Beauty Beyond Words: Explainable Beauty Product Recommendations Using Ingredient-Based Product Attributes

Siliang Liu[1],*, Rahul Suresh[1] and Amin Banitalebi-Dehkordi[1]

[1] *Amazon BeautyTech*

Abstract

Accurate attribute extraction is critical for beauty product recommendations and building trust with customers. This remains an open problem, as existing solutions are often unreliable and incomplete. We present a system to extract beauty-specific attributes using end-to-end supervised learning based on beauty product ingredients. A key insight to our system is a novel energy-based implicit model architecture. We show that this implicit model architecture offers significant benefits in terms of accuracy, explainability, robustness, and flexibility. Furthermore, our implicit model can be easily fine-tuned to incorporate additional attributes as they become available, making it more useful in real-world applications. We validate our model on a major e-commerce skincare product catalog dataset and demonstrate its effectiveness. Finally, we showcase how ingredient-based attribute extraction contributes to enhancing the explainability of beauty recommendations.

Keywords

attribute extraction, beauty recommendation, ingredient analysis, explainability

1. Introduction

The value of the global beauty and personal care market is estimated to be over $646 billion in 2024 [1]. Product discovery and trust are two of the biggest considerations in Beauty customers' shopping journeys in e-commerce stores. Many factors contribute to these problems, such as lack of personalized recommendations, inaccurate or incomplete product benefit and/or ingredient information, lack of targeted curation, etc. Having such information accurately listed in the product catalogue is particularly important for Beauty category of products, as they are topically applied to the skin. Manual curation and sanitization of such metadata is possible at small scales. However, for larger e-commerce stores, with a large portfolio of products, it will be impractical to rely on manual annotation.

The primary objective of our work is to enhance the beauty shopping experience by automatically and accurately extracting beauty attributes at scale. These attributes not only aid customers in comparing and refining product choices but also foster trust in the e-commerce stores. Furthermore, the extracted attributes contribute to building more explainable beauty recommendations, which empower customers to make informed purchasing decisions.

We propose a robust and scalable learning-based solution capable of predicting beauty attributes from product ingredients. To achieve this, we integrate an energy-based implicit strategy to extract 5 skin types, 11 skin concerns, and 17 attributes commonly preferred across beauty products, as elaborated in Section 9.1. In summary, the key benefits of our proposed model are:

- Improved **accuracy** and **precision** compared to the alternatives
- **Explainability** through analysis of the attention weights (§5.4)
- **Robustness** in a low-resource regime via implicit data augmentation (§5.5)
- **Flexibility** when finetuning previously trained models on new labels (§6.2)

Bari'24: Workshop on Strategic and Utility-aware REcommendations held in conjunction with the 18th ACM Conference on Recommender Systems (RecSys), 2024, in Bari, Italy.

*Corresponding author.

celineli@amazon.com (S. Liu); surerahu@amazon.com (R. Suresh); aminbt@amazon.com (A. Banitalebi-Dehkordi)
0009-0007-4561-7548 (S. Liu)

To the best of our knowledge, there has been no prior study on the extraction of beauty-specific attributes based on product ingredients. Our contributions are outlined as follows:

- We introduce a novel energy-based implicit model for extracting beauty attributes from product ingredients and the title. We define *implicit* vs. *explicit* models in Section 3.
- Our proposed approach is assessed using skincare products from a major e-commerce store. We demonstrate its superiority over traditional keyword-based solutions and an explicit classifier baseline on a test dataset annotated by beauty domain experts.
- We document and extensively discuss the key algorithmic and architectural features that contribute to explainability, robustness, and flexibility of our proposed model.
- As a use-case study, we illustrate how ingredient-based extracted attributes can enhance the development of explainable beauty recommendations in Section 7.

2. Related Works

Attribute Value Extraction The problem of product attribute extraction in e-commerce is traditionally solved using named entity recognition (NER). NER approaches typically use beginning-inside-outside (BIO) tagging [2, 3] to segment texts. However, NER-based approaches exhibit substantial limitations due to their reliance on predefined entity types. This rigidity makes it difficult to scale in dynamic environments where attributes are numerous and constantly evolving, such as in beauty product recommendations. Certain research also models the attribute extraction task as a sequential tagging problem [4, 5] using CRF and BiLSTM. [6] describes a method that extracts attributes using a parameterized decoder with pretrained attribute embeddings, through a hypernetwork and a Mixture-of-Experts (MoE) module. [7] also model the attribute to make the prediction task more scalable. Our work is similar to the solution proposed in [7], which uses BERT and Bi-LSTMs to model semantic relations between attribute and product titles on a large-scale dataset. However, the deep learning modules in [7] are primarily used as components in the NER pipeline and the outputs of the model are still the BIO tags. Our work is different in that our proposed model directly outputs the attribute values and the architectural design choices are heavily guided by explainability, robustness, and flexibility.

In the direction of classification tasks, recent advancements utilize multitask framework and multimodality [8, 9, 10]. Furthermore, these models utilize parameter sharing across different attribute prediction tasks, reducing the model's complexity and encouraging generalization. Each attribute has its own output layer, allowing the network to predict multiple attributes simultaneously. On the other hand, prior works have demonstrated that incorporating an implicit method [11, 12] offers unique benefits. In particular, when treating product attribute extraction as an implicit classification problem—where attributes themselves are also part of the input—the model can focus on specific attributes to extract from the product description. This approach helps the model learn more meaningful and relevant embeddings from the input which leads to more accurate attribute value extraction.

Beauty Product Recommendation Extant literature provides limited research on beauty product recommendation that incorporates ingredient analysis [13, 14]. [15] directly uses an ingredient-concern mapping table to provide solutions for users of various skin conditions detected by an object detection computer vision model. However, this mapping table is often supplied by a third party where mappings are constructed independently for each ingredient without accounting for the order and the interactions with other ingredients, leading to inflexible rule-based recommendation methods. [16]'s approach extracts ingredient efficacy based on user reviews and recommends products containing those ingredients for customers across various age groups. Although this method relies on user-generated content, it does not align with our fact-based approach, making it inapplicable to our use-case scenario. [17] employs a method based on ingredient similarity using one-hot encoding to recommend products given a user's past purchase. However, this work does not leverage ingredient data to predict targeted skin types and concerns directly, which is the focus of our work.

