider bars for continuous numeric features and visual facets, being commonplace in many major E-Commerce platforms.

At The Home Depot (THD), our faceted browsing aims at enabling a seamless product discovery and navigation experience for customers by providing several search refinement strategies. Customers can either utilize facets catered towards key product features, or generic filters such as price, brands or based on available fulfilment options, to name a few. Figure 1 shows a subset of facets available to the user for the search term "Circular Saw" at The Home Depot. Users can either use the coarse-grained facets such as *Brand, Price* etc., or employ fine-grained ones such as *Cordless/Corded, Blade Diameter, Battery Platform* that cater to the specific functionality desired in a product. Our faceted navigation provides users a quick rundown of the available options and enables them to filter the results based on their specific requirements.

While these facets serve as excellent functional tools in helping customers refine their search, they are still limited in scope. Customer-facing facets are typically derived from structured *key* → *value* attribute pairs, that are either manually curated or discovered from product metadata such as title and description. Even in the case of automatic discovery of product attributes, the discovery in limited to the *value* of the attribute, while the *key* typically belongs to one of the pre-defined attributes. The manual process involved in extracting product attributes for faceting, severely limits the number of facets that can be associated with a product and as a consequence poses multiple challenges to faceted navigation.

Firstly, since facets are manually curated, they do not represent the maximal set of features available for a product. And even if one were to extract all key features for a product, the current approach to facet organization presents challenges with presentation. Currently, the most prevalent strategy for faceting involves grouping product attributes by their *key* and presenting all distinct *values* as filters available under the facet. Several user interface studies and online tests carried out at THD have shown that there is limited engagement for facets that are outside the visual scope of the user. With increased e-commerce activity from mobile devices and smaller screens, any additional facets presented outside of a device's

SIGIR eCom'21, July 15, 2021, Virtual Event, Montreal, Canada

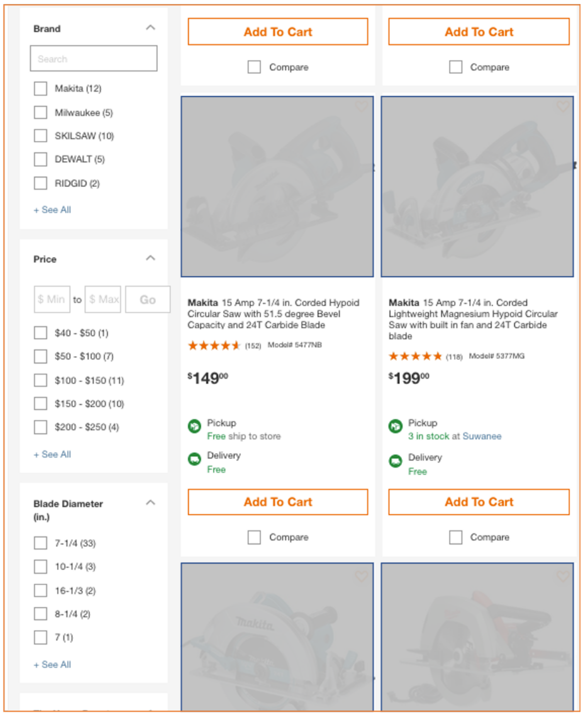


Figure 1: Example Facets at THD for Search Term "Circular Saw"

visual range would add little value in terms of customer engagement and ultimately product discovery. For example, considering our initial search term of "Circular Saw", at THD, there are a total of 18 product-specific facets, including the ones shown in Figure 1, listed in our search results page. Even if our discovery process resulted in, say, 5 additional facets, they would only add to the facet count, with little to no incremental value.

Secondly, though there is active research in dynamic facet ranking by incorporating customer engagement, the attributes used for faceting themselves are usually static key-value pairs that are associated with products and are agnostic of temporal shifts in customer demand. While the mapped key-value pairs might accurately define the salient features of a product, they might not be reflective of what users are actually looking for in a product. For example, in our experiments, we found that the key feature customers often look for in a "Circular Saw" is *Motor Power* $\longrightarrow$ *15 amp*. If *Motor Power* was presented as a facet to the ranking algorithm, it would have been ranked high to indicate customer interest. But it was not a catalogued product attribute and hence was not presented as a facet to customers. By not assimilating user behavior during attribute extraction, it cannot be guaranteed that the facets presented to the user represent the top features associated with a search term.

