MMT4: Multi Modality To Text Transfer Transformer

Amir Tavanaei
atavanae@amazon.com
Amazon
Seattle, WA, USA

Karim Bouyarmane
bouykari@amazon.com
Amazon
Seattle, WA, USA

Iman Keivanloo
imankei@amazon.com
Amazon
Seattle, WA, USA

Ismail Tutar
ismailt@amazon.com
Amazon
Seattle, WA, USA

ABSTRACT

Recent studies have demonstrated the ability of auto-regressive and seq-to-seq generative models to reach state-of-the-art performance on various Natural Language Understanding (NLU) and Natural Language Processing (NLP) tasks. They operate by framing all the tasks in a single formulation: text auto-completion or text-to-text encoding-decoding. These models can be trained on the products corpus in order to understand the information in the e-commerce products listings. In this paper, we present a new generative model to involve different modalities (e.g. text and vision). The proposed model is an encoder-decoder model with the T5 (**T**ext **T**o **T**ext **T**ransfer **T**ransformer) foundation in which the non-text components are fused to the text tokens. Specific relative positional and token type embeddings are used in the encoder part, while the decoder generates new text corresponding to diverse tasks. Hence, we name the proposed model MMT4: **M**ulti **M**odality **T**o **T**ext **T**ransfer **T**ransformer. The experiments are done over our proprietary e-commerce catalog involving image and text, with the rationale that the image of a product provides more information about the product. One of the main advantages of this model is to generate product attributes (product specifications) that can be either solely inferred from the text or the image, or both. In the experiments, we pre-train and fine-tune MMT4 to solve a number of downstream tasks: attribute generation, image-text matching (ITM), and title (product name) generation from product's image (captioning). The experimental results show up to 35% accuracy improvement in comparison with the fine-tuned T5 in the attribute generation task. Product title generation also shows more than 3% higher Rouge-1 recall than the fine-tuned state-of-the-art captioning model. Although we fine-tuned our model on less than 2M samples in a generative mode, its performance is only 2% area under the precision-recall curve lower than the state-of-the-art ITM model.

SIG-KDD, Workshop on Content Understanding and Generation for E-commerce, August 14–18, 2022, Washington, DC

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

CCS CONCEPTS

• **Applied computing** → **Electronic commerce**; • **Information systems** → **Information systems applications**.

KEYWORDS

Multimodal transformers, encoder-decoder, natural language generation

ACM Reference Format:
Amir Tavanaei, Karim Bouyarmane, Iman Keivanloo, and Ismail Tutar. 2022. MMT4: Multi Modality To Text Transfer Transformer. In *Proceedings of SIG-KDD, Workshop on Content Understanding and Generation for E-commerce.* ACM, New York, NY, USA, 7 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Following the success of transformers in encoding the natural language data and text representation [5, 13, 19, 32], recent research focused on using the pre-trained text encoders in downstream tasks such as sentence encoding [26], classification, semantic analytics [23, 26], question answering [28], and named entity recognition [30]. For each task, separate fine-tuning and model customization are required to provide task-specific models. Additionally, in classification tasks, the number of classes is pre-defined and the model should be re-trained after adding a new label to the task. To address these concerns, sequence-to-sequence transformers [15, 18, 25, 33] provide a unique framework to support multi-task learning and additional class labels by generating text using their decoder component. T5 (Text To Text Transfer Transformer) [25] is one of the popular generative models that offers a unified architecture across multiple tasks and has shown great performances in different applications [2, 7, 10, 21].

Encoder-based transformers have been customized to encode other modalities and use the self-attention mechanism [32] in multimodal frameworks. There are a series of multimodal transformers in the literature which perform early/late fusion in vision and language modeling such as ViLBERT [20], ViLT [12], MMBT [11], and ALBEF [16] or two-tower vision and language models such as CLIP [24], and ALIGN [9]. Data2Vec [1] is the most recent transformer that encodes text, image, and speech. It is pre-trained using mask prediction and latent target representation using teacher signal to learn individual representations of the modalities and it does not perform multimodal training. Although these multimodal networks performed well in learning from joint or individual modalities, they inherit the same limitations in multi-task learning and

