SMARTAVE: Structured Multimodal Transformer for Product Attribute Value Extraction

Qifan Wang[1], Li Yang[2], Jingang Wang[3], Jitin Krishnan[1], Bo Dai[2], Sinong Wang[1], Zenglin Xu[4], Madian Khabsa[1] and Hao Ma[1]
[1]Meta AI [2]Google Research [3]Meituan Lab
[4]Harbin Institute of Technology
wqfcr@fb.com lyliyang@google.com

Abstract

Automatic product attribute value extraction refers to the task of identifying values of an attribute from the product information. Product attributes are essential in improving online shopping experience for customers. Most existing methods focus on extracting attribute values from product title and description. However, in many real-world applications, a product is usually represented by multiple modalities beyond title and description, such as product specifications, text and visual information from the product image, etc. In this paper, we propose SMARTAVE, a Structure Mltimodal trAnsformeR for producT Attribute Value Extraction, which jointly encodes the structured product information from multiple modalities. Specifically, in SMARTAVE encoder, we introduce hyper-tokens to represent the modality-level information, and local-tokens to represent the original text and visual inputs. Structured attention patterns are designed among the hyper-tokens and local-tokens for learning effective product representation. The attribute values are then extracted based on the learned embeddings. We conduct extensive experiments on two multimodal product datasets. Experimental results demonstrate the superior performance of the proposed approach over several state-of-the-art methods. Ablation studies validate the effectiveness of the structured attentions in modeling the multimodal product information.

1 Introduction

Product attributes are important features that carry useful information about the product. They form an essential component of e-commerce platforms, which provide guidance for customers to compare products and make purchasing decisions. Product attributes also facilitate retailers on various applications, including product search (Nguyen et al., 2020; Lu et al., 2021), product recommendations (Yu et al., 2021; Truong et al., 2022), and question answering systems (Zhang et al., 2020; Rozen et al., 2021; Huang et al., 2022). However, as shown in previous studies (Dong et al., 2020; Yang et al., 2022), product attributes are often noisy and incomplete with a lot of missing values for most retailers. Therefore, it is an important research problem to extract the product attributes with missing values.

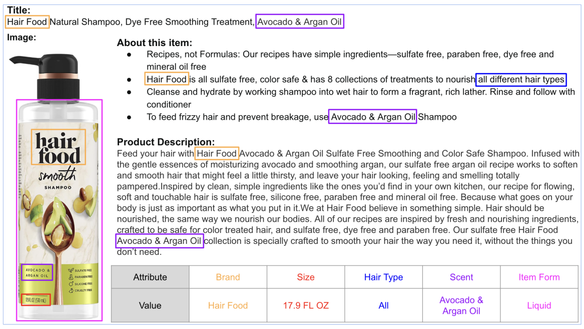
Title:
Hair Food Natural Shampoo, Dye Free Smoothing Treatment, Avocado & Argan Oil

Image:

About this item:

- Recipes, not Formulas: Our recipes have simple ingredients—sulfate free, paraben free, dye free and mineral oil free
- Hair Food is all sulfate free, color safe & has 8 collections of treatments to nourish all different hair types
- Cleanse and hydrate by working shampoo into wet hair to form a fragrant, rich lather. Rinse and follow with conditioner
- To feed frizzy hair and prevent breakage, use Avocado & Argan Oil Shampoo

Product Description:
Feed your hair with Hair Food Avocado & Argan Oil Sulfate Free Smoothing and Color Safe Shampoo. Infused with the gentle essences of moisturizing avocado and smoothing argan, our sulfate free argan oil recipe works to soften and smooth hair that might feel a little thirsty, and leave your hair looking, feeling and smelling totally pampered.Inspired by clean, simple ingredients like the ones you'd find in your own kitchen, our recipe for flowing, soft and touchable hair is sulfate free, silicone free, paraben free and mineral oil free. Because what goes on your body is just as important as what you put in it.We at Hair Food believe in something simple. Hair should be nourished, the same way we nourish our bodies. All of our recipes are inspired by fresh and nourishing ingredients, crafted to be safe for color treated hair, and sulfate free, dye free and paraben free. Our sulfate free Hair Food Avocado & Argan Oil collection is specially crafted to smooth your hair the way you need it, without the things you don't need.

Attribute	Brand	Size	Hair Type	Scent	Item Form
Value	Hair Food	17.9 FL OZ	All	Avocado & Argan Oil	Liquid

Figure 1: An example of product attributes with their corresponding values extracted from multiple modalities of the product.

Attribute value extraction has attracted a lot of attention from both academia and industry in recent years, with a plethora of research (Putthividhya and Hu, 2011a; Zhao et al., 2019; Chen et al., 2019; Shinzato et al., 2022) being proposed to tackle this problem. Most existing works (Zheng et al., 2018; Xu et al., 2019; Wang et al., 2020; Yan et al., 2021) rely solely on the product title and description from the product profiles, which is often insufficient to obtain values for all attributes. In real-world applications, products are usually associated with rich information from other modalities, such as product specification and image. This additional information can improve the attribute value extraction in two main aspects. First, attribute values are sometimes absent in the product title and description. For example, as illustrated in Figure 1, the value '17.9 FL OZ' corresponding to the attribute 'Size' is only mentioned in the OCR text from the product image. The value 'Liquid' of the attribute 'Item Form' is not mentioned in any text source of the

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 263–276
December 7-11, 2022 ©2022 Association for Computational Linguistics

product, but can be inferred from the product visual information. In these cases, the product image is able to recover the missing values of the attributes. Second, product title and description might contain multiple candidate values for an attribute, while other modalities could consolidate the correct value. For example, while both ‘Hair Food’ and ‘Natural’ from the product title are reasonable values for attribute ‘Brand’, the OCR text in the image clearly suggests that ‘hair food’ is the correct answer.

There are a few recent works (Zhu et al., 2020; Lin et al., 2021) that incorporate the product image for attribute value extraction, which achieve promising results. However, these techniques suffer from two major limitations. First, each modality of the product is encoded independently with an individual encoder, followed by a light fusion/cross-modality layer on the top. In this way, the correlations among different modalities are not fully captured, leading to less effective embeddings. Second, the texts from individual modalities are simply concatenated and fed into a single encoder, making these methods inefficient to handle long input.

