Hallucination Detection in LLM-enriched Product Listings

Ling Jiang, Keer Jiang, Xiaoyu Chu, Saaransh Gulati, Pulkit Garg
Amazon
Sunnyvale, CA, USA
{jiangll, kjiang, xiaoychu, saaransg, pulkitg}@amazon.com

Abstract

E-commerce faces persistent challenges with data quality issue of product listings. Recent advances in Large Language Models (LLMs) offer a promising avenue for automated product listing enrichment. However, LLMs are prone to hallucinations, which we define as the generation of content that is unfaithful to the source input. This poses significant risks in customer-facing applications. Hallucination detection is particularly challenging in the vast e-commerce domain, where billions of products are sold. In this paper, we propose a two-phase approach for detecting hallucinations in LLM-enriched product listings. The first phase prioritizes recall through cost-effective unsupervised techniques. The second phase maximizes precision by leveraging LLMs to validate candidate hallucinations detected in phase one. The first phase significantly reduces the inference space and enables the resource-intensive methods in the second phase to scale effectively. Experiments on two real-world datasets demonstrated that our approach achieved satisfactory recall on unstructured product attributes with suboptimal precision, primarily due to the inherent ambiguity of unstructured attributes and the presence of common sense reasoning. This highlights the necessity for a refined approach to distinguish between common sense and hallucination. On structured attributes with clearly defined hallucinations, our approach effectively detected hallucinations with precision and recall surpassing targeted level.

Keywords: large language model, hallucination, e-commerce, product listing enrichment

1. Introduction

In e-commerce, the significance of comprehensive product listings cannot be overstated, as it plays a pivotal role in facilitating informed purchase decisions by customers. However, real-world product listings often suffer from diverse quality issues, such as data incompleteness, information redundancy, and misinformation. These challenges impact customers' shopping experiences. Therefore, product listing enrichment is a critical task in e-commerce to generate compelling product listings.

The product listing enrichment task aims to create concise yet informative product listings given the source product data. Figure 1 illustrates this process. In the initial listing, several essential product attribute values are missing, and the product name contains redundant details. After enrichment, the product name is more succinct and user-friendly, and a correct value was populated for the attribute *Material*. Such enriched listings can help customer reduce cognitive load during product evaluation and improve sales conversion (Purnomo, 2023).

Product listing enrichment involves generating structured data and free text from the source input, which is an essential task in various natural language generation applications, such as summarization (Nenkova et al., 2011) and data-to-text generation (Wiseman et al., 2017). Traditional template-based approaches (Gatt and Reiter, 2009; Reiter et al., 2005) rely on manually crafted rules and lack scalability. Transformers and language models (Vaswani et al., 2017; Devlin et al., 2018;

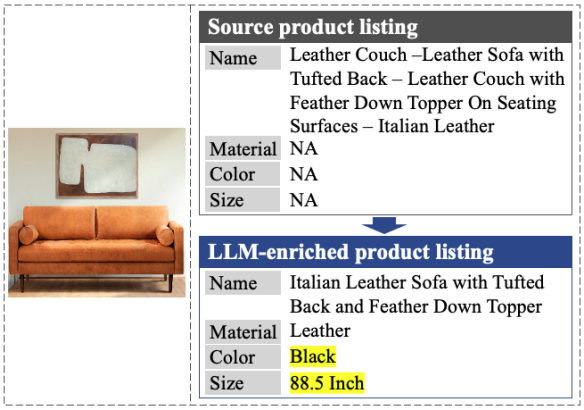

Figure 1: Example of product listing enrichment and hallucination.

Yang et al., 2019; Liu et al., 2019; Radford et al., 2019) have demonstrated exceptional capabilities of generating fluent text. Recently, Large Language Models (LLMs) (Brown et al., 2020; Ouyang et al., 2022; Achiam et al., 2023; Touvron et al., 2023; Chowdhery et al., 2023) have pushed the boundaries of natural language generation to new heights (Bubeck et al., 2023). The remarkable language comprehension of LLMs offers an opportunity for automating the generation and enhancement of product listings (Westmoreland, 2023).

However, a concerning drawback of LLMs is its tendency to hallucinate, when LLMs generate texts that appear fluent and coherent but are nonfactual or unsupported by the input data (Varshney et al.,

2023). In Figure 1, the LLM successfully improved the product *Name* and accurately generated value for *Material*. Conversely, the reliability of the generated values for *Color* and *Size* is questionable when examining the source data alone without additional information. Such hallucinations pose risks by potentially leading to negative user experiences and, more critically, misinformation-induced purchases. In safety-critical scenarios, such as the failure to generate warnings on toy choking hazards, hallucinations may result in legal consequences.

In this work, we address the hallucination problem in LLM-enriched product listings. We define hallucination as the generation of text that is unfaithful to the provided source input (Ji et al., 2023). Some works also consider factual inaccuracies in their definition (Varshney et al., 2023; Zhang et al., 2023). The primary function of product listings is to communicate descriptive details about the items. For example, the value of *Material* is product-specific, inherently contingent upon the source product information. Here, factual accuracy depends on faithfully reflecting the source input for each unique product, assuming the provided product information accurately describes the products. The source input data serves as the definitive reference for truth in this context.

