AttriSAGE: Product Attribute Value Extraction Using Graph Neural Networks

Rohan Potta[*1] **Mallika Asthana**[*1] **Siddhant Yadav**[*1]
Nidhi Goyal[1] **Sai Amrit Patnaik**[2] **Parul Jain**[3]
[1]Mahindra University [2]IIIT Hyderabad [3]IIT Delhi, India
{rohan20ucse145, mallika20ucse086, se20uecm082, nidhi.goyal}@mahindrauniversity.edu.in,
sai.patnaik@research.iiit.ac.in, paruljainfeb@gmail.com

Abstract

Extracting the attribute value of a product from the given product description is essential for e-commerce functions like product recommendations, search, and information retrieval. Therefore, understanding products in e-commerce with greater accuracy certainly gives any retailer the edge. However, they are limited to contextual modeling and do not exploit relationships between the product description and attribute values.

Through this paper, in a world where we move and shift to more complicated models with extensive training time with models like LLMs, we present a novel, more straightforward attribute value extraction from product description leveraging graphs and graph neural networks. Our proposed method demonstrates improvements in attribute value extraction accuracy compared to the baseline sequence tagging approaches while also significantly reducing the computation time leading to lower carbon footprint.

1 Introduction

In the dynamic landscape of e-commerce, where a wide range of products are readily available to consumers, efficient and accurate product understanding plays a pivotal role in facilitating seamless user experiences. The attributes associated with products, including details such as color, material, brand, type, and more, hold the key to enabling users to find their desired items more efficiently.

E-commerce platforms usually provide product descriptions but consumers prefer a quick and intuitive way to narrow down their search and make informed purchasing decisions. Product titles usually contain attributes and their corresponding values but this data is mostly unstructured, noisy, and often contains missing values. For example, in Figure 1, a product along with its context (description) is provided. Along with the description, there are attribute-value pairs for attributes including Gender, Brand, Feature, etc ; But, there also missing attributes for values like Model number (value: 849558), Model name (value: Air VaporMax), etc. Hence we need models that predict attribute values for the attributes that have not been seen before.

New Arrival Original Authentic Nike Air VaporMax Flyknit Running Shoes Men Breathable Sport Outdoor Sneakers 849558

Gender: Men
Brand Name: Nike
Feature: Breathable
Insole Material: NULL
Athletic Shoe Type: Running Shoes

Figure 1: A product description with its attributes and their corresponding values represented as "Attribute: Value".

The critical role of product attributes has driven extensive research efforts to explore innovative methods for their extraction and categorization. Previous works, including those by Ghani et al., 2006, Chiticariu et al., 2010, and Gopalakrishnan et al., 2012, focused on attribute value extraction using a rule-based approach. In this methodology, a domain-specific seed dictionary played a crucial role in identifying key phrases and extracting attribute values. The rule-based systems relied on predefined patterns and heuristics to recognize and capture relevant information from unstructured data, providing a foundational approach to attribute extraction in the context of specific domains. Other works proposed a Named-Entity Recognition (NER) task (Putthividhya and Hu, 2011) for this problem; although NER relies on pre-existing knowledge of named entities. When faced with previously unseen brands, models, or attributes, the system struggles to identify and extract these values accurately. In

[*]These authors contributed equally to this work

Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop, pages 89–94

such situations, a more context-aware approach, like question-answering-based techniques that employ sequence-to-sequence models, might be more effective for attribute value extraction from product descriptions. Later introduced works that employed sequence-to-sequence models performed better than the former models, however, these approaches have a few shortcomings-

(a) they do not exploit the structural relationships between product description and attribute values across the dataset. For example, assume product descriptions C_1 and C_2 share a common attribute value T_1. If there is another attribute value T_2 relevant to C_2 and other similar product descriptions, we can infer that T_2 might also be relevant to C_1. Such transitive cues can be beneficial for identifying missing attribute values.
(b) language models bring high computational costs at massive scales as any task not only involves predicting multiple missing attribute values but also requires precise organization of the most relevant attribute values specific to the product. Graphs are naturally suitable to make the relationships explicit such as product description-attribute value networks.
(c) With the growing popularity of LLMs, we tend to oversee the ecological impact they have on the environment. They consume vast computational resources, leading to significant energy use and high carbon emissions.