To address these challenges, in this paper, we propose a novel Structured Multimodal Transformer for Product Attribute Value Extraction (SMARTAVE), which jointly encodes the structured product information from multiple modalities in a unified Transformer model. We first introduce a set of hyper-tokens to represent all modalities of the product, and use local-tokens to indicate the original text and visual inputs within each modality. Structured attention patterns are designed among the hyper-tokens and local-tokens for modeling the correlation between modalities and learning effective product representation. Intuitively, the hyper-tokens serve as ‘hubs’ routing local-tokens from different modalities to interact with each other. The attribute values are then extracted based on the learned embeddings from the structured encoder. Evaluations on two multimodal product datasets show the superior performance of our model over several state-of-the-art methods. The experimental results also demonstrate the effectiveness of the structured attention mechanism in modeling multimodal product data with large input sequences. We summarize the main contributions as follows:

- We propose a novel structured multimodal Transformer that extracts product attribute values from product title, description, specification, visual information, and texts in the product image.
- We develop a structured attention mechanism to jointly encode the product with multiple modalities, which effectively models the correlation among different modalities and learns high quality embeddings.
- We conduct extensive experiments and demonstrate the effectiveness of the proposed approach over several state-of-the-art baselines.

2 Related Work

Attribute Value Extraction Early works in attribute value extraction (Putthividhya and Hu, 2011b; More, 2016) can be viewed as direct or indirect applications of the Named Entity Recognition (NER) method. The advent of deep learning has paved the way for stronger models such as BiLSTM-CRF (Huang et al., 2015) and OpenTag (Zheng et al., 2018), which formulate the problem as a sequential tagging problem and use a combination of methods such as BiLSTM and CRF. SUOpenTag (Xu et al., 2019) builds on top of these methods by jointly encoding the attribute, making the model more scalable. AdaTag (Yan et al., 2021) uses a hyper-network and a Mixture-of-Experts (MoE) module to parameterize its decoder with pre-trained attribute embeddings. AVEQA (Wang et al., 2020) and MAVEQA (Yang et al., 2022) take a step further by solving for the problem of scalability and generalizability of the task via question answering. Meanwhile, TXtract (Karamanolakis et al., 2020) brings a taxonomy-aware approach to aid attribute extraction.

Using visual clues in attribute extraction has garnered attention in recent works (IV et al., 2017; Anderson et al., 2018; Singh et al., 2019; Tan and Bansal, 2019; Hu et al., 2020). MJAVE (Zhu et al., 2020) designs a multimodal model that jointly predicts product attributes and extract values from textual product descriptions with the help of product images. PAM (Lin et al., 2021) incorporates text descriptions, Optical Character Recognition (OCR) and visual modalities from the product. This model fuses the three modalities into a multimodal Transformer and is framed as a sequence generation task, rather than a sequence tagging task. These multimodal methods use multiple encoders to encode different modalities, but fail to capture the structured information and thus are not able to model the relationship effectively among different modalities.

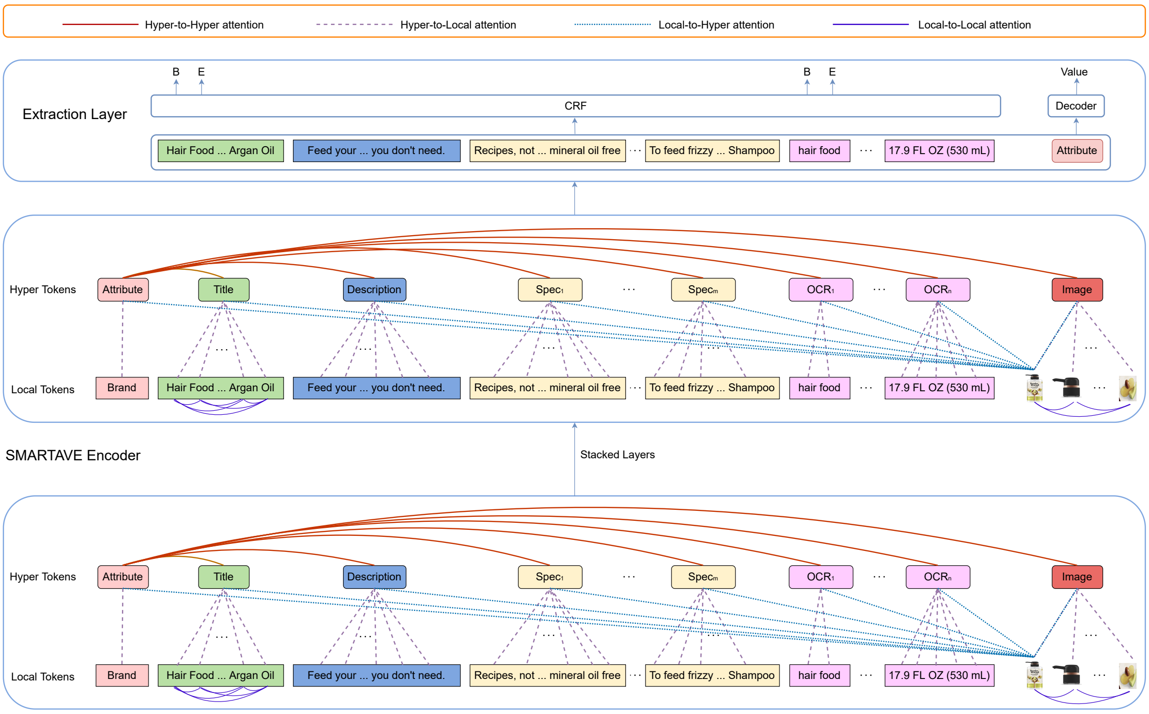

Figure 2: Overview of SMARTAVE model architecture. *Encoder*: our structured encoder jointly encodes the product with multimodal information via structured attentions, and learns effective representation for all the modalities. *Extraction Layer*: the attribute value is decoded from the learned embedding.

Efficient Transformers Our work is also related to those efficient Transformer methods (Beltagy et al., 2020; Rae et al., 2020) that deal with large and structured input. Transformer XL (Dai et al., 2019) designs a mechanism that is able to encode long text sequences beyond a fixed size. Longformer (Beltagy et al., 2020) and ETC (Ainslie et al., 2020) propose to extend the CLS token to multiple global tokens to deal with large text input. HIBERT (Zhang et al., 2019) introduces a hierarchical attention pattern by dividing the input into several blocks with equal sizes. Random sparse attention mechanism is proposed in (Zaheer et al., 2020) which reduces the quadratic attention computations to linear time. These methods achieve promising results on dealing with large and structured text sequences. However, they are not directly applicable to structured data with multiple modalities. A comprehensive study of efficient Transformers can be found in (Tay et al., 2022).

3 SMARTAVE

3.1 Problem Definition

In this section, we formally define the problem of attribute value extraction from the product profile. The product profile contains multiple modalities, such as Title, Description, Specifications, OCR texts, and Image. We denote the product profile as $P = (M_1, M_2, \dots, M_n)$, where M_i represents the i-th modality of the product. For each modality, it is either a text or image sequence, i.e., $M_i = (w_1^i, \dots, w_{m_i}^i)$, where w_j^i is the j-th word or image token in M_i. The goal of attribute value extraction is that given a target attribute A, extract its corresponding values from the product profile.