Previous studies have employed the hidden layer activations or logit values of LLMs to detect hallucinated content (Azaria and Mitchell, 2023; Varshney et al., 2023). Yet, such methods require access to the internal states of LLMs, which is typically not available in state-of-the-art black-box LLMs (e.g. ChatGPT). Some have integrated external knowledge bases with LLMs (Guo et al., 2022; Martino et al., 2023; Peng et al., 2023a; Lee et al., 2022), but this introduces additional cost and complexities. Alternatively, LLMs have been used to autonomously verify their outputs (Wang et al., 2023; Manakul et al., 2023) or have been fine-tuned for specific tasks (Cao et al., 2021; Yu et al., 2023), though LLM-based methods can be costly without in-house models. In-house LLMs, while circumventing some expenses, still demand extensive training data and substantial resources for model development.

While akin to detecting hallucinations in summarization (Cao et al., 2021) or data-to-text generation (Tian et al., 2019), our task involves unique challenges due to the mixture of free text and structured data in both source input and generated text. The former is often noisy and poor-formatted, particularly when sourced from third-party sellers in e-commerce. Furthermore, LLM-generated values may be embedded in various product attribute fields, necessitating a comprehensive examination of all product information for potential evidence. Given that an e-commerce website can sell billions of products, addressing hallucination detection at such a scale requires careful consideration of both performance and cost factors.

In this paper, we present an approach for hallucination detection in LLM-enriched product listings without accessing internal LLM states or relying on external data sources. We propose to detect hallucinations in a two-phase fashion, prioritizing recall in the initial phase and enhancing precision in the subsequent phase. The motivation stems from the high cost associated with LLM-only approaches in massive-scale e-commerce applications. In the initial phase, which we call Lexical and Semantic Screening (LSS), we apply cost-effective unsupervised techniques to detect a broad range of hallucinations. While these methods are efficient, their accuracy may be compromised due to limitations in text comprehension. The second phase, LLM validation, utilizes LLMs to confirm potential hallucinations detected in the first phase. Leveraging the robust language understanding capabilities of LLMs, we optimize precision in the second phase. The initial LSS phase significantly reduces the inference space, allowing the more resource-intensive LLM validation to scale effectively. Experiments on two real-world e-commerce datasets demonstrate the effectiveness of our proposed approach. In addition, we discovered that our approach performs better on structured attributes with concise, deterministic values, as opposed to unstructured attributes presented in long-form free text. We identified directions for future work through analysis of the experimental results.

2. Related Work

2.1. Definition of Hallucination

Varshney et al. (2023) defined hallucination as the generation of text or responses that seem syntactically sound, fluent, and natural but are factually incorrect, nonsensical, or unfaithful to the provided source input. This definition aligns with the taxonomy proposed by Zhang et al. (2023). However, studies on hallucination in various natural language generation tasks (Tian et al., 2019; Maynez et al., 2020; Weng et al., 2023) may emphasize distinct aspects of the phenomenon. Consequently, the definition of hallucination may exhibit variability across tasks. In the product listing enrichment task, product listings primarily convey descriptive information about the products, and factual accuracy is contingent on faithfully representing the source input for each individual product. In this sense, our perception of hallucination aligns more closely with Ji et al. (2023)'s definition, which refers to the generation of text that is nonsensical, or unfaithful to the provided source input. It is further categorized into intrinsic and extrinsic hallucination. Intrinsic hallu-

cination involves output contradicting the source, while extrinsic hallucination refers to output unverifiable against the source. We are concerned with both types, encompassing content that either contradicts or lacks support in the source input, emphasizing the unfaithfulness aspect. After we identify unfaithful hallucinations, we need to further determine the factual accuracy to serve the end business goal. However, we focus this work on the faithfulness aspect and leave the factual part for future work.

2.2. Hallucination Detection

Many recent studies have been focused on mitigating hallucinations in LLMs. Depending on the accessibility of LLM models, there are white-box, grey-box and black-box approaches. A white-box method (Azaria and Mitchell, 2023) used the LLM's hidden layer activations to train a classifier that predicts the probability of a statement being true. Grey-box approaches detect the parts of the output sequence that the LLM is least confident about by examining the logit output values in the response (Varshney et al., 2023). Both white-box and grey-box approaches require access to internal states or token probabilities that may not necessarily be available, e.g. when LLMs are accessed through limited API calls. Black-box approaches (Manakul et al., 2023) are suitable for a wider range of applications when only LLMs responses are available.

Approaches for mitigating hallucination can also be grouped into zero-resource and external knowledge based approaches depending on if an external knowledge base is involved. External knowledge based approaches try to mitigate hallucination through information augmentation from external knowledge sources (Guo et al., 2022; Moiseev et al., 2022; Martino et al., 2023; Peng et al., 2023a). However, knowledge augmented approaches usually come with the cost of additional complexity and resource overhead (Lee et al., 2022).

