Beauty Beyond Words II: Leveraging Attention for Ingredient-Attribute Insights in Beauty Products

Shreya Sriram
Cornell Tech
New York, NY
ss3589

Zachary Decker
Cornell Tech
New York, NY
zad25

Priyanshi Gupta
Cornell Tech
New York, NY
pg485

ABSTRACT

The beauty industry faces a growing need for transparent and explainable methods to map product ingredients to skin-related attributes, addressing consumer demand for informed recommendations. Traditional approaches often rely on manual annotation and domain expertise, which are labor-intensive, costly, and difficult to scale. This study explores the application of transformer-based architectures for multi-label classification of beauty products, predicting attributes such as acne treatment, hydration, and sensitive skin suitability based on product ingredients and descriptions.

Our approach combines explicit model architectures with advanced explainability techniques, including Integrated Gradients [6], SHAP [4], and LIME [5], to uncover the relationships between ingredients and product attributes. By utilizing publicly available Amazon product metadata and reviews alongside a curated skincare glossary, we developed a scalable pipeline to extract and analyze ingredient-attribute relationships in beauty products. This methodology emphasizes both prediction accuracy and model interpretability, bridging the gap between technical performance and actionable insights for industry applications.

While the proposed framework demonstrates potential in linking ingredients to attributes, challenges such as data quality, class imbalances, and limited representation for certain attributes highlight areas for improvement. Nevertheless, explainability methods effectively revealed key contributors to predictions, offering valuable insights into the importance of specific ingredients.

This work underscores the importance of scalable, interpretable machine learning models in the beauty industry, paving the way for enhanced recommendation systems and informed product formulation. Future research should focus on improving dataset quality, leveraging advanced architectures, and exploring semi-supervised techniques to address limitations and enhance the framework's adaptability. By advancing the understanding of ingredient-attribute relationships, this study contributes to the development of more transparent and effective tools for beauty product analysis and recommendation systems.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; • **Information systems** → *Document representation.*

KEYWORDS

attention, machine learning, explainability, ingredient analysis, multi-label classification

ACM Reference Format:
Shreya Sriram, Zachary Decker, and Priyanshi Gupta. 2024. Beauty Beyond Words II: Leveraging Attention for Ingredient-Attribute Insights in Beauty Products. In . ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The beauty industry faces significant challenges in understanding the intricate relationships between product attributes—such as skin benefits, targeted concerns, and suitability for various skin types—and the ingredients that define them. These relationships are critical for making accurate recommendations that meet consumer expectations and ensure satisfaction. For instance, consumers increasingly demand transparency about how specific ingredients contribute to attributes like hydration, anti-aging benefits, or acne reduction, making it essential for brands to establish these connections clearly. However, traditional approaches to addressing this problem often rely heavily on manual annotation and domain expertise to classify products based on their attributes. This process is inherently labor-intensive, costly, and difficult to scale, especially given the ever-expanding volume of beauty products on the market. Additionally, manual annotation introduces variability and potential inconsistencies, as it depends on subjective judgments from human annotators. Compounding these challenges, conventional systems used in this space often lack explainability, making it difficult for brands and consumers alike to understand how or why specific ingredients influence certain attributes. This lack of transparency not only diminishes consumer trust but also hinders the ability to optimize formulations effectively, leaving significant room for improvement in both interpretability and performance.

To address these challenges, we developed a BERT-based [2] machine learning model that achieved a balanced F1 score of 0.61 and precision of 0.75, reflecting its capability to accurately predict product attributes. The model was trained and tested on a manually curated dataset using Amazon product metadata and customer reviews in a multi-label classification setup. This approach enabled simultaneous predictions for multiple attributes, streamlining the classification process while providing actionable insights into ingredient-attribute relationships. Furthermore, we applied advanced explainability techniques such as LIME, SHAP, integrated

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

gradients, and attention-based analysis to uncover how specific product attributes relate to their ingredients. These methods not only enhanced the interpretability of the model's predictions but also provided actionable insights into the importance of individual ingredients, bridging the gap between technical performance and practical application in the beauty industry.

2 RELATED WORK

The *Beauty Beyond Words* paper [1] presents a sophisticated framework, BT-BERT, designed to classify beauty products based on attributes such as skin types, concerns, and preferences. This system leverages the strengths of a bidirectional transformer encoder network, utilizing self-attention mechanisms to derive attribute classifications directly from the model's attention layers. The core innovation lies in its ability to bypass additional classifier layers, instead relying on high-attention tokens to infer relationships between specific ingredients and product attributes. For example, ingredients like salicylic acid are strongly linked to acne treatment, while glycerin is often associated with hydration. By interpreting these attention scores, BT-BERT inherently enhances explainability, providing actionable insights into ingredient-attribute relationships without requiring external interpretability mechanisms. The architecture's flexibility further supports fine-tuning, enabling the model to adapt to new or evolving attributes, a critical capability in the ever-changing beauty industry landscape.

2.1 Implicit and Explicit Architectures in BT-BERT

The *Beauty Beyond Words* paper introduces two distinct architectural paradigms for BT-BERT: the implicit model and the explicit model. These architectures differ in how they handle attribute extraction, with trade-offs in computational efficiency, interpretability, and task-specific optimization.

2.1.1 Implicit Model. The implicit model relies on an energy-based approach, where attribute predictions are derived directly from the self-attention values of the last Transformer encoder layer. This architecture avoids additional feed-forward layers, making it uniquely focused on leveraging the pre-trained Transformer's attention mechanisms.

- **Explainability:** By using attention weights for predictions, the implicit model naturally highlights tokens or ingredients most relevant to specific attributes, providing an interpretable link between input and output.
- **Scalability:** This architecture supports seamless addition of new attributes without requiring retraining of classifier layers, making it adaptable to evolving product catalogs.
- **Computational Complexity:** While it avoids additional classifier layers, deriving predictions directly from attention values can be computationally intensive during both training and inference due to the need to process and interpret high-dimensional attention maps.

2.1.2 Explicit Model. The explicit model, in contrast, introduces a feed-forward classification layer on top of the Transformer embeddings. This layer directly maps the contextualized outputs to the attribute space, optimizing for multi-label classification.

- **Efficiency:** The explicit model requires fewer computations for inference since it avoids relying directly on attention mechanisms. This reduction in complexity makes it more computationally efficient, particularly for large datasets or production environments.
- **Task-Specific Optimization:** By adding a classification head, the explicit model allows for more targeted fine-tuning, improving performance for specific attributes.
- **Interpretability:** Unlike the implicit model, the explicit model does not inherently provide attention-based explanations. However, external explainability tools (e.g., LIME, SHAP) can be used to gain insights into predictions.

