# TACLR: A Scalable and Efficient Retrieval-based Method for Industrial Product Attribute Value Identification

**Anonymous ACL submission**

## Abstract

Product Attribute Value Identification (PAVI) involves identifying attribute values from product profiles, a key task for improving product search, recommendation, and business analytics on e-commerce platforms. However, existing PAVI methods face critical challenges, such as inferring implicit values, handling out-of-distribution (OOD) values, and producing normalized outputs. To address these limitations, we introduce Taxonomy-Aware Contrastive Learning Retrieval (TACLR), the first retrieval-based method for PAVI. TACLR formulates PAVI as an information retrieval task by encoding product profiles and candidate values into embeddings and retrieving values based on their similarity to the item embedding. It leverages contrastive training with taxonomy-aware hard negative sampling and employs adaptive inference with dynamic thresholds. TACLR offers three key advantages: (1) it effectively handles implicit and OOD values while producing normalized outputs; (2) it scales to thousands of categories, tens of thousands of attributes, and millions of values; and (3) it supports efficient inference for high-load industrial deployment. Extensive experiments on proprietary and public datasets validate the effectiveness and efficiency of TACLR. Moreover, it has been successfully deployed in a real-world e-commerce platform, processing millions of product listings daily while supporting dynamic, large-scale attribute taxonomies.

## 1 Introduction

Product attribute values are critical components that support the function of e-commerce platforms. They provide essential structural information, aiding customers in making informed purchasing decisions and enabling product listing (Chen et al., 2024), recommendation (Truong et al., 2022; Sun et al., 2020), retrieval (Magnani et al., 2019; Huang et al., 2014), and question answering (Kulkarni et al., 2019; Gao et al., 2019).

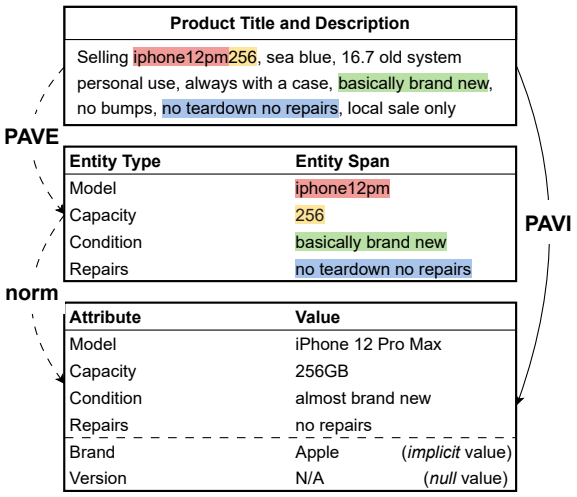



Figure 1: Illustration of the PAVE and PAVI tasks. While an additional normalization step can adapt PAVE methods for PAVI, these methods remain unable to identify implicit values, such as *Apple*.

However, seller-provided attribute values are often incomplete or even inaccurate. This undermines the effectiveness of applications that rely on this information. Consequently, the automatic identification of product attribute values has become a critical challenge. Researchers have explored the task of Product Attribute Value Extraction (PAVE), which extracts spans from product profiles using Named Entity Recognition (NER) (Zheng et al., 2018) or Question Answering (QA) (Wang et al., 2020) models. The upper part of Figure 1 illustrates an example of NER-based PAVE.

Although these approaches effectively extract value spans, the outputs remain raw subsequences. Presenting attribute values in a standardized format is crucial for facilitating data aggregation in business analytics and enhancing the user experience by providing clear and consistent information. To produce standardized values, a normalization step (Putthividhya and Hu, 2011; Zhang et al., 2021) is required to map these spans to predefined formats,

as shown in the lower part of Figure 1. However, implicit values, such as *Apple*, cannot be directly extracted and must instead be inferred from context or prior knowledge.

Therefore, in this work, we focus on the task of Product Attribute Value Identification (PAVI) (Shinzato et al., 2023), which aims to associate predefined attribute values from attribute taxonomy (illustrated in Figure 2) with products. The input to PAVI includes the product category and profile, where the profile comprises textual data, such as the title and description, and may optionally include visual information, such as images or videos. The output is a dictionary with predefined attributes as keys and the inferred attribute values as corresponding entries. In addition, PAVI requires determining when attribute values are missing. For instance, as shown in Figure 1, the value for *Version* is unavailable and is therefore assigned an empty result or *null* value.

Beyond adapting extraction-based approaches, researchers have investigated classification-based (Chen et al., 2022) and generation-based paradigms (Sabeh et al., 2024b) for PAVI. Classification-based methods treat each value as a label and employ multi-label classification models to recognize values across multiple attributes. While straightforward, these methods are fundamentally limited by their inability to identify out-of-distribution (OOD) values not present in the training data, making them impractical for the dynamic and continuously evolving nature of e-commerce platforms. In contrast, generation-based methods reformulate PAVI as a sequence-to-sequence problem. Although these methods can handle implicit and OOD values, they face significant challenges, such as generating uncontrollable outputs and incurring substantial computational costs in high-load scenarios due to their reliance on Large Language Models (LLMs).

In summary, existing approaches face distinct challenges, including difficulties identifying implicit values, generalizing to OOD values, producing normalized outputs, or ensuring scalability and efficiency for large-scale industrial applications. To address these limitations, we propose a novel retrieval-based method, Taxonomy-Aware Contrastive Learning Retrieval (TACLR). Our approach formulates PAVI as an information retrieval task: the product item serves as the query, and the attribute taxonomy acts as the corpus, enabling the efficient retrieval of relevant attribute values as matched documents.

