PAE: LLM-based Product Attribute Extraction for E-Commerce Fashion Trends

Apurva Sinha
apurva.sinha@walmart.com
Walmart Global Tech

Ekta Gujral
ekta.gujral@walmart.com
Walmart Global Tech

Abstract

Product attribute extraction is a growing field in e-commerce business, with several applications including product ranking, product recommendation, future assortment planning and improving online shopping customer experiences. Understanding the customer needs is critical part of online business, specifically fashion products. Retailers use assortment planning to determine the mix of products to offer in each store and channel, stay responsive to market dynamics and to manage inventory and catalogs. The goal is to offer the right styles, in the right sizes and colors, through the right channels to fostering customer loyalty. In this paper we present PAE, a product attribute extraction algorithm for future trend reports consisting text and images in PDF format. Most existing methods focus on attribute extraction from titles or product descriptions or utilize visual information from existing product images. Compared to the prior works, our work focuses on attribute extraction from PDF files where upcoming fashion trends are explained. Our contributions are three-fold: (a) We develop PAE, an efficient framework to extract attributes from unstructured data (text and images); (b) We provide catalog matching methodology based on BERT representations to discover the existing attributes using upcoming attribute values; (c) We conduct extensive experiments with several baselines and show that PAE is an effective, flexible and on par or superior (avg 92.5% F1-Score) framework to existing state-of-the-art for attribute value extraction task.

Keywords

Attribute Extraction, PDF files, Large Language Model (LLM), Text and Images, BERT embeddings

ACM Reference Format:
Apurva Sinha and Ekta Gujral. 2024. PAE: LLM-based Product Attribute Extraction for E-Commerce Fashion Trends. In *Proceedings of the first workshop on Generative AI for E-Commerce 2024, October 25, 2024.* ACM, New York, NY, USA, 7 pages.

1 Introduction

Assortment planning for future products plays a crucial role in the success of e-Commerce as a platform. It involves strategically selecting and organizing a range of products to meet customer demands and maximize sales. This process involves analyzing market trends, customer preferences, and competitor strategies to identify potential gaps and opportunities. By carefully planning the assortment, retailers can ensure they offer a diverse and relevant range of products that cater to different customer segments. This helps in driving customer satisfaction, increasing sales, and staying ahead in the competitive market. Walmart collaborates with trend forecasting company that provides insights and analytics for the fashion and creative industries. They don't release public reports because their insights are provided through a paid subscription service. However, they often share snippets of their forecasts via blog posts or on social media. For example, they might report on upcoming color trends for a particular season, predict consumer behaviors, or identify emerging fashion trends in different regions. The trend forecasting company also provides reports on retail and marketing strategies, textiles and materials innovations, product development and lifestyle and interiors trends. Their reports are typically used by retailers and marketers to plan and develop their products and strategies.

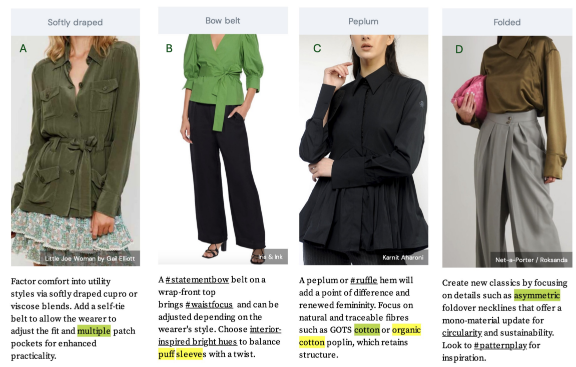


Figure 1: Example of Text and Images for Attribute Extraction

Correctly predicted attributes improve catalog mapping, which in turn facilitates the generation of search tags for products with higher quality content. Customers can filter for products based on their exact needs and compare product variants. Resulting in a seamless shopping experience while searching or browsing a product on an E-commerce platform. The Product attribute Extraction (PAE) engine can help the retail industry to onboard new items or extract attributes from existing catalogs.