The paper demonstrates that while the implicit model excels in scenarios requiring high interpretability and scalability, the explicit model is a better fit for applications prioritizing computational efficiency and task-specific optimization. Both models were evaluated on a skincare dataset with domain-expert annotations, showcasing distinct advantages depending on the use case.

2.2 Our Contributions

Our work builds on the concepts introduced in BT-BERT [1] while addressing some of its practical limitations. Instead of relying on domain experts for labeling, we utilize publicly available Amazon product metadata and customer reviews to construct our dataset. This approach enhances scalability by removing the need for labor-intensive product attribute labeling, making it easier to adapt the system for real-world applications. By leveraging naturally occurring data, we capture a broader variety of ingredient-attribute relationships, reflecting the diversity and dynamic nature of consumer products.

We also opted for the explicit model architecture for our implementation. This decision was guided by practical constraints, as the explicit model requires less training time and is computationally feasible given limited resources. While the implicit model offers direct interpretability through attention values, the explicit model allows for more flexibility in downstream tasks, especially when paired with advanced explainability tools. We employed metrics such as SHAP [4], LIME [5], and Integrated Gradients [6] to analyze and visualize the relationships between product ingredients and attributes, providing actionable insights into which ingredients are most critical for specific product properties. For instance, our analysis highlights key ingredients for attributes like hydration, acne treatment, and sensitive skin, which can inform product formulation and customer recommendations.

Overall, our approach emphasizes scalability, computational efficiency, and transparency, offering a practical alternative to traditional expert-driven models while maintaining the explainability of ingredient-attribute relationships. This balance of performance and interpretability makes our system well-suited for addressing real-world challenges in the beauty industry.

3 DATA PREPARATION

BT-BERT [1] did not make the expertly labeled data used in its training and validation pipeline publicly available. Consequently, we developed and curated our own dataset to establish a comprehensive training and validation pipeline.

In this section, we outline the detailed steps taken to prepare the data for our analysis, including data collection, preprocessing, and the extraction of relevant features. These steps were instrumental in building a dataset capable of analyzing ingredient-attribute relationships and training a multi-label classification model.

3.1 Data Collection

To construct our dataset, we sourced data from the publicly available Amazon Product Reviews dataset hosted by UCSD's dataset repository. This dataset in total comprises approximately 23 million reviews and metadata for over 1 million beauty products, including product titles, descriptions. The rich textual data offered by this dataset served as the foundation for identifying relationships between ingredients, product attributes, and customer feedback. While each review likely contains less meaningful information about makeup attributes than expert labels, if we are able to make explainable connections between them the scale of the data could be very useful in further investigation. Due to computational constraints, for all experiments we tested on a fraction of this data with only 10 thousand entries selected from 100 thousand products based on review availability. (similarly sized to the Beauty Beyond Words dataset). Exploration into the larger dataset can be seen in the Experiments section.

In addition to the Amazon dataset, we utilized a glossary of skincare ingredients sourced from Byrdie's glossary of skincare ingredients [3]. This domain-specific resource was used to extract ingredients from product descriptions.

3.2 Data Preprocessing

The raw data underwent extensive preprocessing to standardize, clean, and extract meaningful features from the combined dataset. These steps were crucial to ensure compatibility between the Amazon data and the external ingredient glossary.

3.2.1 Text Cleaning. To minimize noise and emphasize relevant information, we standardized all text data, including product descriptions and customer reviews, using the following steps:

- Converted all text to lowercase for uniformity.
- Removed punctuation, special characters, and accents to ensure consistent tokenization.
- Excluded stopwords unrelated to skincare, such as "bottle" and "packaging," which do not contribute to attribute classification.
- Tokenized the cleaned text, splitting it into meaningful components for further analysis.

3.2.2 Ingredient Extraction. Using Byrdie's skincare glossary [3], we extracted ingredients from product descriptions and metadata. A rule-based phrase-matching technique was implemented with a regular expression (regex) pattern generated from the ingredient glossary. Key steps in this process included:

- Case-insensitive matching of ingredient names within product descriptions.
- Deduplication and normalization of identified ingredients to ensure consistency across the dataset.
- Creation of structured representations of ingredients for each product, facilitating their use as input features for the classification model.

3.2.3 Attribute Extraction. Attributes such as skin type (e.g., "oily," "dry"), concerns (e.g., "acne," "anti-aging"), and preferences were extracted from customer reviews using a dictionary and regex-based attribute extraction approach. This process aligned with the predefined attribute categories outlined in the *Beauty Beyond Words* paper. Key steps included:

- Developing a dictionary of keywords for each attribute category using related phrases and synonyms (see Table 1).
- Employing a spaCy-based pipeline for efficient phrase matching and extraction of attributes from the text.
- Mapping extracted attributes to the corresponding product descriptions to establish relationships between ingredients and attributes.

Table 1: List of Attribute Keywords Used for Entity Recognition

Attribute	Keywords
Skin Types	
Normal Skin	normal, balanced, healthy, ...
Oily Skin	oily, greasy, shiny, ...
Combination Skin	combination, mixed, dry and oily, ...
Sensitive Skin	sensitive, irritation, reactive, ...
Skin Concerns	
Acne	acne, pimple, blemish, ...
Hydration	hydrating, moisture, moisturizing, ...
Pores	pores, enlarged pores, pore size, ...
Fine Lines and Wrinkles	wrinkles, fine lines, aging, ...
Sagging	sagging, loose, loss of firmness, ...
Dark Spots	dark spots, hyperpigmentation, ...
Redness	redness, red patches, inflammation, ...
Uneven Texture	uneven texture, rough, bumpy, ...
Dark Circles	dark circles, under-eye, eye bags, ...

3.3 Integration of Ingredients and Attributes

To construct the final dataset for training, we combined extracted ingredient and attribute data. Key steps included:

- Aggregating product reviews and metadata by unique product identifiers.
- Grouping extracted ingredients and attributes for each product.
- One-hot encoding the ingredient and attribute lists for compatibility with the classification model.

The resulting dataset represents products as multi-label examples, where ingredients form the input features and attributes serve as the target labels.

Ultimately, following our data preprocessing pipeline as outlined in Flowchart 1, we ended up with a dataset comprising of approximately 3,800 entries of products, with product ingredients and their skin attributes.

