Attribute Value Generation from Product Title using Language Models

Kalyani Roy
IIT Kharagpur, India
kroy@iitkgp.ac.in

Pawan Goyal
IIT Kharagpur, India
pawang@cse.iitkgp.ac.in

Manish Pandey
Carnegie Mellon University
manish.pandey@west.cmu.edu

Abstract

Identifying the value of product attribute is essential for many e-commerce functions such as product search and product recommendations. Therefore, identifying attribute values from unstructured product descriptions is a critical undertaking for any e-commerce retailer. What makes this problem challenging is the diversity of product types and their attributes and values. Existing methods have typically employed multiple types of machine learning models, each of which handles specific product types or attribute classes. This has limited their scalability and generalization for large scale real world e-commerce applications. Previous approaches for this task have formulated the attribute value extraction as a Named Entity Recognition (NER) task or a Question Answering (QA) task. In this paper we have presented a generative approach to the attribute value extraction problem using language models. We leverage the large-scale pretraining of the GPT-2 and the T5 text-to-text transformer to create fine-tuned models that can effectively perform this task. We show that a single general model is very effective for this task over a broad set of product attribute values with the open world assumption. Our approach achieves state-of-the-art performance for different attribute classes, which has previously required a diverse set of models.

1 Introduction

Product attributes and their values play an important role in e-commerce platforms. There are hundreds of thousands of products sold online and each type of product has a different set of attributes. These attributes help customers search for products, compare the relevant items and purchase the product of their choice. While details of a product can be found both in its title as well as its description, commonly, the title includes important attributes of the product. Everyday many new products are added to the product catalogue often with new attributes types and values. However, attribute information is often sparse, noisy and incomplete with missing values. For example, Figure 1 shows a product with its description and attribute value pairs available on the website. It contains attribute values for *Brand Name*, *Type* etc., but there are missing attributes, such as "Dual-coil" for *Pickup Type*, "6" for *Strings* etc. Given the wide diversity of products and new products constantly emerging, it is important that attribute value extraction works with the *open world assumption*, i.e., values for the attributes not seen before.

ammoon Electric Guitar 6 String Solid Wood Brims 23 Frets Basswood Body Dual-coil Pickup Tremolo & Rhythm Control with Pickguard

Brand Name : ammoon
Type : Electric Guitar
Tone Position : 23
Fingerboard Material : NULL
Body Material : Basswood

Figure 1: An example of a product with its title, attributes and values. There is no value for the attribute 'Fingerboard Material' and it is represented as NULL.

Earlier work (Ghani et al., 2006; Chiticariu et al., 2010; Gopalakrishnan et al., 2012) for attribute value extraction use a rule based approach with the help of a domain specific seed dictionary to identify the key phrases. Other work have formulated this as named entity recognition (NER) problem (Putthividhya and Hu, 2011; More, 2016). However, these approaches do not work under the open world assumption. More recently, various neural network based approaches have been proposed and applied to sequence tagging model for attribute value extraction. Huang et al. (2015) is the first to apply the BiLSTM-CRF model for sequence tagging. Zheng et al. (2018) propose an end-to-end tagging model using BiLSTM, CRF and attention without any dic-

Proceedings of the 4th Workshop on e-Commerce and NLP (ECNLP 4), pages 13–17

tionary or hand-crafted features. Most of these approaches create separate models for different attributes. Also, for each attribute a, they have one set of tags to denote beginning (B_a) and inside (I_a) of that attribute. Hence, these methods are not scalable for large set of attributes and these models can not identify emerging values for unseen attributes. Recent works (Xu et al., 2019; Wang et al., 2020) have set up this task as question answering (QA) task. Question answering in machine reading comprehension (MRC) selects a span of text from the given context to answer the question. Xu et al. (2019) considers product title as context, attribute as query, and proposes to find the attribute value using only global set of BIO tags. Although the sequence tagging models (Zheng et al., 2018; Xu et al., 2019) achieve promising result, they do not work well for discovering new attributes values.

In contrast to past extractive or classification based approaches, we have taken a generative approach to identify attribute values. Text generation using language models has several applications in real-world tasks such as text-editing, article writing, sentence completion, etc. Text infilling aims to fill the missing part of a given sentence. Motivated by their success as well as to leverage the large scale pretraining of the language models, we formulate the attribute value extraction as an instance of text infilling task as well as an answer generation task. We utilize Infilling by Language Modeling (ILM) (Donahue et al., 2020) for the infilling approach and we fine-tune Text-to-Text Transfer Transformer (T5) (Raffel et al., 2020) as an answer generation task. We summarize the main contribution of this work as follows:

- We propose a language modeling approach for attribute value extraction.
- We empirically demonstrate that this approach achieves state-of-the-art results on discovering new attribute values.