Zero-resource approaches do not rely on external knowledge to detect hallucinated responses. One line of studies leverage unsupervised metrics scores (Celikyilmaz et al., 2020; Forbes et al., 2023) such as ROUGE (Lin, 2004), BLEU (Papineni et al., 2002), and METEOR (Banerjee and Lavie, 2005) to measure the consistency between the generated text and the source text. While these metrics offer simplicity, they frequently fall short in accurately aligning texts. This leads to sub-optimal performance when semantically relevant text diverges from the reference's surface form. Additionally, these metrics struggle to capture distant dependencies and tend to penalize changes in semantic ordering. BERTScore (Zhang et al., 2019) leverages the pre-trained contextual embeddings from BERT (Devlin et al., 2018) and matches words in candidate and reference sentences by cosine similarity. BERTScore has been shown to correlate well with human judgments. Another approach is converting hallucination detection to classification problem (Chen et al., 2023). Recent studies showed that LLMs can be good evaluators themselves (Wang et al., 2023) and many studies leverage LLMs to detect hallucinations (Mündler et al., 2023; Manakul et al., 2023; Weng et al., 2023; Fu et al., 2023). Another way to mitigate hallucination is fine-tuning LLMs on task-specific data (Cao et al., 2021; Yu et al., 2023). Some of these approaches can complement each other. For example, Guan et al. (2023) combined LLM verification, instruction tuning and retrieval augmentation to verify facts for LLMs outputs. LLM-based approaches can entail significant expenses when utilizing commercially available LLMs, especially on large-scale e-commerce applications. Alternatively, the development of proprietary LLMs within an organization introduces a different set of costs.

In this work, we present a method for detecting hallucinations in LLMs-enriched product listings. Our approach utilizes zero-resource black-box hallucination detection techniques, eliminating the need for external knowledge base or access to LLMs' internal states. This independence allows our system to be agnostic of upstream LLMs and be generalizable to a wider range of LLMs applications. Moreover, our method enhances scalability compared to LLM-only approaches by markedly reducing the inference space prior to LLM validation.

3. Methodology

We propose a two-phase approach for hallucination detection (Figure 2), emphasizing recall in the initial phase and enhancing precision in the subsequent phase. In the first phase, termed Lexical and Semantic Screening (LSS), cost-effective unsupervised techniques are applied to detect a broad spectrum of hallucinations. Although these methods are efficient, their performance may be compromised by text comprehension limitations. In the second phase, LLM validation, we utilize LLMs to validate the candidate hallucinations identified in the first phase. We optimize precision in the second phase by leveraging the robust language understanding capabilities of LLMs. Given the critical need for scalability in hallucination detection, particularly in the context of e-commerce with a vast product inventory, the initial LSS phase significantly reduces the inference space. This reduction enables the more resource-intensive approach to scale effectively in the second phase, addressing the challenge of processing billions of products for product listing enrichment in e-commerce. Figure 3 depicts LLM-generated attribute values,

Brand: Lilly Pulitzer and *Size: 12inchx10inch*, for the given source product listing. Using LSS, we confirmed the legitimacy of *Brand: Lilly Pulitzer* by cross-referencing it with information in the source product name. Subsequently, we evaluated the *Size: 12inchx10inch* attribute and determined it to be hallucinated content employing LLM validation.

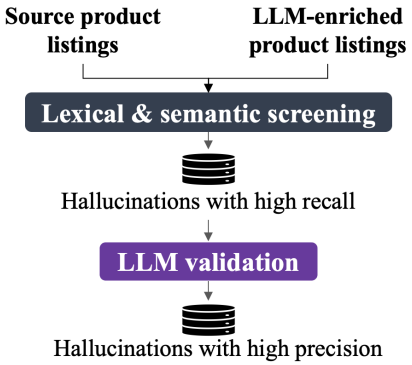

Figure 2: Two-phase hallucination detection.

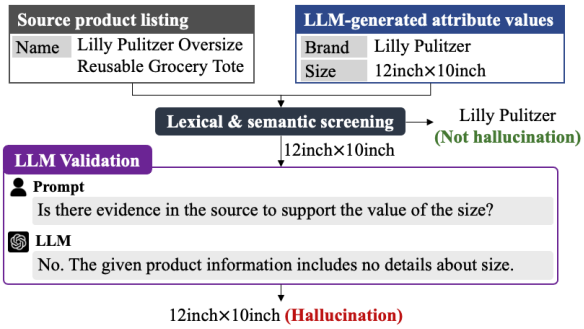

Figure 3: Example of hallucination detection.

3.1. Lexical and Semantic Screening (LSS)

In the initial phase, we employ unsupervised methods to flag all potential hallucinations by detecting information lacking supporting evidence in the source input. This support is traced through exact keywords or similar content, examined at either the token or the entire generated value level. We investigated techniques for locating supporting evidence from the source input:

3.1.1. Token-level LSS

Rebuffel et al. (2022) advocated addressing hallucinations at the word level rather than the instance level. They employed word-level alignment between candidate and inference text to control hallucinations. Their experiments demonstrated that word-level signals improved the fluency, factual accuracy, and relevance of LLM outputs. In our study, we similarly employ token-level alignment between the source input and LLM-enriched values to detect hallucinations.

Exact match. A direct method involves examining the presence of generated content in the source input. In Figure 2, the explicit mention of *Lilly Pulitzer* in the source input rules out hallucination. This method, denoted as T_{exact}, exhibits high recall but low precision in identifying hallucinations.