In this paper, we propose to solve both these challenges and create a seamless faceted browsing experience by introducing a new facet - Featured Keywords - that acts as a unified gateway for presenting popular product features mined from unstructured product metadata. We combine customer search trends and key product features extracted from textual product attributes such as title and description and use them to power a novel facet that enables filtering based on popular product features. Instead of using a separate facet for each *key-value* attribute pair, we present a catch-all search-able facet to include all popular keywords extracted from the product text. Large scale online experiments conducted indicated significant increase customer engagement and overall improvement in session revenue metrics. To the best of our knowledge this is the first attempt at providing a facet to surface popular free-form product features mined from unstructured product metadata at scale.

The rest of the paper is organized as follows: Section 2 provides a brief survey of the related research, Section 3 describes our solution approach, Section 4 details our experimental set up and evaluation and Section 5 concludes the paper with directions for future research.

2 RELATED WORK

The increase in online shopping by customers has led to an increased focus on improving E-Commerce user experience. A better user experience leads to increased customer loyalty and a positive brand perception [1]. Accordingly, there is active research in improving user experience for online shoppers, ranging from providing a personalized shopping experience [5], identifying query intent [12], product discovery [4] and navigation elements [16] to name a few. Since our work is about faceted searching, we will focus on the past research done towards improving facets.

Koren et al [6] aimed to provide a personalized faceting experience by leveraging explicit user ratings and using collaborative filtering to select facets and facet-values that are customized to a user's preferences. Several techniques to dynamically select facets that lead to an optimal drill down experience are proposed in [14] and [13]. Range optimization of numeric facets is studied by formulating it as an optimization problem in [8]. While all these methods improve faceting by optimizing the facet selection process, they do not look to incorporate free-form product features. Our method differs from existing research in faceted navigation by introducing a novel facet that uses free-form keywords extracted from unstructured product metadata and powered by customer behavior to help improve product discovery.

Since our model works on unstructured data, keyword extraction is one of the vital blocks of our work. This is a well studied problem in Natural Language Processing. Several unsupervised approaches for keyword extraction are studied and evaluated in [7]. The common structure is to define POS patterns of interest, filter the extracted set using a combination of syntactic (TF-IDF) or semantic (word2vec) contextual relationships and then use a clustering approach to de-duplicate the extracted keywords. Textrank [9] is a graph-based keyword extraction algorithm that uses co-occurrence relationships between words to run an iterative ranking algorithm to select top-ranked keywords. Topicrank [2] leverages topic modeling concepts by using LDA to identify key topics within the document and selecting one keyword for the most important topics. Ying et al.[15] extend Textrank and Topicrank by including sentence relationships in graph construction and ensuring keyword

Attribute	Value
Brand	Milwaukee
Name	M18 18-Volt Lithium-Ion Cordless 6-1/2 in. Circular Saw (Tool-Only)
Price	$119.00
Ratings	4.5
Cordless / Corded	Cordless
Power Tool Features	Depth Adjustment, Keyed Blade Change, Spindle Lock
Voltage (volts)	18
Batteries Included	No
Blade Diameter (in.)	6-1/2
Saw Drive Type	Sidewinder
Blade Location	Left
Battery Amp Hours	No Battery
Number of Total Batteries Included	0

(a) Categorized Features

The MILWAUKEE 15 Amp 6-1/2 in. Circular Saw may be light in weight at only 8.8 lbs, but packs plenty of power with a 15 Amp motor that can handle your toughest applications. Loaded with features that make cutting quick and easy, this saw has a 2-9/16 in. depth of cut capacity for versatile use and an anti-snag, ball-bearing lower guard for smooth guard operation and added durability. The clear line of sight helps give you blade visibility from any angle and with the integrated dust blower, your line of sight is kept clean for easier and more accurate cutting. Ideal for use on the job or for DIY projects of workshop projects, this saw is a breeze to use and built for long-lasting use.

- 57-degree beveling capacity with stops at 45 degree and 22.5 degree
- 2-9/16 in. depth of cut capacity provides additional versatility of applications
- Anti-snag, ball-bearing lower guard provides smooth guard operation and long life in harsh environments
- Tough cord provides longer durability against cord pull outs
- Integrated dust blower function cleans the line of sight during cutting
- Durable high-grade aluminum smooth base for accurate cuts
- Clear line of sight aids in blade visibility from any angle
- On-board blade wrench storage for convenient wrench access
- Includes: 15 Amp 6-1/2 in. lightweight circular saw, carbide tipped 7-1/4 in. blade, blade wrench and instruction manual
- 3-year limited warranty; 1-year free service contract; 90-day money back guarantee

(b) Product Text with free-form features

Figure 2: Categorized and Free-Form features for a Circular Saw

diversity across all topics. Several such variations for graph based keyword extraction are studied in [18] and [11].