3.2 Model Overview

The overall model architecture of SMARTAVE is shown in Figure 2. Essentially, our model is composed of three key components, the input layer, the SMARTAVE encoder and the extraction layer. The input layer constructs both the hyper and local tokens of SMARTAVE, and initializes their embeddings. The SMARTAVE encoder is the main building block that jointly encodes the multimodal input data with structured attention patterns, including Hyper-to-Hyper, Hyper-to-Local, Local-to-Hyper and Local-to-Local attentions. The extraction layer consists of a decoder that decodes the value from the embedding of the Attribute hypertoken, and a sequential tagging module that extracts the value from the text modalities. Note that the final attribute value is directly obtained from the decoder, while the sequential tagging module provides an auxiliary task for learning better embeddings.

3.3 Input Layer

Existing multimodal attribute value extraction approaches (Zhu et al., 2020; Lin et al., 2021) encode each modality of the product separately with individual encoders. In this work, we jointly model all modalities with texts and images in a unified structure Transformer model. In the input layer of SMARTAVE, we construct two different types of tokens as follows.

Hyper-token For each modality in the product profile, we introduce a Hyper-token in our SMARTAVE. Additionally, we also add one Hyper-token to represent the attribute. The embedding of each Hyper-token can be viewed as a summarization of the information contained in the corresponding modality. For instance, in Figure 2, the embedding of the 'Title' Hyper-token summarizes the text sequence in the product title. The 'Attribute' Hyper-token essentially represents the product-dependent embedding of the target attribute.

Local-token For text modality, the Local-token is the commonly used word representation in natural language models (Vaswani et al., 2017; Devlin et al., 2019). For example, 'OCR_1' contains two Local-tokens, 'hair' and 'food'. For image modality, each Local-token corresponds to an image patch (Dosovitskiy et al., 2021). The creation of these image patches is flexible without any constraint. In our implementation, we adopt the Faster R-CNN model (Ren et al., 2017) to obtain the image patches/regions from the product image, with one additional patch representing the whole image.

In the input layer, every token is represented by a d-dimensional embedding vector. In particular, for a Hyper-token, its embedding is constructed by adding a hyper embedding, a type embedding and a modality embedding. For a Local-token embedding, it is constructed by a word/patch embedding, a type embedding and a modality embedding. The word embedding is widely adopted in the literature (Zou et al., 2013). The patch embedding is directly obtained through ResNet101 (He et al., 2016), which produces a fixed length visual feature. We then learn a linear projection to map it to the d-dimensional embedding space. The hyper embedding is randomly initialized for each Hyper-token. The type embedding is added to indicate which type the token belongs to, i.e. Hyper or Local. The modality embedding is used to distinguish between different modalities, e.g., 'Title', 'Description', 'Image' etc. Note that all the embeddings in our approach are trainable. The word embeddings are initialized from the pre-trained language model.

3.4 SMARTAVE Encoder

The SMARTAVE encoder contains a stack of K identical encoder layers, which bridges the Hyper and Local tokens from multiple modalities with structured attention patterns, and generates effective contextual representations of the product and attribute. To better capture the information contained in different modalities, we design four attention patterns. First, Hyper-to-Hyper attention that encodes the relations among different Hyper-tokens. Second, Hyper-to-Local attention, which connects the Hyper token with its corresponding Local-tokens. Third, Local-to-Hyper attention that passes the information from the Hyper-tokens to the Local-tokens. Fourth, Local-to-Local attention that learns contextual embeddings from other Local-tokens within the same modality.

Hyper-to-Hyper Attention The Hyper-to-Hyper attention is designed to model the relations among different modalities, which essentially computes the attention weights among the Hyper-tokens and propagates the information from one modality to another. In real-world applications, the total number of modalities n for a product is usually small, e.g., $n \leq 20$. Therefore, we adopt the full attention mechanism among the Hyper-tokens, i.e., each Hyper-token is able to attend to all other Hyper-tokens. Formally, given the Hyper-token embedding X^H, the Hyper-to-Hyper full attention is defined as:

$$A^{H2H} = softmax\left(\frac{X^H W_Q^{H2H}(X^H W_K^{H2H})^T}{\sqrt{d}}\right)$$

where W_Q^{H2H} and W_K^{H2H} are learnable weight matrices. d is the embedding dimension.

Hyper-to-Local Attention There are multiple possible choices for designing the Hyper-to-Local attention. For example, a straightforward solution is to enable the attention of a Hyper-token to all Local-tokens. However, the computational cost grows linearly with the length of the total input (i.e., the number of Local-tokens), which is very expensive for large input sequences. Therefore, we adopt local attention by restricting the Hyper-to-Local attention of a Hyper-token only on the Local-tokens that belong to it. For example, in Figure 2, the 'Title' Hyper-token only directly attends to the text tokens within the product title. The information contained in Local-tokens from other modalities, e.g., a patch-token in 'Image', will be propagated to the 'Title' Hyper-token through 'Title'-to-'Image' and 'Image'-to-'patch' attentions. Denote the Local-token embeddings as X^L, the Hyper-to-Local restricted attention is defined as:

$$A_{ij}^{H2L} = softmax\left(\frac{X_i^H W_Q^{H2L}(X_j^L W_K^{H2L})^T}{\sqrt{d}}\right),\ for\ j \in M_i$$

where W_Q^{H2L} and W_K^{H2L} are weight matrices in Hyper-to-Local attention.

Local-to-Hyper Attention In Local-to-Hyper attention, each Local-token communicates with every Hyper-token, which enables the Local-token to receive the high-level representation from these summarization tokens of each modality. For example in Figure 2, each visual Local-token attends to all Hyper-tokens. The definition of the Local-to-Hyper attention is similar to the above Hyper-to-Local attention except that each Local-token attends to all Hyper-tokens.

Local-to-Local Attention The Local-to-Local attention is the traditional attention mechanism used in various existing Transformer models (Devlin et al., 2019; Dosovitskiy et al., 2021; Wang et al., 2022), which learns contextual token embeddings from the input sequence. In our design, to reduce the computational cost, we only allow Local-to-Local attention between two Local-tokens from the same modality. The connections between Local-tokens from two different modalities can be naturally bridged through the structured attention. Note that the efficiency can be further improved by adopting a relative attention pattern with relative position encoding (Shaw et al., 2018, 2019).