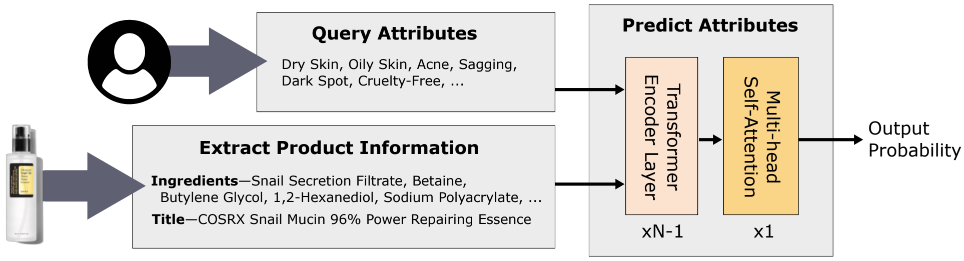

Query Attributes
Dry Skin, Oily Skin, Acne, Sagging, Dark Spot, Cruelty-Free, ...
Extract Product Information
Ingredients—Snail Secretion Filtrate, Betaine, Butylene Glycol, 1,2-Hexanediol, Sodium Polyacrylate, ...
Title—COSRX Snail Mucin 96% Power Repairing Essence
Predict Attributes
Transformer Encoder Layer
xN-1
Multi-head Self-Attention
x1
Output Probability

Figure 1: Overview of beauty product extraction workflow and the BT-BERT architecture. Our model is identical to the BERT Transformer [18] except in the last layer—the initial `N-1` layers remain unmodified. We remove the final MLP from the last layer of the Transformer encoder and directly use the self-attention values to formulate the output probability.

3. System Overview

We approach the beauty attribute extraction problem as a supervised multi-label classification task. Our proposed solution features a bidirectional Transformer encoder network similar to BERT [18], with a slight modification applied to the last attention layer as summarized in Algorithm 1. It is important to note that the network does not use the feed-forward layers in the last Transformer encoder block and does not have any additional classifier modules commonly used in downstream learning tasks. Instead, the logits are directly calculated from the attention values. We refer to our model as **BeautyTech-BERT**, or **BT-BERT** for short.

The model operates by taking as inputs a query attribute, a list of ingredients, and the product title, and producing the probability for the query attribute. Figure 1 shows an example use-case where the user is querying six attributes for a product titled “COSRX Snail Mucin Essence". Based on the product ingredients, the network will make an inference on whether to label the query attributes true or false. In this case, since `Betaine` is an ingredient known for its hydrating properties, the network is likely to predict true for `Dry Skin`, meaning this product likely benefits those who have a dry skin type.

Conceptually, our model can be viewed as an energy-based model (EBM) [19, 20, 21], as it assigns a normalized scalar (or "energy") to each input data point, thereby representing a probability distribution over the training data. We also denote our model as an *implicit model*, as it accepts the query attribute as input and generates a prediction solely for that attribute. This distinguishes it from conventional multi-label classifiers, where the classifier module and the number of output classes must be *explicitly* defined.

Model Input For each product, the query attribute is concatenated with ingredients and title to pass to the model. Maintaining the original sequence order of the ingredient list is essential, as it reflects the standard convention of listing higher potency ingredients first. We first tokenize the query label and pad query tokens up to a length of 3. The product ingredients and title are also tokenized. The entire sequence is truncated or padded such that the final length is 512. We place the query attribute at the beginning of the input sequence so that its position is consistent across all input sequences—similar to the effect of the `[CLS]` token in BERT when using it in downstream tasks—which is important for computing the logits.

4. Data Preparation

Our proposed method is a supervised learning approach and thus requires labeled training data. We first collect a dataset of skincare products from product data available publicly [22, 23]. For each product, attribute labels were meticulously annotated by domain experts based on years of scientific ingredient research. An example is shown in Figure 4. Overall, we collected a total of 11580 data points, where 9334 ($\approx 80\%$) are dedicated to training and 2246 ($\approx 20\%$) to evaluation. Figure 5 shows the distribution of products categorized by product types and attributes in our dataset.

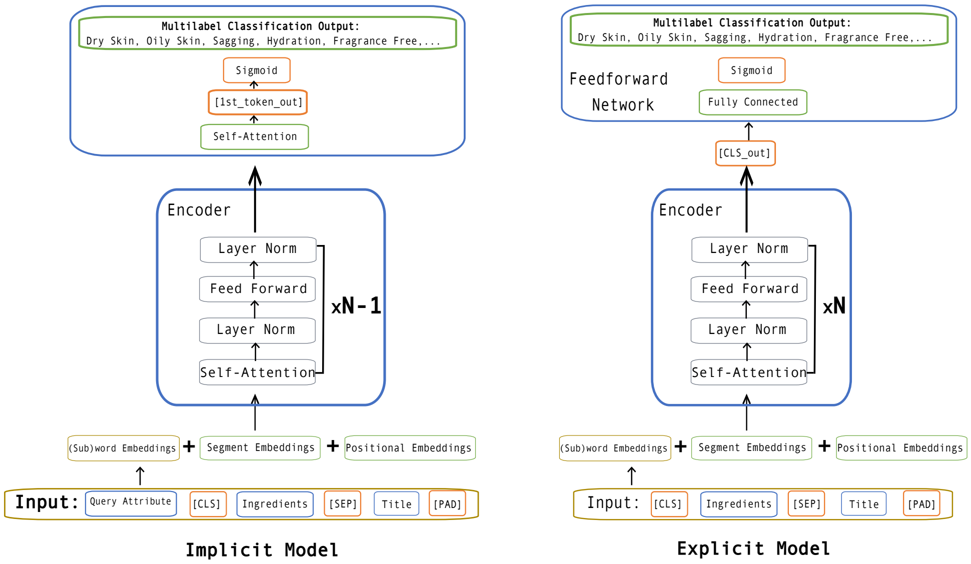


Figure 2: Difference between implicit and explicit models. **Left:** In implicit models, the model intakes query attribute together with product ingredients and title. Note that in our case, the output logits come directly from the self-attention values of the last encoder layer. **Right:** Explicit models represent the standard way of fine-tuning the BERT model, where a classifier is attached to the end of the Transformer.

