Eco-friendly Beauty & Fashion Product Search Engine

Lindsey Dye (dyel), Xinyu "Sylvie" Mei (xinyumei), and Pooja Thakur (tpooja)

Introduction

Our project focuses on developing a specialized, vertical search engine tailored for the beauty and fashion sectors, designed to provide users with a more intuitive and effective product discovery experience compared to general-purpose search engines. Unlike platforms such as Amazon or Google, which prioritize relevance based on sales data, popularity, or basic keyword matching, our search engine emphasizes user-centric features, particularly for eco-conscious consumers.

To enhance the user experience, our platform includes intuitive filters and eco-friendliness tags. These features allow users to seamlessly sort and refine their search results based on sustainability criteria such as "organic," "cruelty-free," or "biodegradable." The eco-friendliness tags serve as visual indicators, making it easier for users to identify sustainable products at a glance.

While general search engines like Amazon or Google may surface eco-friendly products, they often lack dedicated features to emphasize sustainability in their ranking or user interface. Such platforms typically require users to manually assess product descriptions for eco-friendly attributes, which can be time-consuming and inconsistent. By contrast, our search engine integrates sustainability directly into the ranking algorithm, ensuring eco-friendly products are prominently featured for queries where such attributes are relevant.

Additionally, we provide transparency by explicitly tagging products with their sustainability attributes and offering descriptions that highlight their environmental benefits. This approach aligns with the growing demand for ethical consumption while setting our platform apart from generic e-commerce search tools. It bridges the gap between users' intentions and the discoverability of products that match their values.

By prioritizing sustainability in both the ranking system and user interface, our search engine not only aims to empower users to make informed purchasing decisions but also encourage a shift towards environmentally responsible consumption in the beauty and fashion industries. This targeted approach makes our platform uniquely positioned to address the specific needs of eco-conscious consumers, fostering trust and loyalty through its user-centric and sustainability-focused design.

Data

Our dataset comes from Amazon-Reviews-2023-hugging face, which includes the data of over 48 million Amazon items from 33 categories. To make things easier, we chose only two categories among them, All Beauty and Amazon Fashion, with 938.5K items in total.

The first step in data cleaning was removing those items with empty descriptions to enhance data quality and consistency and to minimize the noise in the dataset. Also, from the end-user perspective, empty descriptions might appear confusing and unreliable. So including only items with detailed descriptions will maintain a higher standard of quality and trustworthiness in our search engine. While removing those items, we also combined All Beauty and Amazon Fashion together in one json file, with 493,293 items in total.

While loading the dataset, we also changed the data fields, keeping/adding/removing some of the attributes. The following table is the datafield of our processed dataset. The attributes we decided to remove from the original dataset are noted with breakthrough lines.

Field	**Type**	**Explanation**	**Reason to add/keep/remove**
docid	int	Added docid when filtering and combining the dataset.	To fit our implementation codes.
main_category	str	Main category of the product.	Could serve as a feature in ranking. Or a filter.
title	str	Name of the product.	Important feature in ranking.

average_rating	float	Rating of the product shown on the product page.	Could serve as a feature in ranking. Or a filter.
rating_number	int	Number of ratings in the product.	Could serve as a feature in ranking.
description	str	Combined *description*, *features*, *details* in the original dataset.	This will be the main part of feature engineering in ranking.
price	float	Price in US dollars (at time of crawling).	Could serve as a feature in ranking. Or a filter.
images	list	Images of the product. Each image has different sizes (thumb, large, hi_res). The "variant" field shows the position of the image.	We applied CLIP as a feature in ranking to improve the user interface.
link	str	Combined "https://www.amazon.com/dp/" with *parent_asin*	We added a link directing to the item to improve the user interface but had to generate these. A direct link instead of ASIN code would improve annotation efficiency.
~~videos~~	~~list~~	~~Videos of the product including title and url.~~	We are not planning to include video features. Most items are empty in this attribute.
~~store~~	~~str~~	~~Store name of the product.~~	Limited information.
~~categories~~	~~str~~	~~Hierarchical categories of the product.~~	Most items are empty in this attribute.
~~details~~	~~dict~~	~~Product details, including materials, brand, sizes, etc.~~	Combined to *description*
~~parent_asin~~	~~str~~	~~Item ID~~	Already modified to *link*
~~features~~	~~list~~	~~Bullet-point format features.~~	Combined to *description*
~~bought_together~~	~~list~~	~~Recommended bundles from the websites.~~	It's a great attribute to build network features, however, all items in the dataset are empty.

We crafted 60 keyword-based queries (e.g., "nail polish", "anti-aging night cream for sensitive skin", "maxi dress", "cardigan sweater with pockets") and indexed descriptions using BM25 ranking, selecting the top 50 items for each query to annotate. Annotation was on a 5-point scale, measuring relevance and eco-friendliness.

During annotation, we found that some items' *links* showed "page not found". This might be because the original dataset was collected in 2023, and some items have been removed since then. We first tried to filter our dataset by checking response status code and status text using the Python requests library. However, it took way too long to process the whole dataset (with ~80 hrs for 493K items). More importantly, it got stuck halfway due to frequent requests and website robot checks. For annotation, we pulled several more items from BM25 retrieval, and annotated the top 50 items with valid *link* for each query. We implemented link checking before showing the search results in the final search engine.