Final Attention The final token representation can be computed based on the above structured attention mechanism among Hyper and Local tokens. The output embeddings for Hyper and Local tokens Z^H, Z^L are calculated as follows:

$$Z^H = A^{H2H}(X^H W_V^H) + A^{H2L}(X^L W_V^L)$$

$$Z^L = A^{L2L}(X^L W_V^L) + A^{L2H}(X^H W_V^H)$$

where all the attention weights A are described above. W_V^H and W_V^L are the learnable matrices to compute the values for Hyper and Local tokens respectively. Intuitively, the Hyper-tokens are updated through the Hyper-to-Hyper full attention and Hyper-to-Local restricted attention. The Local-token embedding is learned via Local-to-Hyper full attention and Local-to-Local attention. These structured attention patterns effectively connect the tokens from different modalities, enabling the interactions across modalities efficiently.

3.5 Extraction Layer

The extraction layer of SMARTAVE outputs the final value for the attribute. We apply a Transformer decoder (Vaswani et al., 2017) on the output embeddings of the 'Attribute' token to generate attribute value. We also employ a copy mechanism (See et al., 2017) to allow both copying words from input text sequence, and generating words from a predefined vocabulary during decoding. To further improve the learned embedding, we supplement with an auxiliary task by extracting the text spans from the text modalities via sequential tagging (Xu et al., 2019) as shown in Figure 2. More technical details are provided in Appendix A.

3.6 Discussion

This section provides discussion that connects SMARTAVE with previous methods. If we remove the Hyper-to-Local and Local-to-Hyper attentions and re-organize the Local-to-Local (individual encoder for each modality) and Hyper-to-Hyper (fusion layer), our model architecture degenerates to the multimodal approaches (Zhu et al., 2020; Lin et al., 2021). If we further trim the input by only keeping and concatenating a few text sources (e.g., title and description), our model is similar to those methods that only use the product text features (Wang et al., 2020; Yan et al., 2021; Yang et al., 2022; Zhang et al., 2022). Moreover, if we continue replacing the Transformer with LSTM, our model is similar to the traditional sequential tagging approaches (Zheng et al., 2018; Xu et al., 2019).

4 Experiments

4.1 Datasets

We evaluate our method on two multimodal attribute value extraction datasets, MEPAVE (Zhu et al., 2020) and MAVE (Yang et al., 2022). **MEPAVE**[1] is a multimodal product attribute value extraction dataset with product title and image, collected from a mainstream Chinese e-commerce platform. It contains 87,194 text-image instances consisting of seven categories of products with 26 different attributes such as 'Material', 'Collar Type', 'Color', etc. The dataset is split into train-

[1] https://github.com/jd-aig/JAVE

ing, validation and testing sets with 71,194, 8,000 and 8,000 instances respectively.

MAVE[2] is a large, multi-sourced, diverse dataset for product attribute extraction study, which contains 3 million attribute value annotations across 1257 fine-grained categories created from 2.2 million cleaned Amazon product profiles (Ni et al., 2019). In our experiments, we select 8 root categories 'Amazon Fashion', 'All Beauty', 'Appliances', 'Books', 'Grocery and Gourmet Food', 'Home and Kitchen', 'Sports and Outdoors', 'Toys and Games', and randomly sample 4k products for each category, resulting in total 105K attribute-value instances. We further split them into training, validation and testing sets with 85k, 10k and 10k instances respectively. More details on the datasets are provided in Appendix B.

4.2 Baselines

Our model is compared with six state-of-the-art attribute value extraction baselines, including four text only methods and two multimodal methods.

SUOpenTag (Xu et al., 2019) uses two BiLSTMs to produce separate embeddings for the context and the attribute, followed by a CRF layer.

AVEQA (Wang et al., 2020) adopts the BERT encoder to jointly encode the attribute and the product text profile.

AdaTag (Yan et al., 2021) applies adaptive decoding that is able to extract multi-attribute values.

MAVEQA (Yang et al., 2022) extends the BERT encoder to deal with multiple text sources.

MJAVE (Zhu et al., 2020) utilizes two encoders for text and image separately and predicts product attributes and extract values.

PAM (Lin et al., 2021) incorporates text descriptions, OCR texts and visual modalities from the product and fuses the three modalities into a multimodal Transformer.

4.3 Implementation

Our model is implemented using PyTorch, and is trained on 64 NVIDIA Tesla V100 GPUs. During training, we use the gradient descent algorithm with Adam (Kingma and Ba, 2015) optimizer. During inference, we conduct beam search with beam width 5. The details of all hyper-parameters are reported in Appendix C.

Following previous works, we use precision, recall and F1 score as evaluation metrics denoted as P, R and F1. We follow Exact Match (Rajpurkar et al., 2016) criteria to compute the scores. We repeat each experiment 10 times and report the metrics based on the average over these runs.

Methods	MEPAVE	MAVE
SUOpenTag (Xu et al., 2019)	77.12 ± 0.62	81.74 ± 0.54
AVEQA (Wang et al., 2020)	89.15 ± 0.47	89.20 ± 0.32
AdaTag (Yan et al., 2021)	81.36 ± 0.54	86.19 ± 0.46
MAVEQA (Yang et al., 2022)	88.71 ± 0.38	90.06 ± 0.29
MJAVE (Zhu et al., 2020)	87.17 ± 0.43	88.84 ± 0.55
PAM (Lin et al., 2021)	89.68 ± 0.51	91.51 ± 0.37
SMARTAVE-text	89.21 ± 0.42	91.16 ± 0.41
SMARTAVE	**91.52 ± 0.45**	**93.69 ± 0.34**

Table 1: Overall performance comparison (F1 scores %) with standard deviation on both datasets. Results are statistically significant with p-value < 0.001.

5 Results

5.1 Overall Results

SMARTAVE outperforms the state-of-the-art attribute value extraction methods on both datasets. Table 1 presents the main comparison results on the two datasets. There are several key observations. **First**, the multimodal models SMARTAVE and PAM achieve better results over the text-only methods on both datasets, which demonstrates the usefulness of the information from product images for attribute value extraction. We also observe that MJAVE does not perform well compared to the sophisticated text-only methods AVEQA and MAVEQA. Our hypothesis is that MJAVE only uses a pre-trained BERT to encode the text without better fine-tuning. **Second**, our model significantly outperforms all other multimodal models. For example, the F1 score of SMARTAVE increases over 2.38% and 5.46% compared with PAM and MJAVE on the MAVE dataset. There are two main reasons: 1) Our model adopts a structured attention mechanism which jointly encodes the product information from all modalities, while existing multimodal methods use individual encoders for each modality and fail to capture their connections effectively. 2) Our model efficiently handles long text sequences with the structured modeling, while PAM and MJAVE simply concatenate the text from different modalities. **Third**, our text-only model, SMARTAVE-text, achieves the best performance among all text-only methods. We believe it is also attributed to the advantage of the structured modeling, which allows different text modalities to exchange information for learning better cross-modal