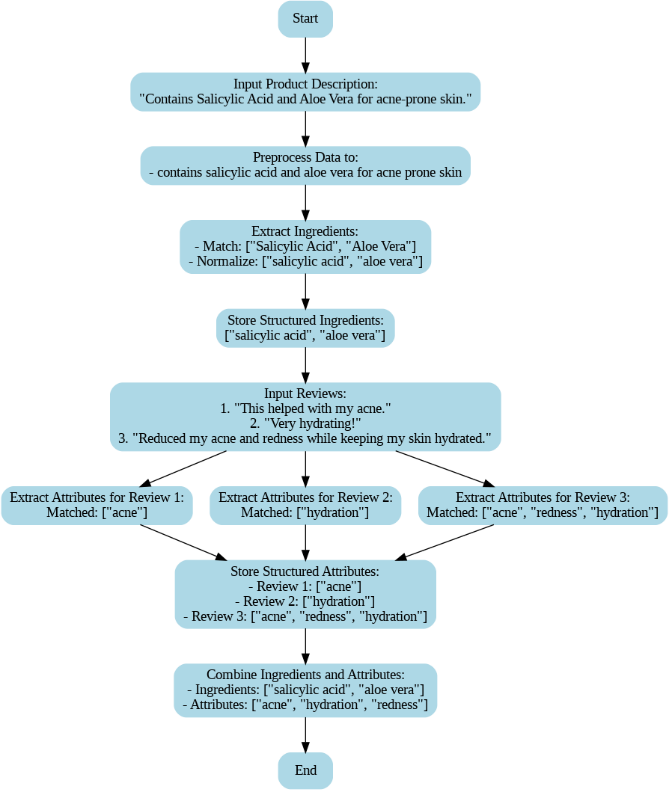

Start
Input Product Description:
"Contains Salicylic Acid and Aloe Vera for acne-prone skin."
Preprocess Data to:
- contains salicylic acid and aloe vera for acne prone skin
Extract Ingredients:
- Match: ["Salicylic Acid", "Aloe Vera"]
- Normalize: ["salicylic acid", "aloe vera"]
Store Structured Ingredients:
["salicylic acid", "aloe vera"]
Input Reviews:
1. "This helped with my acne."
2. "Very hydrating!"
3. "Reduced my acne and redness while keeping my skin hydrated."
Extract Attributes for Review 1:
Matched: ["acne"]
Extract Attributes for Review 2:
Matched: ["hydration"]
Extract Attributes for Review 3:
Matched: ["acne", "redness", "hydration"]
Store Structured Attributes:
- Review 1: ["acne"]
- Review 2: ["hydration"]
- Review 3: ["acne", "redness", "hydration"]
Combine Ingredients and Attributes:
- Ingredients: ["salicylic acid", "aloe vera"]
- Attributes: ["acne", "hydration", "redness"]
End

Figure 1: Flowchart representing the pipeline for extracting ingredients and attributes.

3.4 Challenges in Data Collection

A significant limitation of our data collection approach is the quality of the dataset. Unlike the *Beauty Beyond Words* paper, which utilized expertly annotated labels, our data relies on publicly available metadata and reviews. This trade-off prioritizes scalability and adaptability over precision, reflecting real-world constraints. This also makes it difficult to validate our model’s performance in comparison to BT-BERT’s.

Despite these challenges, our preprocessing pipeline ensured a structured and meaningful dataset capable of supporting multi-label classification tasks.

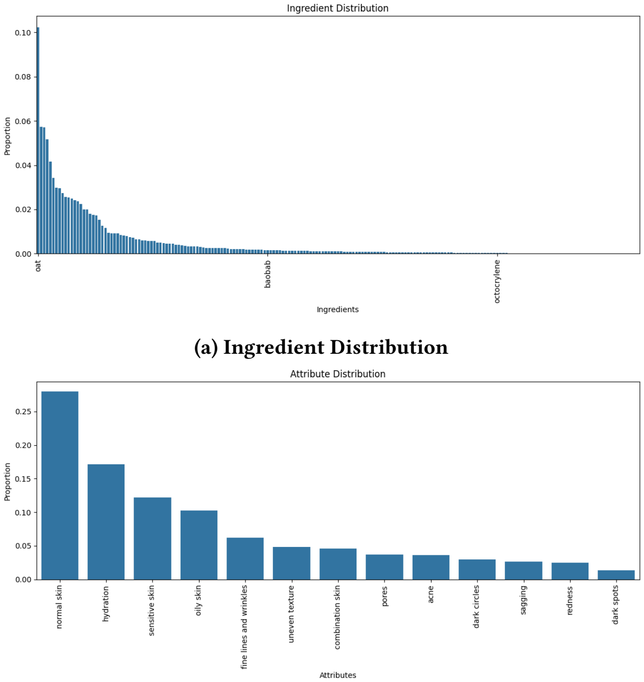
4 METHODS

4.1 Exploratory Data Analysis

As stated in the Data Collection section, the dataset we are working with in total comprises approximately 23 million reviews and metadata for over 1 million beauty products, includ- ing product titles, descriptions. This allows for up to a million data entries, 100x the size of Beauty Beyond Words’ dataset. While Beauty Beyond Words found limited effectivness of larger datasets in training their implicit BERT model, they performed no experiments on explicit models or other larger natural language models. We believe this larger dataset has potential to be leveraged for makeup ingredient explainability and to this end we perform some initial data analysis.

First we want to calculate the amount of usable data we are able to extract from the Amazon dataset. To do this we find the following three figures.

- **Average Number of Reviews per Product**: 23.24 reviews.
- **Average Number of Ingredients per Description**: .51 ingredients.
- **Average Number of Attributes per Review**: .71 attributes.



(a) Ingredient Distribution

(b) Attribute Distribution

Figure 2: Distribution of Ingredients and Attributes in Product Descriptions and Reviews

The average number of reviews per product was 23.24. Even with the average number of attributes mentioned per description being .71, we can expect multiple associated attributes with every product in the dataset. The number of ingredients we are able to extract was not as promising, and may contribute to the low precision and recall scores we saw in our evaluation. Further processing the data, to select for products with more elaborate ingredient descriptions or supplementing this dataset with ingredient descriptions could prove useful in future work.

To further understand the dataset, we present the distribution of ingredients and attributes extracted from product descriptions and reviews in Figures 2a and 2b. As expected there are large imbalances in both the ingredients and attribute distributions. Due to the size of the dataset, a dataset with a balanced distribution of product attributes could easily be made. Ingredients on the other hand are much more difficult, with many ingredients appearing or not at all. Again this could be corrected with supplemental data, or alternatively this data could instead be used to track correlation with fad ingredients (like oat) popular in the makeup industry.

Based on our dataset, some of the top correlated attributes and ingredients, based on our data, are as follows in Table 2. However, more detailed studies of explainability of ingredient-attribute relationships will be performed in upcoming sections.