Exact matching often results in a large number false positive hallucinations, as LLMs may produce semantically similar but distinct words in enriched product listings. Fuzzy matching provides more flexibility, allowing LLMs to generate product information with enhanced fluency and coherence, leveraging advanced vocabulary. We assessed three fuzzy matching techniques:

Edit-distance. Token-level edit-distance (Levenshtein et al., 1966) between the source text and hallucinated text was used in generating synthetic data for hallucination detection and it was found that this approach provided sufficiently high quality training data in practice (Zhou et al., 2020). In our work, we adopt a similar approach by calculating the token-level edit distance between each token in the generated and source texts to pinpoint potential supporting evidence within the source data. This method is denoted as T_{edit}.

N-gram overlap metrics. N-gram matching metrics are commonly used for evaluating text generation by counting the number of n-grams that occur in the reference and candidate text. ROUGE (Lin, 2004) is often used for summarization evaluation, while BLEU is the most widely used metric in machine translation (Papineni et al., 2002). METEOR (Banerjee and Lavie, 2005) introduces flexibility by permitting a transition from strict unigram matching to encompassing word stems, synonyms, and paraphrases. These metrics provide a way to find the evidence for supporting the LLM-generated content from the source input. Therefore, we utilize three metrics: T_{rouge}, T_{bleu}, and T_{meteor}.

Embedding similarity. Token embeddings capture nuanced semantic and syntactic word relationships (Mikolov et al., 2013; Pennington et al., 2014; Bojanowski et al., 2016), and allow for a soft measure of similarity instead of strict string matching between the generated and source text. In this work, we experimented with three embedding models provided in Gensim (Rehurek and Sojka, 2011): *word2vec-google-news-300* (Mikolov et al., 2013) ($T_{word2vec}$), *glove-wiki-gigaword-300* (Pennington et al., 2014) (T_{glove}), and *fasttext-wiki-news-subwords-300* (Bojanowski et al., 2016) ($T_{fasttext}$).

3.1.2. Value-level LSS

Instead of checking evidence at the token level, we can evaluate the semantic similarity between the generated text and the source input as a whole. Significant deviations from the source text in the generated content can signal hallucination.

Sentence embedding. Sentence transformers convert sentences into semantically meaningful embeddings that can be compared using cosine-similarity (Reimers and Gurevych, 2019). In this work, we used four sentence-transformers (Reimers and Gurevych, 2019) models from HuggingFace (Wolf et al., 2019): *all-MiniLM-L6-v2*, *all-mpnet-base-v2*, *gtr-t5-large* (Ni et al., 2021), and *multi-qa-mpnet-base-dot-v1*, denoted as V_{miniLM}, V_{mpnet}, V_{gtr}, and V_{qa} respectively.

BERTScore. BERTScore has been shown to correlate well with human judgments for evaluating natural language generation tasks (Zhang et al., 2019). BERTScore calculates a similarity score between the candidate and reference text by aggregating the cosine similarities of their token embeddings. Unlike traditional metrics such as ROUGE, BLEU and METEOR that rely on string matching or heuristics, BERTScore uses contextualized token embeddings. It demonstrates a stronger capability of accommodating instances where semantically correct phrases deviate from the surface form of the reference. We abbreviate this approach as V_{bert}.

ALIGNSCORE. ALIGNSCORE (Zha et al., 2023) evaluates the factual consistency of generated text against a model input. It is applicable to various factual inconsistency scenarios, as it employs a unified training framework of the alignment function by integrating diverse data sources from seven well-established tasks. We denote this as V_{align}.

3.2. LLM Validation

Recent studies have explored the use of LLMs for evaluating their own generated text (Varshney et al., 2023), showing promising results. However, LLM validation typically incurs a significant expense due to associated API fees, presenting a challenge that impedes the scalability of LLM validation for e-commerce product listing enrichment. This challenge underscores the need to initially filter out a substantial portion of non-hallucinated content in the LSS step, which allows the subsequent LLM validation step to focus on a more manageable number of candidates. In this work, we used Claude 2 (Anthropic, 2023) from Anthropic for LLM validation.

Zero-shot. LLMs have shown great potential in evaluating the factual consistency between a document and its summary in the zero-shot setting (Luo et al., 2023). Thus, we directly prompt an LLM to verify the hallucinations detected in the LSS step.

Chain-of-thought (CoT). Chain-of-thought (CoT) prompting significantly enhances LLMs' complex reasoning abilities (Wei et al., 2022). CoT prompting, involving the presentation of intermediate reasoning steps, prove effective in both zero-shot (Kojima et al., 2022) or in-context learning (Wei et al., 2022) settings. Kojima et al. showed that LLMs demonstrated decent zero-shot reasoning capability by instructing them to think step by step (Kojima et al., 2022). In this work, we adopt a similar approach, instructing LLMs to provide step-by-step reasoning and identify supporting evidence when available. We only use the final decision from LLM as the prediction. Nonetheless, prompting the LLM to seek evidence initiates an underlying reasoning process, which can potentially improve the overall performance (Kojima et al., 2022).