Related Work

In the domain of product search engines, various approaches have been proposed to enhance search relevance and user satisfaction. Our project focuses on developing an Information Retrieval (IR) system specifically for eco-friendly products within beauty and fashion, which introduces unique challenges distinct from general product search engines. This section discusses five key papers that provide context for our project, detailing their methodologies and how our approach builds on and refines them.

Lai and Kao (2018) study sustainable product search engines, proposing an eco-label system to highlight products with verified eco-friendly attributes. Their system emphasizes the importance of environmental data in user queries to improve relevance for eco-conscious consumers. Inspired by this, our project incorporates eco-friendliness as a primary facet, allowing users to filter results by sustainability characteristics. Unlike Lai and Kao, we focus specifically on beauty and fashion products, refining our dataset to just these categories and incorporating parent_asin links to address data gaps where product links were initially missing.

Vandic, Frasincar, and Kaymak (2013) propose a faceted search model that enables users to specify product attributes, such as price range and color, to refine searches. Their model uses an entropy-based approach to rank products based on the user's specific facets. Building on this concept, we extend the faceted search approach by introducing eco-friendliness as a key facet, prioritizing highly specific facets that better align with eco-conscious user intent. By applying this approach to a narrower dataset in beauty and fashion, we improve both the relevance and efficiency of the search, directly addressing the needs of environmentally minded consumers.

Blanco, Cambazoglu, and Mika (2016) investigate ranking models that utilize semantic attributes for e-commerce, such as brand and material, to improve product search relevance. They emphasize attribute-based ranking using known characteristics within product descriptions. In our project, we adapt these ideas to prioritize environmentally relevant facets (e.g., eco-friendly certifications, materials, and production methods). By focusing on the sustainability aspects in beauty and fashion, we tailor the ranking to meet specific consumer demands for eco-friendly products, allowing for nuanced comparisons across products that align with eco-conscious values.

Nguyen, Rao, and Subbian (2020) develop a deep learning-based product search model, utilizing complex relationships between query and product data. While effective, deep learning models often require extensive computational resources to develop and train from scratch. To balance computational feasibility with effective retrieval, we use the pre-developed LambdaMART model which instead uses Machine Learning techniques for listwise ranking. Unlike Nguyen et al., who prioritize query-product relationships at scale, we concentrate on optimizing the LambdaMART model for a focused collection of eco-friendly beauty and fashion products, which allows for relevance improvements without the need for intensive deep learning.

Virginia et al. (2021) combine BM25 with user click data to enhance ranking accuracy through historical interactions. While user interaction data can provide valuable feedback on product relevance, it can be challenging to collect and maintain at scale. Instead, we directly enhance our dataset by linking products with parent_asin identifiers, ensuring complete product information for eco-friendly items in our curated database. By refining BM25 with sustainable facets instead of user click data, we make our ranking model immediately effective in surfacing eco-friendly products without additional data overhead.

In summary, these foundational works provide valuable insights into product search engine design and implementation. Our project builds on these established methodologies, extending traditional models with targeted facets specific to eco-friendly products. By integrating these approaches, we address the growing consumer demand for sustainable choices while ensuring efficient and relevant search results.

Methodology

Baseline Methods

Below are a few methods that were used as a baseline for the system. These were created as basic methods that could be used to compare the final system's performance against.

Baseline 1 (using BM25):

The primary baseline is BM25, a probabilistic retrieval model that ranks documents based on term frequency and document length normalization. BM25 is widely regarded for its effectiveness in various retrieval contexts, making it a suitable baseline for our project.

In the context of our use-case, wherein our ultimate goal is to create an eco-friendly products' search engine, we first created a special product index that would only store the data of eco-friendly products. This was done by filtering out the products by checking if their descriptions contain a certain set of keywords that are generally found within the descriptions of eco-friendly products. This index is then used with a BM25 ranker to rank the available products for a given query.

The keywords found to be most useful were:

"sustainable, organic, biodegradable, recyclable, compostable, recycled, toxic, renewable, vegan, cruelty, FSC-certified, carbon, Fair Trade, climate, upcycled, responsibly sourced, pesticide, ethical, toxin, eco".

Baseline 2 (Naive ranking):

The Naive Ranking baseline leverages a simple term-frequency approach. For each query, it counts the frequency of query terms within each document and uses this as the ranking criterion. In other words, documents are scored based solely on how many times each query term appears in their content, without factoring in more complex relevance metrics like document length or term distribution across the dataset.

This Naive Ranking baseline serves as a straightforward, low-effort approach that helps us gauge the added benefit of more sophisticated scoring methods. For instance, if a user searches for "handmade recycled purse," documents mentioning "handmade" and "recycled" frequently in their descriptions would receive higher scores. This approach also serves as a valuable baseline by showing how much raw term frequency contributes to relevance.

