E-commerce Product Attribute Value Validation and Correction Based on Transformers

Le Yu
The Home Depot
Atlanta, Georgia, USA

Haozheng Tian
The Home Depot
Atlanta, Georgia, USA

Yun Zhu
The Home Depot
Atlanta, Georgia, USA

Simon Hughes
The Home Depot
Chicago, Illinois, USA

Aleksandar Velkoski
The Home Depot
Clarkston, Michigan, USA

ABSTRACT

Accurately representing product attributes is critical to e-commerce performance. Product attribute values are typically entered manually by suppliers during the product on-boarding process, and thus often contain noise and other inconsistencies. The sheer volume of products that are on-boarded makes it difficult to validate product attribute values. Therefore, establishing an automatic and scalable approach to attribute value validation and correction is a crucial step toward enhancing the customer's online shopping experience. In addition, it enables more effective downstream Machine Learning systems (e.g., Search and Recommendation) by virtue of improved data quality. Most existing methods split validation and correction into separate steps whereby researchers build disparate models that have limited transferable domain knowledge. For instance, in attribute validation, multiple algorithms are often built to account for distinct attribute types (e.g., numeric or textual attributes), making a unified solution difficult to generalize. In this paper, we propose a transformer-based approach to automatically validate product attribute values using the product profile, and recommend correct values when errors are observed. Our approach can be applied to all attribute types and adapted to a wide variety of categories. More specially, we extend the RoBERTa based natural language inference (NLI) model to the field of e-commerce product attribute value validation by comparing structured product information against the most relevant content selected from unstructured product profiles. The model treats different data types (e.g., integer, fraction, number and text) as textual inputs to address the issue of scale. Meanwhile, attribute names are concatenated in the input sequences to improve validation quality. In addition to identifying erroneous values, we use the fine-tuned model to recommend correct values for List of Value (LOV) attributes, which expedites the correction process and reduces manual effort. Insights about input pre-processing and training data creation are also explored. The application of our approach in e-commerce demonstrates not only promising results, but also superior performance.

ISIR-eCom 2022 @ WSDM-2022, Feb [isir-ecom.github.io] 25, 2022, Phoenix, AZ
© 2021 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/1122445.1122456

CCS CONCEPTS

• **Computing methodologies → Information extraction**.

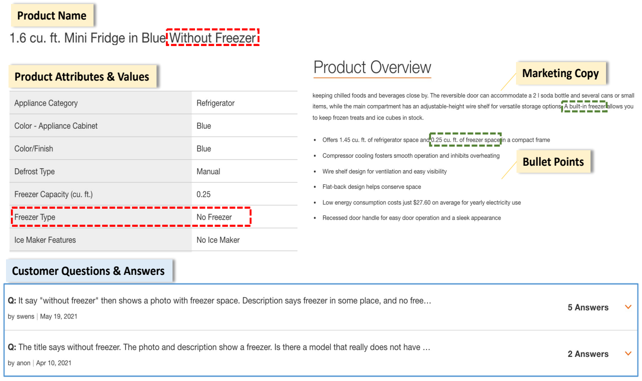
KEYWORDS

information retrieval, data validation, data correction, natural language inference, e-commerce

ACM Reference Format:
Le Yu, Haozheng Tian, Yun Zhu, Simon Hughes, and Aleksandar Velkoski. 2021. E-commerce Product Attribute Value Validation and Correction Based on Transformers. In *Proceedings of International Workshop on Interactive and Scalable Information Retrieval methods for eCommerce (ISIR-eCom) 2022 @ WSDM-2022, https://isir-ecom.github.io/ (ISIR-eCom 2022 @ WSDM-2022).* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