[2] `https://github.com/google-research-datasets/MAVE`

Methods	Brand			Shape			Color			Type		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
SUOpenTag (Xu et al., 2019)	91.25	91.33	91.29	84.36	78.86	81.52	80.84	72.53	76.46	74.08	68.25	71.04
AVEQA (Wang et al., 2020)	96.71	97.13	96.92	93.62	87.78	90.61	89.66	90.17	89.91	85.31	83.25	84.27
AdaTag (Yan et al., 2021)	95.26	94.21	94.73	91.68	90.60	91.14	88.23	85.09	86.63	84.37	82.88	83.62
MAVEQA (Yang et al., 2022)	96.48	97.76	97.11	93.70	88.35	90.95	89.81	91.15	90.47	**85.27**	84.12	84.69
MJAVE (Zhu et al., 2020)	95.52	94.72	95.12	91.94	89.48	90.69	90.02	90.75	90.38	84.17	82.69	83.42
PAM (Lin et al., 2021)	96.39	96.85	96.62	93.63	89.91	91.73	92.41	92.33	92.37	85.01	83.15	84.07
SMARTAVE	**96.87**	**97.84**	**97.35**	**94.55**	**91.38**	**92.94**	**93.39**	**92.80**	**93.09**	84.82	**84.70**	**84.76**

Table 2: Attribute level performance comparison on MAVE.

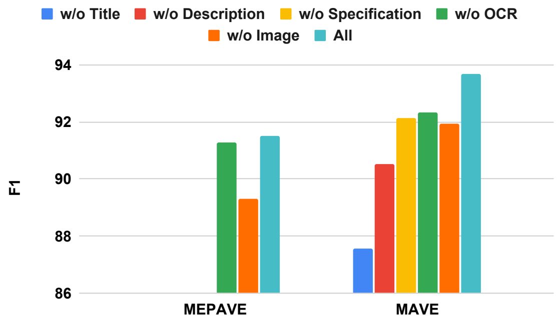

Figure 3: Importance of different modalities.

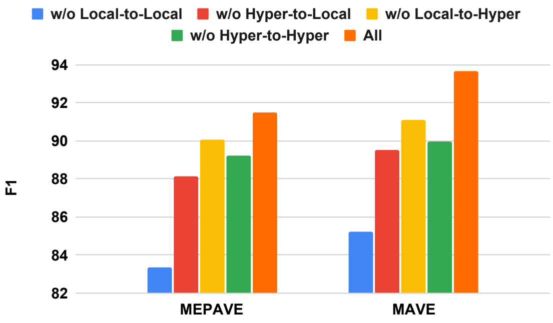

Figure 4: Importance of different attention patterns.

contextual embeddings.

5.2 Attribute Level Results

SMARTAVE generally outperforms the baselines on all attributes, with particularly large improvements over certain attributes. We conduct a fine-grained comparison of SMARTAVE with the baselines on the MAVE dataset using four selected attributes, 'Brand', 'Shape', 'Color' and 'Type'. These attributes are the most common attributes across multiple categories. The attribute level comparison results are reported in Table 2. From these results, we can see that SMARTAVE achieves the best performance, in terms of F1 scores, among all methods on all attributes. However, when comparing the improvements of SMARTAVE with the text-only methods, it is clear that the improvements on 'Brand' and 'Type' are marginal compared to the improvements on 'Shape' and 'Color'. Similar observation is found in (Zhu et al., 2020). The reason is that product image is more useful on a certain group of visual related attributes, such as 'Color' and 'Shape'.

6 Analysis and Discussion

6.1 Importance of Different Modalities

While the text product profile contains the most important information sources for attribute value extraction, OCR texts and visual information from the product image are also valuable sources that boost the extraction performance. To understand the impact of different modalities, we conduct an ablation study by removing each modality from our model. Note that for the MEPAVE dataset, it only contains product title, OCR text and image modalities. The results are illustrated in Figure 3. It can be seen that texts play a crucial role for attribute value extraction tasks. Among all text modalities, 'Title' is the most important source, which is consistent with our expectation. Moreover, it is also clear that the visual feature helps improve the extraction on both datasets. Another interesting observation is that the OCR text modality is less powerful on MEPAVE compared with the results on MAVE. The reason is that the number of products containing the OCR text is very small in the MEPAVE dataset.

6.2 Impact of Different Attention Patterns

Different attentions have different impacts to the model performance, while SMARTAVE with all attentions achieves the best result. In this ablation study, we evaluate the model performance by eliminating different attention patterns. Concretely, we train four additional models by removing one attention pattern per model. The results of these four models and SMARTAVE with all attentions on both datasets are shown in Figure 4. First, we observe that the performance drops significantly without the Local-to-Local attention. This

SMARTAVE	# Parameters	MEPAVE	MAVE
Encoder-2L	51M	89.13	91.45
Encoder-6L	95M	90.68	92.57
Encoder-12L-share	113M	90.44	92.62
Encoder-12L	161M	91.52	93.69
Encoder-24L	282M	**91.71**	**93.77**
Decoder-2L	143M	91.38	93.54
Decoder-4L	161M	91.52	93.69
Decoder-12L	241M	**91.61**	**93.73**
Training time	161M	53m	1h 14m

Table 3: Model performance (F1) over different encoder and decoder configurations.

is because the Local-to-Local attention is used to learn the contextual embeddings for local text and image tokens, which is the fundamental component of our model. We also observe clear performance drops, around 1 to 3 percent in terms of F1 score, if removing one of the other three attention pattern. This observation validates that these structured attentions are crucial for extracting attribute values from multiple modalities. Nevertheless, it is clear that SMARTAVE with all attentions achieves the best performance on both datasets.

6.3 Impact of Different Model Configurations

SMARTAVE with a 12-layer encoder and a 4-layer decoder obtains reasonable performance-scale trade-off. We conduct a series of performance-scale studies on SMARTAVE. The SMARTAVE base model uses a 12-layer encoder with a 4-layer decoder. We evaluate the model performance with a different number of encoder layers in {2L, 6L, 12L-share, 12L, 24L}. The 12L-share encoder means sharing the query and key matrices in different attention patterns. We further evaluate our model by only varying the number of decoder layers in {2L, 4L, 12L}. The F1 results of these models are shown in Table 3. It is not surprising to see that Encoder-24L and Decoder-12L achieve the best performances. However, a larger model usually requires both longer training and inference time, while the SMARTAVE model with a 12-layer encoder and a 4-layer decoder performs reasonably well. The training time of the base model is also reported in Table 3.