Recently, several neural network-based approaches, that model keyword extraction as a sequence learning problem, have been proposed. Two hidden Recurrent Neural Networks(RNN) are used to jointly process keyword ranking, keyphrase generation and keyphrase ranking in [17]. Both LSTM and Bidirectional LSTM(Bi-LSTM) have been popular for entity extraction from textual data. Tang et al use a Bi-LSTM network with attention and leverage BERT [3] to extract keywords from noisy clinical notes.

In our model, keyword extraction is used as a base task for generating candidates for our ranking module which eventually blends customer behavior to rank the extracted keywords. While any of the above mentioned keyword extraction techniques could be reasonably applied, we focused on coverage rather than accuracy for the keyword generation step. After experimenting with Textrank and Topicrank as possible keyword generation techniques, we noticed that we obtained better coverage by sticking to basic POS pattern extraction and use TF-IDF to filter noisy keywords. More details about our keyword extraction technique are given in Section 3.3.

3 METHODOLOGY

This section establishes the problem statement, the challenges and our solution methodology. We will first start with a brief background into the problem and define some terminology to establish context.

The term *Facet* refers to the name of the filter group available to the users and *Facet Value* refers to each individual value for the corresponding facet. Referring back to Figure 1, *Blade Diameter* is a *Facet* while *7-1/4* is a *Facet Value*. The *Facets* and *Facet Values* are derived from the catalogued *Product Attributes* and *Product Attribute Values* respectively. For example, *7-1/4* is an identified value for attribute *Blade Diameter* for a given product. At THD, we employ a mix of manual and automatic extraction process to extract *Attribute-Value* pairs for products. Generation of these *Attribute-Value* is beyond the scope of the paper and we assume each product to be associated with a set of *Attribute-Value* pairs that describe its features.

As stated in Section I, facets are typically extracted from structured attribute-value pairs present in product metadata. Our aim is to go beyond catalogued attributes and present popular free-form keywords extracted from product text, as possible faceting options. For example, Figure 2 shows the categorized attributes for a product and highlights the potential non-categorized free-form features that can be used to increase the range of faceting choices. Rather than strictly limiting faceting to the structured and mapped product features, presenting all free-form product features in an additional independent facet broadens the faceting experience for customers.

Our goal is to extract product features from textual product metadata such as *Title and Description*, rank them based on popularity with respect to a search term and present all free form search terms in a new search-able facet. Instead of separating the product features into different facets by grouping on the attribute key, our solution is to present all top features associated with a search term within one facet, aptly named **Featured Keywords**, to provide customers with a one-stop-shop for searching based on product features. Since the keywords displayed within the facet are ranked based on customer behavior, top ranked keywords for a search term are readily accessible to the customer. In addition, we also provide a search box on the facet to enable customers to search for their desired keywords and not be constrained by the visual dimensions. By providing this unified faceting experience, in addition to the

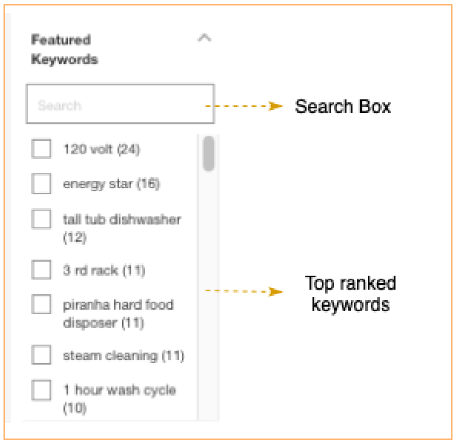


Figure 3: Sample "Featured Keywords" facet for Query = Dishwasher

traditional facets, users have single-point access to all top ranked product features in place place, rather than search across several facets. The **Featured Keywords** facet for search term *Dishwasher* is shown in Figure 3.

3.1 Challenges

At the onset, this may present itself as a trivial keyword/keyphrase extraction problem, which is extensively studied and documented. However, there are two challenges that increase the complexity of our task beyond simple keyphrase extraction.

Firstly, since the extracted keywords are going to be placed in a high-visibility customer-facing asset, it is imperative that they are well-formed and relevant. Existing keyword extraction techniques documented in Section 2 work by exploiting language dependency parsing techniques. They start by defining a set of Parts-Of-Speech (POS) patterns and then extract those word sequences that conform to it. However, due to inconsistencies in the language parsing models, we have observed the output to contain a high degree of noise resulting in sequences that are not well-formed. Textual product metadata like product description, product highlights can also contain other extraneous information about the brand, legal text, shipping information etc., which can result in unrelated keywords being extracted. Our task therefore is not limited to extracting product features, but to also ensure they are complete, well-formed word sequences.