Table 2: Top Correlations Between Ingredients and Attributes

Ingredient	Attribute	Correlation
Sodium Ascorbyl Phosphate	Sagging	0.240
Caffeine	Dark Circles	0.206
Hyaluronic Acid	Fine Lines/Wrinkles	0.179
Hyaluronic Acid	Dark Circles	0.175
Collagen	Dark Circles	0.167
Kojic Acid	Dark Spots	0.159

4.2 Our Approach

Our approach is based on the explicit model architecture introduced in the *BT-BERT* paper. We adapted this methodology for our task by training a multi-label classification model to predict product attributes based on text input derived from product metadata. This section provides a detailed account of the data preparation, model architecture, training process, and the explainability techniques used to interpret model predictions.

4.3 Input and Output Representation

The input to our model consisted of concatenated product ingredients and titles, formatted as a single text sequence. We had approximately 3,800 entries in our dataset. Ingredients, extracted as described earlier, were separated by commas and normalized for consistency (e.g., lowercased and stripped of extra spaces). Product titles, sourced from metadata, were appended to the ingredient list, separated by a special token [SEP]. For example, a product with ingredients *"salicylic acid"* and *"aloe vera"* and the title *'Acne control serum'* was represented as:

```
"salicylic acid, aloe vera [SEP] Acne Control Serum"
```

The output consisted of multi-label predictions for attributes such as *acne*, *hydration*, and *dark circles*, where each attribute was treated as a binary classification task, outputting probabilities between 0 and 1.

4.4 Preprocessing for BERT

To prepare the input for the BERT [2] model, several preprocessing steps were applied:

1. **Text Cleaning:** The concatenated ingredient and title text was standardized by lowercasing, removing punctuation, and normalizing whitespace.
2. **Tokenization:** Using the pre-trained tokenizer (`AutoTokenizer.from_pretrained()`), the text was tokenized into subword units. Special tokens ([CLS] and [SEP]) were added to the beginning and end of each input sequence.
3. **Padding and Truncation:** Input sequences were padded to a maximum length of 128 tokens, and longer sequences were truncated. This ensured uniform input dimensions suitable for batch processing.
4. **Attention Masks:** Binary attention masks were generated to differentiate between real tokens (1) and padded tokens (0). These masks helped the model ignore padding during computations.

The processed input for the BERT model consisted of three components:

- **Input IDs:** Integer-encoded tokens for each subword in the sequence.
- **Attention Mask:** Binary masks indicating valid tokens.
- **Token Type IDs:** Segment identifiers to distinguish between different parts of the input, though these were not used in our case since we only had a single segment.

The output was a vector of probabilities for each attribute, with values indicating the likelihood of the product being associated with that attribute.

4.5 Model Architecture

We implemented an explicit BERT-based model for multi-label classification, consisting of the following components:

- **Transformer Encoder:** The core of the model was a pretrained BERT architecture [2], which served as the backbone to process input text and produce contextualized embeddings for each token. To explore the impact of different transformer architectures, we experimented with three BERT backbones:
  - `BAAI/bge-base-en-v1.5`: Optimized for domain-specific tasks and capable of handling multi-label classification efficiently.
  - `bert-base-uncased`: A widely used general-purpose BERT model, included as a baseline for comparison.
  - `roberta-base`: A robustly optimized BERT variant, designed to enhance downstream task performance through additional pre-training optimizations.
- **Classification Head:** Following the transformer encoder, a fully connected linear layer was employed to map the [CLS] token's embedding (a 768-dimensional vector) to a vector of logits corresponding to the attributes. Each logit represented the raw score predicting the likelihood of a specific attribute being associated with the input.

The model architecture is defined as follows:

```
class ExplicitBTBERT(nn.Module):
    def __init__(self, model_name, num_labels):
        super(ExplicitBTBERT, self).__init__()
     self.bert = AutoModel.from_pretrained(model_name)
        self.classifier = nn.Linear(768, num_labels)

    def forward(self, input_ids, attention_mask):
        outputs = self.bert(input_ids, attention_mask)
        cls_output = outputs.last_hidden_state
        logits = self.classifier(cls_output)
        return logits
```

4.6 Training Process

The training process was divided into two phases:

1. **Classifier Head Training:** During the first four epochs, the BERT backbone was frozen, and only the classifier head was

trained. This allowed the classifier to adapt to the dataset while preserving the pre-trained embeddings.

(2) **Fine-Tuning:** In the subsequent four epochs, the entire model, including the BERT backbone, was fine-tuned using a lower learning rate. This step helped the model better capture domain-specific nuances in the data.

Both phases used binary cross-entropy loss for multi-label classification and the AdamW optimizer with a learning rate of 5e-5 for the classifier training and 2e-5 for fine-tuning.

4.7 Explainability Methods

To interpret the model predictions and uncover the relationships between ingredients and product attributes, we employed several explainability techniques. These methods provided insights into how specific ingredients influenced the likelihood of a product being classified under certain attributes, ensuring both transparency and actionable insights.

- **SHAP (SHapley Additive exPlanations):** This game-theoretic approach assigns importance scores to individual input features (e.g., ingredients or tokens) by evaluating their marginal contributions to the model's output. In our case, SHAP values were computed for each token in the input sequence, allowing us to identify which ingredients or words had the highest influence on a given attribute prediction.
- **LIME (Local Interpretable Model-agnostic Explanations):** LIME generates interpretable explanations by perturbing the input and observing changes in the model's predictions. For each product, we perturbed the tokenized inputs (e.g., removing or altering specific ingredients) and trained a local surrogate model to approximate the behavior of the classifier. This method highlighted the most influential tokens and phrases, by measuring their impact on the predicted probabilities of each attribute.
- **Integrated Gradients:** This gradient-based technique calculates the attribution of each feature by integrating the gradients of the model's output with respect to the input features, along a path from a baseline (e.g., an all-zero input) to the actual input. We applied Integrated Gradients to analyze the contributions of individual tokens, such as ingredients and product titles, to the predicted probabilities of attributes.

4.7.1 *Implementation Details.*

- For **SHAP**, we utilized the `SHAP` Python library, adapting it to the BERT-based multi-label classification model by defining a custom prediction function. This function ensured compatibility with SHAP's internal computations, allowing it to explain the contribution of tokens in the context of multi-label outputs.
- For **LIME**, the `LIMETextExplainer` class was used, with inputs tailored to the tokenized text sequences from the BERT tokenizer. Perturbations were generated by removing tokens (e.g., ingredients) or phrases, and the corresponding changes in predicted probabilities were analyzed to determine feature importance.
- For **Integrated Gradients**, we implemented the method using the `Captum` library, a PyTorch-based explainability framework. Token embeddings were used as inputs, and the attributions were computed with respect to the model logits for each attribute. These attributions were then mapped back to the tokenized input, enabling ingredient-level interpretability.