Previous Works We provide a brief overview of existing Multi Modal Attribute Extraction (MMAE) techniques being used to extract product attributes from Images and Text. MMAE explained in paper [10] talks about returning values for attributes which occur in images as well as text and they do not treat the problem as a labeling problem. Another approach for MMAE explained in paper [5], considers cross-modality comparisons. They leverage pre-trained deep architectures to predict attributes from text or image data. By applying several refinements to leverage pre-trained architectures and build single

Genaiecom '24, October 25, 2024, Boise, ID

modality models like Text only modality model, image only modality model for the task of product attribute prediction. Paper by [14] talks about Multi modal Joint Attribute Prediction and Value Extraction for E-commerce Product. They enhance the semantic representation of the textual product descriptions with a global gated cross-modality attention module that is anticipated to benefit attribute prediction tasks with visually grounded semantics.

Informal Problem 1. Can we develop unsupervised models that require limited human annotation? Additionally, can we develop models that can extract explainable visual attributes, unlike black-box methods that are difficult to debug?

Mitigating Challenges: Current multi-modal attribute extraction solutions [5, 10] are inadequate in the e-commerce field when it comes to handling challenges related to text and image extraction from PDF files and then mapping product attributes to product catalog. Conversely, text extraction solutions that successfully extract attribute values are primarily text-oriented [2, 6, 12, 13] and cannot be easily applied to extracting attributes from images. In this work, we address the central question: how can we perform multi-modal product attribute extraction from upcoming trend PDF reports? The detail description is given in section 3 to handle each challenge. Our proposed method PAE works on extracting upcoming trends from PDF reports generated by the trend forecasting company. This capability provides an insight into upcoming marketing trends and customer preferences. By using trend forecasting reports, catalog can be refined with new classes of products having trending attributes based on external reports, to propel value across the apparel space by accurately indicating attribute trends in the market and increasing customer satisfaction. The contributions of our paper are as follows:

- **Novel Problem Formulation**: We propose the end-to-end model of jointly extracting the trending product attributes from PDF files consisting of text and image data and mapping it back with the product catalog for the final product attributes values.
- **Flexible Framework**: We develop a general framework PAE for extracting text and images from PDF files and then generating product attributes. All the components are easily modified to enhance the capability or to use the framework partially for other applications. See Figure 2.
- **Experiments**: We performed extensive experiments in real-life datasets to demonstrate PAE 's efficacy. It successfully discovers attribute values from text and image data with a high F1-score of 96.8%, outperforming state-of-the-art models. This proves its ability to produce stable and promising results.

2 Problem Definition

The upcoming trend information in PDF files usually looks like figure 1. The text describes the upcoming trends and style types. The images along with it shows how the style will look on different models. We consider each page of PDF file as one product type and that can be Woven Tops, Knitwear etc. We formally define the problem as follows.

Problem definition: With the following information:

> **Given** (a) a PDF file with multiple pages $[1, 2, 3 \ldots, N]$ consist of text T_1^N and image I_1^N data (b) LLM prompt P with target attributes $attr \in$ [color, sleeve style, product type, material, features, categories, age and neck]
> **Find** the value $vals$ for the target attribute $attr$ related for each page.

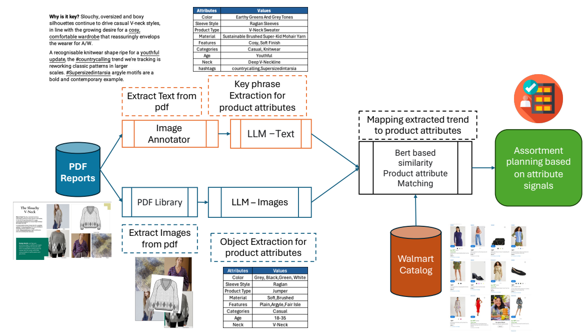


Figure 2: Overview of the proposed Product Attribute Extraction Engine

3 Product Attribute Extraction

In this work, we tackle the attribute value extraction as pair task, i.e., extracting the attribute values from image and text together. The input of the task is a “textual information T, set of images $I_1, I_2 \ldots I_N$ pair per PDF page, and the output is the product attributes values. Our framework is presented in figure 2.