The second challenge is to rank and order these keywords so that only the top-K keywords for a search query are presented to the customer. Though keyword extraction is product-specific, the final set of keyphrases presented to the user is search-specific. Strictly speaking, the keywords extracted from all the products that appear in the search results for a given search query are aggregated to be presented to the user. Hence it is vital to incorporate customer behavior to identify the tok-K distinct keywords that are most relevant to the user's query. The rest of this section details our approach to solve both these challenges.

3.2 Problem Statement

Given a search query, our aim is to retrieve all relevant keywords from the textual attributes of products appearing in the search result and rank the keywords based on the popularity of the feature in relation to the search query.

Problem Definition: *Let P be the set of products and let F_P be the set of keywords extracted from product P. Let Q be a search query and P^Q be the set of products in the recall for query Q. Our aim is to find set of distinct product features K^Q such that*

$$K^Q \subseteq \bigcup \{F_i \in F_P \quad \forall P \in P^Q\} \tag{1}$$

subject to

$$\Theta(K_i^Q) > \Theta(K_j^Q) : \quad i < j \quad \forall i, j \in K^Q$$

where $\Theta(K_i^Q)$ is a function that defines the popularity of the keyword K_i^Q w.r.t. search query Q.

Our problem can be broken down into three parts as follows:

(1) Extract keywords from product text
(2) Rank the keywords based on their popularity with respect to the search query
(3) De-duplicate and present the top-k distinct product features for faceting

3.3 Keyword Extraction

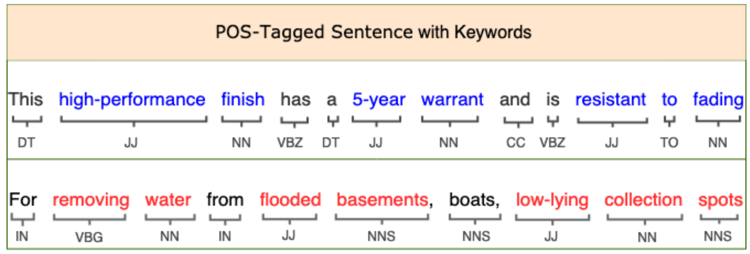
Keyword extraction is a well established technique in Natural Language Processing, as outlined in Section 2. The general framework is to tokenize the text, tag the parts of speech using a POS tagger, define POS patterns that match the intent and extract sequences of words that fit the described patterns. Most keyword extraction techniques that depend on POS pattern extraction look for sequences of nouns and adjectives. For our case, since we are interested in keywords that describe product features, our POS patterns were around different combinations of *Noun*, *Verb* and *Adjective* forms. Figure 4(a) lists a few sample POS patterns used in the extraction process. We also added special cases to identify and extract dimension attributes. Since our goal was to extract crisp, informative keywords with good discriminating value, we extracted keyphrases between 2 to 5 words long, as we found unigrams to be noisy and to add less value. A few sample keyword candidates extracted by our model are highlighted in Figure 4.

While the POS patterns we defined ensured that we extracted all possible word sequences describing product features, it also resulted in noise in the form of irrelevant or incomplete keywords. For example, consider the sentence *"For removing water from flooded basements, boats, low-lying outdoor collection spots"*. Our extraction model extracted the phrases *flooded basements* and *low-lying outdoor collection spots*. As per the POS tags indicated in Figure 4 (a), these are valid keywords, but they are incomplete and do not serve in our context, especially considering their customer facing placement.

We tried several approaches to mitigate this issue, including using topic modeling techniques [2] to form clusters of keywords belonging to key topics in the document and using word2vec [10] embeddings to determine keyphrase similarity with the product

POS Patterns
(<JJ.?>|<NN.?>)? <VB[^Z]>+ <NN.?>+ (extracts adjective, noun and verb combinations)
<NN[^S]?>? <JJ.?>+ <NN[^S]?>* <NN.?> (extracts noun and adjective combinations)

(a) Sample POS Patterns



(b) POS tags and keywords extracted for two sample sentences. Relevant keywords are highlighted in blue and non-relevant ones in red

Figure 4: POS Patterns and Keywords Extracted

text. While these strategies reduced noise, they also resulted in significant decrease in coverage. This is because, product descriptions can sometimes be short texts, with little repetition of words describing product features. So, extraction models that rely on co-occurrence statistics did little to boost such keywords. Similarly, using the semantic similarity of the keywords to the product text also did not improve context due to the loose coupling between product keywords and the products themselves.