E-commerce platforms possess valuable product-oriented textual content, such as product names, attributes, and profiles, which provide customers with important information used in product discovery and purchase decisions. Product information is typically entered manually by suppliers during the product on-boarding process, where inevitably a certain amount of wrong or conflicting information is introduced into the system. As a result, data quality issues may confuse or even mislead customers, which ultimately harm customer's experience. For instance, Figure 1 shows the product information page for a refrigerator. The page contains conflicting information about the presence of a freezer. The data quality issue is significant enough to motivate a number of customers to post questions about the discrepancy on the QA section of the page. Besides customer's experience, downstream Machine Learning systems, such as search or recommendation engines, are also negatively affected by the discrepancy. Anomaly detection is a widely used approach to detect abnormal values among similar items within a group. While numerical anomalies can be identified by looking at value density, distributions, and histograms, it is challenging to implement similar methods to textual attributes, which dominate product attribute values. Another limitation of anomaly detection is that it often leads to the wrong conclusion, because it does not consider product-specific characteristics. For example, the weight of a sink is much heavier if it is made of stone compared to stainless steel. Therefore, the abnormally heavy weight of a stone sink is a product-specific characteristic as opposed to an anomalous attribute value. Upon observing these shortcomings, we chose to validate the attribute values for a given product against its own product profile. In other words, we compared the structured product attribute values against the unstructured product profile to identify inconsistencies.



1.6 cu. ft. Mini Fridge in Blue Without Freezer

Appliance Category	Refrigerator
Color - Appliance Cabinet	Blue
Color/Finish	Blue
Defrost Type	Manual
Freezer Capacity (cu. ft.)	0.25
Freezer Type	No Freezer
Ice Maker Features	No Ice Maker

Product Overview

keeping chilled foods and beverages close by. The reversible door can accommodate a 2 l soda bottle and several cans or small items, while the main compartment has an adjustable-height wire shelf for versatile storage options. A built-in freezer allows you to keep frozen treats and ice cubes in stock.

- Offers 1.45 cu. ft. of refrigerator space and 0.25 cu. ft. of freezer space in a compact frame
- Compressor cooling fosters smooth operation and inhibits overheating
- Wire shelf design for ventilation and easy visibility
- Flat-back design helps conserve space
- Low energy consumption costs just $27.60 on average for yearly electricity use
- Recessed door handle for easy door operation and a sleek appearance

Q: It say "without freezer" then shows a photo with freezer space. Description says freezer in some place, and no free…
by swens | May 19, 2021
5 Answers

Q: The title says without freezer. The photo and description show a freezer. Is there a model that really does not have …
by anon | Apr 10, 2021
2 Answers

Figure 1: Product information includes product name, product attributes and product overview (also known as profile). A data quality issue on an e-commerce website could cause confusion during the online shopping experience. Customers have the ability post questions about the product.

2 RELATED WORK

2.1 Attribute value validation

Attribute value validation is related to the anomaly detection task, which aims to detect data objects that deviate significantly from the norm [3]. Traditional methods are typically unsupervised, and include density-based methods (e.g., DBSCAN), distribution-based methods (e.g., HBOS), and other similar approaches more easily implemented for anomaly detection with numeric attributes. Recently, deep learning anomaly detection models [10] [2] have gained traction due to significant improvements in model performance. For instance, recurrent neural networks (RNNs) and convolutional neural networks (CNNs) are leveraged to learn feature representations and anomaly scores. However, these approaches are often limited to single-sourced features within a group, while product profile information, which we use to identify contradictions, are ignored.

Another related task is natural language inference (NLI), which helps determine the relationship between two texts [13]. Large-scale annotated NLI datasets have been collected, and existing benchmarks such as Stanford Natural Language Inference (SNLI) [1] and Multi-Genre Natural Language Inference (MultiNLI) [15] have promoted development of many distinct NLI models. With the rapid development of transformer-based models such as BERT [5], and following improved models like RoBERTa [9], XLNet [4], BART [7], pre-trained models have achieved superior performance. The taxonomy-aware semantic cleaning method [4] is implemented in the e-commerce domain, and is used to compare attribute values with the corresponding product names. The meta-learning method [14] is used to train a transformer with limited labeled data. The key limitation is that training disparate models for different data types makes it difficult to scale to thousands of categories and millions of products, a volume that is common in a real-world e-commerce systems. Another important limitation is that only a limited amount of information are utilized (e.g., product names) for attribute value validation.

2.2 Attribution value filling/correcting

A number of methods have been developed to extract attribute values from product profiles of e-commerce products. For instance, tagging-based Name Entity Recognition [19] and a Bi-directional LSTM model followed by a conditional random field (CRF) have been explored [16]. Product profile and product images are added together to train a multi-modal transformer [8] to extract product attributes. Typically, the values are extracted directly from the product profiles without any transition. As a result, the various formats that are ultimately extracted could be difficult to use in structured datasets without data standardization. In this case, the domain knowledge learned in attribute value validation has never been transferred into attribute value correction, which is an immediate next step following attribute value validation.