3.1 Text Extraction from PDF

Text extraction from PDF is an important process that involves the conversion of data contained in PDF files into an editable and searchable format. This procedure is crucial for activities like data analysis, content re-purposing, and detecting trends from public reports. However, it can pose certain challenges. The layout complexity of a PDF document can make the extraction process difficult. For instance, the presence of multiple columns, images, tables, and footnotes can complicate extraction of pure text. Another challenge is the use of non-standard or custom fonts in PDFs, which can lead to inaccurate extraction results. Moreover, the presence of 'noise' such as headers, footers, HTML tags and page numbers can also interfere with the extraction process. There are numerous tools available for text extraction from PDF files. Searching for text extraction from PDF on Google yields a plethora of results featuring various tools or pages suggesting such tools as pdfMiner [11], pdfquery[4] etc. However, figure 3 represents the process we used to extract the text from pdf files. First, we split the PDF files into PIL (Python Imaging Library) images using "convert from path" function from the pdf2image [1]. Internally, the function uses the *pdfinfo* command-line tool to extract metadata from the PDF file, such as the number of pages. It then uses the *pdftocairo* command-line tool to convert each page of the PDF into an image. Second, we convert the images to grayscale and perform morphological transformations on each page by applying a morphological gradient operator to enhance and isolate text regions. Finally, we use Image Annotator [7] consists of Optical Character Recognition (OCR) capabilities for text extraction. Once the text is extracted, we use the spell Corrector like language-tool to fix any misinterpreted text from OCR. The Text extracted from PDF report

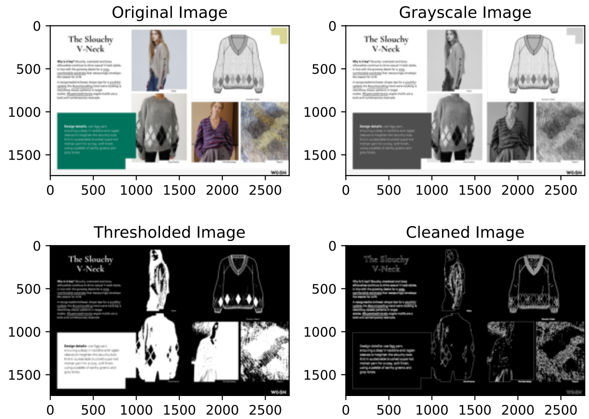


Figure 3: Text Extraction via Image Annotator

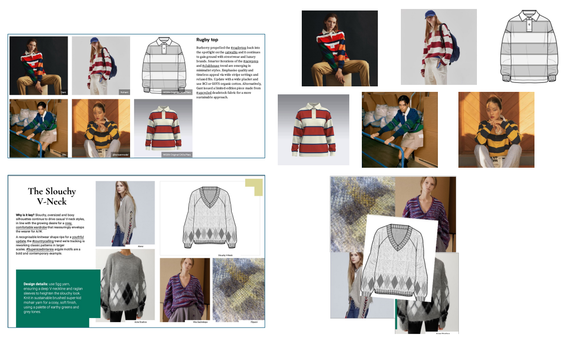


Figure 4: Example of extracted images from PDF files

about product type "Slouchy V-Neck" is shown in Example 1 (italic text).

3.2 Image Extraction from PDF

PDF files can contain images in various formats such as JPEG, PNG, or TIFF. Extracting images from different formats may require multiple techniques. The second challenge could be different types of images in the pdf files, including scanned documents, vector graphics, or embedded images. Third, extracting images from large PDF files efficiently and in a timely manner can be a challenge, especially when dealing with limited system resources. To tackle the above-mentioned challenges, we exploit pure-python PDF library [3], as a standalone library for directly extracting image objects from PDF files. With pure-python PDF library, we identify the pages with images and extract them as raw byte strings. Then, using Pillow, the extracted images are processed and saved in JPG formats. Figure 4 shows extracted images from files.