Proposed Method

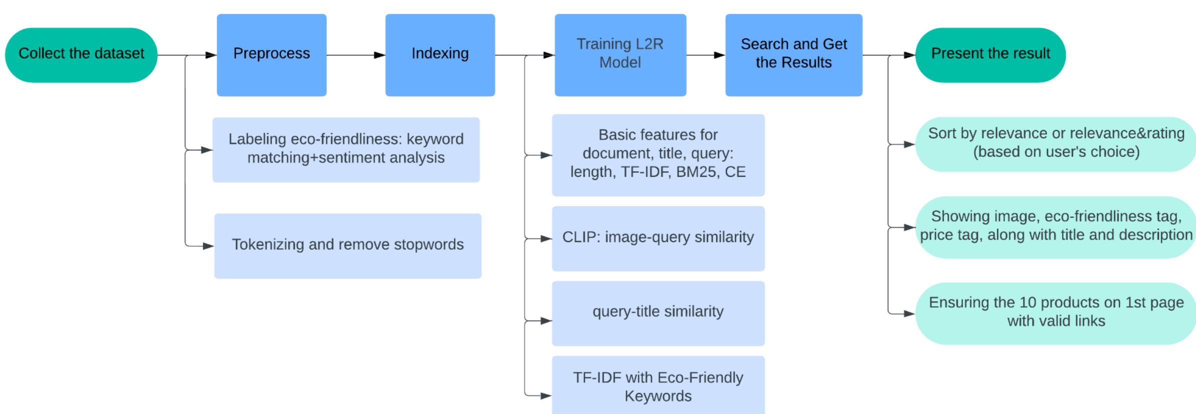

Figure 1: Flow chart of our search engine

Figure 1 is a workflow chart of our search engine.

Our proposed method for this use-case involves training the Learning 2 Rank (L2R) model, LambdaMART, with added features for our particular use-case. A few basic features including product description length, title length, TF, TF-IDF, BM25, Pivoted Normalization and Cross-encoder score are also being used. In addition to this, we add the following features to cater to our particular use-case:

Labeling eco-friendliness:

As eco-friendliness is an important feature of our search engine, we tried several methods to label the eco-friendliness for items as the last step of data preprocessing. Our first approach (also used in the baseline models) was simply matching some eco-friendly keywords we came up with by ourselves. Inspired by Ceren's feedback on our project update, we further tried some more methods to do the labeling.

As a result, we first added a negative keyword list containing words like "disposable" to also match non eco-friendly labels. Further, since the keyword list we came up with only contains tens of words, which is clearly not enough for a good labeling. So we utilized sentiment analysis, which would take into account some thesaurus aside from our keywords. We selected distilbert-base-uncased-finetuned-sst-2-english as the model, which is a text classification model based on DistilBERT and fine-tuned on SST-2. The workflow is as follows:

1. Label the whole corpus with our eco-friendly and non eco-friendly keywords list.
2. Randomly choose 5000 items after the initial labeling, and fine-tune the sentiment analysis model based on them, which would make the model more customized to our task.
3. Utilize the fine-tuned model to update the eco-friendliness labels, and manually check 50 of its updates. Turns out that most updates the sentiment analysis model made look reasonable. So we decided to take the updated labels.

TF-IDF with Eco-Friendly Keywords:

This feature uses a string of pre-defined eco-friendly keywords and calculates their TF-IDF score with product descriptions in the dataset. This is done to ensure that sustainably developed and eco-friendly products are ranked at the top. We know that the intuition behind TF-IDF is that it quantifies the frequency of a term in a given document and how rarely it appears in the rest of the corpus. In the context of our retrieval system, we deduced that this might be useful since -

a) Out of the large dataset of products, only a fraction of them are eco-friendly - meaning that the particular set of eco-friendly keywords are not expected to be observed in most product descriptions
b) More of the eco-friendly keyword terms a product description contains, the more dimensions of eco-friendliness the product satisfies.

Thus, the TF and IDF parts of this metric prove beneficial for our case.

Overlap between query and title:

We calculated the overlapping proportion (words overlapping with item title) in the query as a feature. We chose this feature because for a product search engine, the input queries tend to be keyword-based and to describe the product.

CLIP Similarity Feature:

This feature utilizes the Contrastive Language-Image Pre-training (CLIP) model to compute similarity scores between product images and queries. This multimodal approach allowed us to account for visual representations of eco-friendliness. We utilized *clip-ViT-B-32* model to encode the images and queries, and then computed the cosine similarity between them. As we have around 500K products of the data, which took ~20 hours to encode, we ran the function of image encoding first, and saved a dictionary with docids mapping to image embeddings, to ensure that it won't take too long to launch the search engine.

Learning-to-Rank (L2R):

We trained an L2R model using a combination of baseline features (e.g., BM25, TF-IDF) and custom features (e.g., keyword frequency, image-text similarity). The model was fine-tuned to rank products by relevance and eco-friendliness.

Evaluation and Results

We evaluated our model using several standard metrics, including MAP, NDCG, and F1 Score, to assess the performance of different ranking methods.

Metrics Used:

Mean Average Precision (MAP@10): Evaluates precision across the top 10 results.