Algorithm 1 BT-BERT Forward Pass

```
bert_model = AutoModel.from_pretrained(...)

function FORWARD(input_ids, labels)
    outputs = bert_model(input_ids)

    # extract the last layer's attention, e.g., -1
    # attentions are [batch, heads, seqlen, seqlen]
    attentions = outputs["attentions"][-1]

    # summing attention values over all heads
    # for the first token attending to itself
    # 16 is a hyperparameter multiplication factor
    logits = 16 * attentions[:, :, 0, 0].sum(dim=1)

    L = binary_cross_entropy_with_logits(logits, labels)

end function
```

5. Experiments

5.1. Training Details

For all experiments, we train the network end-to-end with a batch size of 8 until convergence. We use the AdamW optimizer [24] with an initial learning rate of 3×10^{-5}. We follow the standard setup for training Transformer models by splitting the trainable parameters into two categories: decay and non-decay parameters. Non-decaying parameters are biases and LayerNorm [25] parameters; all other parameters are weight decayed. We set $\texttt{beta2} = 0.95$ to improve training stability as recommended in [26].

We explored a few different training recipes but found them to have negligible impact on the final model performance, including using a cosine annealing learning rate scheduler [27], linear decay scheduler, and weighted loss for addressing the class imbalance issue.

Table 1
Model Performance: Explicit vs. Implicit Approach (BT-BERT)

Method	Accuracy	Precision	Recall	F1-Score	Parameters
BT-BERT	**0.964**	**0.987**	**0.958**	**0.960**	109,360,128
Explicit Model	0.946	0.954	0.904	0.912	109,975,296
Fuzzy Search	0.301	0.287	0.356	0.327	–

5.2. Baseline Solutions

We evaluated our method against two simple baseline solutions: Fuzzy Search and the explicit model alternative illustrated in Figure 2.

Fuzzy Search This is a straightforward approach of finding keywords based on edit distance and other heuristics. Specifically, a predefined list of target keywords is established (see Section 9.6) for each of the 33 attributes. Subsequently, a product is categorized as possessing a particular attribute if any of the keywords from the corresponding list are detected within the product information.

We compare to this baseline as an example of highly explainable solution, but we are well aware that it is not state-of-the-art by any means. By examining a few examples, the limitations of the fuzzy search approach is immediately apparent. First, fuzzy search is unable to discern complex textual context. For example, it may overlook the labeling of a product described as *free of perfume, silicones, phthalates, fragrance* as 'Fragrance Free'. Second, it is sensitive to error tolerance threshold. For instance, despite a product being described as *hydra intensive treatment*, the method may not assign the attribute "Hydration" if the error tolerance is set too low.

Explicit Model A common approach for classification tasks often trains an explicit feed-forward network on top of a pre-trained rich embedding, similar to the approach described in [18]. As a benchmark, we experimented with this approach, where the model receives product information as input and outputs the likelihood of the 33 labels. Figure 2 highlights the differences between the implicit and the explicit models. In the explicit model, the classifier's output dimension is predefined to be the same as the number of attributes. For this approach, we use the pre-trained weights and tokenizer of `bge-base-en-v1.5` [28] from HuggingFace. We chose `bge-base-en-v1.5` as it is considered the state-of-the-art text embedding model for retrieval, clustering, reranking tasks in the Massive Text Embedding Benchmark (MTEB) [29]. As a common practice, we freeze the backbone weights and only update the classifier parameters for four epochs to avoid catastrophic forgetting. We find that training end-to-end after four epochs provides the optimal results compared to other configurations.

5.3. Model Results

We evaluate models on the standard classification metrics: Accuracy, Precision, Recall and F1-Score. Although we report recall and F1-score, we prioritize accuracy and precision as the main evaluation metrics. A higher precision aligns more closely with our acceptable risk threshold by minimizing the likelihood of potentially recommending products containing unsuitable ingredients to customers with particularly sensitive skin. This is important as we envision attribute-based beauty recommendations as one of the direct applications on this work.

Table 1 summarizes the results of label prediction across different methods. We observe that both learning-based methods significantly outperform the fuzzy search baseline, as expected. The implicit model performs slightly better than the explicit alternative across all evaluation metrics. Aside from the quantitative edge, the implicit model offers other qualitative advantages that the explicit model does not. We discuss this extensively in the following sections.

Table 2
Attention analysis for ‘Acne’, ‘Fine Lines and Wrinkles’, and ‘Hydration‘ attributes

Attribute	High Attention Sub-word Tokens	Ingredient
Acne	‘sal’, ‘#ic’, ‘#yl’, ‘#ic’, ‘acid’	Salicylic Acid
	‘alcohol’	Alcohol
	‘benz’, ‘#oy’, ‘#l’, ‘per’, ‘#oxide’	Benzoyl Peroxide
	‘beta’, ‘#ine’	Betaine
Lines & Wrinkles	‘#pher’	Tocopheryl Acetate
	‘#ito’	Palmitoyl
	‘baku’, ‘#chio’	Bakuchiol
	‘re’, ‘#tino’	Retinol
Hydration	‘#yal’, ‘#uron’, ‘ate’	Sodium Hyaluronate
	‘#ly’, ‘#cer’, ‘#in’	Glycerin
	‘ni’, ‘#ac’, ‘#ina’, ‘#mide’	Niacinamide

Table 3
Attention analysis for product attributes

Attribute	High Attention Sub-word Tokens	Corresponding Ingredient
Product: *PanOxyl AM Oil Control Moisturizer, NEW Sheer Formula, Absorbs Excess Oil and Reduces Shine, with Mineral Sunscreen for Acne Prone and Oily And All Skin Tones - 1.7 oz*		
Dry Skin	‘#yal’, ‘#uron’, ‘ate’	Sodium Hyaluronate
Sensitive Skin	‘#olo’	Bisabolol
Dark Circles	‘but’, ‘#yl’, ‘#ic’, ‘#yla’, ‘#te’	Butyloctyl Salicylate
Product: *Good Molecules BHA Clarifying Gel Cream - Facial Cream with Salicylic Acid, Green Tea, and Gotu Kola Extract Soothe and Hydrate - Skincare for Face*		
Acne	‘sal’, ‘#ic’, ‘#yl’, ‘#ic’, ‘acid’	Salicylic Acid
Dry Skin	‘#ly’, ‘#cer’, ‘#in’	Glycerin
Redness	‘allan’, ‘#to’	Allantoin
Product: *I DEW CARE Moisturizer Face Cream - Chill Kitten | Moringa Seed, Prickly Pear, Heartleaf Extract, 24 Hour, Aloe Vera Gel for Dry, Red Skin, Cactus Oil-free, 1.69 Fl Oz*		
Redness	‘tea’, ‘ni’, ‘#ac’, ‘#ina’, ‘#mide’	Green Tea, Niacinamide
Fine Lines and Wrinkles	‘as’, ‘#cor’, ‘#bic’	Ascorbic Acid

5.4. Explainability

In this section, we analyze the input tokens with high attention values in the second last layer of the Transformer encoder block. Top tokens are obtained using Algorithm 2.