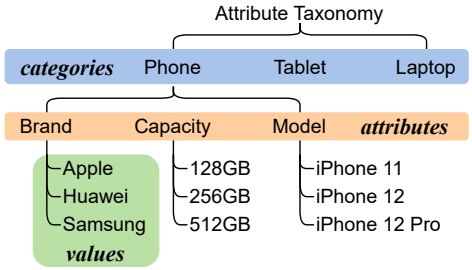



Figure 2: An illustration of a portion of the attribute taxonomy. Each category, such as *Phone*, is linked to multiple attributes, including *Brand*, *Model*, and *Capacity*, with standardized values enumerated for each attribute (e.g., *Apple*, *Huawei*, and *Samsung* for *Brand*).

We use a shared encoder to generate embeddings for the input product and candidate values from the attribute taxonomy. The cosine similarity between these embeddings is computed and normalized to produce a relevance score. Our method adopts a contrastive learning framework inspired by CLIP (Radford et al., 2021). Rather than relying on in-batch negatives, we implement taxonomy-aware negative sampling, which selects hard negative values from the same category and attribute to generate a more challenging and precise contrastive signal. Additionally, learnable null values are explicitly incorporated as the ground truth for product-attribute pairs without associated values. During inference, we address the limitations of static thresholds by introducing dynamic thresholds derived from the relevance score of null values. This adaptive inference mechanism improves the model's ability to generalize across a large-scale attribute taxonomy.

Our contributions are threefold: (1) We propose a novel retrieval-based paradigm for PAVI, introducing a scalable and efficient framework capable of handling implicit values, generalizing to OOD values, and producing normalized outputs. (2) We incorporate contrastive training into TACLR, using a taxonomy-aware negative sampling strategy to improve representation discrimination. Additionally, TACLR features an adaptive inference mechanism that dynamically balances precision and recall in large-scale industrial applications. (3) We validate the effectiveness of TACLR through extensive experiments on proprietary and public datasets. In addition, TACLR has been successfully deployed in a real-world industrial environment, processing millions of product listings and supporting thousands of categories and millions of attribute values.

## 2 Related Work

### 2.1 Product Attribute Value Extraction

**PAVE as Named Entity Recognition.** PAVE can be formulated as NER by identifying subsequences in product texts as entity spans and associating them with attributes as entity types. Early methods, such as OpenTag (Zheng et al., 2018), trained individual models for each category-attribute pair. Subsequent efforts generalized this approach to support multiple attributes or categories. For instance, SUOpenTag (Xu et al., 2019) incorporated attribute embeddings into an attention layer to handle multiple attributes, while AdaTag (Yan et al., 2021) used attribute embeddings to parameterize the decoder. TXtract (Karamanolakis et al., 2020) introduced a category encoder and a category attention mechanism to tackle various categories effectively. Additionally, M-JAVE (Zhu et al., 2020) jointly modeled attribute prediction and value extraction tasks while also incorporating visual information. More recently, Chen et al. (2023) scaled BERT-NER by expanding the number of entity types to support a broader range of attributes.

**PAVE as Question Answering.** The QA framework can also be adapted for PAVE by treating the product profile as context, attributes as questions, and value spans extracted from the context as answers. Wang et al. (2020) first introduced AVEQA for QA-based PAVE. Subsequent work extended this framework by incorporating multi-source information (Yang et al., 2022), multi-modal feature (Wang et al., 2022), and trainable prompts (Yang et al., 2023). Moreover, the question can be extended by appending candidate values as demonstrated by (Shinzato et al., 2022). Combining NER and QA paradigms, Ding et al. (2022) proposed a two-stage framework, which first identifies candidate values and then filters them.

While NER- and QA-based paradigms have proven effective for PAVE, they struggle to identify implicit attribute values. Additionally, both paradigms rely on post-extraction normalization to standardize values, using either string-based methods (Putthividhya and Hu, 2011) or semantic-based techniques (Zhang et al., 2021). Furthermore, QA-based methods require processing a single product multiple times to handle multiple target attributes, leading to inefficiencies in large-scale settings. These limitations underscore the need for novel paradigms integrating extraction and normalization while addressing implicit values.

Table 1: Comparison of different paradigms for identifying implicit, OOD, and normalized values.

| Paradigm | Implicit | OOD | Normalized |
|---|---|---|---|
| Extraction | ✗ | ✓ | ✗ |
| Classification | ✓ | ✗ | ✓ |
| Generation | ✓ | ✓ | ✗ |
| Retrieval | ✓ | ✓ | ✓ |

### 2.2 Product Attribute Value Identification

**Classification-Based PAVI.** A straightforward approach is to frame PAVI as a multi-label classification problem over a finite set of values. Chen et al. (2022) trained a unified classification model that masks invalid labels based on the product category. However, a significant limitation of this classification-based paradigm is its inability to recognize OOD values not included in the training set. This limitation reduces its practicality in dynamic e-commerce environments, where new categories and values frequently emerge.

**Generation-Based PAVI.** Recent advancements in LLM have spurred the exploration of generation-based PAVI methods (Sabeh et al., 2024b). Some methods (Roy et al., 2021; Nikolakopoulos et al., 2023; Blume et al., 2023) construct attribute-aware prompts to generate values for each attribute individually. In contrast, others generate values for multiple attributes simultaneously, either in a linearized sequence format (Shinzato et al., 2023) or as a hierarchical tree structure (Li et al., 2023). Multimodal information has also been integrated into LLMs to identify implicit attribute values from product images (Lin et al., 2021; Khandelwal et al., 2023). More recently, Brinkmann et al. (2024) explored the use of LLMs for both the extraction and normalization of attribute values. Additionally, Zou et al. (2024) introduced the learning-by-comparison technique to reduce model confusion, and Sabeh et al. (2024a) investigated Retrieval-Augmented Generation (RAG) technologies for PAVI.