NDCG@10: Measures ranking quality by considering the positions of relevant items.

F1 Score: Assesses the balance between precision and recall for relevance. It is calculated as a harmonic mean of precision and recall. In our use case, we use this score to judge how well the system is able to classify between a relevant eco-friendly product and a non-relevant one.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall};$$

Where we note that precision is the ratio of relevant documents to retrieved documents and recall is the ratio of relevant retrieved documents to relevant documents.

Results:

Baseline 1 (using BM25):

To evaluate the performance of our BM25 baseline system, we made use of the MAP@10 and NDCG@10 metrics. The evaluation was carried out on the set of annotated query-document pairs. Each member of the team annotated 20 query-document pairs, wherein each query was paired with 50 documents. These 50 potentially relevant products were obtained using a BM25 ranker applied on the usual product index (as opposed to the specially formed eco-friendly product index for this case). The relevance scores were assigned on the basis of: a) The relevance of the product to the query and b) Whether the product satisfied any criteria for being eco-friendly.

The MAP@10 and NDCG@10 results obtained for the BM25 baseline method are depicted using the graph below (with 95% confidence intervals) -

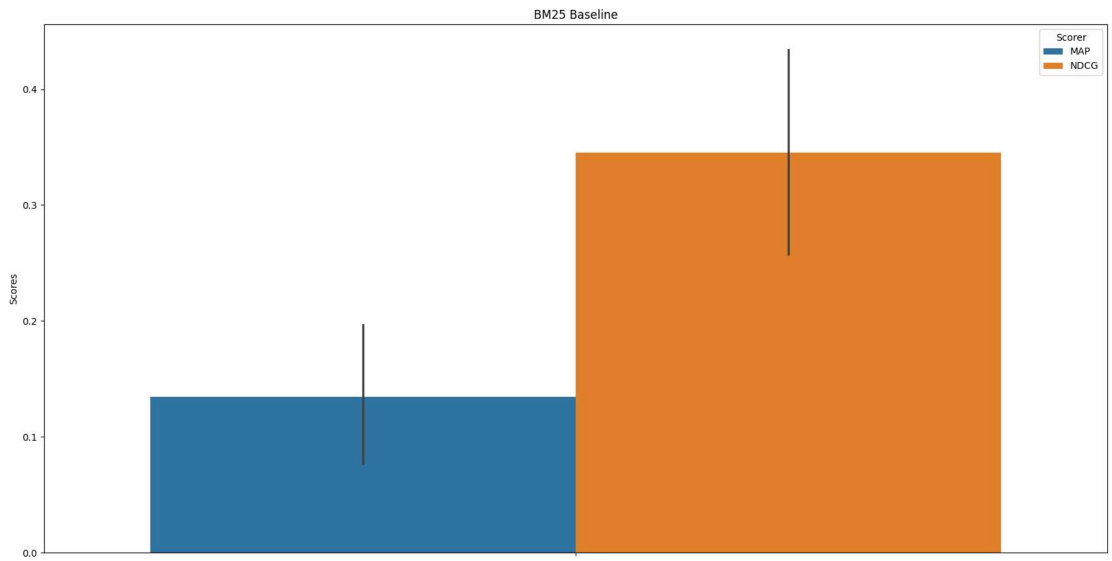

Figure2: Relevance score for baseline BM25 model

From the above graph, we can conclude that even with a baseline BM25 ranker system, we are able to obtain fairly valid results. We also compare the above results with the performance of a product search engine from our related work section, namely the one proposed by Virginia et al.

The comparative results are given below -

Query (Virginia et al.)	**MAP Score (Virginia et al.)**	**Query (our BM25 baseline)**	**MAP Score (our BM25 baseline)**
Laptop asus 8gb	1	Blush brush	0.2988
Jersey bola	0.8052	Eyebrow gel	0.3
Baju basket	1	Eyelash curler	0.2
iPhone 5 putih	0.7463	High waisted leggings with pockets	0.6080
iPhone 5	0.5357	Hydrating lip oil	0.5541
TV LED	0.8681	Leg warmers	0.8
Asus zenfone	0.8333	Long lasting waterproof mascara	0.5810
Sepatu adidas putih	0.6181	Wool scarf	0.4
Average	0.8009	Average	0.4677

The above table shows a comparison of the MAP scores obtained using our baseline BM25 ranker and the method proposed by Virginia et al. The data is presented in a manner similar to how it is given by the authors of [5] for easy comparison. We see that although the baseline model does not perform as well as their proposed model, its performance for certain queries is still comparable to their model.

Baseline 2 (Naive ranker):

We evaluate the Naive Ranker using the same MAP@10 and NDCG@10 metrics that were used for the BM25 baseline. This comparison helps us understand how much better or worse a simple ranker performs compared to the more sophisticated BM25 approach.