So, we resorted to using all keywords extracted from our defined POS patterns and performed a light-weight de-duplication using the stop-words removed, stemmed forms of the keywords. We then used the TF-IDF scores of the keywords to determine relative importance and filter out noisy results by using a parameter ω as the lower limit for the TF-IDF score of the keyword. We also filtered out keywords that were present in less than two products.

3.4 Keyword Ranking

Once we have candidate phrases extracted, the next step is to rank and select top-K keywords for each search term. As previously stated, while the extraction step considers each product in isolation, the ranking consolidates all keywords extracted from all products in the result of a search query. To make the final set of keywords more intuitive and customer-driven, we used historical customer behavior to rank and filter our keywords.

We used two factors in determining the rank of a keyword in the context of a search query. First, the relevance to the search term to the product from which the keyword was extracted, which we call the query-product relevance (QPR). QPR ensures that the keywords from popular products for a query are ranked higher. Second, is the keyword-product relevance (KPR), which is the relevance of the keyword to the product from which it was extracted. KPR guarantees that keyword ranking is driven by customer behavior.

For calculating QPR, we used the normalized value of the number of *orders* a product P has received for search query Q.

$$QPR(Q,P) = \frac{Orders(Q,P)}{\max(\{Orders(Q,P_i) : \forall P_i \in P_i^Q\})} \tag{2}$$

where $Orders(Q,P)$ refers to the number of times product P was ordered when the search query is Q and P_i^Q are the set of all products in the search result for query Q.

QPR helps us rank keywords from top selling products high up the list. However, the bulk of the cleaning and prioritization is done by KPR which measures the popularity of the feature with respect to the search query. We assess the popularity of keyword K extracted from product P by measuring the number of times any search query in which the user engaged with P contained K. Our intuition here is that customers would often include the key feature they are looking for when they search for a product. For example, going back to the *Circular Saw* example, if a customer is particularly looking for a Circular Saw that has a *15 amp Motor*, it is highly likely that their search query is *15 amp Motor Circular Saw*. So, if a large number of users who engage(click, add to cart, order) with a Circular Saw product did so after issuing a search query that contained the term *15 amp Motor*, then we can reliably conclude that the term 15 amp Motor is highly popular in the context of a Circular Saw. We calculate the Keyword-Product Relevance (KPR) as follows:

$$KPR(K,P) = \frac{Clicks(Q^K,P)}{\max(\{Clicks(Q^{K_i},P) : \forall K_i \in F_P\})} \tag{3}$$

where K is a candidate keyword extracted from product P, Q^K is a search query such that $K \subseteq Q^K$ and $Clicks(Q^K,P)$ is the number of times that product P was clicked when the search query is Q^K. We used aggregated 1 year of data from the clickstream to calculate QPR and KPR. In addition to providing a reliable basis for ranking the keywords, KPR also helped remove irrelevant keywords since we only consider a keyword if it is part of a query issued by the user. In other words, we leverage customer search history to identify and remove non-relevant keywords.

Finally, the rank of keyword K extracted from product P with respect to search query Q was computed as

$$rank(K,P,Q) = \alpha \cdot KPR(Q,P) + (1-\alpha) \cdot QPR(K,P) \tag{4}$$

where α is a parameter that controls the contribution of the two terms. In our experiments we have observed better results when α favors KPR. At the end of the ranking process, for each search query, we have a ranked list of keywords as possible candidates to be included in the facet.

3.5 Clustering

Since the keyphrases are extracted from different products, there can be duplicates in the form of lexical and semantic variations. Also, since there is limited real estate, we would like to offer diversity in the keyphrases that we provide for faceting. For example, *resistant*