3 PROBLEM DEFINITION AND PRELIMINARY

3.1 Problem definition

The product displayed on e-commerce websites contains structured and unstructured textual information, as illustrated in Figure 1. Unstructured data includes paragraphs of text called Marketing Copy that describes the overall product features, as well as bullet points highlighting key product features. Meanwhile, one or more tables containing pairs of attribute names and corresponding attribute values are provided as structured data. It is not uncommon that a product has over 20 pairs of attributes and values. Product attributes usually fall into three major types: numeric, List of Values(LOV), and textural. Integers, fractions, and decimals are typical numeric values, e.g., "Product Weight (lb.) is 8". LOV contains a set of predefined values that suppliers need to choose from during product on-boarding, e.g., "Assembled Required" has two optional values as "Yes" or "No". Over 50% of the attributes on our site are LOV values. Textural values are those that suppliers have the freedom to type in, and "Manufacturer Warranty" is a typical textural attribute.

The main goal of the task is to detect inconsistencies between the product's structured attribute-value pairs and its unstructured profile. After discovering inconsistencies, it is possible to provide values correction based on the profile. For example, given the inputs,

- **Attribute Name:** *Sink Shape*
- **Attribute Value:** *Rectangular*
- **Product info:** "Somerton 60 in. Double Bathroom Vanity w/ 4 Drawers 2 Shelves 4 Doors; Granite Top; Antique White, Oval white porcelain under mount sink..."

We want to be able to recognize that *Rectangular* is not the correct value of *Sink Shape* for this particular product. Instead, *Oval* is the correct one. Formally, *Given a product* P*, corresponding Product Name* N*, Marketing Copy* M*, a set of Bullet Points* $B = \{b_1, ..., b_k\}$*, and a set of pairs of attributes/values* $A = \{(a_1, v_1), ..., (a_j, v_j)\}$*, detect the inconsistent value* v_i *and* $i \in [1, j]$*. For a LOV attribute, given the pre-defined values options* $O = \{o_1, ..., o_l\}$ *and if the current value* o_q *and* $q \in [1, l]$ *is detected as incorrect, provide the most likely value* o_r *and* $r \in [1, l]$ *based on the product profile.*

3.2 Natural Language Inference

NLI is a classification task used to establish the relationship between a premise and hypothesis as either entailment, contradiction, or neutral. Recent transformer-based neural network models like BERT [5] have taken the NLP landscape by a storm, outperforming traditional approaches on several key tasks. In addition, XLNet [4] and RoBERTa [9] demonstrated improved performance in NLI tasks. Our method is based on the RoBERTa model structure. However, there are limitations that must be addressed before applying the model to the field of e-commerce.

Limitations: (1) Fine-tuning a transformer-based model requires a large high-quality labeled dataset, which might result in excessive labeling costs. (2) Previous work on the product attribute value validation focuses on either building individual models for different attributes, or applying models to merely a small subset of attributes, which is neither scalable nor appropriate for accommodating a wide range of categories and attribute types. (3) The fact that transformer-based models has a length limit of input makes it hard to consume the lengthy description of products. Alternative methods, such as truncating text into smaller parts, increase computational cost and difficulty of deployment. (4) Knowledge learnt in NLI model is rarely reused in value correction, since a separate model (LSTM, NER, etc.) is built to do the job.

Key ideas of our solution: To address these limitations, we propose an end-to-end solution that contains: (1) High-quality training data selection strategies based on the pre-trained model and active learning to include high-yield training samples while reducing labeling cost; (2) Improved input structure by concatenating the attribute name and value as one input to improve the scalability to different attribute types and categories. Selecting highly relevant content from the product profile to address input length limitation and improve validation performance; (3)Transferable knowledge learned from value validation reused for value correction, reducing development effort in real industrial applications.

4 METHODS

In this section, we first introduce the structure of the value validation models, and then the strategies of generating the training data. Finally, we introduce the re-use of knowledge for value correction.