2 Problem Statement

In this section, we formally define the problem of attribute value generation. Given a product context $T = (w_1^t, w_2^t, ..., w_m^t)$ and its attribue $A = (w_1^a, w_2^a, ..., w_n^a)$, our goal is to generate the value $V = (w_1^v, w_2^v, ..., w_e^v)$. For example, the context of the product in Figure 1 is "ammoon Electric Guitar 6 String Solid Wood Brims 23 Frets Basswood Body Dual-coil Pickup Tremolo & Rhythm Control with Pickguard". Consider the two attributes *Type* and *Fingerboard Material*. We want to generate the value "Electric Guitar" for the attribute *Type* and NULL for the attribute *Fingerboard Material* as this attribute is not present in the context.

Attributes	Train	Valid	test
All	76,970	10,996	21,991
Brand Name	7,969	1,095	2,348
Material	2,824	373	752
Color	735	112	197
Category	662	86	206

Table 1: Statistics of the AV-109K dataset and its four frequent attributes

In this work, first, we formulate this problem as a (i) *text infilling* task and then as an (ii) *answer generation* task. For *text infilling*, we combine the context, T, attribute, A, and the value, V, in a sentence as "T. A is V." where the attribute value V is masked as blank. Our objective is to generate the missing span in this sentence to predict this value. Let the incomplete sentence be $\tilde{S} = (w_1^s, w_2^s, ..., w_p^s)$. Our model outputs the best attribute value sequence $\tilde{V}$ by learning the distribution $\tilde{V} = P(V|\tilde{S})$. In the *answer generation* approach, our aim is to generate V as the answer, considering T as the context and A as the question.

3 Experimental Setup

3.1 Dataset

We have used publicly available dataset[1] which is collected from Sports & Entertainment category of AliExpress (Xu et al., 2019). This dataset contains $110,484$ examples. Each example contains a triple, i.e., context as product title, an attribute and its value. We preprocessed the dataset to handle noisy data, and removed triples with empty values and triples with '-', '/' as value. This led to a dataset comprising of $109,957$ triples which we refer to as AV-109K. There are $2,157$ unique attributes and $11,847$ uniques values in this dataset. Also, not all the attributes have a value in the context and these are represented as NULL. There are $21,461$ such triples in AV-109K. We randomly split the data into 7:1:2 ratio, i.e., we randomly select $76,970$ triples as training set, $10,996$ triples as validation set, and the remaining $21,991$ triples as the test set.

[1] https://raw.githubusercontent.com/lanmanok/ACL19_Scaling_Up_Open_Tagging/master/publish_data.txt

Method	$EM(\%)$	$P(\%)$	$R(\%)$	$F_1(\%)$
SUOTag	68.88	70.81	71.31	71.06
ILM	81.14	83.35	83.38	83.37
T5	**81.35**	**83.89**	**83.75**	**83.82**

Table 2: Performance comparison on the AV-109K dataset

To further examine the model's ability to generate values for unseen attributes, we select five attributes with relatively low frequency ($< 0.1\%$) in the dataset: *Frame Color*, *Lenses Color*, *Shell Material*, *Wheel Material* and *Product Type* and the number of triples for these attributes are 108, 62, 36, 23, and 523, respectively. All the triples with these attributes are included in the test set. From the remainder of the dataset, we pick 10% as validation set and the rest as the training set. We refer to this dataset as AV-zero.

3.2 Evaluation Metrics

To evaluate the models, we use the *Exact Match (EM)* metric on the generated values where the whole sequence of the value must match. Since values can contain more than one tokens and models may generate tokens in any order, we have also computed average bag of word precision, recall and F_1 score as our evaluation measure which are denoted as P, R and F_1, respectively. Let N be the size of the dataset, $V = \{v_1, v_2, .., v_N\}$ be the gold standard values, $G = \{g_1, g_2, ..., g_N\}$ be the generated values, and $|v_i \cap g_i|$ denotes the bag of words overlap between the gold standard and the generated values corresponding to the i^{th} triple. The computation of P and R is shown below:

$$P = \frac{1}{N}\sum_{i=1}^{N}\frac{|v_i \cap g_i|}{|g_i|} \qquad R = \frac{1}{N}\sum_{i=1}^{N}\frac{|v_i \cap g_i|}{|v_i|}$$

3.3 Baselines

We compare our models with BiLSTM-CRF (Huang et al., 2015) and SUOTag (Scaling Up Open Tag) (Xu et al., 2019)[2].