3.3 Attribute Extraction from Text

PDF reports consist of specific products or product categories, providing details on their design, features, materials, colors, and styles. These reports provide information about product innovations, market demand, and consumer preferences in upcoming months or years. Here, we extract 8 product attributes namely [Color, Sleeve Style, Product Type, Material, Features, Categories, Age, Neck]. We are utilising the Text Bison LLM model for extracting the attributes. Below is a sample prompt and example of text extracted:

> **Example 1**
> **Prompt:** Give me color, sleeve style, product type, material, features, categories, age and neck attributes from the following text:
> **Extracted Text T:** *"The Slouchy V-Neck Why is it key? Slouchy, oversized and boxy silhouettes continue to drive casual V-neck styles, in line with the growing desire for a cosy, comfortable wardrobe that reassuringly envelops the wearer for A/W. A recognisable knitwear shape ripe for a youthful update, the #countrycalling trend we're tracking is reworking classic patterns in larger scales. #Supersizedintarsia argyle motifs are a bold and contemporary example. Design details: use 5gg yarn, ensuring a deep V-neckline and raglan sleeves to heighten the slouchy look. Knit in sustainable brushed super-kid mohair yarn for a cosy, soft finish, using a palette of earthy greens and grey tones."*

The output is then processed in dictionary type as follow:

> **Output of above text example**
> *Color*: Earthy Greens And Grey Tones
> *Sleeve Style*: Raglan Sleeves
> *Product Type*: V-Neck Sweater
> *Material*: Sustainable Brushed Super-Kid Mohair Yarn
> *Features*: Cosy, Soft Finish
> *Categories*: Casual, Knitwear
> *Age*: Youthful
> *Neck*: Deep V-Neckline

3.4 Attribute Extraction from Images

The extraction of detailed image attributes from fashion images has a wide range of uses in the field of e-commerce. The recognition of visual image attributes is vital for understanding fashion, improving catalogs, enhancing visual searches, and providing recommendations. In fashion images, the dimensionality can be higher due to the complexity and diversity of fashion items. For instance, a single piece of clothing can have multiple attributes for color, fabric type, style, design details, size, brand, and others. Hence, image attribute extraction has become more complex than text. However, these attributes can be extracted using various computer vision techniques, such as image segmentation, object detection, pattern recognition and deep learning algorithms. In this work, we explore the vision based LLM model. Each extracted image as shown in figure 5 is converted to base64 encoding. Base64 encoding is a method of converting binary data, such as an image, into ASCII text format. This is required as current Gemini LLM model takes text format as input. The ASCII text format example as follow:

> **ASCII text format**
> *'iVBORw0KGgoAAAANSUhEUgAAAU4AAA GwCAIAAABJqRtXAAAACXBIWXMAAA7EAAAOxAGV . . . '*

Next, we use this encoded string along with LLM prompt to generate the product attributes as follow:

Figure 5: Image Example

Example
*text: "**Give** a list format color, sleeve style, product type, material attributes from the the below image. Also give me features, categories, age and neck attributes from below image."*
"inlineData": {
"mimeType": "image/png",
"data": *'iVBORw0KGgoAAAANSUhEUgAAAU4AAA GwCAIAAABJqRtXAAAACXBIWXMAAA7EAAAOxAGV . . . '* }

The output is then processed in dictionary type as follow:

Output of above image example
Color: Multicolor,
Sleeve Style: Long Sleeve
Product Type: Pullover
Material: Wool Blend
Features: V-Neck, Drop Shoulder
Categories: Women's Fashion
Age: Adult
Neck: V-Neck

Another common issue that arises is the presence of noisy and missing labels. It is a challenging task to accurately label and annotate all the relevant information for every page in the PDF. Despite employing various automated and manual annotation processes, it is nearly impossible to obtain perfectly labeled structured data. To address this, we employ image pre-processing or data cleaning techniques to eliminate duplicate, noisy, and invalid images before proceeding with attribute extraction. *Once we extract attributes from text and images on each page, we aggregate the extracted attributes per page for our further analysis.*

3.5 Product attribute Matching

Product attribute matching is a process that ensures extracted attributes meet specific criteria and can be consistently compared to existing product information. One of the challenges of product attribute matching is the presence of multiple variants of representation for the same attribute value. For example, "vneck" and "V-Neck" both represent the same neck product attribute and need to be consolidated as "V-Neck". We leverage a pre-trained BERT model, which is designed to learn deep bidirectional representations from unlabeled text by considering both left and right context in all layers. This allows us to create representations or embeddings for the extracted attributes and existing product attributes. Finally, we use cosine similarity to match similar product attributes from the catalog.