In this work, our focus is on advancing the domain of product attribute value extraction through a novel approach that leverages graph models and graph neural networks (GNNs). Our primary goal is to enhance the generalizability of existing approaches and provide more interpretable predictions. We construct a product data graph using a dataset comprising 110k product title-attribute triples, enabling us to gain deeper insights into the data. Leveraging graph-based neural network architecture we performed a node classification task to classify our title nodes with multiple attribute values.

Through this work, we aim to contribute the following:

- A Graph Neural Network (GNN) based approach for attribute value extraction from a given product description.
- A Knowledge graph that captures the transitive relations and can predict the missing attribute values through these transitive links for up to k-number of hops.
- Using the GraphSAGE model, we are able to reduce the training time significantly.

2 Related Work

Initial works focusing on the attribute value extraction task involved the use of domain-specific rules to detect attribute-value combinations from product descriptions (Zhang et al., 2009). The first learning-based approaches required substantial feature engineering and were limited in their capacity to generalize to unknown features and attribute values.

The initial application of the bidirectional LSTM with a Conditional Random Field layer (BiLSTM-CRF model) for sequence tagging in attribute value extraction was introduced by Huang et al., 2015. Following this, Zheng et al., 2018 proposed an end-to-end tagging model, OpenTag utilizing BiLSTM, CRF, and attention mechanisms, eliminating the need for dictionaries or hand-crafted features. However, this methodology poses scalability challenges when dealing with a large set of attributes and cannot identify emerging values for previously unseen attributes. An extension to OpenTag, SUOpenTag was proposed by Xu et al., 2019 which encodes both a target attribute and the product title using the pre-trained language model, BERT (Devlin et al., 2019). Wang et al., 2020 proposed AVEQA which formulates the attribute value extraction from products task as a multi-task approach via Question Answering.

With the advancements in the field of language models, recent works by Roy et al., 2021 leverage large language models to extract attribute values from product data. They formulated the attribute value extraction as an instance of text infilling task as well as an answer generation task for which they utilized Infilling by Language Modeling (ILM) (Donahue et al., 2020) for the infilling approach and fine-tuned text-to-text transfer transformer (T5) (Raffel et al., 2023) as an answer generation task. These models outperform the existing models but they fail to capture the intricate relations between different products.

3 Problem Formulation

We can formulate this problem of attribute value extraction as follows:

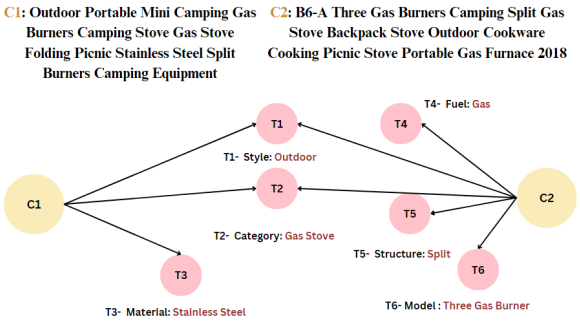

Figure 2: A graph illustrating the Description-Attribute-Value model for a given product and outlining the graph.

Given a product description, $\mathcal{C}$, such that, $\mathcal{C} = \{C_1, C_2, C_3, \ldots, C_i\}$ and an attribute A, the goal is to predict the corresponding attribute value $\mathcal{T}$ associated with A where $\mathcal{T} = \{T_1, T_2, T_3, \ldots, T_j\}$, where i and j are the number of unique source and target nodes respectively. We consider the nodes representing $\mathcal{C}$ as the Source node and $\mathcal{T}$ as the Target node. For constructing the graph, the Product Description, $\mathcal{C}$, and the Attribute Values, $\mathcal{T}$ are arranged in a graph $G = (V, E)$ where V, the nodes represent $\mathcal{C} \cup \mathcal{T}$.

$E = \mathcal{C} \times \mathcal{T}$ is the set of edges denoting the ground truth relation between product descriptions, $\mathcal{C}$, and the attribute values, $\mathcal{T}$. We formulate our problem as a multi-label node classification task which also takes into consideration transitive relations between the nodes. This formulation allows more comprehensive correlations to be inferred. For example, from Figure 2 , we can infer that titles C_1 and C_2 share a common value T_1. If there is another value T_2 relevant to C_2, it can be inferred that T_2 might be relevant to C_1 as well, i.e., one of the labels for C_1 could be T_2. This formulation helps us improve the interpretability of the obtained results. Table 1 depicts statistics of the graph modelled on the entire AE-110K dataset after pre-processing.