Although generation-based methods can infer implicit and OOD attribute values from product profiles, they face several challenges in real-world scenarios. A key issue is the potential for the LLMs to produce uncontrollable or hallucinated outputs, a known limitation of LLMs (Huang et al., 2024). Additionally, these methods often rely on large, computationally intensive models to achieve strong performance, making them inefficient and costly for large-scale industrial deployment.

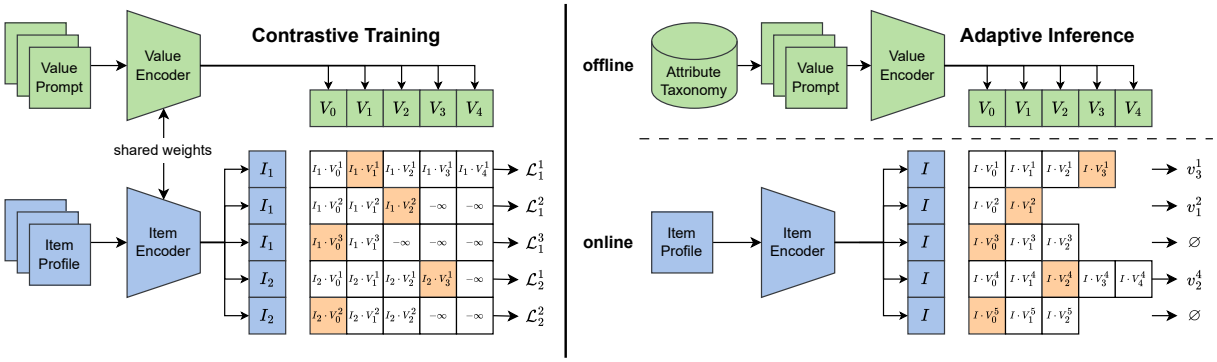



Figure 3: Overview of the training and inference process of our retrieval-based PAVI method. The left section illustrates contrastive training with taxonomy-aware negative sampling, while the right section demonstrates adaptive inference with pre-computed value embeddings.

## 3 Taxonomy-Aware Contrastive Learning Retrieval

This section defines the PAVI task with an attribute taxonomy (§3.1) and presents our retrieval-based paradigm for PAVI (§3.2). We then detail the use of contrastive training with taxonomy-aware negative sampling (§3.3) and an adaptive inference mechanism with dynamic thresholds (§3.4). Figure 3 provides an overview of the approach.

### 3.1 PAVI Task Definition

PAVI is grounded in an attribute taxonomy that encompasses numerous product categories. For each category $c$, the taxonomy specifies a set of attributes $\mathcal{A}_c = \{a_1, a_2, \dots\}$ relevant to products in that category, and for each attribute $a \in \mathcal{A}_c$, it provides a predefined set of standard values $\mathcal{V}_a = \{v_1, v_2, \dots\}$. Figure 2 illustrates this structure.

For a given product item $i$, with its title $t$ and description $d$, the item is assigned to a specific category $c$ with associated attributes $\mathcal{A}_c$. The objective of the PAVI task is to identify a relevant set of values $\mathcal{V}_a^+ \subseteq \mathcal{V}_a$ for each attribute $a \in \mathcal{A}_c$. The set $\mathcal{V}_a^+$ can take one of three forms: a singleton ($\{v\}$), multiple values ($\{v_1, v_2, \dots\}$), or an empty set ($\varnothing$) if no information about $a$ is available in the product profile. Notably, a standard value may not always appear explicitly as a text span in $t$ or $d$; it may be conveyed in other forms. When a value is not explicitly mentioned, such cases are referred to as implicit values.

### 3.2 Retrieval-Based PAVI

In a standard information retrieval setting, given a query, the objective is to retrieve a list of relevant documents from a corpus. Similarly, for PAVI, we treat the input item as the query and the attribute taxonomy as the corpus, aiming to retrieve relevant attribute values as the output documents.

To achieve this, we use encoders to generate embeddings for both the item and its candidate values. The cosine similarity between the item embedding and each candidate value embedding is computed and normalized to the range $[0, 1]$ to measure relevance. For each attribute, the candidate values are ranked based on their similarity scores, and the most relevant values are selected as the output set.

To effectively encode both the item and candidate values, we preprocess them as textual inputs and use a shared text encoder. For each item, we concatenate its title $t$ and description $d$ as the input text. Each candidate value $v$ associated with attribute $a$ under category $c$ is represented as a text prompt, such as "A [phone ($c$)] with [brand ($a$)] being [Apple ($v$)]". [1] We explore the impact of various value prompt templates in §5.3.

### 3.3 Contrastive Training

Inspired by CLIP (Radford et al., 2021), we employ contrastive learning to train the shared encoder. Rather than relying on in-batch negatives, we compare each positive value with hard negative values from the same category and attribute in the taxonomy, providing a more challenging and precise training signal.