These explainability methods collectively provided a comprehensive understanding of how the model associated specific ingredients with attributes, aiding in model evaluation and interpretation. Additionally, the results highlighted actionable ingredient-attribute relationships.

4.8 Alternative Method: Implicit Model

We explored alternative approaches, such as using an implicit model architecture based solely on attention scores without an additional classification layer. However, this approach was computationally intensive and less flexible for downstream tasks compared to the explicit architecture. The explicit model, by contrast, offered a balance between computational efficiency and explainability, making it more suitable for our resource-constrained setup.

5 EXPERIMENTS

5.1 Explicit Model Results

The performance of each of the 3 model architectures with different pre-trained backbones was evaluated using F1 score, precision, recall, and accuracy, as shown in Table 3. These metrics provided a comprehensive evaluation of the models' ability to accurately predict product attributes while minimizing false positives and false negatives.

Table 3: Performance Metrics for Different Transformer Architectures

Architecture	F1 Score	Precision	Recall	Accuracy
`BAAI/bge-base-en-v1.5`	0.6059	0.7527	0.5658	0.3413
`bert-base-uncased`	0.5348	0.9146	0.4563	0.3307
`roberta-base`	0.1460	0.1738	0.1312	0.0040

5.1.1 *Comparing Model Architecture Performances.*

As per Table 3, when we compare the results of `BAAI/bge-base-en-v1.5` to those of other popular transformer models, specifically `bert-base-uncased` and `roberta-base`, we see marked differences in performance that underscore the complexity of the multi-label classification task.

`bert-base-uncased`, while showing a relatively high precision (0.9146), struggles with recall (0.4563), which leads to a significantly lower F1 score of 0.5348. This indicates that `bert-base-uncased` is highly selective in its predictions—it classifies true positives correctly when it does so, but it misses a substantial portion of them. The low recall value suggests that the model is biased toward predicting negatives, possibly due to an overemphasis on precision. This is reflected in its overall accuracy of 0.3307, indicating that it is not capturing the breadth of the attribute classes well. The high precision, however, means that when `bert-base-uncased` does make a positive prediction, it is often correct. This could imply that the model is particularly adept at identifying rare, well-defined patterns but struggles when faced with more diverse or ambiguous examples, which is typical for fine-tuned models in multi-label classification scenarios.

On the other hand, roberta-base exhibits particularly poor performance, with an F1 score of just 0.1460, precision of 0.1738, and recall of 0.1312. The near-zero accuracy (0.0040) suggests that this model is severely misaligned with the task, possibly due to insufficient fine-tuning or issues related to hyperparameter settings. It is not successfully distinguishing between relevant and irrelevant attributes, as indicated by its low precision and recall values. These results point to a model that, despite being powerful in other NLP tasks, is unable to generalize well on this particular multi-label task, potentially due to either the fine-tuning process or the inherent complexity of the task.

In comparison, BAAI/bge-base-en-v1.5 outperforms both bert-base-uncased and roberta-base across most metrics. The relatively balanced F1 score and better overall recall and precision values highlight BAAI/bge-base-en-v1.5's ability to generalize effectively across a wider range of product attributes. It demonstrates a stronger ability to cope with class imbalances in the dataset, as it is able to consistently predict attributes such as acne, oily skin, and sensitive skin, which have higher class frequencies. In contrast, bert-base-uncased's higher precision suggests that it can be more accurate when predicting positive samples but fails to capture as many of the true positives as BAAI/bge-base-en-v1.5, especially for underrepresented classes. roberta-base's results, as noted, are concerning and highlight the importance of careful fine-tuning and understanding the architecture's suitability for specific tasks.

These comparative results illustrate the challenges faced in multi-label classification tasks, especially when dealing with class imbalance. While BAAI/bge-base-en-v1.5 stands out as the most effective model overall, the performance of bert-base-uncased and roberta-base reinforces the need for targeted adjustments in model architecture, fine-tuning strategies, and data preprocessing. To improve the performance of these models, we could explore strategies such as class balancing, data augmentation, or the incorporation of more domain-specific features that may better align with the target attributes.

The performance metrics for the best-performing model, BAAI/bge-base-en-v1.5, are summarized in Table 4. These metrics include accuracy, precision, recall, and F1-score for each attribute, showcasing the model's effectiveness across various product characteristics.

5.1.2 *Comparing Performance Metrics for Each Attribute using Best Model* BAAI/bge-base-en-v1.5.

According to Table 4, the BAAI/bge-base-en-v1.5 model demonstrates good class-specific performance, with F1-scores for the highest-performing classes—**acne**, **hydration**, **oily skin**, and **sensitive skin**—ranging from 0.67 to 0.68. These attributes also achieved strong precision, with values exceeding 0.82. Recall values, however, remained slightly lower, indicating the model's ability to identify products targeting these attributes (positive cases) with reasonable accuracy but a tendency to miss some relevant predictions. Accuracy values for these attributes, however, remain relatively low (around 0.38–0.41), reflecting challenges in correctly identifying both positive and negative cases across the dataset.

Table 4: Performance Metrics for Each Attribute using BAAI/bge-base-en-v1.5

Attribute	Accuracy	Precision	Recall	F1-Score
Acne	0.4112	0.8866	0.6278	0.6833
Hydration	0.4107	0.8684	0.6215	0.6797
Sensitive Skin	0.3924	0.8300	0.6140	0.6746
Oily Skin	0.3838	0.8212	0.6091	0.6721
Combination Skin	0.3759	0.8077	0.5033	0.5993
Fine Lines and Wrinkles	0.3438	0.7639	0.5354	0.5143
Pores	0.3016	0.7153	0.5647	0.6573
Redness	0.2756	0.6878	0.5761	0.5570
Normal Skin	0.3939	0.6043	0.5820	0.6266
Dark Circles	0.2922	0.6525	0.5220	0.4585
Dark Spots	0.2939	0.5509	0.5439	0.5198
Sagging	0.2910	0.7720	0.5406	0.6477
Uneven Texture	0.2709	0.8247	0.5147	0.5865

In contrast, attributes such as **uneven texture**, **sagging**, and **dark spots** exhibited lower F1-scores and significantly reduced precision and recall. For example, **uneven texture** and **dark spots** recorded F1-scores below 0.60, with accuracies around 0.27–0.29. This suggests that the model struggles to generalize for these less frequent classes, leading to higher rates of false negatives (e.g., missing products designed for these attributes) and false positives (e.g., incorrectly associating unrelated products with these attributes).