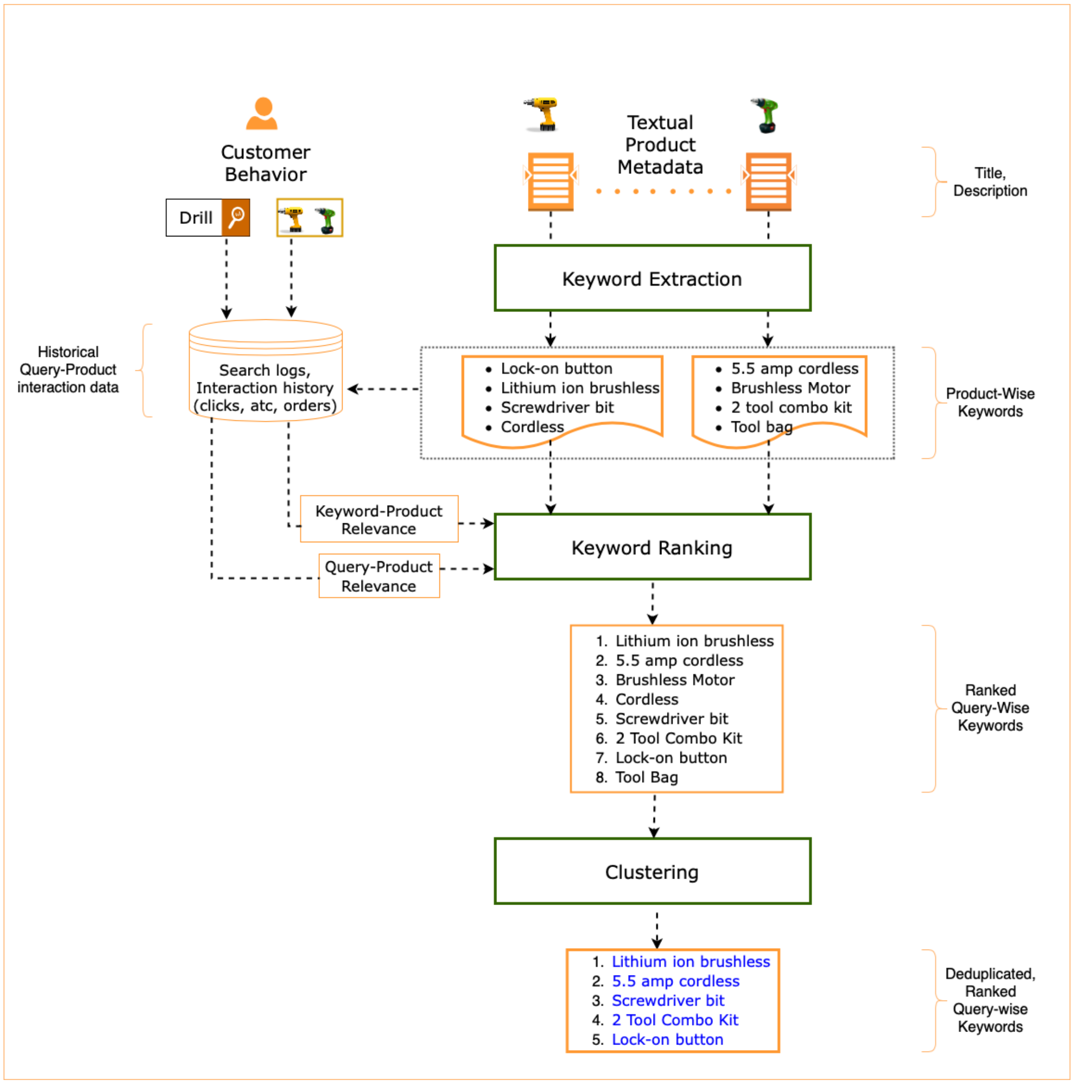


Figure 5: System Architecture

to rust and *rust resistant* are lexical variations of the same keyword. *mini refrigerator* and *small refrigerator* are semantically equivalent terms. We perform a round of clustering after the ranking process to group keywords by their semantic and lexical similarity and choose a representative keyword for each cluster to be presented in the facet. Let us first define the similarity metrics used in the clustering process.

To measure lexical similarity, we first run our keywords through the standard NLP pipelines of *Tokenization*, *Stop Work Removal*, *Stemming* and use the stemmed tokens to represent each keyword. We use *Jaccard Similarity* to measure the lexical similarity between two keywords.

$$Sim_{lexical}(K_i, K_j) = \frac{|Tokens(K_i) \cap Tokens(K_j)|}{|Tokens(K_i) \cup Tokens(K_j)|} \quad (5)$$

where $Tokens(K_i)$ represents the stemmed and stop word-removed tokens of keyword K_i.

To compute the semantic similarity, we used all the available textual attributes of a product as the corpus to train a word2vec model to learn vector representations of tokens. Vector embeddings for keywords were calculated by averaging the vectors of the individual tokens of the keyword. We used these vector representations to calculate the semantic similarity of the keywords.

$$Sim_{semantic}(K_i, K_j) = Cosine\_Sim(V(K_i), V(K_j)) \quad (6)$$

where $V(K_i)$ is the vector representation for keyword K_i. We defined the combined keyword similarity as:

$$\begin{aligned} Sim(K_i, K_j) =& \beta \cdot Sim_{lexical}(K_i, K_j) \\ &+ (1-\beta) \cdot Sim_{semantic}(K_i, K_j) \end{aligned} \quad (7)$$

where β is a factor that controls the weight of the lexical and semantic similarities. Having defined the measures of similarity used, let us walk through the clustering process.