4.1 Overall Architecture: RoBERTa based model

As shown in Figure 2, the overall validation architecture is an NLI model based on the RoBERTa structure. The model detects whether a pair $[(a_i, v_i), (N, M, B)]$ is aligned given a product (i.e., whether a pair of product attribute a_i and product value v_i from A is true (entailment), false (contradiction), or undetermined (neutral) given the information from product profile including Product Name N, Marketing Copy$M$ and Bullet Points B). Let T_{ki} be the top k sentence chunks relevent to (a_i, v_i) (See Section 4.1.1). The raw input sequence S of the model is the concatenation of a_i, v_i and T_{ki}:

$$S = concat(< s >, a_i, v_i, < /s >, T_{ki}, < /s >) \tag{1}$$

where a special token $< s >$ is used as an indicator for a classification task, and $< /s >$ is used as a separation token between two input sequences. For the v-th token in the sequence, an embedding vector e_v is the summation of three embedding vectors with the same dimension:

$$e_v = e_v^{Tokenizer} + e_v^{Segment} + e_v^{Attention} \tag{2}$$

where a built-in BPE tokenizer [12] is applied to get the tokenizations. Segment tokenization uses binary numbers to separate the first and second sequence. Attention masks are used to inform the model the locations of both the padding and the text. We care most about contradiction outputs because it indicates the existence of a discrepancy between the input sequences, and further signals the need of correction.

4.1.1 *Top-K relevant content selection.*

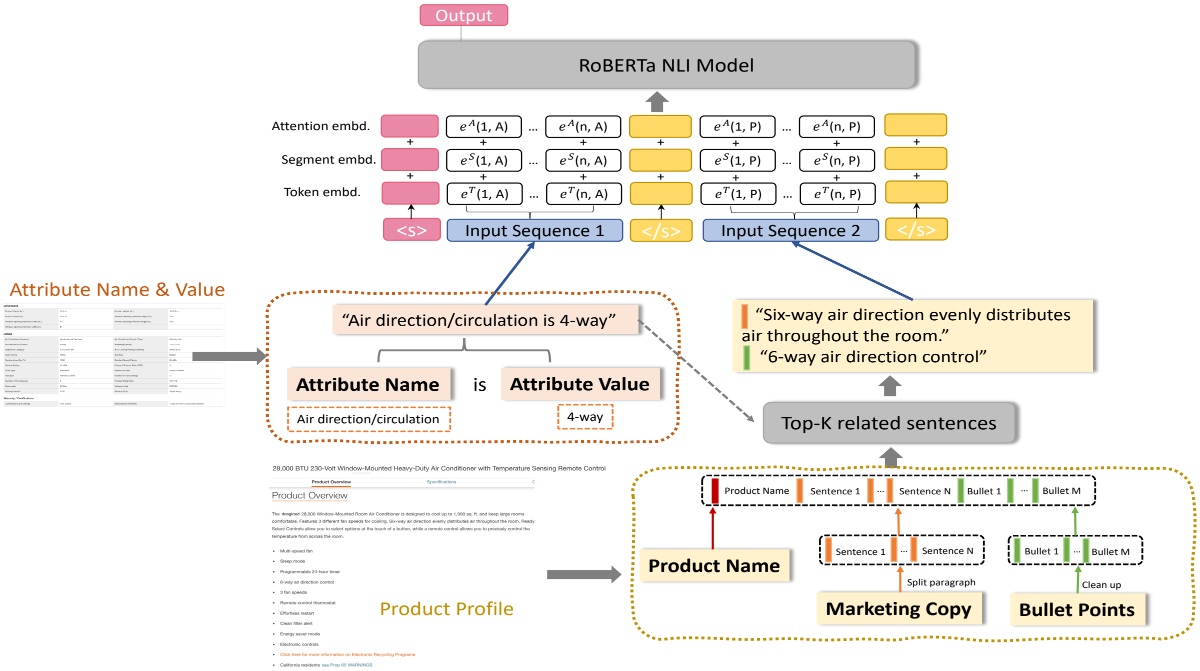
Considering that the total length of the product profile could be much longer than the regular model input length limit of 512 tokens, and the position of the relevant content varies, we apply a pre-processing step to first clean the product profile and then select at most the top K relevant chunks as the second input sequence of the NLI model.