In-context learning. LLMs demonstrate impressive ability to do in-context learning (Brown et al., 2020). They can generalize to unseen data by leveraging a limited set of training examples in the prompt, without explicit pre-training for the specific task (Xie et al., 2021). We asked domain experts to select examples of product listings with and without hallucinations and include them in prompts. By augmenting the context with these selected examples, it is anticipated that the LLM will discern the underlying pattern present in the demonstrations, thus enabling accurate predictions.

Many studies have investigated instruction-tuning for LLMs to enhance alignment with specific tasks (Ouyang et al., 2022; Peng et al., 2023b). In contrast to in-context learning, wherein examples are presented during inference without updating the LLMs' parameters, instruction tuning involves utilizing a set of examples to adjust the parameters during training. However, no demonstrations are employed during inference in instruction tuning (Duan et al., 2023). Although instruction tuning has shown promising results, it requires human-annotated prompts and feedback on a specific task. In our work, we do not experiment with instruction-tuning but consider it a potential future direction.

4. Experiments

In this section, we first introduce the dataset employed for evaluating hallucination detection in LLM-enriched product listings. We then compare the performance of different approaches, demonstrating the effectiveness of our proposed approach.

Finally, we discuss our findings, providing insights that guide our future research.

4.1. Dataset

We conducted experiments on two sets of human-annotated LLM-enriched product listings as described in Table 1. D_S exclusively contains structured product attributes, whereas D_U comprises only unstructured attributes. Structured attributes typically encompass product features characterized by enumerated, categorical, numerical, or keyword values, whereas unstructured attributes comprise long-form free-text values. In Table 1, we provide a summary of word counts for both structured and unstructured attributes. Given that the majority of product attributes are structured, we consolidate these into a single total count rather than listing each attribute individually. Conversely, we detail the three most prevalent unstructured attributes. Structured attributes normally contain 1-2 words, while unstructured attributes can include more than one hundred words. Additionally, attributes in D_S exhibits deterministic values, enabling annotators to identify hallucinations through a direct examination of the source input. In contrast, D_U introduces greater ambiguity. For instance, D_S primarily includes attributes such as *Color*, making it straightforward to assess the faithfulness of generated values to the source. On the contrary, D_U contains descriptive attributes, where values are less deterministic. Annotators' perceptions play a critical role in human judgments, contributing to increased uncertainty in hallucination detection.

Dataset	Attribute type	#Listings	#Entries
D_S	Structured	200	2,765
D_U	Unstructured	4,042	12,126

Table 1: Dataset description.

Type	Attribute	#Words		
		$Avg.$	$50p$	$75p$
Structured	-	2	1	2
Unstructured	U-Attribute 1	17	16	21
	U-Attribute 2	91	81	102
	U-Attribute 3	138	130	160

Table 2: Number of words by attribute type.

The datasets include original product listing alongside LLM-enriched values for one or multiple attributes in that listing. Our approach focuses on detecting hallucination at the attribute level. For instance, Figure 3 displays a listing with 2 LLM-enriched attributes: *Brand* and *Size*. We make independent decisions for each attribute. Domain experts audited the dataset to pinpoint hallucinations in LLM-generated values, and we use these human labels as the gold standard to evaluate our proposed approaches.

4.2. Experimental results

We utilize precision, recall, and F1 score to assess model performance. In e-commerce, distributing hallucinated product listings may lead to negative user experiences or legal issues in critical scenarios. Nevertheless, rejecting LLM-enriched product listings based on mistakenly identified hallucinations carries substantial costs. The precision-recall trade-off allows optimizing the balance between the consequences of false positive and false negative predictions in practice.

4.2.1. LSS

Table 3 and 4 detail the performance of various LSS models on D_S and D_U. In the results, P, R, and $F1$ represent precision, recall, and F1 score, respectively. Precision and recall targets, established by domain experts, are denoted as p and r. The target $f1$ is calculated with p and r. We present the performance of various approaches compared against the targets. As we optimize recall in the LSS phase to maximize the coverage of hallucinations, we choose the optimal model from each method with $recall >= min(recall_{max}, r)$.

LSS Model	P/p	R/r	$F1/f1$	$\%Inf$
T_{exact}	1.05	1.09	**1.07**	7.4
T_{edit}	1.05	1.09	**1.07**	7.4
T_{rouge}	0.88	1.06	0.95	9.0
T_{bleu}	1.05	1.07	1.06	6.8
T_{meteor}	0.68	1.06	0.82	11.6
$T_{word2vec}$	1.09	1.04	1.06	6.4
T_{glove}	1.09	1.03	1.06	6.4
$T_{fasttext}$	1.09	1.04	**1.07**	6.4
V_{miniLM}	0.85	1.06	0.94	9.8
V_{mpnet}	0.81	1.07	0.91	116
V_{gtr}	0.84	1.06	0.93	9.5
V_{qa}	0.81	1.06	0.91	10.6
V_{bert}	0.40	1.08	0.57	48.5
V_{align}	0.80	1.07	0.91	10.6

Table 3: Performances of LSS models on D_S.