Formally, the subset of values matched with the item is referred to as the ground truth value set, $\mathcal{V}_a^+ \subseteq \mathcal{V}_a$. If no matched values exist for a given attribute, i.e., $\mathcal{V}_a^+ = \varnothing$, we assign a specific *null value* $v_0^a$ for this attribute as the positive value, i.e. $v_a^+ = v_0^a$. Otherwise, a positive value is ran-

[1] This framework can be extended to multimodal scenarios by replacing the text encoder with a multimodal encoder to incorporate features like images.

domly drawn from the ground truth value set, i.e. $v_a^+ \sim \mathcal{V}_a^+$. For negative sampling, we select values as $\mathcal{V}_a^- = \{v_1^-, v_2^-, \dots\} \subseteq \mathcal{V}_a - \mathcal{V}_a^+$, ensuring a maximum of $K$ values. The contrastive loss is then computed as follows:

$$\mathcal{L}_a = -\log\left(\frac{\exp(\frac{s(i,v_a^+)}{\tau})}{\exp(\frac{s(i,v_a^+)}{\tau}) + \sum_{v\in\mathcal{V}_a^-}\exp(\frac{s(i,v)}{\tau})}\right),$$

where $s(i,v) = \frac{I\cdot V}{\|I\|\|V\|}$ denotes the cosine similarity between the item embedding $I$ and the value embedding $V$, and $\tau$ is the temperature hyperparameter. It is important to note that each item typically includes multiple attributes, all of which share the same item embedding $I$ while being individually compared against corresponding values. Therefore, the loss for item $i$ is the sum of losses over all attributes from $\mathcal{A}_c$:

$$\mathcal{L}_i = \sum_{a\in\mathcal{A}_c}\mathcal{L}_a.$$

An example logit matrix is depicted on the left side of Figure 3. Note that the item embedding $I_1$ contributes to the loss computations of $\mathcal{L}_1^1$, $\mathcal{L}_1^2$, and $\mathcal{L}_1^3$, which correspond to the attributes $a_1$, $a_2$, and $a_3$ within the same product category. We also pad the logit matrix with negative infinity for batched computation if fewer than $K$ values are available.

### 3.4 Adaptive Inference

During retrieval, relevance scores are assigned to every candidate values. To filter output values, a static threshold $T$ can be applied to these scores. However, in real-world e-commerce platforms with a vast number of category-attribute pairs, using a single threshold across all pairs is often suboptimal. Moreover, defining a unique threshold for each pair is tedious or even impractical.

To address this, we introduce an adaptive inference method that uses dynamic thresholds to make cutoff decisions. As discussed in §3.3, we add an explicit null value $v_0^a$ for each category-attribute pair, with its embedding learned during training. In the inference phase, we compute the similarity $s(i, v_0^a)$ between the item and the null value, using it as a dynamic threshold $T_a'$ to exclude candidate values for attribute $a$ that have lower scores:

$$\mathcal{V}_a^{\text{pred}} = \{v \mid s(i,v) > T_a'\}.$$

Since most category-attribute pairs have exclusive values, meaning that each product can have at most one value for a given attribute, we focus on the top-1 predicted value in this work. The output can be further simplified as follows:

$$v_a^{\text{pred}} = \begin{cases} \arg\max_{v\in\mathcal{V}_a} s(i,v) & \text{if } \max_{v\in\mathcal{V}_a} s(i,v) > T_a' \\ \textit{null} & \text{otherwise} \end{cases}.$$

The inference process is illustrated on the right side of Figure 3, demonstrating how candidate value embeddings from the attribute taxonomy are pre-computed and stored offline to enhance efficiency. During online inference, the item profile is encoded into an item embedding $I$, which is then compared against groups of candidate value embeddings for various attributes. In this example, the predictions for $a_3$ and $a_5$ are determined to be empty because the highest-scoring value for these attributes is the null value.

## 4 Experiment Settings

### 4.1 Datasets

To evaluate PAVI under the settings described in §3.1, we benchmark our proposed method against baselines using both proprietary and public datasets with normalized values.[2] Table 2 presents statistics of the attribute taxonomies and the datasets.

**Ecom-PAVI**. This dataset, derived from a real-world e-commerce platform, is designed to evaluate the scalability and generalization of PAVI methods. The attribute taxonomy in the e-commerce platform comprises 8,803 product categories, 26,645 category-attribute pairs, and 6.3 million category-attribute-value tuples. For our experiments, we sampled 1 million products for training, 10,000 for validation, and 10,000 for testing, ensuring that the samples span different time periods to reflect real-world scenarios. To ensure data quality, annotators manually verified the assigned product categories, discarded incorrectly categorized products, and selected the corresponding attribute-value pairs from the taxonomy as the ground truth.

**WDC-PAVE** (Brinkmann et al., 2024). This dataset consists of products distributed across 5 categories. The training set includes 1,066 products and 8,832 product-attribute pairs, of which 3,973 have null values. The test set contains 354 products and 2,937 product-attribute pairs, with 1,330 null pairs.

[2] Other popular benchmarks such as AE-110k (Xu et al., 2019) and MAVE (Yang et al., 2022) provide only unnormalized values as spans extracted from product profiles, making them unsuitable for our experiments.

Table 2: Statistics of the attribute taxonomies and dataset splits from Ecom-PAVI and WDC-PAVE. "CA Pairs" refers to category-attribute pairs, "CAV Tuples" denotes category-attribute-value tuples, "PA Pairs" represents product-attribute pairs, and "Null Pairs" indicate product-attribute pairs with null values. "Excl." refers to the test split excluding measurement attributes.