Dataset	$\#P$	$\#T$	$\#I$	$\#H$	GT
Boys Apparel	7	11	32	0	Y
Women's Cut Sew	7	30	24	5	Y
Women's Woven Tops	6	28	24	6	Y
Country Life Boys	12	12	66	3	Y
Knitwear Core	6	11	31	4	Y
Knitwear Fashion	12	21	70	7	Y
Woven Tops Core	6	13	35	0	Y
Woven Tops Fashion	13	24	74	4	Y

Table 1: Datasets used in the experiments. For each PDF file, we extracted all the pages. $\#P$ represents number of pages, $\#T$ represents number of text blocks, $\#I$ represents number of total images present in pdf file. Here $\#H$ represents hashtags available in pdf file and GT is ground truth attributes are available for the pdf file.

4 Experiments

Although our work is mostly related to retail business, we will compare the performance of our PAE's sub-parts with different baselines on real-life datasets. We evaluate our approach on 8 upcoming trend reports. In particular, we want to answer the following questions:

- **(Q1)** How accurate is our proposed method PAE when compared to other baselines?
- **(Q2)** How sensitive is PAE w.r.t different parameters?
- **(Q3)** How time consuming is PAE?

4.1 Data-set description

We provide the datasets used for evaluation in Table 1. These are trend reports used for 2023 assortment planning. Assuming the attribute value is applicable to each pdf page, if the attribute value information can not be observed from the given text description and images, we will assign "Not Mentioned" as the corresponding value.

4.2 Evaluation Measures

We use Accuracy, True Positive Rate (Recall) and F1 score as the evaluation metrics. We compute Accuracy (denoted as P) as percentage of correct value generated by our framework; True positive rate (denoted as TPR) as percentage of ground truth value retrieved by our framework; F1 score (denoted as $F1$) as harmonic mean of Precision and Recall.

4.3 Baselines

To evaluate our proposed framework, we choose the following models as baselines for text: topic rank [2] and sOpenTag [12]. Our attribute value extraction task for images is highly related to the visual question answering tasks. Thus, we used two baselines, Vilt [8], and BLIP [9] for visual attribute value extraction.

4.4 Accuracy of PAE

For all datasets we compute F1-score (%) for text and images. The results for qualitative measure for data is shown in Table 2. We observed that F1-score (Image) is perfect for all the dataset as images provide clear visual attributes for future trends. However, text data has missing attributes and PAE is able to extract average 92.5% of attributes from the all the PDF files.

Dataset	F1-score (Text)	F1-score (Image)
Boys Apparel	94.3%	100%
Women's Cut Sew	88.6%	100%
Women's Woven Tops	100%	100%
Country Life Boys	89.4%	100%
Knitwear Core	92.5%	100%
Knitwear Fashion	97.8%	100%
Woven Tops Core	96.8%	100%
Woven Tops Fashion	98.9%	100%

Table 2: Test accuracy for multiple datasets for PAE

Dataset	**PAE**	**Topic Rank** [2]	**sOpenTag**[12]
Precision	100%	93.3%	61.4%
True Positive Rate	93.9%	42.4%	73.4%
Accuracy	95.3%	54.7%	86.2%
F1-Score	96.8%	59.5%	66.7%

Table 3: Text Attribute extraction accuracy for 'Woven Tops Core' datasets for PAE and state-of-art-methods.