The aim of the clustering is to remove duplicates and to increase diversity among the results. For each search query, we consider the top N keywords from our ranking algorithm as input for the clustering process. Our problem does not lend well to using traditional clustering techniques like K-Means or K-NN because the dynamic nature of the data makes it difficult to designate either a fixed number of clusters or a fixed number for cluster memberships. Since the number of candidate keywords for each search query varies widely depending on the number of products the search query returns, setting a fixed value of K, in K-Means or K-NN would result in unsatisfactory de-duplication results.

So, we used the Connected Components (CC) algorithm to perform our clustering. Since CC works by taking a pair-wise similarity graph as input and forms clusters by detecting the connected components in the graph, it dynamically adjusts the number of clusters based on available data. For each search query, we computed the pairwise similarity for the top-N ranked keywords and fed all pairs with the combined similarity score Sim above a threshold γ as input to the CC algorithm. The diversity factor γ can be used to control the cohesiveness of each cluster.

Once cluster assignments are made, we chose the top ranked keyword within each cluster as the cluster representative and return all cluster representatives as values to be listed in the *Featured Keywords* facet.

4 IMPLEMENTATION AND ONLINE EVALUATION

Our pipeline runs on Spark, an open source general-purpose distributed data processing system. All components of our system can be executed as offline, batch jobs that produce static {search query,keyword list} pairs that are consumed by the front end API. The system uses all the available products in our catalog as the corpus to extract keywords and uses 1 year's worth of customer interaction data from the click stream to compute popularity scores. Our experiments showed setting the thresholds $\alpha = 0.7$, $\beta = 0.7$ and $\gamma = 0.8$ yielded the best results.

We performed an A/B test for 2 weeks across the site to test our model and obtain a deeper understanding of the model performance. Our feature is primarily aimed at enhancing user experience and aiding faster product discovery, rather than affect the relevancy or ranking of the search results themselves. Accordingly, the metric we would most like to improve would be *Customer Engagement*, which measures the percentage of sessions in which a customer interacted with the facets. The results showed that our feature improved customer facet engagement across the site by **2%**, compared to the control group without the **Featured Keywords** facet.

Another interesting factor we observed was its impact in Average Order Value (AOV) and Revenue Per Visit (RPV). While the facet does not seem to impact conversion, by curating and listing popular features for products, it seems to have a positive consequence of increasing AOV and RPV. We observed a **2.9%** increase in AOV and **3.6%** increase in RPV. This can be attributed to the higher adoption of the facet for products with higher value ticket items where customers tend to deeply consider features before making a purchase. This is further corroborated by the fact that the top 5 search queries with the *Featured Keywords* facet were *Refrigerator, Ceiling Fan, Toilet, Counter Depth Refrigerator and Vinyl Plank Flooring*.

5 CONCLUSION AND FUTURE WORK

In this paper, we presented our approach towards designing a novel facet that serves as the one-stop shop for product discovery based on free-form product features. We use unstructured product metadata to extract key features for a product and use customer behavior signals to intelligently rank the most popular features for a given search query. We showcase the utility of this feature through our online experiments that showed overall increase in customer engagement and revenue.

In its current form, our methodology suffers from the cold-start problem and does not cater well to search queries without sufficient customer interaction data. In future, we would like to model this as a sequence learning problem and experiment using Bi-LSTM to automatically extract popular key phrases with limited customer engagement data.

REFERENCES

[1] Maritza Cieza Anchahua, Luis Vives Garnique, and Javier Alvarez Tarazona. 2018. User Experience Maturity Model for Ecommerce Websites. In *2018 Congreso Internacional de Innovación y Tendencias en Ingeniería (CONIITI)*. 1–6. https://doi.org/10.1109/CONIITI.2018.8587080

[2] Florian Boudin. 2016. pke: an open source python-based keyphrase extraction toolkit. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*. Osaka, Japan, 69–73. http://aclweb.org/anthology/C16-2015

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018). arXiv:1810.04805 http://arxiv.org/abs/1810.04805