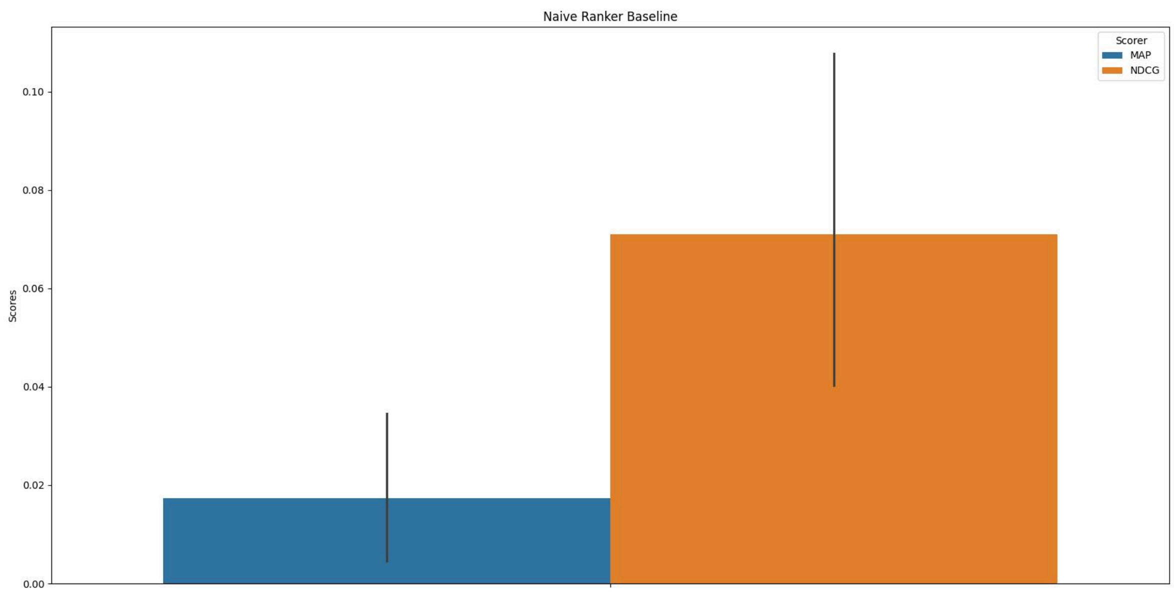

Figure3: Relevance score of naive baseline model

The above graph shows us that the Naive Ranker tends to produce lower MAP and NDCG scores compared to the BM25 baseline, as expected. The Naive Ranker performs well for simpler queries since it only considers the frequency of query tokens in the document. This tells us that a simplistic term-matching approach may struggle to capture the nuanced relevance of documents for more complex queries.

The Naive Ranker provides valuable insight into the importance of term matching in basic ranking tasks, but more sophisticated models like BM25 are better suited for more complex document relevance evaluations. The BM25 baseline remains more reliable overall for more diverse query types.

Proposed Method Performance:

We evaluated the L2R system using three standard metrics: MAP@10, NDCG@10, and F1 Score. These metrics assess the relevance, ranking quality, and balance between precision and recall. Learning-to-Rank (L2R), BM25, and Naive Ranking results are as follows:

	L2R	BM25	Naive
MAP@10	0.034	0.097	0.017
NDCG@10	0.145	0.34	0.07
F1 Score	0.097	0.094	0.048