3.1 Implementation

All the models are implemented using PyTorch (Paszke et al., 2019).

For each product description $i \in C$, and attribute value $j \in T$, we generated a D dimensional initial representation of their textual features capturing the semantic information of these values.

These initial features, which we could call word embeddings were generated using pretrained FastText (Bojanowski et al., 2017) and BERT (Devlin et al., 2019). These word embeddings provide a dense representation of words in a continuous vector space, enabling the model to capture semantic relationships and nuances. Additionally, BERT works well with numerical text hence if the value is composed of numbers the model can grasp the semantics of the value well. For implementing the graph neural network to process the graph-structured data, we have implemented the GraphSAGE (Graph Sample and Aggregation) model (Hamilton et al., 2017), which performs neighbor sampling and aggregation to generate embeddings for each node in the graph. Our model architecture can be explained as follows:

Property	Value
Nodes	52,028
Source Nodes	39,445
Target Nodes	12,586
Edges	85,872
Avg Degree	3.3009
Density	0.0634

Table 1: Graph Statistics

Let $G = (V, E)$ be the input graph, where V is the set of nodes and E is the set of edges. For each node $v_i \in V$, there is an initial node feature vector x_i representing the textual features:

$$x_i \in R^D$$

where D is the dimensionality of the word embeddings. In our case, D equals 768, representing the dimension of the BERT embeddings. Sampling neighbors of each node v_i is done as :

$$N(v_i) = \{v_{i,1}, v_{i,2}, \ldots, v_{i,k}\} \quad (1)$$

where k is the number of sampled neighbors. Then a mean aggregator is applied for aggregating information from the node and its neighbors:

$$h'_i = Aggregate(\{h_{i,1}, h_{i,2}, \ldots, h_{i,k}\}) \quad (2)$$

Then the aggregated representation is concatenated with the initial node embedding:

$$h_i = Concat(h'_i, x_i) \quad (3)$$

Finally, the model is trained to minimize the difference between predicted and ground truth attribute values:

$$Minimize \sum_{i \in N} Loss(h_i, ground\_truth_i) \quad (4)$$

Attributes	Train	Dev	Test
Brand Name	50,413	5,601	14,055
Material	22,814	2,534	6,355
Color	5,594	621	1,649
Category	5,906	590	1,649
All	77,207	10,920	22,169

Table 2: The table represents the most frequently occurring attributes (Brand Name, Material, Color, Category) from the AE-110K dataset.

The Cross-Entropy Loss is calculated between the predicted probabilities and the true labels. Training using backpropagation and stochastic gradient descent (SGD) is performed and the model parameters are updated.

For k-hop architecture, repeat sampling and aggregation for k hops:

$$h_i^{(1)}, h_i^{(2)}, \ldots, h_i^{(k)} \tag{5}$$

The representations from each hop can be concatenated as:

$$h_i = Concat(h_i^{(1)}, h_i^{(2)}, \ldots, h_i^{(k)}) \tag{6}$$

4 Experiment Setup

4.1 Dataset

We have used the publicly available AE-110K dataset[1] from The Sports and Entertainment category of AliExpress (Xu et al., 2019). This dataset contains 110,484 triples, wherein each triple consists of the product title (context), attribute, and value each separated by a delimiter. For our task, we pre-processed the dataset to handle triples with empty values as well as triples where the attribute value was denoted by '-' and '/'. The resultant dataset consists of 110,296 triples with 2761 unique attributes and 12,607 unique attribute values. We divided the data randomly into a 7:1:2 ratio. Specifically, we chose 77,207 triples as our training set, 10,920 triples as the validation set, and the remaining 22,169 triples as our test dataset. Table 2 shows the most frequently occurring attributes in the AE-110K dataset.