In Table 2, we choose three query attributes—‘Acne’, ‘Fine Lines and Wrinkles’ and ‘Hydration’—and show that tokens with high attention values are ingredients that address the target skin concerns. This means that our model has learned the effects of different ingredients and how they are associated to different skin concerns and skin types. We chose these labels, as they are the most popular filter criteria for beauty products.

We also assess the high attention tokens for each predicted label of a single product and show that these tokens are different across attributes of a given product. This means that our model has learned to pay attention to different tokens when it is being asked about different attributes. Table 3 demonstrates some of the examined products.

5.5. Robustness in Low Data Regime

In this section, we present empirical evidence demonstrating the robust performance of BT-BERT even when the volume of training data is limited. Figure 6 shows the validation accuracy across various degrees of data scarcity, namely when the model is trained using the full dataset, as well as 1/2, 1/4, and 1/8 of the full training corpus. In each training run, we systematically down-sample the training set and keep the validation set constant, i.e., it still contains the same 2246 products.

Note that for the 1/8 training, the model is trained with only 1167 products and yet still the validation accuracy only drops by less than 1.25%. We hypothesize that the robust performance of BT-BERT in such a low-resource regime can be attributed to the fact that it is an energy-based implicit model, as opposed to an explicit classifier. The same scaling pattern is observed in other energy-based models [12]. Additionally, we attribute part of such robustness to the implicit data augmentation strategy employed in training—specifically, each product is paired with all 33 query attributes, exposing our model to diverse input contexts. We have not yet fully characterize the scaling behaviors of implicit and explicit models. It is possible that with improved training techniques, the explicit approach can close the gap in low-resource regimes.

6. Discussion

6.1. Does the choice of logits transformation matter?

Our early experiments indicate that scaling the probability linearly with 16 achieves better results than not employing it. We explored an alternative scaling formulation using $f(x) = \log(x/(1-x))$, where x represents the attention value of the first query token from all attention heads. The design is inspired by probability theory, where $x/(1-x)$ is commonly referred to as the odds or odds ratio when x is a probability. Taking the logarithm of the odds ratio is a common transformation used in logistic regression to convert probability into logits.

Additionally, we experimented with using the summation and average of the attention values from the first three query tokens as x before applying the log transformation. However, these variations did not produce better results. Ultimately, we chose the linear scaling method of multiplying by 16 due to its simplicity and slightly faster computation times.

6.2. Finetuning on Additional Attributes

In this section, we discuss the adaptability of implicit models in incorporating new labeled attributes as they become available. We design a scenario mirroring real-world dynamics, where an initial dataset comprises 30 out of 33 labels, with the remaining 3 labels introduced in a subsequent release. Such scenarios are commonplace in the beauty industry, where emerging trends and evolving consumer preferences necessitate the addition of new product attributes. For instance, the advent of clean beauty as a trend in 2023 [30] underscores the relevance of this work. Through comprehensive analysis and experimentation, we assess and highlight the implicit model's efficacy in seamlessly incorporating new attributes.

We removed the labels for 'Fragrance Free' (generally-preferred), 'Oily Skin' (skin type), and 'Acne' (skin concern) from the full dataset ($\mathcal{D}_{\text{full}}$) and trained a model on the remaining 30 labels ($\mathcal{D}_{30}$). Then, we add back the removed labels and finetune the previously trained model with the complete dataset for only one epoch. Table 4 shows the validation accuracies before and after the finetuning step. When finetuning on only the three additional labels ($\mathcal{D}_3$), we observe a significant drop in validation accuracy for the existing 30 labels in the validation set. We believe this is due to the *catastrophic forgetting* problem and could potentially be alleviated by using more advanced finetuning algorithms [31, 32, 33].

When finetuning with $\mathcal{D}_{\text{full}}$, we observe only a slight drop of performance when predicting the existing 30 labels, but the accuracy for the new labels is drastically improved. It is important to note that this finetuning procedure is impossible when using explicit models, since the number of output classes is different and therefore the classifier must be replaced and retrained.

Table 4
Model performance on partially held out data. In this experiment, we evaluate the model's ability to incorporate additional labels when they become available.

	Train $\mathcal{D}_{30}$	Finetune $\mathcal{D}_3$	Finetune $\mathcal{D}_{\text{full}}$
Acc. on 30 labels	93.9%	82.4%	93.4%
Acc. on 3 labels	59.6%	94.7%	93.5%

6.3. Alternating Query Attribute Tokens

In this section, we highlight the benefit of our implicit model during inference time. First, we show that it can handle similar but not identical query attributes. We take 'Fine Lines and Wrinkles' as an example and replace the query attribute with just a single word 'Lines' for a commonly available anti-wrinkle renewal skin cream. We use Algorithm 2 to extract the high attention tokens and track how they change when the attribute tokens are replaced.

We observed a number of overlapping tokens especially those addressing lines and wrinkles—'#chio', 'pu', 'soy', 'lines', 'baku', 're', and '#tino'. We also identified non-overlapping tokens such as water, after, cleansing, fine, cart, and wr. It is important to note that the non-overlapping tokens, such as 'water' and 'cleansing,' are more general and not as directly relevant to the specific skin concern. We believe that this approach can help us better understand the ingredients and their target uses.