The results indicate that LSS models exhibited promising abilities in detecting hallucinations within D_S. T_{exact}, T_{edit} and $T_{fasttext}$ yielded the highest F1 score at 1.07$f1$. While other LSS models demonstrated slightly lower performance, the majority maintained recall rates above 1.05r and precision within the range of 0.8p-1.05p. In contrast, these models exhibited a significant drop in performance when applied to D_U. While most maintained over 1.05r recall, precision struggled, falling below 0.25p. In e-commerce, ensuring high recall in hallucination detection is crucial, as false negatives

LSS Model	P/p	R/r	$F1/f1$	$\%Inf$
T_{exact}	0.23	1.11	0.36	12.5
T_{edit}	0.23	1.11	0.36	12.5
T_{rouge}	0.23	1.11	0.37	13.2
T_{bleu}	0.23	1.11	0.36	12.5
T_{meteor}	0.26	1.08	**0.41**	7.2
$T_{word2vec}$	0.23	1.11	0.36	12.4
T_{glove}	0.23	1.11	0.36	12.4
$T_{fasttext}$	0.23	1.11	0.36	12.4
V_{miniLM}	0.27	0.96	**0.41**	2.9
V_{mpnet}	0.20	0.90	0.31	2.5
V_{gtr}	0.17	0.74	0.27	1.7
V_{qa}	0.24	1.06	0.37	6.0
V_{bert}	0.25	0.34	0.29	0.3
V_{align}	0.22	1.05	0.35	6.8

Table 4: Performances of LSS models on D_U.

directly affect customer experiences and can hurt brand reputation.

The $\%Inf$ column denotes the portion of the inference space identified by LSS as hallucination candidates, serving as input for LLM validation. LSS models effectively reduced the inference space to less than 12.5%, and some models further narrowed this down to below 10% for LLM validation.

V_{bert} is an anomalous case among the models for D_S, with a performance of only 0.4p. Our observations indicate that V_{bert} tends to give lower scores to generated values significantly shorter than the reference text, causing an increase in false positives. Conversely, it assigns high similarity scores to unstructured attribute values compared to the source text, which substantially reduces recall.

The primary difference between the datasets is their attribute types. The unstructured attributes in D_U contain substantial free-text content, differing significantly from the concise nature of structured attribute values in D_S. Detecting hallucination from unstructured attribute values poses a greater challenge compared to structured ones. Also, the difference between token-level and value-level approaches is larger on structured attributes than that on unstructured attributes. This indicates that word-to-word comparison approaches are more suitable when the LLM-generated values are a bag of keywords, rather than coherent paragraphs.

Combining LSS models improves performance on D_S through an AND operation on their predictions. We explored every possible pairing of two models, and Table 5 displays the top three combined LSS models for D_S, demonstrating notable enhancements. Intuitively, the combined models generated enhance precision but decreased recall. All the optimal combined models consist of a token-level and a text-level model. ALIGNSCORE significantly contributed to the top combined models D_S. On the contrary, combined LSS models demonstrated inferior performance compared to individual models on dataset D_U, as evidenced in Table 5. Notably, this combination led to a significant reduction in recall without improving precision, resulting in a decreased F1 score.

Dataset	Combined model	P/p	R/r	$F1/f1$
D_S	T_{exact}+V_{align}	1.12	1.06	1.09
	T_{edit}+V_{align}	1.12	1.06	1.09
	T_{bleu}+V_{align}	1.12	1.05	1.09
D_U	T_{bleu}+V_{bert}	0.23	0.31	0.26
	T_{edit}+V_{bert}	0.22	0.31	0.26
	T_{exact}+V_{bert}	0.22	0.31	0.26

Table 5: Combination of LSS models.

4.2.2. LLM validation

Table 6 presents the F1 scores of applying LLM validation to hallucination candidates identified by LSS models. Overall, LLM validation enhanced the performance on D_S, while yielding marginal improvement on D_U. On D_S, solely employing zero-shot LLM validation improved the performance for each LSS model. The CoT approach generally achieved higher F1 scores, with the exceptions of T_{bleu}, $T_{fasttext}$ and V_{bert}, where the zero-shot approach outperformed. For D_U, LLM validation with in-context examples consistently outperformed zero-shot and CoT. Optimal performance was achieved by combining LSS models with subsequent LLM validation for D_S, all models demonstrated comparable F1 scores for D_U post-LLM validation. Selection of models can be tailored based on specific business requirements for precision and recall.

It is known that LLM in-context learning faces a robustness challenge (Liu et al., 2021), with outcomes highly depend on the chosen in-context examples. We observed this dependence in our experiments. We asked domain experts to select examples of product listings with and without hallucinations. The selected examples aim to represent different situations where hallucinations may occur. Despite efforts to cover various scenarios, it remains challenging to encompass all possibilities within a limited set of examples. Our observation indicates that the LLM model tends to replicate the behavior of the provided examples during validation response generation. Consequently, it predominantly identifies semantically-close samples as hallucinations. A future direction would be strategically select examples based on their similarity to the query instance.

4.3. Discussions

Next, we discuss some key findings during the experiments and talk about a few open questions not covered by this work. This sheds lights on directions for future work.