[4] Ruocheng Guo, Xiaoting Zhao, Adam Henderson, Liangjie Hong, and Huan Liu. 2020. Debiasing Grid-Based Product Search in E-Commerce. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery Data Mining* (Virtual Event, CA, USA) *(KDD '20)*. Association for Computing Machinery, New York, NY, USA, 2852–2860. https://doi.org/10.1145/3394486.3403336

[5] Clare-Marie Karat, Jan Blom, and John Karat. 2003. Designing Personalized User Experiences for ECommerce: Theory, Methods, and Research. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA) *(CHI EA '03)*. Association for Computing Machinery, New York, NY, USA, 1040–1041. https://doi.org/10.1145/765891.766138

[6] Jonathan Koren, Yi Zhang, and Xue Liu. 2008. Personalized Interactive Faceted Search. In *Proceedings of the 17th International Conference on World Wide Web* (Beijing, China) *(WWW '08)*. Association for Computing Machinery, New York, NY, USA, 477–486. https://doi.org/10.1145/1367497.1367562

[7] Feifan Liu, Deana Pennell, Fei Liu, and Yang Liu. 2009. Unsupervised Approaches for Automatic Keyword Extraction Using Meeting Transcripts. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (Boulder, Colorado) *(NAACL '09)*. Association for Computational Linguistics, USA, 620–628.

[8] Xueqing Liu, Chengxiang Zhai, Wei Han, and Onur Gungor. 2017. Numerical Facet Range Partition: Evaluation Metric and Methods. In *Proceedings of the 26th International Conference on World Wide Web Companion* (Perth, Australia) *(WWW '17 Companion)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 662–671. https://doi.org/10.1145/3041021.3054195

[9] Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing Order into Text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Barcelona, Spain, 404–411. https://www.aclweb.org/anthology/W04-3252

[10] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. arXiv:1301.3781 [cs.CL]

[11] Zhou Qingyun, Fang Yuansheng, Shang Zhenlei, and Zhong Wanli. 2020. Keyword Extraction Method for Complex Nodes Based on TextRank Algorithm. In *2020 International Conference on Computer Engineering and Application (ICCEA)*. 359–363. https://doi.org/10.1109/ICCEA50009.2020.00084

[12] Parikshit Sondhi, Mohit Sharma, Pranam Kolari, and ChengXiang Zhai. 2018. A Taxonomy of Queries for E-Commerce Search. In *The 41st International ACM SIGIR Conference on Research Development in Information Retrieval* (Ann Arbor, MI, USA) *(SIGIR '18)*. Association for Computing Machinery, New York, NY, USA, 1245–1248. https://doi.org/10.1145/3209978.3210152

[13] Damir Vandic, Steven Aanen, Flavius Frasincar, and Uzay Kaymak. 2017. Dynamic Facet Ordering for Faceted Product Search Engines. *IEEE Trans. on Knowl. and Data Eng.* 29, 5 (May 2017), 1004–1016. https://doi.org/10.1109/TKDE.2017.2652461

[14] 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* (San Francisco, California, USA) *(CIKM '13)*. Association for Computing Machinery, New York, NY, USA, 2327–2332. https://doi.org/10.1145/2505515.2505664

[15] Yan Ying, Tan Qingping, Xie Qinzheng, Zeng Ping, and Li Panpan. 2017. A Graph-based Approach of Automatic Keyphrase Extraction. *Procedia Computer Science* 107 (12 2017), 248–255. https://doi.org/10.1016/j.procs.2017.03.087

[16] Fan Zhang, Siyuan Lin, Xuan Li, Yingbin Shuai, Hao Jiang, Cheng Yao, Fangtian Ying, and Feng Gao. 2017. Navigation Configuration and Placement Influences the Visual Search Efficiency and Preference. In *Proceedings of the 26th International Conference on World Wide Web Companion* (Perth, Australia) *(WWW '17 Companion)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 871–872. https://doi.org/10.1145/3041021.3054236

[17] Qi Zhang, Yang Wang, Yeyun Gong, and Xuanjing Huang. 2016. Keyphrase Extraction Using Deep Recurrent Neural Networks on Twitter. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, 836–845. https://doi.org/10.18653/v1/D16-1080

[18] Yong Zhang, Fen Chen, Wufeng Zhang, Haoyang Zuo, and Fangyuan Yu. 2020. Keywords Extraction Based on Word2Vec and TextRank. In *Proceedings of the 2020 The 3rd International Conference on Big Data and Education* (London, United Kingdom) *(ICBDE '20)*. Association for Computing Machinery, New York, NY, USA, 37–42. https://doi.org/10.1145/3396452.3396460