4.2 Evaluation Metrics

The model's performance was assessed on the test set, by employing a comprehensive set of metrics. We calculated average metrics for F1-score, precision (P), and recall (R). The objective is to assess the model's ability to accurately predict the attributes associated with each product title node in the graph. The metrics are represented by F_1 score, P, and R respectively. Let u_i and g_i be the gold standard and generated values for the i-th sample respectively and let N be the total number of samples in the test set, then:

$$P = \frac{1}{N}\sum_{i=1}^{N}\frac{|v_i \cap g_i|}{|g_i|} \tag{7}$$

$$R = \frac{1}{N}\sum_{i=1}^{N}\frac{|v_i \cap g_i|}{|v_i|} \tag{8}$$

4.3 Baselines

We compare our models with SUOTag (Scaling Up Open Tag) Xu et al., 2019 and ILM-T5 (Roy et al., 2021).

- **SUOTag** (Xu et al., 2019) employs a BiLSTM-based architecture with attention and CRF components. It utilizes pre-trained BERT embeddings for word representation and employs two separate BiLSTMs for title and attribute modeling. An attention layer is applied to capture the semantic interaction between attributes and titles. The output layer utilizes a CRF layer to predict tag sequences, considering dependencies between output tags. (Lafferty et al., 2001).

- **ILM-T5** (Roy et al., 2021) presents the problem formulation to generate product attribute values as two tasks - (i) an instance of text infilling task leveraging the Infilling by Language Modeling (ILM) and pre-trained GPT-2 small (Radford et al., 2019) model and (ii) as an answer generation task using the text-to-text transfer transformer (T5) model.

4.4 Result

Table 3 presents the performance of the AttriSAGE model in comparison to the baseline models on the AE-110K dataset. AttriSAGE works well on a large set of attributes. With even a simple and compact graph-based network like ours, we can achieve performance comparable to LLMs, which demonstrates substantial improvements compared to sequence tagging models. Our model achieved an F1 score of 80.45, signifying a notable improvement over the sequence tagging models.

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

Model	Precision	Recall	F1
SUOTag	70.81	71.31	71.06
ILM	83.35	83.38	83.37
T5	83.89	83.75	83.82
AttriSAGE	79.06	81.90	80.45

Table 3: Performance of Different Models on AE-110K

Additionally, our AttriSAGE model significantly reduces overall training time and efficiently manages computational resources compared to Large Language Models. The model was trained on the NVIDIA DGX A100 GPU and it took 2-3 hours to execute, showing improvement in terms of both time and resource utilization.

5 Discussions

Our model's success in capturing the essence of the dataset can be attributed to its interpretability. We have utilized the structured format of a graph to restructure the data, which aligns with the analysis capabilities of a graph neural network. Unlike an LLM, which predicts the next token in the same dataset, our graph neural network excels in analyzing structured data and making accurate predictions leveraging a graph's ability to learn from its neighborhood. By capturing the relationships between data points through the graph, our model has achieved significant levels of accuracy. Moreover, our model's interpretability allows us to comprehend the rationale behind its predictions, which is crucial for maintaining its dependability and credibility.

6 Conclusion and Future Work

In this work, we have proposed a novel approach to extract attribute values from unstructured product data with the help of graphical representation. Representing the e-commerce data as graphs and leveraging graph techniques to extract the attribute values helped in understanding the underlying relationships between different products and forming transitive relations between products and their corresponding values.

We plan on extending this work to build an advanced multi-hop model architecture that can make better predictions under diverse scenarios, including handling missing values, exploring strategies for imputing the most frequent values, and addressing other issues and datasets.

Limitations

The current method has only been tested on a single dataset, which is the primary limitation of this work. Although the results are promising in this particular context, the generalizability of the method across diverse datasets and under different scenarios remains untested. To overcome this limitation, future work would expand the experiments to include a more varied selection of datasets. Furthermore, this work currently only focuses on the GraphSAGE architecture. Alternative graph-based architectures with different configurations and hyperparameter settings could be explored to enhance the current findings and results.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

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.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. 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.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging.

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.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library.

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, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer.

Kalyani Roy, Pawan Goyal, and Manish Pandey. 2021. Attribute value generation from product title using language models. pages 13–17.

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.

Liyi Zhang, Mingzhu Zhu, and Huang Wei. 2009. A framework for an ontology-based e-commerce product information retrieval system. *Journal of Computers*, 4.

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 '18, page 1049–1058, New York, NY, USA. Association for Computing Machinery.