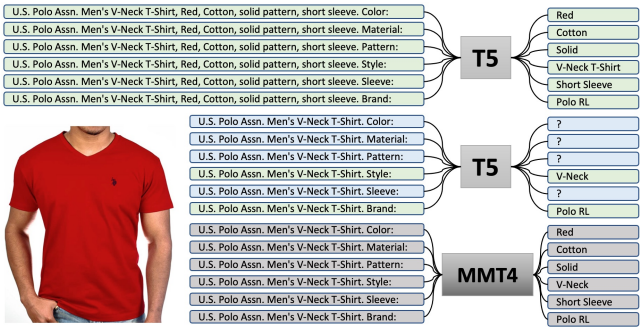

Figure 1: T5 versus MMT4 facing complete and incomplete product titles to generate product attributes. The MMT4 uses image information to compensate information lack in the text. '?' means that the model fails to predict correct response.

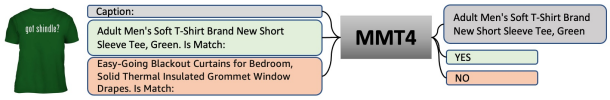

Figure 2: Title generation and image-title/text matching decision using MMT4.

unseen label recognition. Thus, in this work, we propose a new generative model to encode different modalities and generate desired output texts. The proposed model architecture is a customized T5 in which the non-text (e.g. image) components are fused to the text tokens with specific relative positional and token type embeddings in the encoder section while the decoder generates new texts corresponding to diverse tasks specified by keywords (prompts). We name our model Multi Modality To Text Transfer Transformer (MMT4). Fig. 1 shows an example of the MMT4's application and its superiority over T5 where the image input provides information to compensate the lack of information in the text. For instance, if the text input does not include the shirt's color or pattern, the image input can help to generate (or correct) these important attributes in the product's details. MMT4 can also be used in other downstream tasks such as title generation from other sources of data, title completion, cross-modality matching, and so on. Fig. 2 shows two examples of title generation from image and image-title matching (as a specific case of cross-modality matching).

This work is inspired from the multimodal generative model introduced in [3] which unifies tasks involving both text and image using generative models. [3] uses 36 Faster-RCNN [27] image regions as regions of interest to calculate the vision embeddings and pre-trains the model using self-supervised learning approaches and downstream tasks given specific prompts. Using an object detection algorithm for visual embedding preparation is computationally expensive and this network limits the inputs to only text and image. The proposed approach in MMT4, on the other hand, does not need to know the regions of interest as input and it learns to find them through self and cross attention layers which results in latency reduction in inference.

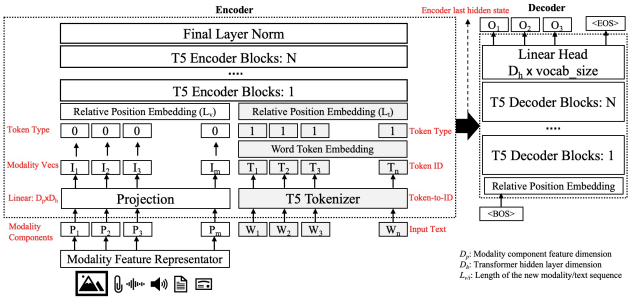

Figure 3: MMT4 architecture. D_p is the non-text feature dimension, D_h is the transformer hidden dimension, and L is the sequence length. Text and non-text inputs are fused in the encoder using early fusion method and the encoder's last hidden state is used in the decoder for text generation.

In summary, our contributions in this work are to build a general sequence-to-sequence generative model architecture that can support different kinds of modalities, provide an efficient multitask, multimodal framework to improve the text-only generative models, and address a number of problems in e-commerce using the proposed model (MMT4).