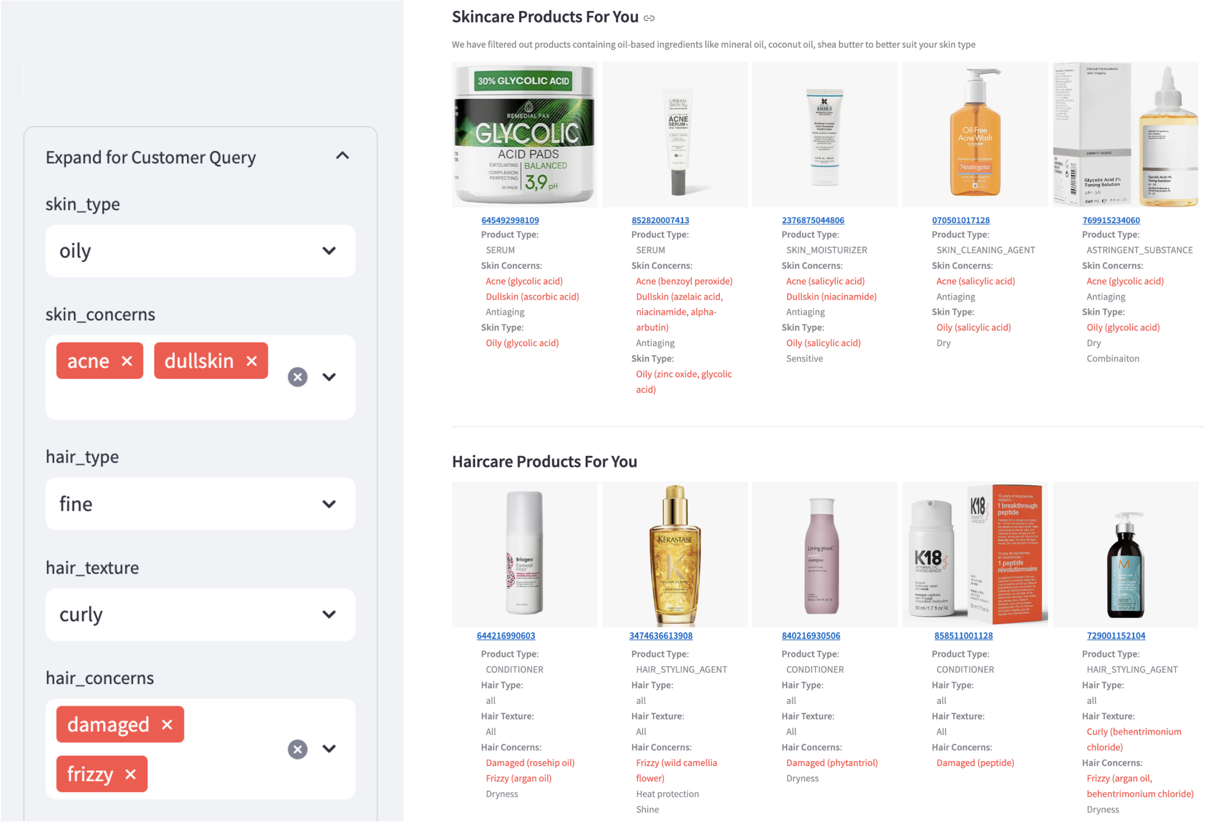
7. Explainable Beauty Recommendation and Customer Understanding

Explainable Beauty Recommendation One critical application of ingredient-based attribute extraction lies in delivering explainable recommendations to beauty customers.

In the ever-evolving beauty industry, where personalization is key, transparency and clarity in product suggestions are vital. As illustrated in Figure 3, skincare recommendations are made using a point-wise approach, where each product is individually assessed based on the customer's specific skin type and concerns. Here, the customer has selected "oily" skin and concerns of "acne" and "dullskin". The recommended products not only contain ingredients intended to address these issues but are also compatible with the customer's stated skin type, enhancing the trustworthiness and relevance of each suggestion. Each product is annotated with its predicted target skin concerns and skin types, alongside the ingredients intended to address those concerns, using Algorithm 2 discussed in Section 5.4. For example, Salicylic Acid is highlighted for its anti-acne properties across various product types like cleansers, pads, and serums. Furthermore, the system strategically omits products with oil-based ingredients that could exacerbate oily skin, ensuring that recommendations are appropriate for the user's concerns.

By providing fact-based explanations for recommended products, this approach offers clear and transparent justifications for the recommendations. As customers purchase and use products with effective ingredients, they are more likely to achieve the desired skin results, fostering long-term trust and encouraging repeat engagement with the e-commerce store. This method not only empowers customers to make informed purchasing decisions but also strengthens their trust in the recommendation system. This approach is versatile and can be applied broadly across most beauty catalogs, including haircare and makeup, where ingredients stay on the skin for extended periods. In the context of strategic and utility-aware recommendations, explainability is crucial for aligning personalized suggestions with both individual needs and broader objectives. This alignment ultimately enhances customer confidence, satisfaction, and long-term audience growth.

Customer understanding Conversely, customer propensity toward specific attributes—such as preferred skin type, skin concerns, and ingredient preferences—can be inferred from their past purchases. Our future work focuses on understanding customer skin types and concerns by building upon



Expand for Customer Query

skin_type: oily

skin_concerns: acne, dullskin

hair_type: fine

hair_texture: curly

hair_concerns: damaged, frizzy

Skincare Products For You

We have filtered out products containing oil-based ingredients like mineral oil, coconut oil, shea butter to better suit your skin type

645492998109
Product Type: SERUM
Skin Concerns: Acne (glycolic acid), Dullskin (ascorbic acid), Antiaging
Skin Type: Oily (glycolic acid)

852820007413
Product Type: SERUM
Skin Concerns: Acne (benzoyl peroxide), Dullskin (azelaic acid, niacinamide, alpha-arbutin), Antiaging
Skin Type: Oily (zinc oxide, glycolic acid)

2376875044806
Product Type: SKIN_MOISTURIZER
Skin Concerns: Acne (salicylic acid), Dullskin (niacinamide), Antiaging
Skin Type: Oily (salicylic acid), Sensitive

070501017128
Product Type: SKIN_CLEANING_AGENT
Skin Concerns: Acne (salicylic acid), Antiaging
Skin Type: Oily (salicylic acid), Dry

769915234060
Product Type: ASTRINGENT_SUBSTANCE
Skin Concerns: Acne (glycolic acid), Antiaging
Skin Type: Oily (glycolic acid), Dry, Combinaiton

Haircare Products For You

644216990603
Product Type: CONDITIONER
Hair Type: all
Hair Texture: All
Hair Concerns: Damaged (rosehip oil), Frizzy (argan oil), Dryness

3474636613908
Product Type: HAIR_STYLING_AGENT
Hair Type: all
Hair Texture: All
Hair Concerns: Frizzy (wild camellia flower), Heat protection, Shine

840216930506
Product Type: CONDITIONER
Hair Type: all
Hair Texture: All
Hair Concerns: Damaged (phytantriol), Dryness

858511001128
Product Type: CONDITIONER
Hair Type: all
Hair Texture: All
Hair Concerns: Damaged (peptide)

729001152104
Product Type: HAIR_STYLING_AGENT
Hair Type: all
Hair Texture: Curly (behentrimonium chloride)
Hair Concerns: Frizzy (argan oil, behentrimonium chloride), Dryness

Figure 3: Skincare recommendation with explainable ingredient for each attribute.

existing attribute extraction methodologies. This advancement will enable further refinement of our recommendation algorithms, particularly in the ranking layer.

