TeamCMU at Touché: Adversarial Co-Evolution for Advertisement Integration and Detection in Conversational Search

Notebook for the Touché Lab at CLEF 2025

To Eun Kim, João Coelho[†], Gbemileke Onilude[†] and Jai Singh[†]

Carnegie Mellon University, Pittsburgh, PA, USA

Abstract

As conversational search engines increasingly adopt generation-based paradigms powered by Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG), the integration of advertisements into generated responses presents both commercial opportunities and challenges for user experience. Unlike traditional search, where advertisements are clearly delineated, generative systems blur the boundary between informational content and promotional material, raising concerns around transparency and trust. In this work, we propose a modular pipeline for advertisement management in RAG-based conversational systems, consisting of an ad-rewriter for seamless ad integration and a robust ad-classifier for detection. We leverage synthetic data to train high-performing classifiers, which are then used to guide two complementary ad-integration strategies: supervised fine-tuning of the ad-rewriter and a best-of-N sampling approach that selects the least detectable ad-integrated response among multiple candidates. Our evaluation focuses on two core questions: the effectiveness of ad classifiers in detecting diverse ad integration strategies, and the training methods that best support coherent, minimally intrusive ad insertion. Experimental results show that our ad-classifier, trained on synthetic advertisement data inspired by marketing strategies and enhanced through curriculum learning, achieves robust detection performance. Additionally, we demonstrate that classifier-guided optimization, through both fine-tuning and best-of-N sampling, significantly improves ad stealth, enabling more seamless integration. These findings contribute an adversarial co-evolution framework for developing more sophisticated ad-aware generative search systems and robust ad classifiers.

Keywords

Conversational Search, Retrieval-Augmented Generation, LLM, Advertisement, Classification

1. Introduction

Conversational search engines powered by Large Language Models (LLMs) [1] and Retrieval-Augmented Generation (RAG) [2, 3] are increasingly integrating advertisements into responses to enhance monetization. As these systems shift toward generation-driven paradigms, the inclusion of advertising content in LLM outputs has become both a timely and underexplored area, especially as state-of-the-art industry systems move toward ad-supported deployments [4, 5]. Given that advertising has historically served as the primary revenue stream for search engines [6], this transition raises critical questions about how to embed ads in generated content without compromising response utility or user trust. Unlike traditional search interfaces, where sponsored content is explicitly demarcated, generative systems risk blurring the line between organic information and promotional material, potentially obfuscating ad presence in the absence of clear markers [7].

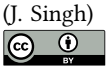
Despite its significance for the future of commercial LLM systems, advertisement integration and transparency in LLM-generated responses remain insufficiently studied. While prior work has introduced auction frameworks for generative ads and investigated methods for detecting LLM-generated

CLEF 2025 Working Notes, 9 – 12 September 2025, Madrid, Spain

[†] Equal contribution.

toeunk@cs.cmu.edu (T. E. Kim); jmcoelho@cs.cmu.edu (J. Coelho); gonilude@cs.cmu.edu (G. Onilude); jsingh2@andrew.cmu.edu (J. Singh)

0000-0002-2807-1623 (T. E. Kim); 0009-0001-6207-1934 (J. Coelho); 0009-0006-7264-1693 (G. Onilude); 0009-0007-6276-3029 (J. Singh)



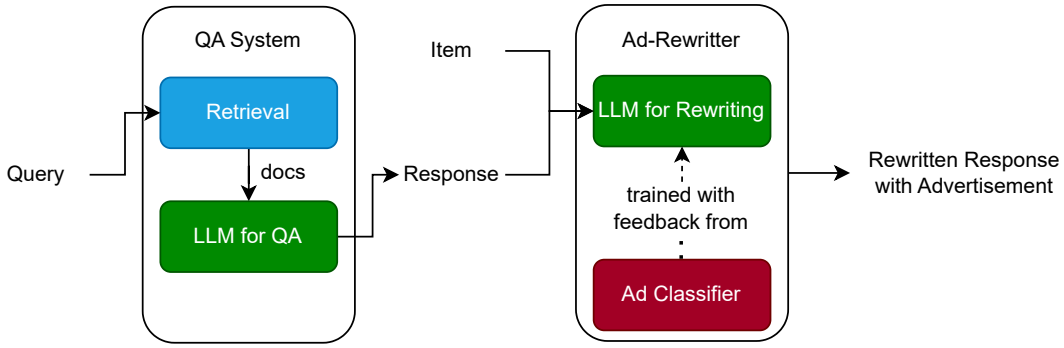


Figure 1: Overall Pipeline. A user query is first processed by the `QA System` to generate a base response. When an item is specified for advertisement, it is passed along with the base response to the `Ad-Rewriter`, which produces an ad-integrated version of the response. To further reduce ad detectability, we apply a best-of-N sampling strategy, selecting the rewritten response with the lowest ad probability as predicted by the `Ad-Classifier`.

advertisements [8, 9], comprehensive generation-side strategies remain limited. In addition, foundational insights from marketing research, such as the distinctions between explicit vs. implicit advertising and soft vs. hard selling [10, 11, 12], have yet to be meaningfully incorporated into generative model design. It also remains unclear whether existing ad-detection systems [13, 14], originally developed for traditional media, can generalize to the diverse and increasingly subtle forms of ad integrated in LLM-generated contents. Furthermore, recent efforts that rely on naive ad insertion strategies [9] may risk compromising response quality and user experience.