2 MMT4 MODEL ARCHITECTURE

MMT4 is a customized and augmented version of T5 [25] where the input is not limited to text and can be chosen from different modalities. Fig 3 shows the model architecture where the input is the combination of an input text and another source of input (e.g. corresponding image to the text). The "Modality Feature Representator" pre-processes the non-text input and extracts a sequence of feature vectors representing the new modality. For instance, the image input can be divided into image patches flattened to D_p-dimensional feature vectors; or a voice signal can be represented by a number of Mel Frequency Cepstral Coefficient (MFCC) [22] feature vectors. This component can also be a neural network architecture processing the raw input. For example, a vision transformer or convolutional neural network (CNN) can extract sequential feature vectors from images. The "Projection" component adjusts the modality feature vector dimension to the T5 hidden layer dimension (e.g. 768 for T5-base) using a linear layer followed by a normalization layer (LayerNorm). At this point, the new modality's input features are ready to be passed to the embedding and transformer layers next to the embedded text tokens. Eq. 1 shows the modality component (P_i) obtained by the modality (V) feature representator (f_v) that is projected by a linear NN (W_{proj}) followed by normalization, γ.

$$P_i = f_v(V)_i \,, \quad I_i = \gamma(W_{proj}.P_i) \tag{1}$$

The token type (segment) embedding (SE) in this architecture separates different modalities and helps the model distinguish information flow from different input segments. Each modality in this architecture gets a unique token type id (in this figure 0 and 1) as shown in Eq 2. The embedded inputs for the text (X^t) and the other modality (X^v) are concatenated to prepare a single sequence

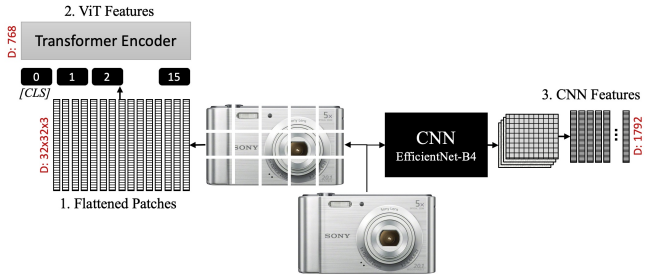

Figure 4: Three approaches for image feature representation before fusion. 1: Simple image patching and flattening. 2: ViT over image patches. 3: CNN feature map vectors.

fed to the T5 encoder.

$$X_i^v = I_i + \text{SE}(0) \tag{2}$$

$$X_i^t = \text{WE}(T_i) + \text{SE}(1), \quad \text{WE : Word Embedding}$$

Following the T5 architecture, we also employ the relative positional embedding (RPE) [25, 29] computed according to the input length for each input segment/modality (as shown in Eq. 3) and is incorporated in the self-attention computation. The positional embedding utilizes the relative pairwise distance between tokens so that it can handle long sequences and better generalize to sequences with different lengths than the lengths seen in the training data [29]. The RPE of the input vectors (a_{ij}) is shared with the other encoder blocks.

$$a_{ij}^v = \text{RPE}(head, L_v, L_v) \ , \quad a_{ij}^t = \text{RPE}(head, L_t, L_t) \tag{3}$$

$$a_{ij} = \text{concat}(a_{ij}^v, a_{ij}^t)$$

The embedded features are concatenated and fed to the encoder layers including multi-head self-attention layers. The last hidden state of the encoder is a sequence of $(L_v{+}L_t)$ D_h-dimensional feature vectors passed to the decoder as key-value for the cross-attension calculation in the decoder layers. The "Linear Head" of the decoder section maps the decoder's last hidden state to token IDs based on the casual language modeling masking. In this study, we only focus on the text generation, thus, the "Linear Head" is used in all the experiments. In the case that the other modality generation is expected, the "Linear Head" may or may not be used according to the modality type and the task.

3 EXPERIMENTAL METHOD: IMAGE+TEXT

In this paper, we focus on image as the additional modality so that an encoder input involves a product's title and its corresponding image. In this study we experiment three different image feature representation approaches as explained in the following paragraphs.

Image Patching: The image patching component divides the image to squared patches with the shape of $[patch \times patch \times channel]$ where $channel = 3$ (RGB channels). The flattened patches represent the image component features, P_i, as demonstrated in Fig. 4.1.