8. Conclusion

We present an energy-based implicit model for extracting beauty-specific attributes trained using end-to-end supervised learning. We empirically show that the implicit approach outperforms traditional explicit classifiers in terms of accuracy, precision, and other evaluation metrics. Aside from better performance, we show that the implicit model is explainable, robust to low-data scenarios, and easy to incorporate new attributes as they become available. Using the explainability feature of our model, we propose novel ways to use the predictions without additional training by comparing and contrasting the high value tokens across different products and attributes. We have not yet fully characterized the limits of the model's capabilities. Currently, we only qualitatively identify the high attention value tokens and discuss how they are related to the specific skin concerns and skin types in our attention analysis. We wish to better quantify the correlations between all predicted ingredients and the attributes. Although our work focuses on beauty attribute extraction, we believe the simplicity of our approach and comprehensiveness of our analysis provide a solid foundation for future research in designing more capable and explainable models in all domains of machine learning. In future work, we will validate the generated attributes within downstream recommendation systems and conduct a thorough evaluation. Furthermore, we will assess the impact of explainability for end users through A/B testing.

9. Appendices

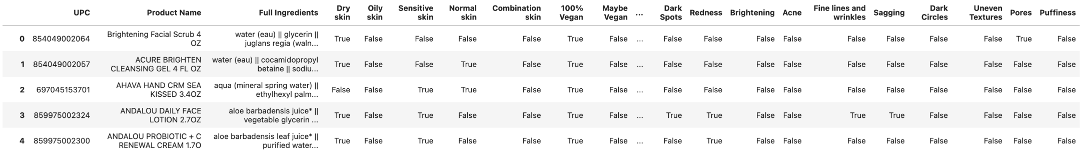
9.1. Labels for Skincare Products

We define 33 labels for skincare products that include 5 skin types, 11 skin concerns, and 17 attributes that are generally preferred across beauty products.

- Target skin types: Dry Skin, Normal Skin, Oily Skin, Combination Skin, Sensitive Skin
- Target skin concerns: Acne, Hydration, Pores, Fine Lines and Wrinkles, Sagging, Dark Spots, Dullness, Redness, Uneven Texture, Dark Circles, Puffiness
- General preferred beauty attributes: 100% Vegan, Cruelty Free, Fragrance Free, Hypoallergenic, Paraben Free, Mineral Oil Free, Palm Oil Free, Oil Free, Alcohol Free, Sulphate Free, Gluten Free, Silicone Free, Phthalate Free, Talc free, Non Comedogenic, Aluminum Free, Fluoride Free.

9.2. Product information and Labels

Each product comes with a title, list of ingredients, and a Boolean label for each attribute. An example is shown in Figure 4.



	UPC	Product Name	Full Ingredients	Dry skin	Oily skin	Sensitive skin	Normal skin	Combination skin	100% Vegan	Maybe Vegan	...	Dark Spots	Redness	Brightening	Acne	Fine lines and wrinkles	Sagging	Dark Circles	Uneven Textures	Pores	Puffiness
0	854049002064	Brightening Facial Scrub 4 OZ	water (eau) || glycerin || juglans regia (waln...	True	False	False	False	False	True	False	...	False	False	False	False	False	False	False	False	True	False
1	854049002057	ACURE BRIGHTEN CLEANSING GEL 4 FL OZ	water (eau) || cocamidopropyl betaine || sodiu...	True	False	False	True	False	True	False	...	False	False	False	False	False	False	False	False	False	False
2	697045153701	AHAVA HAND CRM SEA KISSED 3.4OZ	aqua (mineral spring water) || ethylhexyl palm...	False	False	True	True	False	True	False	...	False	False	False	False	False	False	False	False	False	False
3	859975002324	ANDALOU DAILY FACE LOTION 2.7OZ	aloe barbadensis juice* || vegetable glycerin ...	True	False	True	False	False	True	False	...	True	True	False	False	True	True	False	False	False	False
4	859975002300	ANDALOU PROBIOTIC + C RENEWAL CREAM 1.7O	aloe barbadensis leaf juice* || purified water...	True	False	True	False	False	True	False	...	False	True	False	False	False	False	False	False	False	False

Figure 4: Sample Pandas dataframe with product ingredient list (`Full Ingredients`) and title (`item_name`) for each product.