6.4 Varying Maximal Sequence Length

SMARTAVE efficiently handles large input sequences. We evaluate our model by varying the maximal input sequence length from {64, 128, 256, 512, 1024, 2048} on MAVE (as input sequences in MEPAVE are all very short) . Note that the baseline Transformer methods, such as (Wang et al., 2020; Lin et al., 2021), require a significant amount of time to train a model with input length beyond 512, as they simply feed the concatenated text into the standard Transformer. On the other hand, our model adopts the structured attention mechanism which dramatically reduces the computational cost. The model performance with respect to the maximal sequence length is shown in Figure 5. It is clear that the performance of SMARTAVE improves as maximal sequence length increases, and saturates around 1024.

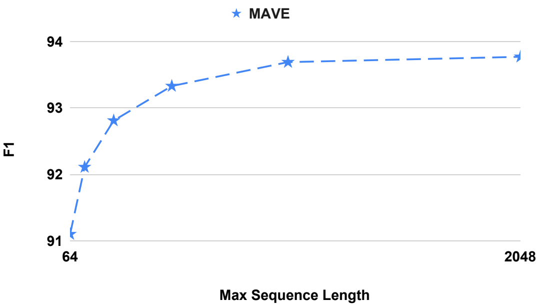

Figure 5: Model performances with different maximal sequence lengths.

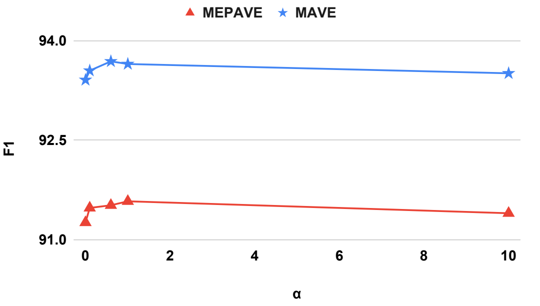

Figure 6: Impact of multi-task learning.

6.5 Impact of Multi-task Learning

Sequential tagging task generally improves the model performance. To understand the impact of the auxiliary sequential tagging task, we conduct a set of experiments by varying the weight parameter α (see Appendix A) from {0, 0.1, 0.6, 1, 10}. We illustrate the model performance with different values of α in Figure 6. Note that $\alpha = 0$ essentially stands for only applying the decoding task. It can be seen from the Figure that adding the sequential tagging task helps improve the attribute value extraction on both datasets. The model performance is relatively stable on a wide range of α.

7 Conclusions

This paper presents a novel Structured Multimodal Transformer model for Product Attribute Value Extraction. A structured attention mechanism is designed to encode the product information from multiple modalities. These structured attention patterns enable effective and efficient interactions among the text and visual tokens from different product modalities. The attribute values are then extracted based on the learned embeddings from the structured encoder. Evaluations are conducted on two multimodal datasets, which show the superior performance of the proposed approach over several state-of-the-art methods. Ablation studies also demonstrate the effectiveness of the structured attention patterns in modeling products with multimodal data and large input sequences.

Limitations

There are two limitations of our current approach. First, our model focuses on attribute value extraction for a single product. However, there are a few cases where an attribute contains multiple different values corresponding to different parts of a product. For example, for a 'Lamp' product, it has a 'white' lampshade with a 'blue' lamp holder. In this scenario, our model might be confused and extract an incorrect or partial value for the attribute 'Color'. Second, our model generates attribute-dependent product embeddings. In other words, we need to run the model inference once for each attribute to extract its values, even for a same product. This may increase the inference time for real-world applications, especially for a product with a large number of attributes. To alleviate this problem, we are working on jointly encoding multiple attributes associated with one product in the SMARTAVE model.

References

Joshua Ainslie, Santiago Ontañón, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. 2020. ETC: encoding long and structured inputs in transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 268–284. Association for Computational Linguistics.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 6077–6086. Computer Vision Foundation / IEEE Computer Society.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *CoRR*, abs/2004.05150.

Ke Chen, Lei Feng, Qingkuang Chen, Gang Chen, and Lidan Shou. 2019. EXACT: attributed entity extraction by annotating texts. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, pages 1349–1352. ACM.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc Viet Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2978–2988. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Xin Luna Dong, Xiang He, Andrey Kan, Xian Li, Yan Liang, Jun Ma, Yifan Ethan Xu, Chenwei Zhang, Tong Zhao, Gabriel Blanco Saldana, Saurabh Deshpande, Alexandre Michetti Manduca, Jay Ren, Surender Pal Singh, Fan Xiao, Haw-Shiuan Chang, Giannis Karamanolakis, Yuning Mao, Yaqing Wang, Christos Faloutsos, Andrew McCallum, and Jiawei Han. 2020. Autoknow: Self-driving knowledge collection for products of thousands of types. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 2724–2734. ACM.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision* *and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society.

Ronghang Hu, Amanpreet Singh, Trevor Darrell, and Marcus Rohrbach. 2020. Iterative answer prediction with pointer-augmented multimodal transformers for textvqa. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9989–9999. Computer Vision Foundation / IEEE.

Guanhuan Huang, Xiaojun Quan, and Qifan Wang. 2022. Autoregressive entity generation for end-to-end task-oriented dialog. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 323–332, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991.

Robert L. Logan IV, Samuel Humeau, and Sameer Singh. 2017. Multimodal attribute extraction. In *6th Workshop on Automated Knowledge Base Construction, AKBC@NIPS 2017, Long Beach, California, USA, December 8, 2017*. OpenReview.net.

Giannis Karamanolakis, Jun Ma, and Xin Luna Dong. 2020. Txtract: Taxonomy-aware knowledge extraction for thousands of product categories. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8489–8502. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Rongmei Lin, Xiang He, Jie Feng, Nasser Zalmout, Yan Liang, Li Xiong, and Xin Luna Dong. 2021. PAM: understanding product images in cross product category attribute extraction. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, pages 3262–3270. ACM.

Hanqing Lu, Youna Hu, Tong Zhao, Tony Wu, Yiwei Song, and Bing Yin. 2021. Graph-based multilingual product retrieval in e-commerce search. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Papers, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 146–153. Association for Computational Linguistics.

Ajinkya More. 2016. Attribute extraction from product titles in ecommerce. *CoRR*, abs/1608.04670.

Thanh V. Nguyen, Nikhil Rao, and Karthik Subbian. 2020. Learning robust models for e-commerce product search. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6861–6869. Association for Computational Linguistics.

Jianmo Ni, Jiacheng Li, and Julian J. McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 188–197. Association for Computational Linguistics.

Duangmanee Putthividhya and Junling Hu. 2011a. Bootstrapped named entity recognition for product attribute extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1557–1567. ACL.

Duangmanee Putthividhya and Junling Hu. 2011b. Bootstrapped named entity recognition for product attribute extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1557–1567. ACL.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. 2020. Compressive transformers for long-range sequence modelling. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics.

Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2017. Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1137–1149.

Ohad Rozen, David Carmel, Avihai Mejer, Vitaly Mirkis, and Yftah Ziser. 2021. Answering product-questions by utilizing questions from other contextually similar products. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 242–253. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1073–1083. Association for Computational Linguistics.

Peter Shaw, Philip Massey, Angelica Chen, Francesco Piccinno, and Yasemin Altun. 2019. Generating logical forms from graph representations of text and entities. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 95–106. Association for Computational Linguistics.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 464–468. Association for Computational Linguistics.

Keiji Shinzato, Naoki Yoshinaga, Yandi Xia, and Wei-Te Chen. 2022. Simple and effective knowledge-driven query expansion for qa-based product attribute extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 227–234. Association for Computational Linguistics.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards VQA models that can read. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 8317–8326. Computer Vision Foundation / IEEE.

Hao Tan and Mohit Bansal. 2019. LXMERT: learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 5099–5110. Association for Computational Linguistics.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2022. Efficient transformers: A survey. *ACM Comput. Surv.*

Quoc-Tuan Truong, Tong Zhao, Changhe Yuan, Jin Li, Jim Chan, Soo-Min Pantel, and Hady W. Lauw. 2022. Ampsum: Adaptive multiple-product summarization towards improving recommendation captions. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pages 2978–2988. ACM.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Qifan Wang, Yi Fang, Anirudh Ravula, Fuli Feng, Xiaojun Quan, and Dongfang Liu. 2022. Webformer: The web-page transformer for structure information extraction. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pages 3124–3133. ACM.

Qifan Wang, Li Yang, Bhargav Kanagal, Sumit Sanghai, D. Sivakumar, Bin Shu, Zac Yu, and Jon Elsas. 2020. Learning to extract attribute value from product via question answering: A multi-task approach. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 47–55. ACM.

Huimin Xu, Wenting Wang, Xin Mao, Xinyu Jiang, and Man Lan. 2019. Scaling up open tagging from tens to thousands: Comprehension empowered attribute value extraction from product title. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5214–5223. Association for Computational Linguistics.

Jun Yan, Nasser Zalmout, Yan Liang, Christan Grant, Xiang Ren, and Xin Luna Dong. 2021. Adatag: Multi-attribute value extraction from product profiles with adaptive decoding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4694–4705. Association for Computational Linguistics.

Li Yang, Qifan Wang, Zac Yu, Anand Kulkarni, Sumit Sanghai, Bin Shu, Jon Elsas, and Bhargav Kanagal. 2022. MAVE: A product dataset for multi-source attribute value extraction. In *WSDM '22: The Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event / Tempe, AZ, USA, February 21 - 25, 2022*, pages 1256–1265. ACM.

Sanshi Yu, Zhuoxuan Jiang, Dongdong Chen, Shanshan Feng, Dongsheng Li, Qi Liu, and Jinfeng Yi. 2021. Leveraging tripartite interaction information from live stream e-commerce for improving product recommendation. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, pages 3886–3894. ACM.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontañón, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. In *Advances in Neural*

Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.

Wenxuan Zhang, Yang Deng, Jing Ma, and Wai Lam. 2020. Answerfact: Fact checking in product question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 2407–2417. Association for Computational Linguistics.

Xingxing Zhang, Furu Wei, and Ming Zhou. 2019. HIBERT: document level pre-training of hierarchical bidirectional transformers for document summarization. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5059–5069. Association for Computational Linguistics.

Xinyang Zhang, Chenwei Zhang, Xian Li, Xin Luna Dong, Jingbo Shang, Christos Faloutsos, and Jiawei Han. 2022. Oa-mine: Open-world attribute mining for e-commerce products with weak supervision. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pages 3153–3161. ACM.

Jie Zhao, Ziyu Guan, and Huan Sun. 2019. Riker: Mining rich keyword representations for interpretable product question answering. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 1389–1398. ACM.

Guineng Zheng, Subhabrata Mukherjee, Xin Luna Dong, and Feifei Li. 2018. Opentag: Open attribute value extraction from product profiles. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pages 1049–1058. ACM.

Tiangang Zhu, Yue Wang, Haoran Li, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2020. Multimodal joint attribute prediction and value extraction for e-commerce product. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 2129–2139. Association for Computational Linguistics.

Will Y. Zou, Richard Socher, Daniel M. Cer, and Christopher D. Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1393–1398. ACL.

A More Technical Details

We provide more technical details on our SMARTAVE in this section.

Input Layer In the input layer, each Hyper-token and Local-token will be mapped to an embedding vector. The embeddings for the Hyper-tokens are randomly initialized $E^H = \{E_1^H, \ldots, E_n^H\}$. For the text Local-tokens, they are initialized from a pre-trained model $E^{i,L} = \{E_1^{i,L}, \ldots, E_{m_i}^{i,L}\}$. For the visual Local-tokens, each patch w_j^i is converted/mapped into a d-dimensional embedding vector with a projection matrix W_p, i.e., $E_j^{i,L} = W_p w_j^i$. We use $E^L = \{E^{1,L}, \ldots, E^{n,L}\}$ to represent the embeddings of all Local-tokens.

SMARTAVE Encoder The SMARTAVE encoder is a stack of K identical layers:

$$X^k = \text{SMARTAVE}(X^{k-1}), \;\; 1 \leq k \leq K$$

where $X^0 = \{E^H, E^L\}$ is the input embedding for the first layer. Each SMARTAVE encoder layer contains a structured attention layer followed by a standard feed forward network:

$$Z^k = \text{Attention}(X^{k-1}), \;\; X^k = \text{FFN}(Z^k)$$

The Attention layer uses the structured attention mechanism described in the main paper. We now present the full format of all attentions.

Hyper-to-Hyper Attention The Hyper-to-Hyper full attention is formulated as:

$$\alpha_{ij}^{H2H} = \frac{\exp(e_{ij}^{H2H})}{\sum_{\ell} \exp(e_{i\ell}^{H2H})}, \; for \; 1 \leq j \leq n$$

$$e_{ij}^{H2H} = \frac{x_i^H W_Q^{H2H} (x_j^H W_K^{H2H})^T}{\sqrt{d}}$$

which is equivalent to the compact matrix format in the main paper.

