Using Large Language Models to Improve Product Information in E-commerce Catalogs

Gang Luo*
Amazon
University of Washington
Seattle, WA, USA
luogang@uw.edu

Julien Han
Amazon
Seattle, WA, USA
julienhan1795@gmail.com

Hayreddin Ceker
Amazon
Seattle, WA, USA
hayro@amazon.com

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

ABSTRACT

To give customers good experience, an e-commerce retailer needs high-quality product information in its catalog. Yet, the raw product information often lacks sufficient quality. For a large catalog that can contain billions of products, manually fixing this information is highly labor-intensive. To address this issue, we propose using the tool use functionality of large language models to automatically improve product information. In this talk, we show why existing data cleaning methods are not well suited for this task and how we designed our automated system to improve product information. When evaluated on a random sample of products from an e-commerce catalog, our system improved product information completeness by 78% with no major drop in information accuracy.

CCS CONCEPTS

• Applied computing → Online shopping; • Computing methodologies → Machine learning; Natural language processing.

KEYWORDS

E-commerce; large language model; tool use; product information

ACM Reference format:

Gang Luo, Julien Han, Hayreddin Ceker, and Karim Bouyarmane. 2025. Using Large Language Models to Improve Product Information in E-commerce Catalogs. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM'25), November 10-14, 2025, Seoul, Republic of Korea.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3746252.3761437

1 INTRODUCTION

To ensure a good customer experience, an e-commerce retailer needs high-quality product information in its catalog. Yet, the raw product information often lacks sufficient quality. Usually, a product's information covers several dozen to several hundred attributes. >80% of attribute values can be missing. Many attribute values can be wrong. Also, long-text attribute values (e.g., title, description, and feature bullets) could include undesired content, omit key details, or not be well written. As a large catalog can contain billions of products, manually fixing the product information in it is highly labor-intensive. Thus, we want to automatically improve product information.

Many data cleaning methods exist [2-4]. Traditional methods use statistical, database, data mining, and machine learning techniques [2]. Several recent methods, some beating traditional methods, use large language models (LLMs) [3, 4]. Yet, no existing data cleaning method is well suited for improving product information in e-commerce catalogs:

1) Existing methods usually handle relational database tables with a fixed schema. In our case, the attributes associated with a product vary by product type.
2) Existing methods ignore long-text attributes and usually aim to fix wrong values and fill in missing values for numerical, categorical, and short-text attributes. In our case, we want to fix wrong values and fill in missing values for all types of attributes, as well as improve the quality of long-text attributes via content rewriting, e.g., by distilling and retaining key information to make the content more concise.
3) Typical existing data cleaning methods cannot extract information from long-text attribute values to fix wrong values and fill in missing values for numerical, categorical, and short-text attributes, though this is needed in our case. Existing data repair methods mainly work by finding another database that is consistent and differs minimally from the given database, but such a database may not exist in our case.

* Gang Luo is an Amazon Scholar as well as a Professor at the University of Washington. This paper describes work done at Amazon.

4) Existing methods that do not use LLMs usually do (e.g., mean) imputation to fill in missing values for numerical attributes. The imputed values are often imprecise. In our case, we want to fill in precise values for numerical attributes.
5) Many existing methods that use neither machine learning nor LLMs require user-provided inputs such as constraints, regular expressions, and dictionaries. It is time-consuming for users to provide such inputs and difficult for them to fully cover all possible cases. In our case, no such input should be required.
6) Existing methods using LLMs often focus on fixing a few fixed types of errors. In our case, we want to fix all types of errors.
7) The method described in [3] requires the user to input a data analysis purpose for which the cleaned data will be used. In our case, no such requirement should be imposed.

To address the above-mentioned issue, we propose using the tool use functionality of LLMs to automatically improve product information. This serves as a test case. The principle behind our proposed method is general and can be applied to clean any data set where the schema may vary across data instances.

2 METHODS

In this section, we outline two new methods that use LLMs to improve product information in e-commerce catalogs: the tool use method and the vanilla LLM method.

Tool use method: This method uses the tool use functionality [1] that many LLMs provide. A tool is a function with zero or more input parameters. Given a list of tools and a user query, the tool use functionality selects an appropriate tool to help with the query and extracts the tool's input parameter values from it. Our idea is to use this functionality to obtain improved product information by linking a tool's input parameters to a product's attributes and embedding the product's data from various sources into the user query.

More specifically, given a product, we find its associated attributes based on its product type. Our goal is to improve the content of these attributes. For each attribute, we retrieve its name, type, and description. If the attribute is of array type, we retrieve the minimum and maximum number of items allowed in the array, if any. If the attribute is of string type or an array of strings, we retrieve the minimum and maximum lengths allowed for the string or for each string in the array, respectively, if any. If the attribute is of string, array of strings, or Boolean type, we retrieve the list of possible values for the string, each string in the array, or the Boolean attribute, respectively, if any. We use all this information to create the JSON (JavaScript Object Notation) schema for the input parameters of a tool, with each input parameter linked to a distinct attribute. We provide this tool, the product's catalog information, and any additional data retrieved from various websites for the product to the LLM's tool use functionality and require the LLM to select this tool. The LLM's tool use functionality then outputs a JSON object listing this tool's input parameter values, which we use as the improved attribute values for the product.

Vanilla LLM method: This method uses an LLM without invoking the tool use functionality. Given a product, we provide its attribute schema (as described above), its catalog information, and any additional data retrieved from various websites for it to the LLM. We then ask the LLM to output a JSON object containing the product's improved attribute values. We check the generated result. If it is not a valid JSON object, we ask the LLM to correct it.

3 EVALUATIONS

We implemented both the tool use and vanilla LLM methods in an automated system to improve product information. We used the Claude 3.7 Sonnet LLM and evaluated our methods on 234 products randomly sampled from an e-commerce catalog. Human experts judged the correctness of the attribute values.

Table 1 shows the performance of different methods. Completeness is the percentage of product attributes with non-missing values. Accuracy is the percentage of product attributes with correct values. Compared to the raw product information in the catalog, the tool use method improved completeness by 78% with a 1.1% drop in accuracy. Compared to the vanilla LLM method, the tool use method improved completeness by 26.7% with a 2% drop in accuracy. Due to the huge improvement in completeness and minor difference in accuracy, we prefer the tool use method over the vanilla LLM method.

Table 1. Performance of different methods.

Method	Completeness	Accuracy
Raw product information in the catalog (baseline)	16.8%	77.3%
Vanilla LLM method	68.1%	78.2%
Tool use method	94.8%	76.2%

4 SPEAKER BIO

Gang Luo is an Amazon Scholar at Amazon as well as a professor at the University of Washington. He holds a PhD degree in computer science from the University of Wisconsin-Madison. He works on machine learning, health informatics, information retrieval, database systems, and data mining.

5 RELEVANCIES TO CIKM

Our talk covers several topics of interest to the CIKM community: practical industry challenges, system design from industry practitioners, and connections with academia to solve interesting problems.

REFERENCES

[1] Anthropic. 2025. Tool use with Claude. https://docs.anthropic.com/en/docs/build-with-claude/tool-use/overview.
[2] Ihab F. Ilyas and Xu Chu. 2019. *Data Cleaning*. ACM Books. New York, NY.
[3] Lan Li, Liri Fang, Vetle I. Torvik. 2024. AutoDCWorkflow: LLM-based data cleaning workflow auto-generation and benchmark. arXiv:2412.06724
[4] Shuo Zhang, Zezhou Huang, Eugene Wu. 2024. Data cleaning using large language models. arXiv:2410.15547