Attributes	**PAE**	**Vilt** [8]	**BLIP**[9]
Color	100%	87.5%	87.5%
Sleeve Style	100%	00.0%	50.0%
Product Type	100%	62.5%	62.5%
Material	100%	75.0%	75.0%
Features	100%	50.0%	25.0%
Categories	100%	75.0%	100%
Age Group	100%	75.0%	00.0%
Neck	100%	12.5%	62.7%

Table 4: Accuracy per attribute for 'Woven Tops Core' dataset for PAE and state-of-art-methods on images.

Further we compare the performance of PAE with the aforementioned baseline values for text and images separately. We use "Woven Tops Core" dataset with 8 pages. We evaluate text extraction on 6 pages out of 8 with 7 attributes namely Sleeve Style, Features, Product Type, Material, Neck Style, Product Categories and Color. From these 7 attributes, we have total 33 attribute values and 9 not mentioned values. Table 3 shows the performance of our method compared to baseline methods. Topic rank and sOpenTag performed very well on 'Woven Tops Core' dataset. However, PAE outperforms baselines in-terms of F1 score and accuracy. We also note that our proposed method is 2.7× faster as compared to Topic rank and sOpenTag.

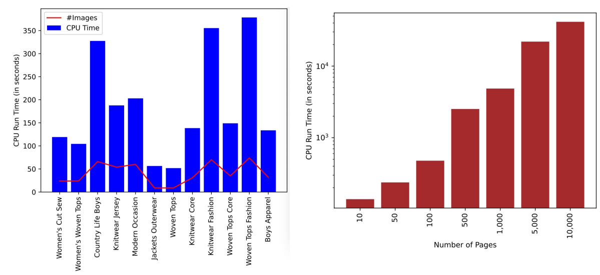
We evaluate image attribute extraction on all 8 attributes namely Sleeve Style, Product Type, Material , Neck, Categories, Age, Features and Color for all 8 pages with image information. From these 8 attributes, we have total 64 visual attribute values. Table 4 shows that PAE outperforms the baseline methods. Both the baselines were limited to pass single attribute in prompt, therefore results in consuming more time in producing the final attribute values per page. This answer our question **Q1**.

4.5 Parameter of sensitivity for PAE

In this study, we evaluate the performance of temperature parameter of LLM models. Temperature is a parameter in large language models (LLMs) that controls the randomness of the model's responses, ranging from 0 to 1. A higher temperature means more creative and diverse output, while a lower temperature means more predictable output. As we need predictable output, we kept the temperature parameter below 0.5. The table 5 shows that at temperature = 0.2, method performance is high. Therefore, we chose to keep 0.2 as parameter value. Due to limited space, we provide sensitivity analysis

Dataset	0.05	0.1	0.2	0.4
Precision	85.7%	93.7%	100%	100%
True Positive	72.2%	90.9%	93.9%	90.9%
Accuracy	78.5%	88.1%	95.3%	92.8%
F1-Score	84.2%	92.3%	96.8%	95.2%

Table 5: Sensitivity to LLM temperature parameters for attribute extraction from text data in 'Woven Tops Core' dataset.



(a) (b)

Figure 6: CPU Time Analysis: (a) Running time for each PDF report. (b) Running time for synthetic PDF report.

in supplementary document. To summarize, as expected, PAE is sensitive to both LLM prompt and temperature parameter. This answer our question **Q2**.

4.6 CPU Time Analysis

In this work, we present two experiment results. First, we provide CPU time for each dataset in figure 6. We observe that CPU time is directly proportional to number of images in the PDF files. "Women Tops Fashion" PDF file has 74 images and took around 350 seconds to get attributes for all 13 pages.

Second, we created synthetic PDF files with each consisting of 500-1000 words and 4-6 images per page. We created PDF files with $[10, 50, 100, 500, 1000, 5000, 10000]$ pages. The figure 6 shows that CPU time is linear w.r.t size of the PDF file. This answer our question **Q3**.

5 CONCLUSIONS AND FUTURE WORK

This paper introduces our attribute extraction framework, PAE, designed for e-commerce applications. Given a PDF containing trend information in form of text and images, PAE accurately extracts predefined attributes. This extracted data empowers businesses to plan future assortments by predicting relevant attributes and associating them with existing catalog products. We acknowledge opportunities for further development. One avenue is exploring Large Language Models (LLMs) capable of processing combined image and text sets to generate comprehensive attribute sets. Additionally, we aim to improve the product matching system, that can be utilized images to find the products to enhance customer experience during e-commerce product searches.