Hyper-to-Local Attention The Hyper-to-Local restricted attention is formulated as:

$$\alpha_{ij}^{H2L} = \frac{\exp(e_{ij}^{H2L})}{\sum_{\ell \in M_i} \exp(e_{i\ell}^{H2L})}, \; for \; j \in M_i$$

$$e_{ij}^{H2L} = \frac{x_i^H W_Q^{H2L} (x_j^L W_K^{H2L})^T}{\sqrt{d}}$$

Local-to-Hyper Attention The Local-to-Hyper attention is formulated as:

$$\alpha_{ij}^{L2H} = \frac{\exp(e_{ij}^{L2H})}{\sum_{\ell} \exp(e_{i\ell}^{L2H})}, \; for \; 1 \leq j \leq n$$

$$e_{ij}^{L2H} = \frac{x_i^L W_Q^{L2H} (x_j^H W_K^{L2H})^T}{\sqrt{d}}$$

Local-to-Local Attention The Local-to-Local constrained attention is formulated as:

$$\alpha_{ij}^{L2L} = \frac{\exp(e_{ij}^{L2L})}{\sum_{\ell \in M_i} \exp(e_{i\ell}^{L2L})}, \; for \; j \in M_i$$

$$e_{ij}^{L2L} = \frac{x_i^L W_Q^{L2L} (x_j^L W_K^{L2L})^T}{\sqrt{d}}$$

Final Attention The final computation of Z^H and Z^L can be written as:

$$z_i^H = \sum_{1 \leq j \leq n} \alpha_{ij}^{H2H} x_j^H W_V^H + \sum_{\ell \in M_i} \alpha_{i\ell}^{H2L} x_k^L W_V^L$$

$$z_i^L = \sum_{j \in M_i} \alpha_{ij}^{L2L} x_j^L W_V^L + \sum_{1 \leq \ell \leq n} \alpha_{ij}^{L2H} x_k^H W_V^H$$

Extraction Layer The attribute value decoder is a standard Transformer decoder, which consumes the embedding of 'Attribute' Hyper-token from the SMARTAVE encoder and generates the value word by word:

$$\bar{w_t} = \arg\max_{w_t}(softmax(W_v X_{de}^t))$$

where X_{de}^t is the decoder output at word position t. W_v is the output matrix which projects the final embedding to the logits of vocabulary size. As mentioned in the main paper, we further employ a copy mechanism to allow copying words from input text sequence via pointing. For the sequential tagging component, we pass the embeddings of each text sequence into the CRF layer (Xu et al., 2019; Yan et al., 2021) to predict tags for each token in $\{B, I, O, E\}$. The total loss is defined as:

$$\mathcal{L} = \mathcal{L}_D + \alpha \mathcal{L}_{Seq}$$

where $\mathcal{L}_D$ is the decoder loss and $\mathcal{L}_{Seq}$ is the sequential tagging loss. α is a hyper-parameter.

Category	# Product	# Instance	# Attribute	# Value
Clothes	12,240	34,154	14	1,210
Shoes	9,022	20,525	10	1,036
Bags	3,376	8,307	8	631
Luggage	1,291	2,227	7	275
Dresses	4,567	12,283	13	714
Boots	713	2,090	11	322
Pants	2,832	7,608	13	595
Total	34,041	87,194	26	2,129

Table 4: The statistics of the MEPAVE dataset.

Category	# Product	# Instance	# Attribute	# Value
Amazon Fashion	4k	16.6k	18	1.1k
All Beauty	4k	17.3k	23	1.3k
Appliances	4k	12.5k	16	0.8k
Books	4k	12.7k	9	0.8k
Grocery and Gourmet Food	4k	9.8k	12	0.7k
Home and Kitchen	4k	11.2k	15	0.8k
Sports and Outdoors	4k	12.7k	16	0.9k
Toys and Games	4k	12.2k	15	1.0k
Total	32k	105k	65	3.6k

Table 5: The statistics of the MAVE dataset.

B Dataset

Data Processing For the MEPAVE dataset, it already has the product image associated with each product. For the MAVE dataset, they do not directly provide the product images. We join the product 'id' from MAVE with the product 'asin' from the original Amazon Review Data[3] to obtain the high-resolution product images. We further remove the text sources that directly represent the attribute and value. For example, {'source': 'brand', 'text': 'Kalso'}. We believe this is one of the main reasons why AVEQA and MAVEQA methods achieve very high scores on the original MAVE dataset. For both datasets, in order to compute the sequential tagging loss, we match the attribute values with the OCR texts to generate the tags on the OCR text sequences (for other text modalities, both datasets provide full tag annotations).

Statistics The statistics of both datasets are shown in Table 4 and 5.

C Implementation Details

The language of the texts on the datasets are different. MEPAVE contains Chinese product profiles, while MAVE has English product profiles. Moreover, the text characteristics are also very different for these two datasets. MEPAVE only includes the product title, with very limited OCR texts. The maximal number of Chinese words/tokens in the dataset is 56. On the other hand, MAVE consists of multiple text modalities with much larger text length (can even go over 1024). Therefore, the vocabularies used for different datasets are different, as well as certain other parameters. The model parameters used for both datasets are provided in Table 6.

Parameters	MEPAVE	MAVE
encoder layers	12	12
encoder heads	12	12
encoder hiden size	768	768
encoder hidden units (FFN)	3,072	3,072
max input sequence length	64	1024
decoder layer	4	4
decoder heads	6	6
decoder hiden size	768	768
decoder hidden units (FFN)	3,072	3,072
max output sequence length	10	10
beam width	5	5
batch size	128	32
training epochs	28	20
optimizer	Adam	Adam
learning rate schedule	linear decay	linear decay
learning rate	$2e^{-5}$	$2e^{-5}$
learning rate warmup steps	2,000	2,000
vocab	Chinese vocab from MJAVE	BERT-base
vocab size	2,772	30,522
α	0.6	0.6

Table 6: Model Hyper-parameters details.

batch size	MEPAVE	MAVE	learning rate	MEPAVE	MAVE
16	91.46	93.62	$1e^{-5}$	91.62	92.95
32	91.42	93.69	$2e^{-5}$	91.52	93.69
64	91.48	93.21	$4e^{-5}$	91.17	93.24
128	91.52	-	$8e^{-5}$	91.03	92.86
512	90.85	-	$2e^{-4}$	90.80	92.51

Table 7: F1 results with different batch sizes and learning rates on both datasests.

D Impact of Batch Size and Learning Rate

To evaluate the model performance with different training batch sizes and learning rates, we conduct experiments to train a set of SMARTAVE models with a hyper-parameter sweep consisting of learning rates in $\{1e^{-5}, 2e^{-5}, 4e^{-5}, 8e^{-5}, 2e^{-4}\}$ and batch sizes in 16, 32, 64, 128, 512 on both datasets. The F1 results with different learning rates and batch sizes are reported in Table 7. Note that for MAVE, we are not able to train on large batch size, i.e., 128 and 512, as the maximal input sequence length is set to 1024. It can be seen from the table that smaller batch size and learning rate usually lead to better model performance.

[3] https://nijianmo.github.io/amazon/index.html