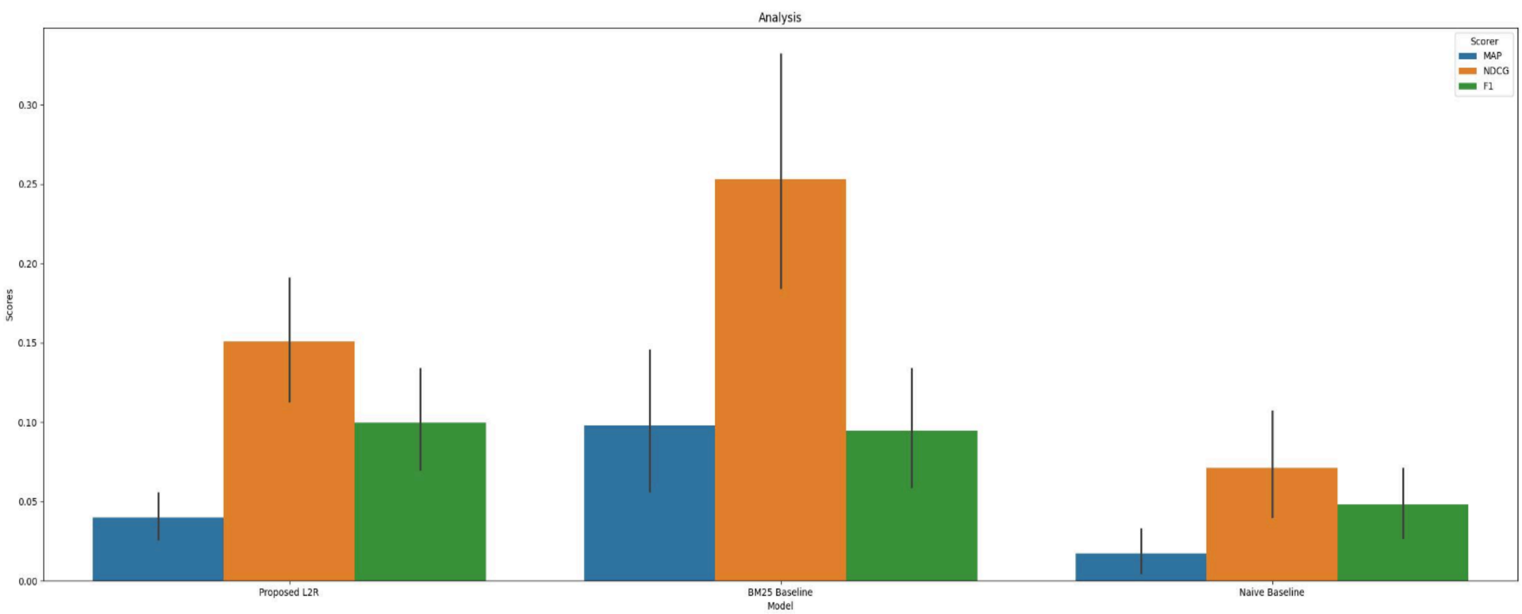

Figure 4: Analysis of scores of all models used with 95% confidence interval

The L2R system achieved a MAP@10 score of 0.034 and NDCG@10 score of 0.145, indicating moderate effectiveness in identifying relevant results and ranking them appropriately. The F1 Score of 0.097 demonstrates a reasonable balance between precision and recall, emphasizing L2R's capability to recognize relevant products while minimizing false positives. However, the performance of L2R was constrained by computational limitations during feature extraction and model training. The small size of the dataset and the use of resource-intensive features, such as embeddings for similarity calculations, impacted its ability to scale effectively.

Although the BM25 system achieved a better performance than L2R in this case, this may be attributed to the limited data that L2R was trained on. Being a highly sophisticated method in comparison, it is expected that it would require a good amount of training data to tune its hyperparameters for the best results. Since we had a total of 3000 annotated query-document pairs (only 70% of these were used for training) this could have potentially led to underfitting of the model. Despite these challenges, L2R successfully incorporated features specific to eco-friendly products, laying the groundwork for future improvements.

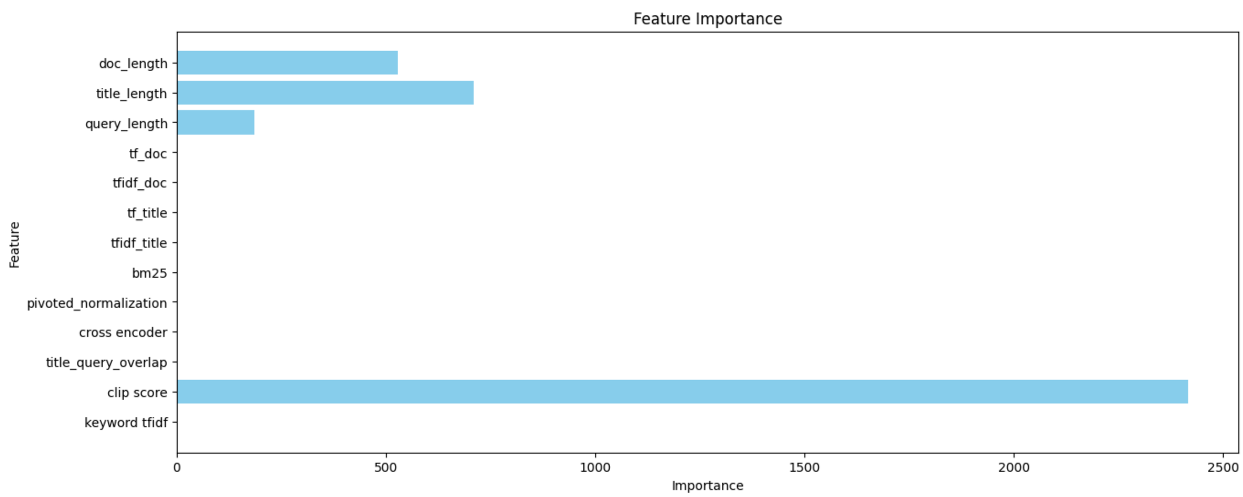

Figure 5: L2R Feature Importance Score

The feature importance plot highlights CLIP score as the most influential feature by a significant margin, suggesting the model heavily relies on the similarity between images and queries. Moderately important features include title_length and doc_length, while others, like query_length, have minimal impact, and features such as tf_doc and bm25 contribute almost nothing. This dominance of CLIP score raises concerns about over-reliance on a single feature. In the original dataset, there are several images for each item, while we only chose the image with label “MAIN” to calculate the CLIP score. In this way, the quality of the “MAIN” image would impact a lot on the result, which could be one of the reasons for

L2R’s worse performance than BM25. Evaluating the robustness of CLIP score and ensuring balanced feature contributions are essential next steps for optimization.

User Interface

Based on the web page implementation in assignments, we added some features to make it more user-friendly.

Sorting option bar:

In the home page of the search engine, we added a bar for users to select the sorting option. Aside from relevance, we implemented “Relevance & Rating” here, which weighted relevance score and item rating as 7:3 to compute a new score list.