Vision Transformer (ViT): The Vision Transformer (ViT) model [6] trains a transformer encoder on ImageNet [4] by dividing the image to squared patches (e.g. 16x16 or 32x32) as input feature

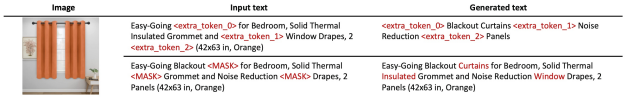

Figure 5: Pre-training sample row examples. The first row shows a span prediction example and the second row shows an MLM example.

vectors. ViT has outperformed the CNN models in image classification tasks and has been widely used for visual and language transformers [12, 16, 24]. In this study, we use the ViT pre-trained on 224x224 images that are divided into 32x32 patches. This model represents an image by $m+1$ feature vectors where m is the number of patches and 1 is for the [CLS] token in this architecture as shown in Fig. 4.2.

Convolutional Neural Networks: CNN as a well known NN architecture for image processing [8, 14] has shown great success in cooperating with the multi-modal transformers for encoding visual and natural language data [11, 16]. In this study, we use EfficientNet-B4 [31] as a high performance and efficient CNN model to extract visual features of images before fusing them to the text token features in the encoder. The visual features are acquired from the last convolutional layer of the EfficientNet-B4 in which each point across 1792 feature maps represent one image component. Fig. 4.3 depicts the image components extracted by the EfficientNet-B4. Given an image with the size of [380x380], the last convolutional layer's output consists of 1792 feature maps with the size of [12x12]. Thus, this image feature representator extracts 144 features vectors with the dimension of 1792.

4 EXPERIMENTS AND RESULTS

The modalities in the experiments are product's title and image where the image modality is represented by the three approaches mentioned above. The training dataset includes 1.92M titles and their corresponding images and the validation dataset includes 7720 titles and corresponding images from different products than the training dataset. Pre-training and fine-tuning tasks are performed using the same training hyper-parameters with learning rate=2e-4 decayed linearly, training epochs=2, and batch size=288. This section describes the experiments and results of the pre-training and fine-tuning tasks.

4.1 Pre-training

Pre-training of MMT4 involves self-supervised span prediction and masked language modeling (MLM). The span prediction replaces several, random token spans by special tokens and the MLM replaces random tokens by a mask token. T5 tokenizer provides 100 extra special tokens names <extra-token-0> to <extra-token-99> where 99 of those are used for the span prediction task and <extra-token-99> is used as the [MASK] token in MLM. Fig. 5 shows an example of the pre-training data and expected generated texts. In the span prediction task, the noise density is 0.2 and the average span length is 2 tokens. The MLM mask density is 0.3. In both tasks, the cross entropy loss (ce) of the generated text and the target text is calculated and equally weighted for pre-training. Eq. 4 shows the

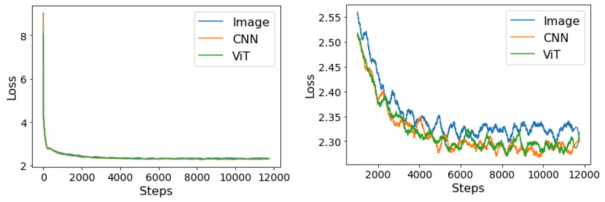

Figure 6: Loss value trend during training. Left: all the steps. Right: steps 1000-12000.

loss function.

$$\text{Loss} = \text{ce}(p_{sp}, y_{sp}) + \text{ce}(p_{mlm}, y_{mlm}) \,, \tag{4}$$

$$\text{ce} = -\Sigma_{t=1}^{L}\Sigma_{c=1}^{M} y_{t,c} \log(p_{t,c})$$

where, p denotes the softmax of the language model head's output, L is the length of output text, M is the vocabulary size, and y shows the target tokens. The smoothed training loss values for three MMT4 architectures including different image representation components are shown in Fig. 6. The CNN and ViT visual feature extraction methods have shown lower loss values than the raw image patches during training.

4.2 Fine-tuning: Attribute Generation

The aim of MMT4 is mainly to improve the performance of downstream tasks and to provide a new network architecture to address questions that cannot be solved by text-only models. One of the main concerns in attribute generation/extraction from the product title (and description) is the lack of relevant information pinpointing the attributes. For instance, generating the "sleeve-type" attributes from a shirt's title (and other descriptions) that does not reveal any information about the sleeve type of the shirt is almost impossible. However, the other modalities (e.g. image of the product) can solve this problem. Additionally, even if the textual data include required target tokens, additional modality can improve the model's performance and correct the attribute values. To assess the impact of the image in information generation, we pre-process the dataset according to the task and remove the tokens of interest from the input data.