References

[1] Edouard Belval. pdf2image. https://pypi.org/project/pdf2image/, 2017.

[2] Adrien Bougouin, Florian Boudin, and Béatrice Daille. Topicrank: Graph-based topic ranking for keyphrase extraction. In *International joint conference on natural language processing (IJCNLP)*, pages 543–551, 2013.

[3] claird. Pypdf4. https://pypi.org/project/PyPDF4/, 2018.

[4] Jack Cushman. Pdfquery. https://github.com/jcushman/pdfquery/tree/master, 2013.

[5] Aloïs De la Comble, Anuvabh Dutt, Pablo Montalvo, and Aghiles Salah. Multi-modal attribute extraction for e-commerce. *arXiv preprint arXiv:2203.03441*, 2022.

[6] Pushpendu Ghosh, Nancy Wang, and Promod Yenigalla. D-extract: Extracting dimensional attributes from product images. In *WACV 2023*, 2023.

[7] Google. Google cloud vision api. https://cloud.google.com/python/docs/reference/vision/latest.

[8] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *International conference on machine learning*, pages 5583–5594. PMLR, 2021.

[9] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022.

[10] Robert L Logan IV, Samuel Humeau, and Sameer Singh. Multimodal attribute extraction. *arXiv preprint arXiv:1711.11118*, 2017.

[11] PYusuke Shinyama. pdfminer. https://www.unixuser.org/~euske/python/pdfminer/, 2004.

[12] Huimin Xu, Wenting Wang, Xinnian Mao, Xinyu Jiang, and Man 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*, pages 5214–5223, 2019.

[13] Guineng Zheng, Subhabrata Mukherjee, Xin Luna Dong, and Feifei Li. Opentag: Open attribute value extraction from product profiles. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1049–1058, 2018.

[14] Tiangang Zhu, Yue Wang, Haoran Li, Youzheng Wu, Xiaodong He, and Bowen Zhou. Multimodal joint attribute prediction and value extraction for e-commerce product. *arXiv preprint arXiv:2009.07162*, 2020.

6 Supplementary Information

7 Introduction

Motivating Example Retailers can use upcoming market trends to decide on product catalog assortment planning based on upcoming seasons like spring, fall and summer. For a concrete example, refer to Figure 1, a classic shirt (with unstructured text and image data), it talks about *peplum* or *ruffle hem* being used on Global Organic Textile Standard *(GOTS) cotton* or *organic cotton poplin*. This can be referenced to having classic shirts in a catalog made of Organic/GOTS cotton and peplum/ruffle hem as features for the shirt. Based on these attribute insights, assortment planners would work closely with suppliers and designers to curate a collection of such clothing items to complete the look, including leggings, sports bras, sweatshirts, and sneakers. They would consider factors such as quality, affordability, and inclusively to ensure that the assortment caters to a wide range of customers. The images shows popular prints, innovative fabrics, and style variations within the classic shirt category. This would enable retailers to offer a diverse selection of classic shirt options that align with the latest trends. Additionally, once retailers have an assortment ready for selling, they could collaborate with fashion influencers who align with the upcoming trend to create exclusive collections or promote the existing assortment. This would help to generate excitement among customers and drive high engagement. By incorporating the recommended color palettes, visual elements, and messaging, the retailer could create an immersive shopping experience.