(a) Statistics of the attribute taxonomies.

| Statistic | Ecom | WDC |
|---|---|---|
| # Categories | 8,803 | 5 |
| # Attributes | 3,326 | 24 |
| # CA Pairs | 26,645 | 37 |
| # CAV Tuples | 6,302,220 | 2,297 |

(b) Statistics of the datasets.

| Statistic | Ecom-PAVI | | | WDC-PAVE | | |
|---|---|---|---|---|---|---|
| | Train | Valid | Test | Train | Test | Excl. |
| # Products | 809,528 | 81,699 | 85,024 | 1,066 | 354 | 354 |
| # PA Pairs | 3,584,462 | 358,582 | 458,954 | 8,832 | 2,937 | 2,285 |
| # Null Pairs | 2,345,577 | 228,534 | 272,285 | 3,973 | 1,330 | 916 |

Table 3: Confusion matrix comparing labeled value set with predicted value and their corresponding outcomes.

| Label | Prediction | Outcome |
|---|---|---|
| $\varnothing$ | $\varnothing$ | True Negative (TN) |
| $\varnothing$ | $v$ | False Positive (FP) |
| $\mathcal{V}$ | $v \in \mathcal{V}$ | True Positive (TP) |
| $\mathcal{V}$ | $\varnothing$ | False Negative (FN) |
| $\mathcal{V}$ | $v' \notin \mathcal{V}$ | FP & FN |

We conduct two evaluations: the first on the original test set, which includes all attributes, and the second on a test split that excludes measurement attributes to focus on tasks not requiring complex reasoning for unit conversion.

### 4.2 Metrics

Since most attributes in the taxonomy are exclusive (i.e., each product can have at most one value per attribute), we evaluate PAVI methods using micro-averaged Precision@1, Recall@1, and F1@1 scores.

For each attribute, the ground truth is a set of values $\mathcal{V}$ from the taxonomy. If the ground truth is empty ($\varnothing$), a correct prediction (True Negative, TN) occurs when the model also predicts an empty set; otherwise, it is a False Positive (FP). When the ground truth is not empty, the model's top-1 output is a True Positive (TP) if it matches any ground truth value. Predicting an empty set in this case results in a False Negative (FN), while mismatched predictions are both False Positives (FP) and False Negatives (FN), as it simultaneously introduces an error and misses the correct value.[3] Table 3 summarizes these outcomes. Final precision, recall, and F1 scores are computed by aggregating TP, FP, and FN counts across the dataset for a comprehensive performance evaluation.

[3]In prior work (Shinzato et al., 2023), metrics did not account for the FP case, and FP & FN cases were counted as FP only. We adopt more stringent metrics.

### 4.3 Baselines

We evaluate our retrieval-based method TACLR against classification and generation baselines.[4] For implementation details, refer to Appendix A.

**BERT-CLS**. This baseline frames PAVI as a multi-label classification task, treating each category-attribute-value tuple as an independent label. The model is fine-tuned to predict matches, with label masking applied to exclude irrelevant labels for each category, following (Chen et al., 2022). The model outputs a probability distribution over values and selects the highest probability value for each attribute. If no probability exceeds a specified threshold, the prediction is set to be empty.

**LLMs**. For generation-based baselines, we utilize state-of-the-art open-source LLMs, including Llama3.1-7B (Llama Team, 2024) and Qwen2.5-7B (Qwen Team, 2024). These models are initially evaluated in zero-shot and few-shot settings using a template adapted from (Brinkmann et al., 2024), which incorporates the category, attribute, and product profile along with detailed value normalization guidelines. We also fine-tune the LLMs on task-specific data to predict attribute values in JSON format. A greedy decoding strategy is applied to ensure reproducibility.

## 5 Results

### 5.1 Main Results

Table 4 presents the performance comparison between our retrieval-based method TACLR and classification- and generation-based baselines on Ecom-PAVI and WDC-PAVE. On Ecom-PAVI, TACLR achieves the highest F1 score of 86.2%, surpassing the fine-tuned Llama3.1, which obtains an F1 score of 84.7%. Notably, TACLR excels in

[4]Extraction-based baselines, such as NER or QA models, are widely used for PAVE, but they are excluded from our comparison due to the lack of a standard normalization method, which makes fair evaluation challenging.

Table 4: Performance comparison of different methods on Ecom-PAVI and WDC-PAVE. "F1 Excl." refers to the F1 score calculated without measurement attributes (e.g., width and height), which require unit conversion reasoning.

| Paradigm | Method | Ecom-PAVI | | | WDC-PAVE | | | |
|---|---|---|---|---|---|---|---|---|
| | | Precision | Recall | F1 Score | Precision | Recall | F1 Score | F1 Excl. |
| Classification | BERT-CLS | 50.9 | 50.1 | 50.5 | 68.9 | 12.0 | 20.5 | 23.4 |
| Generation | Llama3.1 (zero-shot) | 29.1 | 46.2 | 35.7 | 56.6 | 60.8 | 58.6 | 64.6 |
| | Llama3.1 (few-shot) | 31.0 | 51.1 | 38.6 | **76.0** | **74.1** | **75.0** | 79.0 |
| | Llama3.1 (fine-tune) | **86.9** | 82.7 | 84.7 | 57.7 | 60.4 | 59.0 | 64.5 |
| | Qwen2.5 (zero-shot) | 42.7 | 55.7 | 48.4 | 51.9 | 60.3 | 55.8 | 60.8 |
| | Qwen2.5 (few-shot) | 45.8 | 58.6 | 51.4 | 72.2 | 72.3 | 72.2 | 76.2 |
| | Qwen2.5 (fine-tune) | 84.5 | 79.1 | 81.7 | 54.1 | 60.0 | 56.9 | 61.7 |
| Retrieval | TACLR | 85.4 | **87.1** | **86.2** | 74.3 | 70.9 | 72.6 | **80.3** |