The attributes selected for the attribute generation task are color, brand, style, material, sleeve-type, number-of-items, and pattern. These attributes are the examples of attributes that may be discovered from the image if the attribute is not in the text input (title). The number of samples for each task may be different from other tasks because not all attributes are valid for all the products (for instance, shoes do not have sleeve-type).

The MMT4 is fine-tuned to generate the attributes mentioned above given the modified titles followed by task prompts (e.g. color:). To evaluate the fine-tuned MMT4, the validation dataset including 7700 samples is used. As shown in Table 1, in average, using ViT as image representation slightly outperforms the other methods. The accuracy of the number-of-items generation is greater than 99% because most of the values are 1. If we only take the number-of-items>1 into account, our model is 31% (12 out of 39) matched with the labels.

Table 1: Attribute generation performance of MMT4 based on incomplete titles.

Model	Attributes (Evaluation Accuracy %)						
	Color	Material	Pattern	Style	Brand	Sleeve	Items
MMT4 (Image Patch)	78.00	71.54	63.37	47.78	30.32	81.94	94.89
MMT4 (EffNet)	74.75	72.61	63.38	51.45	29.59	85.17	95.53
MMT4 (ViT)	79.92	71.20	69.43	50.87	30.15	86.34	95.53
Number of samples	3018	2337	314	519	6518	681	626

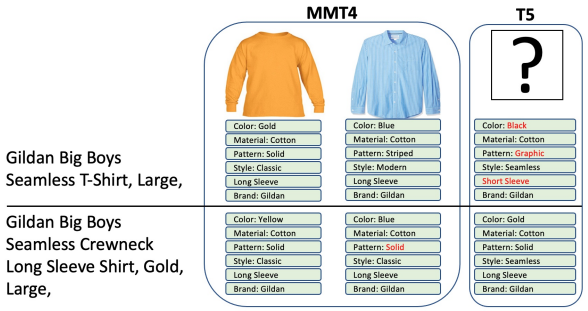

Figure 7: MMT4's impact in generating relevant attributes by relying on both image and title. This figure shows examples for color, pattern, and sleeve-type attribute generation.

To compare MMT4 with T5 in this task, we fine-tuned the pre-trained T5 (t5-base) using the same training dataset and hyper-parameters as MMT4. Table 2 shows the T5 and MMT4 performances on the validation dataset in three folds: 1) all the titles are pre-processed as explained in Section 4.2, 2) only 20% of titles are pre-processed, and 3) the original titles are used. As expected, MMT4 outperforms T5 since it receives more information (image and text) from input, especially in detecting/generating attributes that can be easily obtained from the image. Even if the original titles are used (fold 3 in Table 2), MMT4 outperforms T5 in all the attribute generation tasks.

Figs. 7 and 8 visualize the impact of MMT4 in correcting and completing the product's attributes using both title and image. This impact is better demonstrated in the titles that miss many attributes. Table 3 shows a detailed comparison between fine-tuned T5 and MMT4. In this analysis, for each attribute, we reported the true positive rates for frequent values belonging to each attribute. For example, out of 73 different "patterns", the test dataset has 31 products with "Striped Pattern" and out of 31 striped pattern products, MMT4's correctness ratio was 20/31 and T5's correctness ratio was 1/31. In both comparisons, MMT4 outperforms T5 due to its multimodal framework. To further compare our model with the state-of-the-art generative models, we tested pre-trained multimodal VL-T5 [3] for image-text matching and color generation tasks. However, the performance of the zero-shot VL-T5 was very poor in these tasks on our validation dataset (<60% PR-AUC) so that we did not take that comparison into account in this manuscript.

4.3 Fine-tuning: Title Generation from Image

To evaluate MMT4 for title generation from image, we used 6819 product images from the validation dataset. The generated titles are

Table 2: Fine-tuned T5 versus fine-tuned MMT4 for attribute generation. First, all the titles are pre-processed (i.e. the attribute values in the titles are removed). Second, 20% of titles are pre-processed. Third, Original titles are used.