- **BiLSTM-CRF** (Huang et al., 2015) is considered to be the state-of-the-art sequence tagging model for NER tasks. It uses the word embedding from pretrained BERT model and applies a BiLSTM layer over it to the contextual representation. Finally a Conditional Random Fields (CRF) (Lafferty et al., 2001) layer is applied over this BiLSTM.
- **SUOTag** (Xu et al., 2019) uses two separate BiLSTMs over the BERT based pretrained word embeddings to represent the context and attribute. Then, it applies a cross attention between these two representations followed by a CRF layer.

[2]AVEQA (Attribute Value Extraction via Question Answering) (Wang et al., 2020) is also a recent work that could potentially be a baseline, but we could not get the numbers as the code was not publicly available.

	Models	SUOTag	ILM	T5
NULL	Precision (%)	41.73	75.25	**77.32**
	Recall (%)	**93.10**	78.99	74.09
	$F_1(\%)$	57.63	**77.07**	75.67
EM(%) when attributes appear in context		28.86	**61.11**	54.57
EM(%) when attributes does not appear in context		69.53	81.22	**81.78**
Values having multiple words EM (%)		47.00	62.74	**62.96**
Numerical values		43.24	66.56	**72.06**

Table 3: Performance of models on AV-109K dataset in different scenarios.

3.4 Implementation Details

All the models are implemented with PyTorch (Paszke et al., 2019). We train each model for 5 epochs. The model that performs the best on the validation set is used for evaluating the test set. The minibatch size is fixed to 32. We use AdamW optimizer and a learning rate of 5e-5. We use pretrained GPT-2 small (Radford et al., 2019) model to train ILM and we use the validation set perplexity of the model on the masked token. We fine-tune T5-Base for the answer generation framework.

3.5 Results and Discussion

We conduct experiments on different settings to (1) explore the scalability on large attribute set, (2) compare the performance on four frequent attributes, and (3) examine the model's ability to discover new attributes.

Table 2 reports the performance on the AV-109K dataset. Since BiLSTM-CRF requires to tag each of the attributes a with separate B_a and I_a tags, it is not suitable for a large attribute set. So, we did not consider this model. The overall result shows that both ILM and T5 have the capability to a handle large number of attributes. Next, we examine the models for various interesting cases such as (a) when the values are NULL, (b) when the attributes appear in the context vs. when the attributes do not appear in the context, (c) when the values contain multiple words, and (d) when value has numerical

Attributes	Model	$EM(\%)$	$P(\%)$	$R(\%)$	$F_1(\%)$
Brand Name	BiLSTM-CRF	85.77	80.99	86.37	83.59
	SUOTag	91.05	92.53	92.35	92.44
	ILM	94.72	94.93	94.89	94.91
	T5	**94.97**	**95.35**	**95.29**	**95.32**
Material	BiLSTM-CRF	65.03	65.20	67.08	66.13
	SUOTag	68.09	72.21	72.36	72.28
	ILM	**85.24**	88.59	**88.10**	**88.34**
	T5	84.57	**88.94**	87.48	88.20
Color	BiLSTM-CRF	42.64	40.74	42.64	41.67
	SUOTag	42.64	43.15	43.09	43.12
	ILM	75.63	80.29	79.8	80.04
	T5	**76.65**	**80.63**	**81.02**	**80.82**
Category	BiLSTM-CRF	48.06	51.25	50.08	50.66
	SUOTag	52.43	56.55	55.26	55.90
	ILM	**79.13**	81.56	**81.96**	**81.76**
	T5	74.27	**81.67**	80.18	80.92

Table 4: Performance comparison of different models on four frequent attributes.

Attributes	Model	$EM(\%)$	$P(\%)$	$R(\%)$	$F_1(\%)$
Frame Color	SUOTag	71.30	71.76	72.22	71.99
	ILM	69.44	69.44	69.44	69.44
	T5	**74.07**	**74.07**	**74.07**	**74.07**
Lenses Color	SUOTag	64.52	64.52	64.52	64.52
	ILM	67.74	67.74	67.74	67.74
	T5	**69.35**	**69.35**	**69.35**	**69.35**
Shell Material	SUOTag	30.56	41.2	52.78	46.28
	ILM	47.22	59.72	72.22	65.38
	T5	**58.33**	**68.06**	**77.78**	**72.59**
Wheel Material	SUOTag	47.83	52.90	60.87	56.60
	ILM	69.57	69.57	69.57	69.57
	T5	**78.26**	**78.26**	**78.26**	**78.26**
Product Type	SUOTag	20.84	21.63	21.8	21.71
	ILM	**57.17**	**68.84**	**68.59**	**68.72**
	T5	52.20	62.01	64.15	63.06