Model	D_S			D_U		
	Z	C	I	Z	C	I
T_{exact}	1.08	1.09	1.08	0.37	0.36	0.37
T_{edit}	1.08	1.09	1.08	0.37	0.36	0.37
T_{rouge}	1.01	1.05	1.00	0.36	0.36	0.37
T_{bleu}	1.08	1.07	1.05	0.36	0.36	0.37
T_{meteor}	0.97	0.99	0.99	0.36	0.35	0.39
$T_{word2vec}$	1.07	1.07	1.01	0.37	0.37	0.37
T_{glove}	1.07	1.07	1.01	0.37	0.37	0.37
$T_{fasttext}$	1.08	1.07	1.03	0.37	0.37	0.37
V_{miniLM}	1.01	1.01	1.02	0.37	0.37	0.37
V_{mpnet}	1.01	1.04	1.00	0.37	0.36	0.37
V_{gtr}	1.01	1.04	1.00	0.37	0.36	0.37
V_{qa}	1.00	1.04	1.00	0.37	0.36	0.37
V_{bert}	1.02	1.00	1.00	0.32	0.33	0.34
V_{align}	1.07	1.04	1.01	0.37	0.36	0.36
T_{exact}+V_{align}	1.10	1.07	1.04	-	-	-

Table 6: LLM validation performance. Z denotes zero-shot, C denotes CoT, and I denotes in-context learning.

Factual hallucination. This study focuses on identifying hallucinated product information that lack support from the source input. However, not all detected hallucinations are necessarily incorrect; some may align with factual information (Cao et al., 2021). As illustrated in Figure 4(a), the LLM-generated *Color* value lacks support in the source input, but aligns with the product image. It is worth noting that the product image was not part of the source input but included here for illustrative purposes. Conversely, the generated value of *Number of pocket* for the tote bag in Figure 4(b) is both unsupported and non-factual. In this study, we aim to identify hallucinated content based on the given source product listings. Distinguishing between factual and non-factual hallucinations could facilitate taking follow-up actions on the detected hallucinations. However, verifying the correctness of hallucinations necessitates external knowledge sources, like supplementary product details or images. We leave this for future work.

Source product listing	
Name	Big Scoop Dump Truck Toy with Sandbox
Gender	Unisex
Weight	1.96 LB
LLM-generated attribute values	
Color	Yellow, Green

(a) factual hallucination

Source product listing	
Name	Simply Cool Reusable bags
Size	14.5 X 14 X 6 inches
Color	Beige
LLM-generated attribute values	
Number of pockets	2

(b) non-factual hallucination

Figure 4: Factual and non-factual hallucination.

Common sense. We observed that annotators relied on common sense to evaluate hallucination in some cases. In Figure 5, LLM suggested *Walking* as the *Recommended use* for the sandal. Human annotators considered this non-hallucinatory, given the common understanding that sandals are suitable for walking rather than activities like running or jumping. However, our method flagged this as hallucination because there was no corresponding information in the source input supporting *Walking*, and the LLM validation step failed to capture it. Unlike the factual hallucination in Figure 4(a), which is clearly unfaithful to the source input even if factual, determining hallucination becomes challenging when common sense is a factor.

Source product listing	
Name	Women's Thong Sandal
Fabric	Polyester
Color	Light Brown Leopard
LLM-generated attribute values	
Recommended use	Walking

Figure 5: Common sense.

Common sense is a subjective and evolving concept, varying among individuals based on their experiences and knowledge. For instance, what is common knowledge today, such as Apple being the manufacturer of the iPhone, may not have been widely known several years ago. To distinguish common sense from hallucination in LLMs, we can leverage their hidden knowledge or external sources. However, a precise definition of common sense versus hallucination for different use cases is essential for effective hallucination detection.

Hallucination in LLM validation. The LLM validation phase, like other LLM applications, is prone to hallucinations. Instead of developing a new hallucination detection solution, a potential strategy to address this issue is to utilize multiple responses from one or more models. However, cost is a crucial consideration in real-world industrial applications, particularly in large-scale e-commerce settings. Another alternative is fine-tuning a task-specific LLM, but this necessitates high-quality training labels.

5. Conclusions

This paper introduces an effective approach for identifying hallucinations from LLM-enriched product listings. We proposed a two-phase approach, prioritizing recall in the initial phase and enhancing precision in the subsequent phase. Our experiments on two real-world e-commerce datasets demonstrate the efficacy of our proposed approach, with better performance observed on structured attributes compared to unstructured ones. We also highlight the challenge introduced by common sense when human annotators label the data, providing valuable insights for future work.

6. Bibliographical References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Anthropic. 2023. *Claude 2*. https://www.anthropic.com/index/claude-2 [Accessed: 2024-01-08].

Amos Azaria and Tom Mitchell. 2023. The internal state of an llm knows when its lying. *arXiv preprint arXiv:2304.13734*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Meng Cao, Yue Dong, and Jackie Chi Kit Cheung. 2021. Hallucinated but factual! inspecting the factuality of hallucinations in abstractive summarization. *arXiv preprint arXiv:2109.09784*.

Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2020. Evaluation of text generation: A survey. *arXiv preprint arXiv:2006.14799*.