Titles Pre-processed	Model	Attributes (Evaluation Accuracy)%					
		Color	Sleeve_type	Pattern	Material	Style	Brand
All of the titles	T5	39.5	78.9	55.7	66.3	43.2	21.2
	MMT4	**74.8**	**85.2**	**63.4**	**72.6**	**51.4**	**29.6**
20% of the titles	T5	75.9	85.8	69.1	74.2	54.1	58.8
	MMT4	**87.3**	**88**	**72.6**	**77.2**	**58.4**	**61**
No pre-processing	T5	84.2	87.7	72	76.2	57	68.3
	MMT4	**89.8**	**88.1**	**73.9**	**78**	**60**	**68.7**

Attribute	Image	Title	Target Attribute Value	T5 Generated Attribute	MMT4 Generated Attribute
Material		Rarido Ring Waldorf Ribbon Hand Kite Toy Swirl Stremers FLY ME Birthyday Party Favors - (Color: Random Mixed Color),	Wooden	Other	Wood
Style		Western 8" Ceramic Decorative Plate, Rusty Stars on Wooden Background Aged Antique Vintage Country Design Dinner Plate Accessory for Dining Table Tabletop Home, Dark Orange Warm Taupe,	Art Deco	Casual	Art Deco
Brand		Smartphone Protective Case Slim PC Hard Cover Case for Samsung Galaxy S6 Active G890A CECELL Phone case, Leather Blue Vibrant Diamond Pattern	CECELL Phone case	Topgs	CECELL phone case
Brand		iPhone 5s Cases & Covers - Crane TPU Soft Case Cover Protector for iPhone 5s - White,	custom phone cases	i5s & Covers	custom phone cases

Figure 8: Material, style, and brand attribute generation examples in which MMT4 uses visual features to outperform T5. The visual features of the product image help MMT4 to pick the right value for the attributes.

compared with the reference titles using the Rouge-1 metrics [17]. Rouge-1 shows what percentage of unigrams are in both generated and reference texts. The MMT4 with EfficientNet and ViT image representators showed 18% Rouge-1 recall. It means, in average, 18% of the words in each title is generated by our model. We compared the generated titles by MMT4 with the generated titles by the state-of-the-art captioning model, X-VLM [34]. The zero-shot X-VLM reported 13.3% and the fine-tuned (using the same data as MMT4) X-VLM reported 14.9% Rouge-1 recall. Fig. 9 shows some examples of the reference and generated titles. As shown in this figure, the type of shirt (t-shirt or tank top), item type (curtain, shows, shirt, pants, ...), phone model (Samsung, iPhone), and other attributes such as color and pattern are well addressed in the generated titles.

4.4 Fine-tuning: Image-Text Matching

Another application of the multimodal generative model is classification in generative mode. That is, the output class name is generated instead of being selected from a list of classes. In this task, the input data includes the product image and title followed by "match:" prompt. The output in this task is either "yes" or "no". The performance of MMT4 for image-text matching of 6819 validation data is shown in Table 4. The confusion matrix, precision-recall curve, and ROC of this test are shown in Fig. 10. As shown in Table 4, CLIP outperforms MMT4. This is mostly because of different training process where CLIP is pre-trained on 400 million image-text pairs using metric learning while MMT4 is only trained on 1.9 M samples in a multi-task generative schema. Thus, only 2% PR-AUC performance drop is a green light showing that MMT4 can be applied to classification tasks in multi-task problems. Additionally, using task-specific discriminators for classification is limited to a pre-defined set of classes and cannot extract out-of-box information from the input whereas MMT4 can generate new words that well explain the input data from multiple modalities.

5 CONCLUSION

This paper introduces a new multimodal generative model named MMT4 to generate text given different modalities as input. Involving more than one modality in catalog data processing improves the downstream tasks performance by providing more information about the product. The experimental results of attribute generation showed that MMT4 outperforms the text-only model (T5) in attribute generation, given that the attribute values do not always exist in the text. Thus, the results of this application can be used for