Table 5: Performance comparison of different models on AV-zero for identifying values of unseen attributes.

data. The details are summarized in Table 3. ILM performs better than other models in identifying triples having NULL values. Specifically, language models give a much better precision in this case. There are 19.26% NULL values in AV-109K, but SUOTag predicts 43.83% data as NULL. Hence, it has such high recall. There are very few triples where the attributes appear in the context - only 1.50% in train dataset and 1.59% in test dataset. So, when the attribute appears in the context, the performance of all the models is poor in comparison with when the attribute does not appear in the context. In the AV-109K dataset, there are $4,058$ triples whose value consist of multiple words. T5 performs the best in finding the values having more than one word. There are 8.5% numerical data in the test set and T5 gives much better results than other models in identifying them.

The second experiment is conducted on the four most frequent attributes of the AV-109K dataset. Table 4 shows the result. T5 performs better than other models in *Brand Name* and *Color*. For *Material* and *Category*, ILM has the best performance. We have looked into the predictions of the values in these two categories and found that T5 is not correctly identifying the NULL values. On closer look at the dataset, we find that most of those NULL values are incorrectly annotated, e.g., "new 1pcs Golf Sports Mens Right Left Hand Golf Gloves Sweat Absorbent Microfiber Cloth Soft Breathable Abrasion Gloves" - the material of this product is microfiber, but it is annotated as NULL. T5 has predicted the category as Bicycle Saddle for the title "INBIKE Soft Wide Bicycle Saddle Comfortable Bike Seat Vintage Bicycle Leather Saddle Pad", but the annotation is NULL. Although T5 has identified the correct value of the attribute, it is marked as incorrect due to faulty annotation.

The last experiment is performed on AV-zero dataset. Table 5 shows the result of discovering values of five new attributes. ILM is the best in identifying "Product Type". The value of most of the "Product Type" is *Fishing Float*, but T5 either predicted the product type to be NULL or the type of the float, e.g., Luminous Fishing Float, Ice Fishing Float, etc. For the remaining three attributes, T5 outperforms other models.[3] Both T5 and ILM perform better than SUOtag in discovering unseen attribute values.

4 Conclusion

In this work, we present a formulation to generate product attribute values as (i) an instance of text infilling task and (ii) as an answer generation task. We show that we can leverage GPT-2 based and T5 text-to-text transformer models for this task. The models achieve strong results over a broad set of attributes. T5 performs better at multi-word values, and ILM is better at predicting null values. Additionally, our approach outperforms the state-of-the-art models for discovering new attribute values.

[3]We would like to note that in Table 5, for some of the attributes, all the evaluation metrics are identical. This occurs because for those attributes, the predicted value is a single token.

References

Laura Chiticariu, Rajasekar Krishnamurthy, Yunyao Li, Frederick Reiss, and Shivakumar Vaithyanathan. 2010. Domain adaptation of rule-based annotators for named-entity recognition tasks. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1002–1012, Cambridge, MA. Association for Computational Linguistics.

Chris Donahue, Mina Lee, and Percy Liang. 2020. Enabling language models to fill in the blanks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2492–2501, Online. Association for Computational Linguistics.

Rayid Ghani, Katharina Probst, Yan Liu, Marko Krema, and Andrew Fano. 2006. Text mining for product attribute extraction. *SIGKDD Explor. Newsl.*, 8(1):41–48.

Vishrawas Gopalakrishnan, Suresh Parthasarathy Iyengar, Amit Madaan, Rajeev Rastogi, and Srinivasan Sengamedu. 2012. Matching product titles using web-based enrichment. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, page 605–614, New York, NY, USA. Association for Computing Machinery.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Ajinkya More. 2016. Attribute extraction from product titles in ecommerce. *arXiv preprint arXiv:1608.04670*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037.

Duangmanee Putthividhya and Junling Hu. 2011. Bootstrapped named entity recognition for product attribute extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1557–1567, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

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 *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 47–55, New York, NY, USA. Association for Computing Machinery.

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 Annual Meeting of the Association for Computational Linguistics*, pages 5214–5223, Florence, Italy. Association for Computational Linguistics.

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 and Data Mining*, KDD '18, page 1049–1058, New York, NY, USA. Association for Computing Machinery.