Figure 6: User interface of the search engine homepage

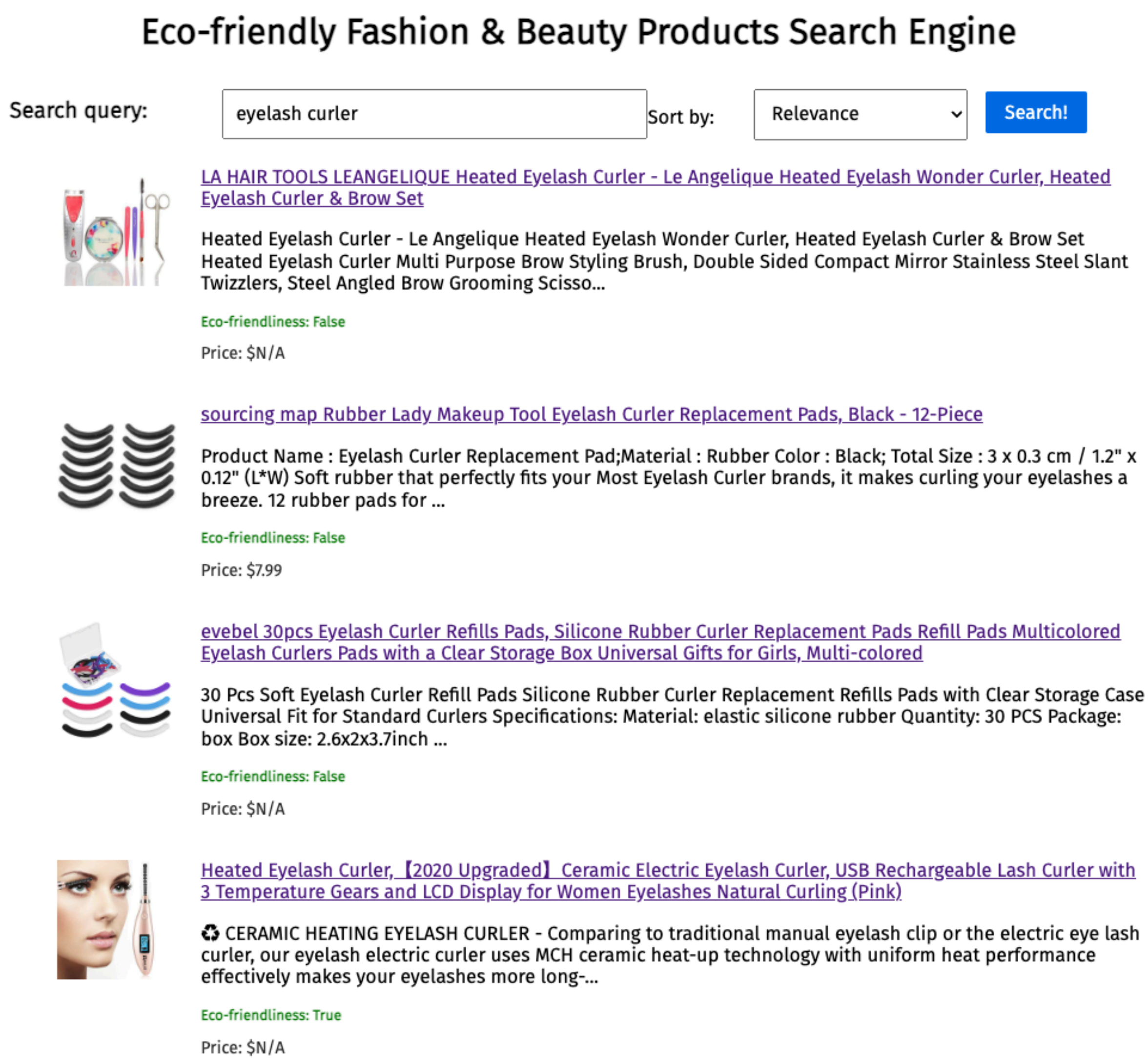

Figure 7: Sample of search engine result page

As shown in Figure 7, we also added some features in the result page:

Eco-friendliness tag and price tag:

We added an eco-friendliness tag and a price tag under the description in the search result page. The eco-friendliness tag shows whether the product is classified as eco-friendly in earlier steps. For the price tag, those items without price value in the dataset are labeled as 0.

Product image:

We added the product images on the left of other information.

Product link validation in first page:

As mentioned above, there are a lot of items whose links are already invalid in our dataset. And due to the data size and the CAPTCHA, it was not possible to run through the whole dataset to filter those with valid links. So we checked top results' links before printing out, to make sure that the 10 items on the first page have valid links.

Discussion

BM25 emerged as the most reliable ranking method, balancing performance and computational efficiency. Its success underscores the importance of robust term-based ranking methods in the absence of more advanced, resource-intensive models. While L2R incorporated additional features to improve relevance, its moderate performance reveals the challenges associated with large-scale data and feature complexity. Computational limitations hindered its ability to fully leverage its advanced capabilities, suggesting a need for optimization in future iterations. The underperformance of the Naive Ranker highlights the inadequacy of simplistic approaches for handling complex and context-sensitive queries. This reaffirms the importance of employing more sophisticated ranking algorithms for real-world applications.