Input: Image	Output: Generated Title	Reference Title
	got symphony? - Adult Men's Soft T-Shirt Brand New Short Sleeve Tee, Green, XX-Large	got shindle? - Adult Men's Soft T-Shirt Brand New Short Sleeve Tee, Green, Medium
	Women's Tank Tops Black X-Large	MADE IN JAPAN Women's Tank Top Shirt
	YUAZHOQI Shower Curtain,Seasonal Waterproof Polyester Fabric Shower Door Curtains,Waterproof Showers Curtain for Bathroom,W72 x L72	Georgia Barnard 72" x 79" Shower Curtain, Bath Decor, Ethnic Style Bathroom Decor, Bathroom Curtain
	YUAZHOQI Women's Casual Loose Fit Casual Casual Pants Shirts,Light Blue,XXL	MOUTEN Women's Overalls Jeans Stretch Destroyed Ripped Hole Bib Long Jumpsuits Light Blue L
	New Arrival Case Cover With YZYYJJKJ Case For Iphone 6 Plus(sunrise)	New Style Tpu 6 Protective Case Cover/ Iphone Case - Peach
	Personalized Phone Case for Samsung Galaxy S5 I9600, - Printed Phone Cover w/Customized Design & Customized Style XL	K-K-Q- Soul Mate Matching Couple True Best Friend Phone Case For Samsung Galaxy note 3 N9000 [Pattern-5]
	XJJXXL Stainless Steel Watches for Men and Women, Personalized Watch for Women	Paper Priinted Wrist Watches XWDS238 New Volkswagen VW Golf Metal Mens Watch
	Xiaoyong Women's Canvas Shoes,Men'S Canvas Sneakers,Beautiful Summer Shoes	JIUDUIDODO Men's Game of Thrones Lace-up High-top Pop Canvas Shoes Black Sneakers,US12
	I'm A Yuri. To Save Time Let's Just Assume I Am Always Right. - 15oz Ceramic White Coffee Mug Cup, White	I Make The Name Oneika Look Good - 11oz Ceramic White Coffee Mug Cup, White

Figure 9: Examples of the generated titles by MMT4 given product images.

Table 3: Examples of attribute values generated by MMT4 and T5. The "Labels" for each attribute shows the number of valid values in the ground truth for that specific attribute. Data size shows the number of products with the corresponding attribute value in the ground truth. [Brand names are hidden for the sake of privacy].

Attribute	Attribute Value	Labels	Data Size	T5 TP	MMT4 TP
Sleeve-Type	Long Sleeve	5	220	172	**206**
	Short Sleeve		344	320	**322**
	Sleeveless		53	39	**41**
Color	Red	27	187	27	**159**
	White		514	210	**369**
Pattern	Striped	73	31	1	**20**
	Solid		74	59	**63**
	Print		88	**82**	**81**
Material	Wood	230	53	28	**37**
	Ceramic		71	61	**64**
	Cotton		373	277	**305**
Style	Classic	102	45	**28**	**29**
	Modern		96	65	**68**
	Art Deco		35	28	**35**
Brand	P***	4000	51	7	**33**
	S***		58	**58**	**58**
	H***		45	**45**	**45**

Table 4: Performance of MMT4 in image-text matching in comparison with CLIP.

Model	ROC-AUC	PR-AUC	R@0.80	R@0.85	R@0.90	R@0.95	R@0.97
MMT4	0.979	0.975	0.996	0.991	0.972	0.873	0.763
CLIP [24]	0.996	0.995	0.999	0.996	0.993	0.980	0.963

attribute correction and validation, especially where text sources miss attribute values. Additionally, title generation from image outperformed the state-of-the-art captioning model by generating key phrases in the title. Those can be used for improving Search in e-commerce. Although image-text matching as a generation problem

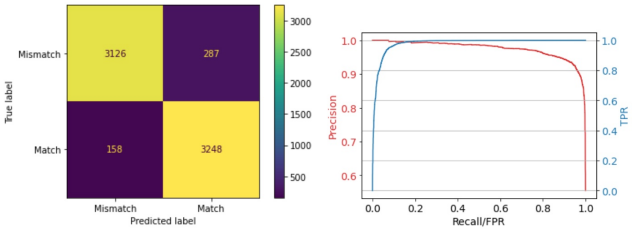

Figure 10: Confusion matrix and ROC/PR curves showing the performance of the image-text matching task using MMT4.

did not perform better than framing it as classification problem with CLIP, the 98% PR-AUC attained after only fine-tuning MMT4 on less than 2M samples warrants taking the next steps in fine-tuning the model on larger datasets.