Table 5: Inference efficiency comparison on Ecom-PAVI (Throughput in samples/second).

| Method | Time (ms) | Throughput |
|---|---|---|
| BERT-CLS | 8.6 | 930 |
| Llama3.1 (zero-shot) | 101.3 | 80 |
| Llama3.1 (few-shot) | 124.8 | 64 |
| Qwen2.5 (zero-shot) | 84.0 | 95 |
| Qwen2.5 (few-shot) | 98.4 | 81 |
| TACLR | 12.7 | 630 |

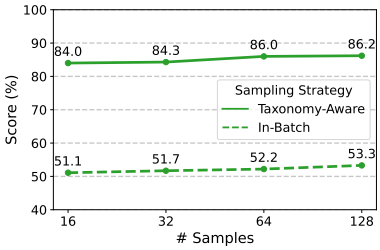



Figure 4: Comparison of negative sampling strategies with increasing number of samples.

recall, achieving 87.1% compared to Llama3.1's 82.7%. On WDC-PAVE, TACLR achieves the highest F1 Excl. score of 80.3%, which excludes measurement attributes requiring reasoning ability for unit normalization. This result highlights TACLR's effectiveness and robustness in addressing general PAVI across diverse datasets.

The classification-based method, BERT-CLS, shows the weakest performance on both datasets. It achieves an F1 score of 50.5% on Ecom-PAVI, but its performance drops drastically on WDC-PAVE, where it only attains an F1 score of 20.5%. This underscores the limitations of classification-based approaches in generalization, including their inability to adapt to OOD values.

Among the generation-based methods, few-shot and fine-tuning consistently improve performance over zero-shot settings. For example, on Ecom-PAVI, Llama3.1 achieves F1 scores of 35.7%, 38.6%, and 84.7% in zero-shot, few-shot, and fine-tuned settings, respectively. Similarly, Qwen2.5 achieves F1 scores of 48.4%, 51.4%, and 81.7% in the corresponding settings. On WDC-PAVE, however, fine-tuned LLMs exhibit weaker generalization due to the scarcity of training data. Few-shot learning proves more robust in this dataset, achieving an F1 score of 75.0% for Llama3.1 and 72.2% for Qwen2.5.

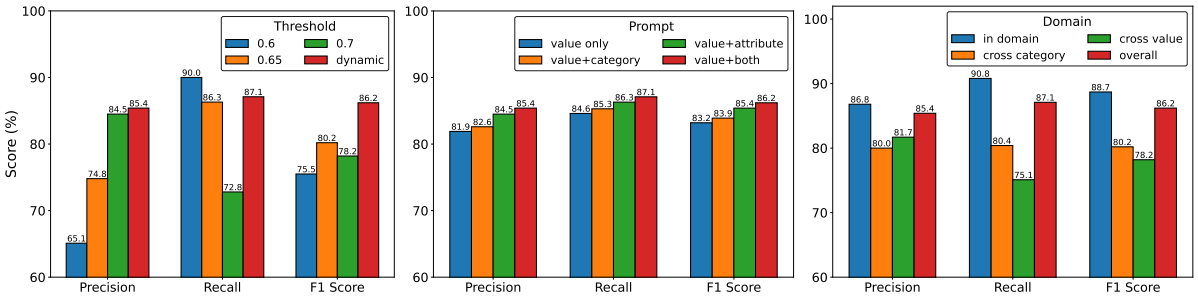
### 5.2 Inference Efficiency

Table 5 compares the inference efficiency of PAVI methods under identical evaluation conditions. We employed a naive PyTorch implementation without speed optimizations and used the largest batch size that avoids out-of-memory errors. All experiments were conducted on a machine equipped with one NVIDIA V100 GPU.

TACLR achieves a strong balance between performance and efficiency, with an inference time of 12.7 ms and a throughput of 630 samples per second. In contrast, generation-based methods, such as Llama3.1 (few-shot) and Qwen2.5 (few-shot), exhibit significantly longer inference times (124.8 ms and 98.4 ms, respectively) and lower throughputs (64 and 81 samples per second), highlighting the computational overhead of LLMs in high-load scenarios. While BERT-CLS delivers the fastest inference time and highest throughput, its inability to handle OOD values and limited capacity restrict its effectiveness in practical applications.

### 5.3 Analysis

**Impact of Taxonomy-Aware Negative Sampling.** Figure 4 compares the proposed taxonomy-aware sampling (§3.3) with in-batch sampling across different sample sizes. As the number of sampled




(a) Comparison of inference thresholds. (b) Comparison of prompt templates. (c) Comparison on data domains.

Figure 5: Performance analysis across inference thresholds, prompt templates, and data domains.

values increases, F1 score consistently improve, corroborating findings from (Chen et al., 2020). With in-batch sampling as the baseline, the model achieves an F1 score of 53.3% with a sample size of 128. In contrast, taxonomy-aware sampling significantly outperforms this baseline, with an F1 score improving from 84.0% to 86.2% as the sample size increases from 16 to 128. These results highlight the superiority of taxonomy-aware sampling, which leverages the structure of attribute taxonomy to generate more challenging negative examples, enhancing the model's recognition capabilities.