The overall low accuracy across all classes suggests the model's difficulty in reliably predicting both "Present" and "Not Present" labels for attributes, likely due to the multi-label nature of the task and imbalances in the dataset. The higher F1-scores for certain attributes highlight the model's strength in focusing on relevant positive cases, such as acne and hydration, but also reveal gaps in comprehensive coverage for less frequent attributes.

This performance discrepancy underscores the importance of addressing dataset imbalances and class representation in multi-label classification tasks. Enhancements such as data augmentation, class weighting, or improved architecture designs could help improve both precision and recall for underrepresented attributes, while boosting the model's overall accuracy across the task.

5.2 Confusion Matrices for the Explicit Model

The confusion matrices below summarize the performance of the BAAI/bge-base-en-v1.5 model in predicting the presence or absence of key skin conditions. The results indicate varying levels of accuracy, with observable false positives and false negatives across the conditions.

The confusion matrices highlight the variability in model performance across the three best-predicted skin conditions. The model demonstrates strong precision for all three attributes, indicating its ability to correctly identify "Present" labels. This is perhaps the most important metric if this methodology is used for integration into a recommender system, a sentiment shared by the original paper. This is because if the model's precision is low, irrelevant attributes (False Positives) will result in recommendations for products that do not address the user's actual needs. This undermines the system's utility and credibility.

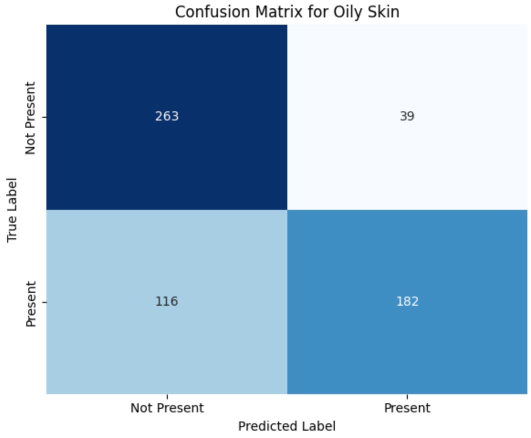


Figure 3: Confusion Matrix for Oily Skin

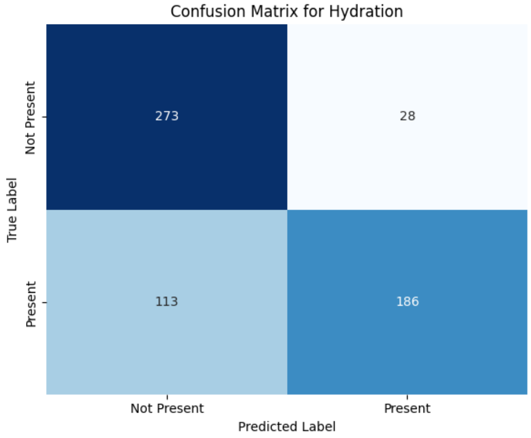


Figure 4: Confusion Matrix for Hydration

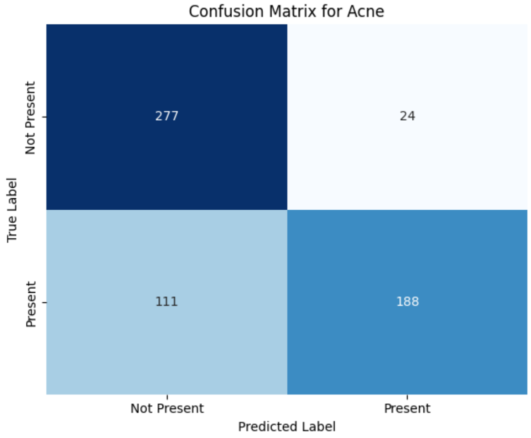


Figure 5: Confusion Matrix for Acne

However, the presence of False Negatives across attributes suggests the need for improvement in recall, particularly for **Oily Skin**.

We will look into the explainability of these 3 skin attributes in upcoming sections using relevant metrics.

5.3 LIME-Based Insights on Ingredient Importance

The Local Interpretable Model-Agnostic Explanations (LIME) method was applied to analyze the contribution of individual ingredients to the model predictions for **hydration** and **acne**. LIME provides interpretable, local explanations by perturbing input features and observing their impact on predictions, making it ideal for identifying key ingredients that influence the model outputs.

Hydration Analysis. The results for hydration, as shown in Figure 6, highlight the importance of ingredients widely recognized for their moisturizing properties:

- **Hyaluronic Acid**: A well-known humectant that draws moisture into the skin, contributing significantly to hydration scores.
- **Shea Butter**: A common emollient known for its ability to soften and moisturize the skin.
- **Glycerin**: Another humectant frequently used in skincare products to retain moisture and improve hydration levels.

These ingredients align well with existing dermatological knowledge, demonstrating the model's ability to identify scientifically validated hydration enhancers.

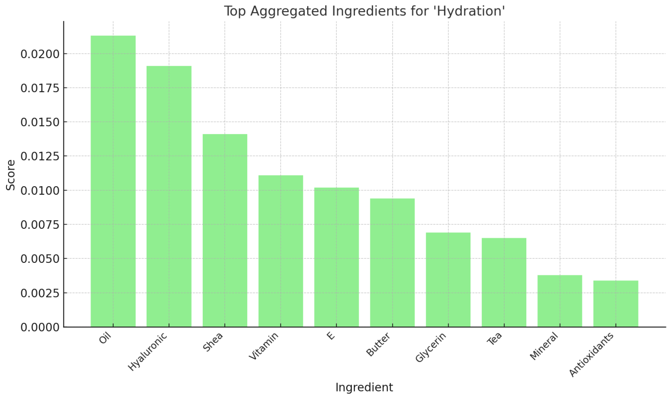


Figure 6: LIME-based ingredient importance for hydration predictions.

Acne Analysis. For acne predictions, the LIME analysis, as shown in Figure 7, identified ingredients commonly used for treating acne:

- **Salicylic Acid**: A beta hydroxy acid (BHA) known for its exfoliating properties, which unclog pores and reduce acne formation.
- **Glycolic Acid**: An alpha hydroxy acid (AHA) that promotes exfoliation, removes dead skin cells, and improves skin texture.

These findings demonstrate the model's alignment with established acne treatment ingredients, reinforcing its interpretability and reliability.