Conclusion

This project demonstrates the potential of a domain-specific search engine to bridge the gap between consumer intent and product discoverability in the beauty and fashion sectors. By emphasizing sustainability, we empower consumers to make environmentally responsible choices more easily, supporting their values while fostering market demand for eco-friendly products. For consumers, this simplifies the search for ethical products, saving time and effort while aligning purchases with personal values. For businesses, this offers a platform to showcase sustainable products, enhancing brand visibility and appealing to eco-conscious customers. For the environment, our search engine encourages sustainable consumption patterns, potentially reducing the environmental footprint of the beauty and fashion industries.

Our framework could be integrated into existing e-commerce platforms to extend its reach and influence. For instance, collaboration with major retailers or integration with online marketplaces could bring eco-conscious shopping tools to a wider audience, amplifying the impact of sustainability-focused initiatives across industries.

Other Things We Tried

To ensure that the model ranks eco-friendly products higher in the context of any user query, we tried to use a cosine similarity feature between the product description and title embeddings and the keywords. The goal was to add this feature to the final L2R ranker. However, after all the product description and title data was encoded into embeddings using the 'sentence-transformers/msmarco-MiniLM-L12-cos-v5' model via the Sentence Transformer module the resulting file had a very large size of 1.5 GB. This possibly caused the model to throw the 'out of memory' error. Hence, although the method was theoretically sound - it could not be used in our context.

At first, we also wanted to add "sort by price" in the user interface, but in the dataset there are only ~9000 items with actual price value.

What You Would Have Done Differently or Next

Our current scoring mechanism could be improved by integrating more robust data sources, such as third-party certifications (e.g., Fair Trade, USDA Organic, Leaping Bunny) as key ranking factors. Evaluating the environmental impact of products across their entire lifecycle would also provide a more comprehensive eco-friendliness score, but would require more comprehensive data on these products.

Implementing user personalization is a high-priority enhancement. This feature would tailor search results based on user behavior, preferences, and past interactions, such as prioritizing vegan or cruelty-free options for users consistently selecting such products. Personalization would further refine relevance and improve user satisfaction.

Query modification could also be utilized to further enhance search relevance and user satisfaction. Query expansion, for instance, could be implemented to add synonymous or related terms to user queries, ensuring that the search captures a broader set of relevant results. Additionally, leveraging user feedback and click-through data, personalized query refinement strategies could be developed to adapt to individual preferences over time.

We did not use network features for this project because the "bought_together" attribute in the dataset is empty for all the items. If we could get data for this attribute, it would be helpful to create a network between all the items in the dataset, and include features like pagerank and HITS scores in L2R.

To address potential scalability challenges, future work could focus on automating periodic link validation to ensure the accuracy and availability of product data and incorporating APIs to fetch live data for inventory changes, pricing, and new product entries.

Team Work Distribution

Our team collaborated effectively by dividing tasks based on individual strengths and expertise. Sylvie extracted and combined the datasets and implemented the possible data search part to get preparation for annotation. She also implemented the function checking link validation, and tried on the whole dataset once (but failed). Each member annotated 20 queries, evaluating 50 documents per query using the BM25 ranking system.

For implementation, Pooja developed the BM25 baseline, cosine similarity for eco-friendliness (did not work out however), TF-IDF for eco-friendliness, and also helped debug the CLIP system and run evaluation tests of all systems along with introducing the F1 score. Lindsey implemented the naive ranking system baseline for comparison, while Sylvie utilized CLIP for image processing, implemented a sentiment analysis model to better label eco-friendliness, and implemented the user interface design.

In report writing, Sylvie authored the Introduction, Data, and the User Interface sections, drew a workflow chart, and modified What You Would Have Done Differently or Next section. Lindsey wrote the Related Work, Conclusion, and What You Would Have Done Differently or Next sections, and collaborated with Pooja and Sylvie on the Methodology and Evaluation sections. Pooja also created the Work Plan and kept it updated.

All team members contributed to proofreading and editing the report and code, ensuring clarity and technical accuracy. This division of labor enabled us to deliver a comprehensive and impactful project.

References

[1] Vandic, D., Frasincar, F., and Kaymak, U. "Facet selection algorithms for Web product search." *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*, 2013, pp. 2327–2332.

[2] Lai, T., and Kao, Y. "An eco-label-based search engine for sustainable product recommendations." *Journal of Environmental Management*, 2018, pp. 405-415.

[3] Blanco, R., Cambazoglu, B.B., and Mika, P. "Attribute-based ranking models for e-commerce search engines." *ACM Transactions on Information Systems*, 2016.

[4] Nguyen, T. V., Rao, N., and Subbian, K. "Learning robust models for e-commerce product search." *arXiv preprint arXiv:2005.03624*, 2020.

[5] Virginia, G., et al. "Search engine optimization by re-ranking product search results based on user click data." *Proceedings of the 3rd International Conference on Advanced Information Science and System*, 2021.