**Comparison of Dynamic and Static Thresholds.** Figure 5a evaluates the dynamic, learnable thresholds (§3.4) against the best static thresholds of 0.6, 0.65, and 0.7, which were chosen based on their performance on the validation set. The dynamic threshold achieves the highest F1 score of 86.2%, surpassing the static thresholds, which yield F1 scores of 75.5%, 80.2%, and 78.2%, respectively. Static thresholds exhibit a clear trend: as the threshold increases, precision rises (from 65.1% to 84.5%) while recall diminishes (from 90.0% to 72.8%). In contrast, the dynamic threshold balances precision (85.4%) and recall (87.1%) effectively, eliminating the need for extensive hyperparameter tuning across category-attribute pairs. This adaptability makes dynamic thresholds a practical choice.

**Performance Gains from Context-Rich Prompts.** The influence of varying value prompt templates on the PAVI task is shown in Figure 5b. Using only the value as a prompt achieves an F1 score of 83.2%. Adding category information raises the F1 score to 83.9%, while incorporating attribute information further improves it to 85.4%. The most comprehensive template, combining category, attribute, and value information (e.g., "A {category} with {attribute} being {value}"), achieves the highest F1 score of 86.2%. These results are consistent with prior work (Radford et al., 2021), highlighting that context-rich prompts enhance the model's discriminative performance.

**Zero-Shot Generalization Across Data Domains.** Figure 5c presents the results of zero-shot transfer experiments, evaluating the model's generalization across unseen categories and values. The in-domain split achieves an F1 score of 88.7%, while performance decreases for cross-category and cross-value splits, which attain F1 scores of 80.2% and 78.2%, respectively. These declines reflect the inherent challenges of adapting to dynamic attribute taxonomies in OOD domains. Nevertheless, the overall F1 score of 86.2% demonstrates the robust generalization capabilities of TACLR.

## 6 Conclusion

In this work, we present TACLR, a novel approach for retrieval-based PAVI. By formulating PAVI as an information retrieval problem, TACLR enables the inference of implicit values, generalization to OOD values, and the production of normalized outputs. Building on this framework, TACLR employs contrastive training with taxonomy-aware sampling and adaptive inference with dynamic thresholds to enhance retrieval performance and scalability.

Comprehensive experiments on proprietary and public datasets demonstrated TACLR's superiority over classification- and generation-based baselines. Notably, TACLR achieved an F1 score of 86.2% on the large-scale Ecom-PAVI dataset. Efficiency analysis further highlighted its advantage, achieving significantly faster inference speeds than generation-based methods. Beyond these experimental results, TACLR has been successfully deployed on a real-world e-commerce platform, processing millions of product listings daily and seamlessly adapting to dynamic attribute taxonomies, making it a practical solution for large-scale industrial applications.

## 7 Limitations

While TACLR demonstrates effectiveness in processing textual product profiles, it does not currently leverage multimodal information, such as images or videos. Multimodal data could provide valuable complementary context for attributes that are challenging to infer from text alone (e.g., visual attributes like color or texture). Incorporating multimodal capabilities may further enhance the model's ability to identify attribute values with greater accuracy and comprehensiveness.

## References


Ansel Blume, Nasser Zalmout, Heng Ji, and Xian Li. 2023. Generative models for product attribute extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 575–585, Singapore. Association for Computational Linguistics.

Alexander Brinkmann, Nick Baumann, and Christian Bizer. 2024. Using llms for the extraction and normalization of product attribute values. In *European Conference on Advances in Databases and Information Systems*, pages 217–230. Springer.

Kang Chen, Qing Heng Zhang, Chengbao Lian, Yixin Ji, Xuwei Liu, Shuguang Han, Guoqiang Wu, Fei Huang, and Jufeng Chen. 2024. IPL: Leveraging multimodal large language models for intelligent product listing. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 697–711, Miami, Florida, US. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

Wei-Te Chen, Keiji Shinzato, Naoki Yoshinaga, and Yandi Xia. 2023. Does named entity recognition truly not scale up to real-world product attribute extraction? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 152–159, Singapore. Association for Computational Linguistics.

Wei-Te Chen, Yandi Xia, and Keiji Shinzato. 2022. Extreme multi-label classification with label masking for product attribute value extraction. In *Proceedings of the Fifth Workshop on e-Commerce and NLP (ECNLP 5)*, pages 134–140, Dublin, Ireland. Association for Computational Linguistics.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 657–668, Online. Association for Computational Linguistics.

Yifan Ding, Yan Liang, Nasser Zalmout, Xian Li, Christan Grant, and Tim Weninger. 2022. Ask-and-verify: Span candidate generation and verification for attribute value extraction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 110–110, Abu Dhabi, UAE. Association for Computational Linguistics.

Shen Gao, Zhaochun Ren, Yihong Zhao, Dongyan Zhao, Dawei Yin, and Rui Yan. 2019. Product-aware answer generation in e-commerce question-answering. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, page 429–437, New York, NY, USA. Association for Computing Machinery.

Junshi Huang, Si Liu, Junliang Xing, Tao Mei, and Shuicheng Yan. 2014. Circle & search: Attribute-aware shoe retrieval. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 11(1).

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2024. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.* Just Accepted.

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*, pages 8489–8502, Online. Association for Computational Linguistics.

Anant Khandelwal, Happy Mittal, Shreyas Kulkarni, and Deepak Gupta. 2023. Large scale generative multimodal attribute extraction for E-commerce attributes. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 305–312, Toronto, Canada. Association for Computational Linguistics.

Ashish Kulkarni, Kartik Mehta, Shweta Garg, Vidit Bansal, Nikhil Rasiwasia, and Srinivasan Sengamedu. 2019. Productqna: Answering user questions on e-commerce product pages. In *Companion Proceedings of The 2019 World Wide Web Conference*, WWW '19, page 354–360, New York, NY, USA. Association for Computing Machinery.