As the next steps, we are planning to pre-train and fine-tune MMT4 on larger datasets (with more attributes). The fine-tuned model will be used for a series of downstream tasks such as product data inconsistency detection and attribute correction and the results will be compared with the state-of-the-art models. Finally, we will use the fine-tuned MMT4 to provide more information for other models to improve their performance (teacher pseudo-labeling). For example, generated product type from image and text can help the product type classification pipeline in the e-commerce data warehouse.

ACKNOWLEDGMENTS

Special thanks to Changhe Yuan and Shioulin Sam from Amazon IRIS-Science, and Yi Xu from M5-Search team for their crucial reviews and comments.

REFERENCES

[1] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. 2022. Data2vec: A general framework for self-supervised learning in speech, vision and language. *arXiv preprint arXiv:2202.03555* (2022).

[2] Jordan J Bird, Anikó Ekárt, and Diego R Faria. 2021. Chatbot Interaction with Artificial Intelligence: human data augmentation with T5 and language transformer ensemble for text classification. *Journal of Ambient Intelligence and Humanized Computing* (2021), 1–16.

[3] Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. 2021. Unifying vision-and-language tasks via text generation. In *International Conference on Machine Learning*. PMLR, 1931–1942.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[7] Isha Ganguli, Rajat Subhra Bhowmick, Shivam Biswas, and Jaya Sil. 2021. Empirical Auto-Evaluation of Python Code for Performance Analysis of Transformer Network Using T5 Architecture. In *2021 8th International Conference on Smart Computing and Communications (ICSCC)*. IEEE, 75–79.

[8] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, et al. 2018. Recent advances in convolutional neural networks. *Pattern Recognition* 77 (2018), 354–377.

[9] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*. PMLR, 4904–4916.

[10] Kelvin Jiang, Ronak Pradeep, and Jimmy Lin. 2021. Exploring listwise evidence reasoning with t5 for fact verification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. 402–410.

[11] Douwe Kiela, Suvrat Bhooshan, Hamed Firooz, Ethan Perez, and Davide Testuggine. 2019. Supervised multimodal bitransformers for classifying images and text. *arXiv preprint arXiv:1909.02950* (2019).

[12] Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. Vilt: Vision-and-language transformer without convolution or region supervision. In *International Conference on Machine Learning*. PMLR, 5583–5594.

[13] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942* (2019).

[14] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436–444.

[15] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461* (2019).

[16] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. 2021. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in Neural Information Processing Systems* 34 (2021).

[17] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.

[18] Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics* 8 (2020), 726–742.

[19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[20] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems* 32 (2019).

[21] Antonio Mastropaolo, Simone Scalabrino, Nathan Cooper, David Nader Palacio, Denys Poshyvanyk, Rocco Oliveto, and Gabriele Bavota. 2021. Studying the usage of text-to-text transfer transformer to support code-related tasks. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. IEEE, 336–347.

[22] Lindasalwa Muda, Mumtaj Begam, and Irraivan Elamvazuthi. 2010. Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques. *arXiv preprint arXiv:1003.4083* (2010).

[23] Mohiuddin Md Abdul Qudar and Vijay Mago. 2020. Tweetbert: A pre-trained language representation model for twitter text analysis. *arXiv preprint arXiv:2010.11091* (2020).

[24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*. PMLR, 8748–8763.

[25] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683* (2019).

[26] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

[27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems* 28 (2015).

[28] Taihua Shao, Yupu Guo, Honghui Chen, and Zepeng Hao. 2019. Transformer-based neural network for answer selection in question answering. *IEEE Access* 7 (2019), 26146–26156.

[29] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155* (2018).

[30] Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. 2019. Portuguese named entity recognition using BERT-CRF. *arXiv preprint arXiv:1909.10649* (2019).

[31] Mingxing Tan and Quoc Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*. PMLR, 6105–6114.

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[33] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mT5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934* (2020).

[34] Yan Zeng, Xinsong Zhang, and Hang Li. 2021. Multi-Grained Vision Language Pre-Training: Aligning Texts with Visual Concepts. *arXiv preprint arXiv:2111.08276* (2021).