Marketing Copy typically contains a collection of descriptive sentences, and its length varies. We split the copy into chunks, while avoiding splitting units of measurement (e.g., oz., ft., sq.ft.) as these are important characteristics of numeric attributes. Bullet Points are already represented as short text, so we only remove irrelevant content like web links (e.g., URLs linking to product warranties and manuals). Product names are represented as concise text containing at most 120 characters in length. Therefore, no cleanup is needed. The cleanup process also includes some format standardization: (1) fraction to decimal conversion (e.g., 1/5 to 0.2); (2) decimal precision conversion (e.g., 0.200 to 0.2); (3) unit of measurement difference detection and conversion (e.g., 1ft. to 12in.).

After the cleanup process, we apply term frequency–inverse document frequency (TF-IDF) to get the feature matrix of sentence chunks from the product profile $P = [p_1, p_2, ..., p_n] \in \mathbb{R}^{nxk}$ and the vector matrix of a pair of attribute name and value $V = [v_1] \in \mathbb{R}^{1xk}$, where n is the total number of sentence chunks captured, and k is the size of a single vector. Then we use the linear kernel to calculate pairwise distances of the sentence chunks and the pair of attribute name and value $k(P, V) = P^T V \in \mathbb{R}^{nx1}$. Based on the pairwise distances, the top-k most relevant chunks are concatenated as the input sequence 2 which is sent into the NLI model as shown in Figure 2. The input sequence 1 combines the pair of attribute name and value. The above steps are repeated for each pair of attribute name and value to get the corresponding most relevant product info. If no relevant chunks are found, the product name itself is used.

4.2 Training data preparation

As mentioned previously, fine-tuning RoBERTa NLI model to adapt the domain knowledge in e-commerce requires a sufficient training dataset, so we automatically generate most of the training data by using distant supervision [6] and active learning[11]. Only a small set of complex examples is labeled by humans to reduce the total costs. In reality, a large portion of the attributes and values are aligned with their product profiles, and the pre-trained NLI model(e.g., RoBERTa large mnli[9]) is better at detecting these entailed pairs than contradictory ones. Therefore, we generate entailed pairs by selecting high-confidence entailed pairs outputted from the pre-trained NLI model. For each entailed pair, we randomly


Output
RoBERTa NLI Model
Attention embd.
Segment embd.
Token embd.
$e^A(1, A)$ ... $e^A(n, A)$
$e^S(1, A)$ ... $e^S(n, A)$
$e^T(1, A)$ ... $e^T(n, A)$
$e^A(1, P)$ ... $e^A(n, P)$
$e^S(1, P)$ ... $e^S(n, P)$
$e^T(1, P)$ ... $e^T(n, P)$
<s>
Input Sequence 1
</s>
Input Sequence 2
</s>
Attribute Name & Value
"Air direction/circulation is 4-way"
Attribute Name
is
Attribute Value
Air direction/circulation
4-way
"Six-way air direction evenly distributes air throughout the room."
"6-way air direction control"
Top-K related sentences
Product Name
Sentence 1
...
Sentence N
Bullet 1
...
Bullet M
Split paragraph
Clean up
Product Name
Marketing Copy
Bullet Points
Product Profile

28,000 BTU 230-Volt Window-Mounted Heavy-Duty Air Conditioner with Temperature Sensing Remote Control

Product Overview | Specifications

Product Overview

The desgined 28,000 Window-Mounted Room Air Conditioner is designed to cool up to 1,900 sq. ft. and keep large rooms comfortable. Features 3 different fan speeds for cooling. Six-way air direction evenly distributes air throughout the room. Ready Select Controls allow you to select options at the touch of a button, while a remote control allows you to precisely control the temperature from across the room.

- Multi-speed fan
- Sleep mode
- Programmable 24-hour timer
- 6-way air direction control
- 3 fan speeds
- Remote control thermostat
- Effortless restart
- Clean filter alert
- Energy saver mode
- Electronic controls
- Click here for more information on Electronic Recycling Programs
- California residents see Prop 65 WARNINGS

Figure 2: Overview of Product Attribute value validation architecture. (1) Input sequence 1 comes from both attribute names and attribute values. (2) Input sequence 2 is the top related content selected from product profile. (3) Two input sequences are passed through RoBERTa based Natural Language Inference model.