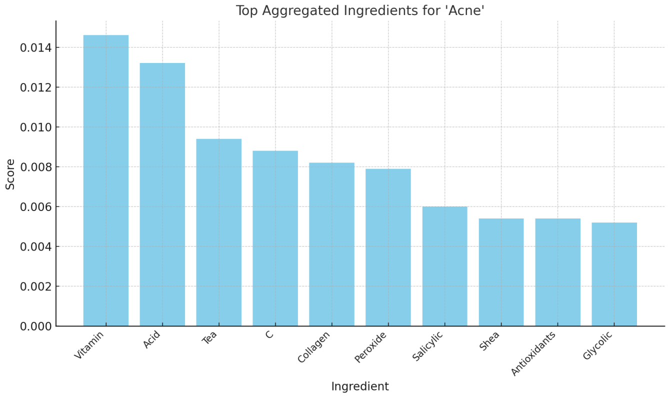


Figure 7: LIME-based ingredient importance for acne predictions.

Overall, the LIME-based analysis shows that the model effectively captures domain-relevant ingredient contributions for both hydration and acne.

5.4 SHAP Insights on Ingredient Importance

The SHapley Additive exPlanations (SHAP) method was applied to analyze the contribution of individual ingredients to the model predictions for acne and hydration. SHAP provides global and local interpretability by assigning each feature a value representing its contribution to a given prediction. This method is especially effective for identifying key ingredients that influence the model's outputs and ensuring transparency in the prediction process.

Acne Analysis. The SHAP analysis for acne, as shown in Figure 8, reveals the positive correlations of ingredients that contribute significantly to acne predictions. Notably, ingredients known for their therapeutic or anti-inflammatory properties emerged as key factors:

- **Cyclopentasiloxane**: A silicone-based compound with emollient properties, contributing positively to acne predictions with a SHAP value of 0.2735.
- **Doxycycline**: An antibiotic used in the treatment of acne, exhibiting a SHAP value of 0.2288, indicating its positive influence on reducing acne-related symptoms.
- **Pycnogenol**: A potent antioxidant that helps in reducing inflammation and oxidative stress, contributing positively to acne treatment with a SHAP value of 0.2087.
- **Dihydroxyacetone**: A compound used in tanning products, with a modest contribution to acne prediction (0.0453).
- **Probiotics**: Known for their gut health benefits, probiotics are shown to have a minor but positive effect on acne (0.0347).

All these ingredients reflect established therapeutic approaches for acne management, reinforcing the model's alignment with dermatological science. Although in this case, most of these ingredients are rare and most commonly used for treating symptoms other than acne although they all help with acne. It may be that these have the higest correlation due to small sample size.

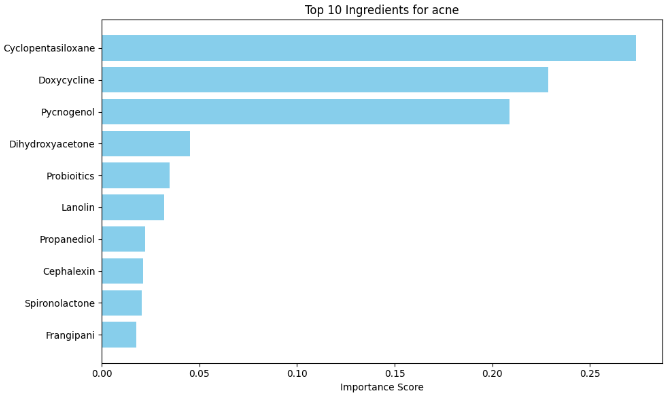


Figure 8: SHAP-based ingredient importance for acne predictions.

Hydration Analysis. The SHAP analysis for hydration, shown in Figure 9, highlights the contribution of ingredients that are widely recognized for their moisturizing and skin-enhancing properties:

- **Pycnogenol**: A powerful antioxidant with hydration-promoting benefits, contributing significantly to hydration predictions with a SHAP value of 0.1336.
- **Caffeine**: Known for its stimulating and anti-inflammatory effects, caffeine was positively correlated with hydration (SHAP value: 0.1145).
- **Panthenol**: A pro-vitamin of B5 that helps in maintaining skin hydration, contributing positively with a SHAP value of 0.1139.
- **Doxycycline**: In addition to its acne treatment properties, doxycycline also showed a mild positive correlation with hydration (0.0922).
- **Zinc**: A mineral known for its anti-inflammatory and healing properties, it contributed positively to hydration predictions (0.0812).

Pycnogenol and Zinc ingredients are well-known for their skin hydration properties, but Caffine and Panthenol are not. While both used to improve skin health and appearance, they are not related directly to skin hydration. Due to the nature of our review dataset, the higher correlation in the fad ingredients may represent a perceived benefit in the user more than any chemical correlation. Overall, the SHAP-based analysis demonstrates that the model effectively captures domain-relevant ingredient contributions for both acne and hydration. These findings ensure the model's transparency and support its alignment with established skincare principles. With more work into analyzing the Shapley values returned by our model, greater insights could be made into the connection between ingredients and satisfying various skin attributes.

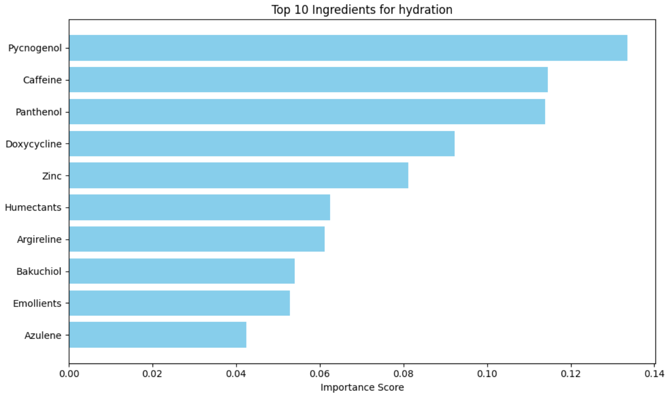


Figure 9: SHAP-based ingredient importance for hydration predictions.

5.5 Integrated Gradients Insights on Important Sub-word Tokens for Key Attribute

The tables 5 and 6 highlights the top tokens contributing to the model's predictions for specific skin attributes of acne and hydration, identified using the Integrated Gradients method. For each token in the input text, an attribution score was calculated on the validation set by measuring its contribution to the model's prediction for the "acne" or "hydration" label. These scores were derived by averaging the Integrated Gradients attribution scores across all validation examples where a given label was active. Subword tokens have been mapped to their full ingredients or contextual meanings wherever applicable.

5.5.1 Acne Analysis. Table 5 highlights the top tokens contributing to the model's predictions for acne, identified using the Integrated Gradients method.