9.3. Training data label distribution

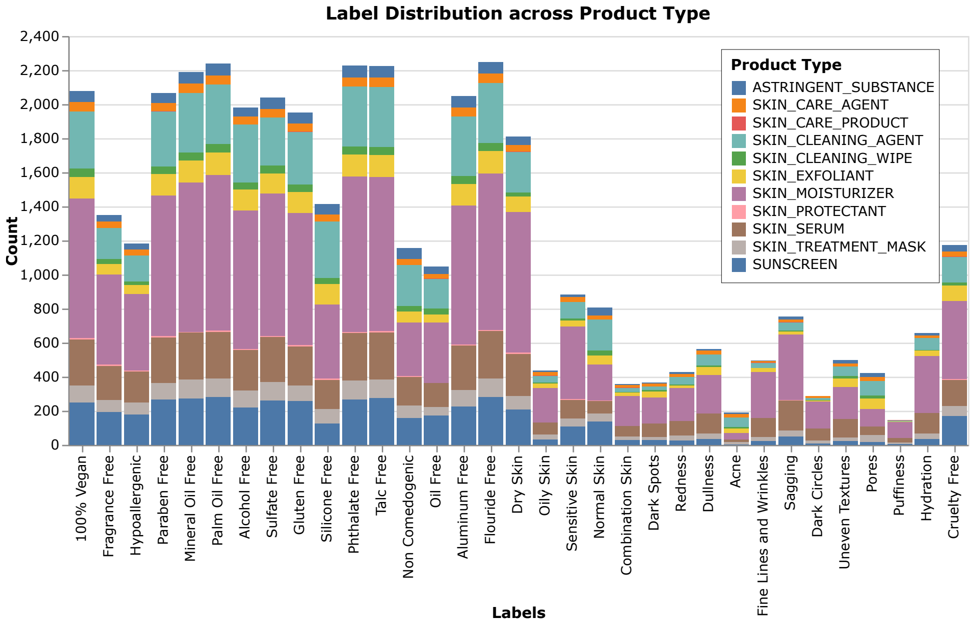


Figure 5: Label Distribution across Product Type in our dataset. The height of each bar indicates the number of products associated with the respective attribute. For instance, there are a total of 1809 out of 11580 products for Dry Skin.

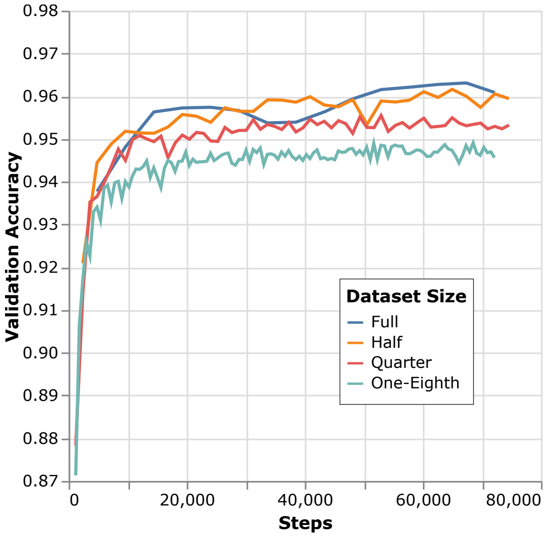


Figure 6: Validation accuracy training on various sizes of dataset

9.4. Learning Curves for Robustness in Low Data Regime Experiment

9.5. Algorithm for Key Token Extraction Based on Attention Values

Algorithm 2 Key Token Extraction Based on Attention Values

```
function GetTopAttentionTokens(input_ids, attentions, topk)
    # input_ids is a tensor of shape (seqlen,)
    # attentions is a tensor of shape (heads, seqlen, seqlen)

    # get index of top-k attention per row across all heads
    topk_indices = attentions.flatten(0, 1).topk(topk).indices
    topk_indices = topk_indices.unique()

    # convert col indices to token strings
    topk_tokens = convert_ids_to_tokens(input_ids[topk_indices])

    # remove non-meaningful tokens
    TO_REMOVE = [‘,’, ‘[CLS]’, ‘[SEP]’, ‘(’, ‘)’, ‘[PAD]’]
    topk_tokens = [k for k in topk_tokens if k not in TO_REMOVE]

end function
```

9.6. FuzzySearch Attribute Key Words

For FuzzySearch method, We define keywords for each of the 33 labels.

- Dry Skin: "dry", "all", "universal".
- Normal Skin: "normal", "all", "universal".
- Oily Skin: "oil", "all", "universal".
- Combination Skin: "combination", "all", "universal".
- Sensitive Skin: "sensitive", "all", "universal".
- Acne: "anti acne", "blackheads", "salicylic acid", "Glycolic Acid", "Benzoyl Peroxide", "breakouts treatment", "acne preventing", "skin clarifying".
- Hydration: "dehydration", "dryness", "hydrating", "rehydrate", "soothing", "moisturizing", "nourishing", "softening", "replenishing".
- Pores: "pore", "oil control".

- Fine Lines and Wrinkles: "wrinkle", "anti-aging", "anti aging", "anti-aging", "wrinkle treatment", "wrinkles treatment", "skin cell renewal", "skin-cell-renewal", "plumping", "refine skin texture", "refine-skin-texture", "repairing", "fine line", "anti aging", "plumping", "skin cell renewal", "replenishing", "octinoxate", "octisalate", "avobenzone".
- Sagging: "firming", "wrinkle", "anti aging", "skin cell renewal".
- Dark Spots: "hyperpigmentation", "melasma", "dyschromia", "brown spot", "age spot", "dark spot", "brightening", "even toning", "color correction", "lightening", "antioxidant", "oxygenating", "whitening".
- Dullness: "even toning", "dull skin", "lightening", "brightening", "colour correction", "skin cell renewal", "rejuvenating", "exfoliating", "plumping".
- Redness: "redness", "anti inflammatory", "soothening", "soothing", "redness reduction", "redness removal", "oxygenating".
- Uneven Texture: "uneven texture", "uneven skin".
- Dark Circles:"puffiness", "dark circles", "color correction", "lightening", "antioxidant", "radiant skin", "brightening".
- 100% Vegan: "vegetarian", "plantbased", "vegan", "animalbyproductfree".
- Cruelty Free: "crueltyfree".
- Fragrance Free: "unscented", "fragrancefree".
- Hypoallergenic: "preservativefree", "latexfree", "chemicalfree", "formaldehydefree", "slesfree".
- Paraben Free: "preservativefree", "slesfree", "slsfree", "parabenfree".
- Mineral Oil Free: "palmoilfree", "mineraloilfree".
- Palm Oil Free: "palmoilfree".
- Oil Free: "oilfree", "palmoilfree", "mineraloilfree".
- Alcohol Free: "alcoholfree".
- Sulphate Free: "sulfatefree".
- Gluten Free: "glutenfree".
- Silicone Free: "siliconefree".
- Phthalate Free: "phthalatefree".

References

[1] L. Wood, Beauty & personal care-worldwide, https://www.statista.com/outlook/cmo/beauty-personal-care/worldwide, 2024. Accessed: 2024-04.

[2] L. Chiticariu, R. Krishnamurthy, Y. Li, F. Reiss, S. Vaithyanathan, Domain adaptation of rule-based annotators for named-entity recognition tasks, in: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Cambridge, MA, 2010, pp. 1002–1012. URL: https://aclanthology.org/D10-1098.