apply one of the following methods to generate a contradictory pair: (1) replace the attribute value with a randomly selected value (except itself) from the vocabulary of itself (e.g., replace "100" to "80" in the vocabulary of "Product Weight (lb.)"); (2) replace the attribute value with a randomly selected value from the vocabulary of another attribute (e.g., replace product weight value "100" to "Red" in the vocabulary of "Color"); (3) replace the attribute value with a randomly selected n-gram word(s) ($n \leq 3$) from Bullet Points. As the attribute distribution among products and taxonomies is unbalanced, and similar products, which only vary in color, size, etc, could be selected as training data together, we apply active learning - diversity selection[11] to select more diverse examples to get better coverage during automatic generation and human labeling processes. More formally, for $X \subseteq Y$, the property is $f(X+v) - f(x) \geq f(Y+v) - f(Y)$. When these functions represent a notion of diversity, finding the subset of examples that maximize these functions corresponds to finding a minimally redundant set. We use textual featurization (e.g., n-gram TF-IDF or some pre-trained neural embedding) of the product info to select diverse examples for training data preparation.

4.3 Classification based value correction

Instead of building another model for attribute value correction, the entailment labels could help locate the potential correct values by re-using the knowledge learned in NLI model. Inspired by the zero-shot classification task [17], we apply our fine-tuned RoBERTa NLI model as a classification model to recommend a potential correct LOV value o_r from $O = \{o_1, ..., o_l\}$, $q \in [1, l]$ and $r \neq q$, where o_q is the current value and identified as incorrect. As value options O and the length l are different for different LOV attributes, the regular classification task which has fixed classes is not feasible in our case. Zero-shot classification takes the text and a class at each time as the input and output "Yes" or "No" to indicate whether or not the input class is the true label for the text. Similarly, in our case, we replace the current LOV value with an option from the LOV value list and pass it into the fine-tuned RoBERTa NLI model with the original selected top relevant content.

Specially, as shown in Figure 3, we use the example from Figure 2 to explain it in more detail. The attribute "Air direction/circulation" is a LOV attribute and has 5 value options as "1-way; 2-way; 4-way; 6-way; 8-way". The current value "4-way" is predicted as contradictory to the product profile - "Six-way air direction evenly distributes air throughout the room...", we then replace the value "4-way" with an option at a time from the list and get the classification label for each option. The entailment prediction with the highest confidence score would be the candidate value for the LOV attribute. In the real application, we apply a threshold to the option with the highest confidence to further improve the correction performance and only recommend values for LOV attributes with less than 10 options, to reduce computational costs.

We adopt the assumption that the product profile provides true information and the LOV value list is exhaustive. Specifically, in case of a discrepancy, the true value can be found in the product profile and should already be included in the list.

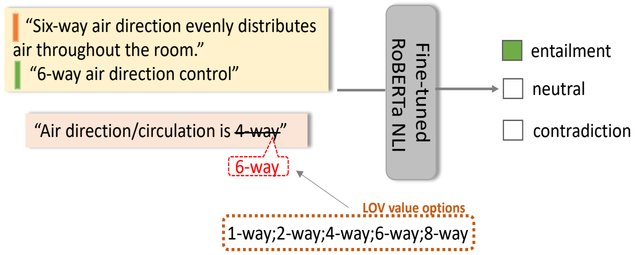

"Six-way air direction evenly distributes air throughout the room."
"6-way air direction control"
"Air direction/circulation is ~~4-way~~"
6-way
LOV value options
1-way;2-way;4-way;6-way;8-way
Fine-tuned RoBERTa NLI
entailment
neutral
contradiction

Figure 3: An example to illustrate how to use RoBERTa NLI for LOV value correction.

5 EXPERIMENTS SETUP AND RESULTS

We now present the training dataset used to fine-tune the RoBERTa NLI model, the parameter setup, and the evaluation results.

5.1 Dataset and performance

The training data has 8247 examples over 100 categories and 1023 unique attributes, and it includes 2658 contradictory examples, 3091 neutral examples and 2517 entailed examples. We fine-tune the RoBERTa-large-mnli model with the max input length of 256 tokens to reduce the computational resource cost in both training and deployment steps, and at most $k(k = 5)$ chunks selected as the relevant content with format standardization as described in Section 4.1.1. The model is fine-tuned on Telsa P100 GPUs. We compare the fine-tuned model with the pre-trained RoBERTa large mnli model in different input sequence settings:

- Only title is used as the input selected from product profile.
- Top 5 relevant sentences are selected as the input sequence.
- Format standardization is applied to attributes and sentences selected from product profile.