Token	Average Attribution Score	Ingredient
##tino	0.8290	Retinol (from `retino`)
sulfur	0.4662	Sulfur
##eth	0.4478	Ethanol
zinc	0.4271	Zinc
##bu	0.4199	Butylene glycol
per	0.3930	Peroxide
sal	0.3625	Salicylic acid (from `sal`)
benton	0.3493	Bentonite clay
##oxide	0.3470	Peroxide or zinc oxide
##jic	0.3370	Kojic acid
tree	0.3270	Tea tree oil

Table 5: Top tokens contributing to acne predictions, with subwords mapped to full ingredients or contextual meanings where applicable.

Analysis: The results demonstrate that the model identifies relevant acne treatment ingredients, such as sulfur, zinc, and salicylic acid, as key contributors. Subword tokens, such as `##tino` (retinol) and `##oxide` (zinc oxide or peroxide), align well with known skincare formulations.

5.5.2 Hydration Analysis. Table 6 below highlights the top tokens contributing to the model's predictions for hydration, identified using the Integrated Gradients method.

Token	Average Attribution Score	Ingredient
##eth	0.9065	Ethanol
##anu	0.5271	Manuka (honey)
petroleum	0.4896	Petroleum
##bu	0.3558	Butylene glycol
jelly	0.3519	Petroleum jelly
##cta	0.2791	Possibly cetrimonium
##ram	0.2785	Possibly ceramid
goat	0.2754	Goat milk
honey	0.2636	Honey

Table 6: Top tokens contributing to hydration predictions, with subwords mapped to full ingredients or contextual meanings where applicable.

Analysis: The results demonstrate that the model identifies relevant hydration-related ingredients, such as petroleum jelly, goat milk, and honey, as key contributors. Subword tokens, such as `##eth` (ethanol) and `jelly` (petroleum jelly), align with known hydrating formulations. However, some tokens, such as `##cta`, and `##ram`, remain ambiguous and suggest potential areas for refinement in token mapping or further analysis.

Furthermore, ethanol is known for its drying properties, rather than hydrating. Hence, it is odd that it achieves the highest average attribution score. This could be because of its role as a solvent or antimicrobial agent rather than as a direct contributor to moisture retention.

6 CONCLUSION

This study demonstrates the potential of transformer-based architectures, particularly the `BAAI/bge-base-en-v1.5` model, to predict beauty product attributes with competitive performance while maintaining explainability. By leveraging an explicit multi-label classification approach, the model successfully identified key attributes such as acne, hydration, and sensitive skin with F1-scores ranging from 0.67 to 0.68 and precision values exceeding 0.82. These results highlight the system's reliability in identifying relevant product features, crucial for applications like recommendation systems where irrelevant suggestions can undermine user trust.

Our application of explainability techniques, including Integrated Gradients, LIME, and SHAP, provided actionable insights into ingredient-attribute relationships. For instance, the model attributed salicylic acid and retinol as primary drivers for acne-related predictions, while hydration-related predictions were linked to ingredients like petroleum jelly and goat milk. These findings align with established dermatological knowledge, reinforcing the model's interpretability and practical relevance.

Despite these successes, the model exhibited lower performance on underrepresented attributes such as uneven texture and dark spots, underscoring challenges posed by class imbalances in multi-label tasks. Additionally, certain subword tokens, such as ##eth and ##cta, remained ambiguous, suggesting opportunities for refining feature extraction and token interpretation.

6.1 Limitations

Several limitations of the study must be acknowledged. First, the quality and limited size of the dataset posed significant challenges. Unlike the expertly curated datasets used in prior research, our dataset relied on publicly available metadata and reviews, which lack validation against ground truth beauty data. This likely impacted the model's performance, particularly for underrepresented classes, where the lack of sufficient positive examples hindered generalization.

Furthermore, the original dataset from the UC San Diego database is extremely large, a lot of the data was unstructured and lacked ingredient information or any mention of our relevant attributes. As such, only a small subset of the dataset was usable. Furthermore, due to computational constraints, we were only able to parse portions of the dataset without encountering memory issues.

Second, while we experimented with attention-based mechanisms for explainability, their effectiveness was limited, potentially due to the smaller dataset size and uneven representation of attributes. However, LIME and SHAP successfully circumvented this limitation by providing interpretable insights even in the absence of strong attention signals.

Third, the multi-label nature of the task presented additional complexity, as the model had to balance predicting multiple attributes simultaneously while managing data imbalances. Enhancements such as better data augmentation, targeted sampling, and class weighting could mitigate these issues in future work.

6.2 Future Work

Future research should aim to address these limitations by leveraging a much larger dataset with significantly more reviews per product.

This would ensure more robust training and better representation of underrepresented classes, thereby improving the model's ability to generalize across diverse attributes. Expanding the dataset with evenly distributed classes and exploring advanced transformer architectures, semi-supervised learning techniques, or domain-specific pretraining could further enhance model performance. These efforts would ultimately lead to more reliable ingredient-attribute insights, fostering innovation in beauty product analysis and recommendation systems.

REFERENCES

[1] Celine Liu, Rahul Suresh, Amin Banitalebi-Dehkordi. 2024. *Beauty Beyond Words: Explainable Beauty Product Recommendations Using Ingredient-Based Product Attributes.* arXiv preprint. Available at: https://arxiv.org/pdf/2409.13628

[2] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. 2019. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT).* Association for Computational Linguistics, 4171-4186. Available at: https://arxiv.org/abs/1810.04805.

[3] Byrdie Editors. 2024. *The Complete Guide to Skincare Ingredients.* Available at: https://www.byrdie.com/skincare-ingredients-glossary-4800556. Accessed: December 19, 2024.

[4] Lundberg, S. M., and Lee, S.-I. 2017. *A Unified Approach to Interpreting Model Predictions.* In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17).* Curran Associates Inc., 4765–4774. Available at: https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf

[5] Ribeiro, M. T., Singh, S., and Guestrin, C. 2016. *"Why Should I Trust You?": Explaining the Predictions of Any Classifier.* In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD).* ACM, 1135–1144. Available at: https://arxiv.org/abs/1602.04938

[6] Sundararajan, M., Taly, A., and Yan, Q. 2017. *Axiomatic Attribution for Deep Networks.* In *Proceedings of the 34th International Conference on Machine Learning (ICML).* PMLR, 3319–3328. Available at: https://arxiv.org/abs/1703.01365.

A GITHUB REPOSITORY

The complete codebase, including preprocessing, training, evaluation scripts, and explainability metrics, is available in the following GitHub repository. This repository also includes instructions for replicating experiments and using the model for multi-label classification.

The entire repository has been developed by our team from scratch, with no direct use of code from other repositories or papers. `https://github.com/shreyasriram4/transformers_beauty`