Challenges Despite the potential, leveraging PDF reports consisting text and images for attribute value extraction remains a difficult problem. We highlight few challenges faced during designing and executing extraction framework:

- **C1: Text Extraction from PDF**: PDF reports can be a combination of multiple images, overlapping text elements, annotations, metadata and unstructured text integrated together in no specific PDF format. Extracting text from such reports can be difficult, challenging and lead to misspelled text and loss of specific topic-related context. Another issue is missing and noisy attributes. Text data might not have all the attributes which we are looking for. Therefore, visual attribute extraction plays an important role.
- **C2: Image Extraction from PDF**: Images in PDF reports can be embedded, compressed down to reduce size, in various formats like JPEG, PNG etc. Extracting images while maintaining the resolution and quality of images requires specialized handling to accurately preserve the original appearance. Also, images could bring multi-labeled attributes which can confuse the model but can be mitigated by merging certain attribute values to help with model inferences.
- **C3: Extracting Product Attributes**: Product tags extracted from text/images needs to be carefully mined to match product attributes. The attributes differ based on the category of products we are referring to and can have multi-labeled attributes. For example, women's tops will have sleeve related attribute whereas women's trousers will have type of fit attribute and sleeve attribute will be irrelevant.
- **C4: Mapping Product Attributes to Product Catalog**: E-commerce catalog has specific products and attributes mapped

Dataset	0.05	0.1	0.2	0.4
Precision	85.7%	93.7%	100%	100%
True Positive	72.2%	90.9%	93.9%	90.9%
Accuracy	78.5%	88.1%	95.3%	92.8%
F1-Score	84.2%	92.3%	96.8%	95.2%

Table 7: Sensitivity to LLM temperature parameters for attribute extraction from text data in 'Woven Tops Core' dataset.

Dataset	**Prompt 1**	**Prompt 2**	**Prompt 3**
Precision	83.3%	100%	90.4%
True Positive Rate	15.5%	93.9%	57.6%
Accuracy	30.9%	95.3%	61.9%
F1-Score	25.6%	96.8%	70.3%

Table 6: Sensitivity to LLM Prompt for 'Woven Tops Core' dataset.

to them. On-boarding new attributes based on PDF reports, requires new attribute creation/refactoring existing attributes.

Informal Problem 2. Can we develop unsupervised models that require limited human annotation? Additionally, can we develop models that can extract explainable visual attributes, unlike black-box methods that are difficult to debug?

7.1 Parameter of sensitivity for PAE

7.1.1 Sensitivity to LLM Prompt for Text data. Large Language Models (LLMs) have the ability to learn new tasks on the fly, without requiring any explicit training or parameter updates. This mode of using LLMs is called in-context learning. It relies on providing the model with a suitable input prompt that contains instructions and/or examples of the desired task. Therefore, we evaluate our proposed method PAE for multiple prompts for text and image attribute extraction. Here, we present prompt analysis for attribute extraction from text data only. We keep the temperature parameter constant for this experiment.

- *Prompt 1*: "Give me all clothing characteristics of a product from the following text:"
- *Prompt 2*: "Give me color, sleeve style, product type, material, cloth features, categories, and neck attributes from the following text:"
- *Prompt 3*: "I want you to act as a product attribute extractor in retail space.Given the unstructured text data, you need to find different product attributes in the text. For example: For Input as 'Long contrast fabric Sleeve red cotton adult polo shirts for men with contemporary design element', the attribute extractor will return color attribute is red, sleeve attribute is Long, style sleeve attribute is contrast fabric, product type attribute is polo shirts, material attribute is cotton, feature attribute is contemporary,categories is polo shirts, gender attribute is men and neck attribute is NA. Give me attributes like color, sleeve style, product type, material, features, categories, and neck attributes from the following text: "

The table 6 shows that *Prompt 2* is more effective and efficient way to extract given attributes from the text data. Too vague (*Prompt 1*) and too much (*Prompt 3*) information/context confuses the LLM model and therefore performance degrades.

7.1.2 Sensitivity to LLM Parameters. In this study, we evaluate the performance of temperature parameter of LLM models. Temperature is a parameter in large language models (LLMs) that controls the randomness of the model's responses, ranging from 0 to 1. A higher temperature means more creative and diverse output, while a lower temperature means more predictable output. As we need predictable output, we kept the temperature parameter below 0.5. The table 7 shows that at temperature = 0.2, method performance is high. Therefore, we chose to keep 0.2 as parameter value. In summary, as expected, PAE is sensitive to both LLM prompt and temperature parameter. This answer our question **Q2**.