Table 1 shows the precision/recall performance on a dataset of 1357 human-labeled examples, we can clearly see that our proposed approach largely improves the precision and recall performance of attribute value validation task. Product title holds limited info, but the relevant content selected from product profile can capture more info and boost the validation quality and performance. If using the same top 5 relevant sentences as the inputs, our fine-tuned model improves PRAUC from 0.578 to 0.843, which achieves 46% improvement, in terms of the contradiction class. Furthermore, the format standardization achieves 5.46% improvement over pure relevant content selection. It is worth noting that the pre-trained NLI model is better at identifying consistent than contradictory info, thus we took this advantage to generate contradictory pairs as described in Section 4.1. Our purposed method also achieves more than 10% performance in consistency detection, which is applied to recommend the correct values for LOV values.

We also test the LOV value correction performance on a human-validated dateset of 209 different LOV examples, the length of available LOV values ranges from 2 to 30. The precision results of pre-trained and fine-tuned RoBERTa models are shown in Table 2. We could see that the fine-tuned model achieves 46.33% improvement over the pre-trained model in terms of title inputs and 12.46% in terms of inputs of top 5 relevant sentences from product profile.

Table 1: The performance comparison of different methods.

Model	Class	PRAUC	R@.7P	R@.8P	R@.9P	R@.95P
RoBERTa large-mnli + title	contr. neu. entail.	0.481 0.422 0.743	0.005 0.004 0.606	0.006 0.003 0.566	0.004 0.001 0.403	0.002 0.000 0.316
RoBERTa large-mnli + top 5	contr. neu. entail.	0.578 0.596 0.855	0.106 0.269 0.808	0.031 0.193 0.740	0.013 0.117 0.640	0.011 0.035 0.405
Fine-tuned + title	contr. neu. entail.	0.602 0.498 0.792	0.412 0.017 0.661	0.288 0.013 0.624	0.133 0.011 0.581	0.042 0.007 0.536
Fine-tuned + top 5	contr. neu. entail.	0.843 0.804 0.945	0.792 0.832 0.936	0.706 0.623 0.914	0.576 0.14 0.862	0.419 0.082 0.794
Fine-tuned + top 5 + standardization	contr. neu. entail.	**0.889** **0.864** **0.950**	**0.872** **0.885** **0.943**	**0.803** **0.709** **0.929**	**0.656** **0.594** **0.878**	**0.506** **0.289** **0.808**

Similar to attribute value validation, our proposed top relevant sentence selection method can provide more info for value correction process and boost the performance. By leveraging the fine-tuned RoBERTa NLI model for LOV value correction, we could automatically correct around 20% of the inconsistencies detected, which reduces manual effort and improves the speed and efficiency of the value correction process. Table 4 shows some examples of LOV value correction results.

In the real application, we set a threshold as 0.9 for the output confidence scores to get a higher precision of 0.89 while retaining a reasonable recall of 0.67 for downstream applications. Batch prediction/correction is deployed to run through all products from different categories on 4 Telsa P100 GPUs.

Table 2: LOV attribute value recommendation precision

Model	Title inputs	Top5 inputs
RoBERTa-large-mnli	0.354	0.698
Fine-tuned	**0.518**	**0.785**

Table 3: Contradictory detection comparison between NLI and anomaly detection

Contradiction	Fine-tuned + top 5 + standardization (>90% conf.)	Anomaly Detection
LOV attrs	62K	994
numeric attrs	13K	-
text attrs	4K	-
total	79K	994

5.2 Comparison with anomaly detection

We also run the model over a category that has 222K products and 4M pairs of attributes/values, and our proposed method detects

79K contradictory attribute values as shown in Table 3. 54% of the attributes and 3% of the products have one or more contradictory values detected. On average, the LOV attributes have more contradictory values than other attribute types. We had around 12K contradictory values go through the internal cleanup process, and the model prediction precision was around 90%.

We compare the performance of the fine-tuned RoBERTa NLI model with anomaly detection. In particular, DBSCAN, Histogram-based Outlier Score, and Isolation Forest [18] are used to detect the anomalies in numeric attributes. The majority voting method is then used to get the final results and improve the detection precision. 994 numeric attribute values are detected as anomalies which is much lower than the total number of 13K contradictory numeric values detected by fine-tuned RoBERTa NLI model. Anomaly detection usually uses unsupervised methods, and the anomalies detected can be hard to interpret and to correct in real e-commerce system, while NLI model typically has some product-specific information to explain the contradiction to a certain extent. The NLI based method is capable of handling all types of attributes and treating all of them as textual input to improve the coverage.