Yuyan Chen, Qiang Fu, Yichen Yuan, Zhihao Wen, Ge Fan, Dayiheng Liu, Dongmei Zhang, Zhixu Li, and Yanghua Xiao. 2023. Hallucination detection: Robustly discerning reliable answers in large language models. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 245–255.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

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*.

Hanyu Duan, Yixuan Tang, Yi Yang, Ahmed Abbasi, and Kar Yan Tam. 2023. Exploring the relationship between in-context learning and instruction tuning. *arXiv preprint arXiv:2311.10367*.

Grant C. Forbes, Parth Katlana, and Zeydy Ortiz. 2023. Metric ensembles for hallucination detection.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166*.

Albert Gatt and Ehud Reiter. 2009. Simplenlg: A realisation engine for practical applications. In *Proceedings of the 12th European workshop on natural language generation (ENLG 2009)*, pages 90–93.

Jian Guan, Jesse Dodge, David Wadden, Minlie Huang, and Hao Peng. 2023. Language models hallucinate, but may excel at fact verification. *arXiv preprint arXiv:2310.14564*.

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *URL https://arxiv. org/abs/2205.11916*.

Nayeon Lee, Wei Ping, Peng Xu, Mostofa Patwary, Pascale N Fung, Mohammad Shoeybi, and Bryan Catanzaro. 2022. Factuality enhanced language models for open-ended text generation. *Advances in Neural Information Processing Systems*, 35:34586–34599.

Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and

reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*.

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*.

Zheheng Luo, Qianqian Xie, and Sophia Ananiadou. 2023. Chatgpt as a factual inconsistency evaluator for abstractive text summarization. *arXiv preprint arXiv:2303.15621*.

Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*.

Ariana Martino, Michael Iannelli, and Coleen Truong. 2023. Knowledge injection to counter large language model (llm) hallucination. In *European Semantic Web Conference*, pages 182–185. Springer.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. *arXiv preprint arXiv:2005.00661*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

Fedor Moiseev, Zhe Dong, Enrique Alfonseca, and Martin Jaggi. 2022. Skill: structured knowledge infusion for large language models. *arXiv preprint arXiv:2205.08184*.

Niels Mündler, Jingxuan He, Slobodan Jenko, and Martin Vechev. 2023. Self-contradictory hallucinations of large language models: Evaluation, detection and mitigation. *arXiv preprint arXiv:2305.15852*.

Ani Nenkova, Kathleen McKeown, et al. 2011. Automatic summarization. *Foundations and Trends® in Information Retrieval*, 5(2–3):103–233.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y Zhao, Yi Luan, Keith B Hall, Ming-Wei Chang, et al. 2021. Large dual encoders are generalizable retrievers. *arXiv preprint arXiv:2112.07899*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, et al. 2023a. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813*.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023b. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Yudiyanto Joko Purnomo. 2023. Digital marketing strategy to increase sales conversion on e-commerce platforms. *Journal of Contemporary Administration and Management (ADMAN)*, 1(2):54–62.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training. *OpenAI*.

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

Clément Rebuffel, Marco Roberti, Laure Soulier, Geoffrey Scoutheeten, Rossella Cancelliere, and Patrick Gallinari. 2022. Controlling hallucinations at word level in data-to-text generation. *Data Mining and Knowledge Discovery*, pages 1–37.

Radim Rehurek and Petr Sojka. 2011. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2).

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Ehud Reiter, Somayajulu Sripada, Jim Hunter, Jin Yu, and Ian Davy. 2005. Choosing words in computer-generated weather forecasts. *Artificial Intelligence*, 167(1-2):137–169.

Ran Tian, Shashi Narayan, Thibault Sellam, and Ankur P Parikh. 2019. Sticking to the facts: Confident decoding for faithful data-to-text generation. *arXiv preprint arXiv:1910.08684*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Neeraj Varshney, Wenlin Yao, Hongming Zhang, Jianshu Chen, and Dong Yu. 2023. A stitch in time saves nine: Detecting and mitigating hallucinations of llms by validating low-confidence generation. *arXiv preprint arXiv:2307.03987*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Jiaan Wang, Yunlong Liang, Fandong Meng, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023. Is chatgpt a good nlg evaluator? a preliminary study. *arXiv preprint arXiv:2303.04048*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. 2023. Large language models are better reasoners with self-verification.

Mary Beth Westmoreland. 2023. *Amazon launches generative AI to help sellers write product descriptions*. https://www.aboutamazon.com/news/small-business/amazon-sellers-generative-ai-tool [Accessed: 2023-12-07].

Sam Wiseman, Stuart M Shieber, and Alexander M Rush. 2017. Challenges in data-to-document generation. *arXiv preprint arXiv:1707.08052*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Zhang Ze Yu, Lau Jia Jaw, Wong Qin Jiang, and Zhang Hui. 2023. Fine-tuning language models with generative adversarial feedback. *arXiv preprint arXiv:2305.06176*.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. Alignscore: Evaluating factual consistency with a unified alignment function. *arXiv preprint arXiv:2305.16739*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023. Siren's song in the ai ocean: A survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Paco Guzman, Luke Zettlemoyer, and Marjan Ghazvininejad. 2020. Detecting hallucinated content in conditional neural sequence generation. *arXiv preprint arXiv:2011.02593*.