Yanzeng Li, Bingcong Xue, Ruoyu Zhang, and Lei Zou. 2023. AtTGen: Attribute tree generation for real-world attribute joint extraction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2139–2152, Toronto, Canada. Association for Computational Linguistics.

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 *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 3262–3270, New York, NY, USA. Association for Computing Machinery.

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*, abs/1907.11692.

Meta Llama Team. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Alessandro Magnani, Feng Liu, Min Xie, and Somnath Banerjee. 2019. Neural product retrieval at walmart.com. In *Companion Proceedings of The 2019 World Wide Web Conference*, WWW '19, page 367–372, New York, NY, USA. Association for Computing Machinery.

Athanasios N. Nikolakopoulos, Swati Kaul, Siva Karthik Gade, Bella Dubrov, Umit Batur, and Suleiman Ali Khan. 2023. Sage: Structured attribute value generation for billion-scale product catalogs. *Preprint*, arXiv:2309.05920.

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.

Alibaba Qwen Team. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Kalyani Roy, Pawan Goyal, and Manish Pandey. 2021. Attribute value generation from product title using language models. In *Proceedings of the 4th Workshop on e-Commerce and NLP*, pages 13–17, Online. Association for Computational Linguistics.

Kassem Sabeh, Mouna Kacimi, Johann Gamper, Robert Litschko, and Barbara Plank. 2024a. Exploring large language models for product attribute value identification. *Preprint*, arXiv:2409.12695.

Kassem Sabeh, Robert Litschko, Mouna Kacimi, Barbara Plank, and Johann Gamper. 2024b. An empirical comparison of generative approaches for product attribute-value identification. *Preprint*, arXiv:2407.01137.

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)*, pages 227–234, Dublin, Ireland. Association for Computational Linguistics.

Keiji Shinzato, Naoki Yoshinaga, Yandi Xia, and Wei-Te Chen. 2023. A unified generative approach to product attribute-value identification. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6599–6612, Toronto, Canada. Association for Computational Linguistics.

Changfeng Sun, Han Liu, Meng Liu, Zhaochun Ren, Tian Gan, and Liqiang Nie. 2020. Lara: Attribute-to-feature adversarial learning for new-item recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, WSDM '20, page 582–590, New York, NY, USA. Association for Computing Machinery.

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 *Proceedings of the ACM Web Conference 2022*, WWW '22, page 2978–2988, New York, NY, USA. Association for Computing Machinery.

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.

Qifan Wang, Li Yang, Jingang Wang, Jitin Krishnan, Bo Dai, Sinong Wang, Zenglin Xu, Madian Khabsa, and Hao Ma. 2022. SMARTAVE: Structured multimodal transformer for product attribute value extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 263–276, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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.

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 (Volume 1:*

*Long Papers)*, pages 4694–4705, Online. Association for Computational Linguistics.

Li Yang, Qifan Wang, Jingang Wang, Xiaojun Quan, Fuli Feng, Yu Chen, Madian Khabsa, Sinong Wang, Zenglin Xu, and Dongfang Liu. 2023. MixPAVE: Mix-prompt tuning for few-shot product attribute value extraction. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9978–9991, Toronto, Canada. 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 *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, WSDM '22, page 1256–1265, New York, NY, USA. Association for Computing Machinery.

Danqing Zhang, Zheng Li, Tianyu Cao, Chen Luo, Tony Wu, Hanqing Lu, Yiwei Song, Bing Yin, Tuo Zhao, and Qiang Yang. 2021. Queaco: Borrowing treasures from weakly-labeled behavior data for query attribute value extraction. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, page 4362–4372, New York, NY, USA. Association for Computing Machinery.

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.

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)*, pages 2129–2139, Online. Association for Computational Linguistics.

Henry Zou, Gavin Yu, Ziwei Fan, Dan Bu, Han Liu, Peng Dai, Dongmei Jia, and Cornelia Caragea. 2024. EIVEN: Efficient implicit attribute value extraction using multimodal LLM. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, pages 453–463, Mexico City, Mexico. Association for Computational Linguistics.

## A Implementation Details

We utilize pre-trained RoBERTa-base models (Liu et al., 2019; Cui et al., 2020), augmented with a linear projection layer, to encode both the item profile and the value prompt. The embedding dimension is set to 256. For each product-attribute pair, we sample up to 128 values, which include a null value, an optional positive value (no positive value is selected when absent for the product-attribute), and negative values sampled from the same category-attribute pair. The temperature parameter for contrastive learning is set to 0.05. The models is fine-tuned using the AdamW optimizer with a batch size of 32 and a learning rate of 2e-5, over a maximum of 5 epochs. Hyperparameters and the best model checkpoints are selected based on the F1 score on the validation set.

## B Deployment

The proposed TACLR has been successfully integrated into key functionalities of an e-commerce platform, including product listing, search, recommendation, and price estimation. The system is designed to be highly scalable and efficiently process millions of products daily.

In the product listing process, TACLR automatically identifies attribute-value pairs from user-provided titles and descriptions, significantly reducing manual effort and errors while improving the quality of structured information.

For product search, improved structured information directly enhances lexical retrieval, leading to more accurate matching with user queries. Meanwhile, it enriches product features, consequently enhancing personalized recommendations.

In the area of price estimation, TACLR identifies key attributes that influence pricing, leading to more accurate price predictions. This provides both sellers and buyers with reliable, market-aligned information.