6 CONCLUSION AND FUTURE WORKS

In this paper, we present one unified model to solve the problems of detecting and correcting incorrect product attribute values on e-commerce websites. Instead of building multiple disparate models, one single transformer-based RoBERTa NLI model is trained to enable automatic discrepancy detection and value correction, which fully leverages the domain knowledge in both tasks. The end-to-end framework includes automatic high-quality training data generation for fine-tuning the NLI model, input selection for NLI model based on relevancy and the pipeline of discrepancy correction.

The method could be applied to all types of attributes across different products and categories, which features our model on scalability. Our experimental results and real-world application demonstrates that our method achieves superior performance, and can be used in the e-commerce domain to reduce manual efforts in detection and correction.

Future work in the logic calculation and automatic Unit of Measurement detection space could help us enhance the incorrect value detection process. Moreover, product images could be used as an additional source to evaluate the attribute values. As for automatic data correction, a transformer-based QA model could be used to auto-fill the numeric attributes with retraining and post-processing steps. Further, the selection of top relevant content could be embedded into the transformer-based model itself.

REFERENCES

[1] Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326* (2015).

[2] Raghavendra Chalapathy and Sanjay Chawla. 2019. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407* (2019).

[3] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *ACM computing surveys (CSUR)* 41, 3 (2009), 1–58.

[4] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860* (2019).

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

[6] Xin Luna Dong, Xiang He, Andrey Kan, Xian Li, Yan Liang, Jun Ma, Yifan Ethan Xu, Chenwei Zhang, Tong Zhao, Gabriel Blanco Saldana, et al. 2020. AutoKnow: Self-driving knowledge collection for products of thousands of types. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2724–2734.

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

[8] 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. *arXiv preprint arXiv:2106.04630* (2021).

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

[10] Guansong Pang, Chunhua Shen, and Anton van den Hengel. 2019. Deep anomaly detection with deviation networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 353–362.

[11] Jacob M Schreiber, Jeffrey A Bilmes, and William Stafford Noble. 2020. apricot: Submodular selection for data summarization in Python. *J. Mach. Learn. Res.* 21 (2020), 161–1.

[12] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909* (2015).

[13] Shane Storks, Qiaozi Gao, and Joyce Y Chai. 2019. Recent advances in natural language inference: A survey of benchmarks, resources, and approaches. *arXiv preprint arXiv:1904.01172* (2019).

[14] Yaqing Wang, Yifan Ethan Xu, Xian Li, Xin Luna Dong, and Jing Gao. 2020. Automatic Validation of Textual Attribute Values in E-commerce Catalog by Learning with Limited Labeled Data. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2533–2541.

[15] Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426* (2017).

[16] 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. *arXiv preprint arXiv:2106.02318* (2021).

[17] Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. *arXiv preprint arXiv:1909.00161* (2019).

[18] Yue Zhao, Zain Nasrullah, and Zheng Li. 2019. PyOD: A Python Toolbox for Scalable Outlier Detection. *Journal of Machine Learning Research* 20, 96 (2019), 1–7. http://jmlr.org/papers/v20/19-011.html

[19] 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*. 1049–1058.

Table 4: Examples of LOV value correction.

Attr	Value	LOV list	Info from product profile	Recommended value	Confidence score
Drain Location	Center	Center;Front;Left;Rear;Right	Drain Position: Rear	Rear	0.995
Sink Shape	Round	Rectangular;Round;Specialty;Square	Offset modern undermount rectangular ceramic sink basin	Rectangular	0.814
Sink Gauge	16 Gauge	16 Gauge;18 Gauge;20 Gauge;22 Gauge;No Gauge Applicable; 14 Gauge; 12 Gauge	Durable and dent-resistant sink: sturdy 18-Gauge	18 Gauge	0.987
Flushing Type	Single Flush	Dual Flush;Single Flush	Double Flush Elongated Toilet	Dual Flush	0.98
Mirror Orientation	Horizontal	Horizontal; Vertical; Vertical / Horizontal	this mirror is intended to be hung vertically	Vertical	0.99