[3] D. Putthividhya, J. Hu, Bootstrapped named entity recognition for product attribute extraction, in: EMNLP, 2011, pp. 1557–1567. URL: http://www.aclweb.org/anthology/D11-1144.

[4] Z. Huang, W. Xu, K. Yu, Bidirectional LSTM-CRF models for sequence tagging, CoRR abs/1508.01991 (2015). URL: http://arxiv.org/abs/1508.01991. arXiv:1508.01991.

[5] G. Zheng, S. Mukherjee, X. L. Dong, F. Li, Opentag: Open attribute value extraction from product profiles, CoRR abs/1806.01264 (2018). URL: http://arxiv.org/abs/1806.01264. arXiv:1806.01264.

[6] J. Yan, N. Zalmout, Y. Liang, C. Grant, X. Ren, X. L. Dong, Adatag: Multi-attribute value extraction from product profiles with adaptive decoding, arXiv preprint arXiv:2106.02318 (2021).

[7] H. Xu, W. Wang, X. Mao, X. Jiang, M. Lan, Scaling up open tagging from tens to thousands: Comprehension empowered attribute value extraction from product title, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 5214–5223.

[8] A. Cardoso, F. Daolio, S. Vargas, Product characterisation towards personalisation: Learning attributes from unstructured data to recommend fashion products, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 80–89.

[9] Q. Wang, L. Yang, J. Wang, J. Krishnan, B. Dai, S. Wang, Z. Xu, M. Khabsa, H. Ma, SMARTAVE: Structured multimodal transformer for product attribute value extraction, in: Y. Goldberg, Z. Kozareva, Y. Zhang (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2022, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 263–276. URL: https://aclanthology.org/2022.findings-emnlp.20. doi:10.18653/v1/2022.findings-emnlp.20.

[10] F. T. Dezaki, H. Arora, R. Suresh, A. Banitalebi-Dehkordi, Automated material properties extraction for enhanced beauty product discovery and makeup virtual try-on, arXiv preprint arXiv:2312.00766 (2023).

[11] Y. Du, I. Mordatch, Implicit generation and modeling with energy-based models, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 32, Curran Associates, Inc., 2019.

[12] P. Florence, C. Lynch, A. Zeng, O. A. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, J. Tompson, Implicit behavioral cloning, in: 5th Annual Conference on Robot Learning, 2021.

[13] P. Afshar, J. Yeon, A. Levitskyy, R. Suresh, A. Banitalebi-Dehkordi, Improving the accuracy of beauty product recommendations by assessing face illumination quality, arXiv preprint arXiv:2309.04022 (2023).

[14] T. Alashkar, S. Jiang, S. Wang, Y. Fu, Examples-rules guided deep neural network for makeup recommendation, in: Proceedings of the AAAI conference on artificial intelligence, volume 31, 2017.

[15] H.-H. Li, Y.-H. Liao, Y.-N. Huang, P.-J. Cheng, Based on machine learning for personalized skin care products recommendation engine, in: 2020 International Symposium on Computer, Consumer and Control (IS3C), 2020, pp. 460–462. doi:10.1109/IS3C50286.2020.00125.

[16] Y. Nakajima, H. Honma, H. Aoshima, T. Akiba, S. Masuyama, Recommender system based on user evaluations and cosmetic ingredients, in: 2019 4th International Conference on Information Technology (InCIT), 2019, pp. 22–27. doi:10.1109/INCIT.2019.8912051.

[17] R. S, H. S, K. Jayasakthi, S. D. A, K. Latha, N. Gopinath, Cosmetic product selection using machine learning, 2022 International Conference on Communication, Computing and Internet of Things (IC3IoT) (2022) 1–6. URL: https://api.semanticscholar.org/CorpusID:248753814.

[18] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, CoRR abs/1810.04805 (2018). arXiv:1810.04805.

[19] Y. W. Teh, M. Welling, S. Osindero, G. E. Hinton, Energy-based models for sparse overcomplete representations, J. Mach. Learn. Res. 4 (2003) 1235–1260.

[20] Y. Song, D. P. Kingma, How to train your energy-based models, arXiv preprint (2021). URL: https://arxiv.org/abs/2101.03288.

[21] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, F. Huang, A tutorial on energy-based learning, Predicting structured data 1 (2006).

[22] Skillsmuggler, Amazon ratings dataset, 2024. URL: https://www.kaggle.com/datasets/skillsmuggler/amazon-ratings, accessed: 2024-08-26.

[23] C. Feeds, Amazon usa beauty products dataset, 2024. URL: https://data.world/crawlfeeds/amazon-usa-beauty-products-dataset, accessed: 2024-08-26.

[24] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, arXiv preprint arXiv:1711.05101 (2017).

[25] J. L. Ba, J. R. Kiros, G. E. Hinton, Layer normalization, arXiv preprint arXiv:1607.06450 (2016).

[26] X. Zhai, B. Mustafa, A. Kolesnikov, L. Beyer, Sigmoid loss for language image pre-training, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 11975–11986.

[27] I. Loshchilov, F. Hutter, Sgdr: Stochastic gradient descent with warm restarts, arXiv preprint arXiv:1608.03983 (2016).

[28] S. Xiao, Z. Liu, P. Zhang, N. Muennighoff, C-pack: Packaged resources to advance general chinese embedding, 2023. arXiv:2309.07597.

[29] N. Muennighoff, N. Tazi, L. Magne, N. Reimers, Mteb: Massive text embedding benchmark, arXiv preprint arXiv:2210.07316 (2022). URL: https://arxiv.org/abs/2210.07316. doi:10.48550/ARXIV.2210.07316.

[30] K. MCGRATH, Did clean beauty go too far?, https://www.allure.com/story/is-clean-beauty-over, 2023. Accessed: 2024-04.

[31] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, Lora: Low-rank adaptation of large language models, arXiv preprint arXiv:2106.09685 (2021).

[32] S.-Y. Liu, C.-Y. Wang, H. Yin, P. Molchanov, Y.-C. F. Wang, K.-T. Cheng, M.-H. Chen, Dora: Weight-decomposed low-rank adaptation, arXiv preprint arXiv:2402.09353 (2024).

[33] L. Zhang, A. Rao, M. Agrawala, Adding conditional control to text-to-image diffusion models, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 3836–3847.