To address these challenges, we participate in both sub-tasks (generation and classification) of the *Advertisement in Retrieval-Augmented Generation* shared task at the Touché lab [15], CLEF 2025, where our systems were submitted via the TIRA platform [16]. We propose a modular pipeline (Figure 1) for advertisement management in RAG-based conversational systems. Our architecture consists of a standalone RAG-based `QA System`, followed by an `Ad-Rewriter` that integrates advertisements into the generated responses, and an `Ad-Classifier` trained to detect them.

The `Ad-Rewriter` is experimented in three variants: a zero-shot version prompted directly for ad integration, a supervised fine-tuning (SFT) variant trained with feedback from a robust ad-classifier, and a zero-shot version enhanced with best-of-N sampling, where the final response is selected from multiple candidates based on the classifier's ad probability scores.

In this *adversarial co-evolution* setup, the robustness of the `Ad-Classifier` is critical to the effectiveness of the `Ad-Rewriter`. To enhance the robustness of a classifier, we augment the provided dataset with carefully curated synthetic data, including hard positive and hard negative instances. The enhanced classifier is then used as feedback mechanisms to guide the optimization of the `Ad-Rewriter` across different implementation strategies.

Our study explores the feasibility of this framework through two central research questions:

- **RQ1**: How can we train an `Ad-Classifier` that achieves robust classification performance across diverse types of ad-integrated responses?
- **RQ2**: How can we develop an `Ad-Rewriter` that enables seamless ad integration while minimizing the likelihood of ad detection?

Through our experiments, we demonstrate the effectiveness of training the `Ad-Classifier` using hard positive and hard negative synthetic data to improve robustness. We also show that training the `Ad-Rewriter` in an adversarial setup (*i.e.,* using feedback from the robust classifier) leads to more effective and less detectable ad integration. We publicly release our code for further research.[1]

[1] https://github.com/kimdanny/TeamCMU-AdRAG

2. Related Work

In this section, we survey related work on open domain QA and emerging strategies for advertising in LLM-based search applications.

2.1. Open Domain QA

Early QA systems relied on extracting answer spans from retrieved documents and machine reading comprehension models [17]. Models such as DrQA [18] set the foundation for modern QA by improving retrieval and answer span prediction. Large language models (LLMs) revolutionized QA by enabling zero-shot and few-shot learning [19]. While these models provide high-quality answers, challenges remain in evaluation, due to large but semantically sound answers that differ from the gold label [20], and other LLM-related problems such as hallucination [21]. Retrieval-Augmented Generation (RAG) [2], a specialized method for generation as part of a retrieval-enhanced machine learning strategy [3, 22, 23] combines retrieval mechanisms with LLMs to improve factual correctness and response relevance, especially in knowledge-intensive task such as QA and fact-checking [24].

The MS-MARCO dataset [25] has driven significant web QA advancements, with state-of-the-art methods employing dense retrieval [26, 27] and contrastive learning to optimize both response quality and retrieval accuracy. Recent hybrid architectures that strategically combine LLMs with dense retrievers have demonstrated measurable improvements over standalone GPT-3.5 or LLaMA-7B prompting [28]. Similar approaches [29, 30] have also obtained state-of-the-art results on other benchmarks [31, 32], showing the versatility of retrieval-enhanced language models across domains.

2.2. Advertisement in the Era of LLMs

Search systems are actively employing LLMs to display search results [33]. In the era of LLMs, revenue generation through online advertising within LLM-generated response is gaining attention. In response, researchers are starting to investigate auctions and advertising strategies in the context of LLM-based search systems. Dubey et al. [8] studied an auction framework ensuring higher bidders receive greater ad placement in LLM outputs. Inspired by them, Hajiaghayi et al. [34] examined advertisement auctions with a focus on RAG, by considering both relevance (from the retriever) and bids when allocating ads within generated responses. Soumalias et al. [35] proposed an auction framework where advertisers influence LLM responses through reinforcement learning from human feedback. The detection of generated ad content is also a growing research area. Schmidt et al. [9] introduced the Webis Generated Native Ads 2024 dataset, focusing on identifying LLM-generated ads.

3. Problem Definition

In this section, we restate the two sub-tasks for the *Advertisement in Retrieval-Augmented Generation* shared task at the Touché lab [15] with more details.

3.1. Sub-task 1: Ad-Augmented QA

In Sub-task 1, the QA system is provided with an open-domain web query, a set of relevant passages, and a set of external items to be advertised. The objective is to build a system that leverages the passages to answer the query while incorporating an advertisement for one of the provided items. If m items are given, the system should generate m independent answers, each integrating a distinct item. These advertisements should be seamlessly woven into the response and difficult to detect as ads. Additionally, the system must be capable of generating standard, non-advertising answers when no items are provided, ensuring those responses do not exhibit ad-like characteristics.

Touché-25 Advertisement-in-Retrieval-Augmented-Generation (Ad-RAG) Dataset The Ad-RAG dataset comprises approximately 3,000 queries, for which systems are required to generate both ad-augmented and standard responses.[2] The queries are typically short phrases that describe a topic or product (*e.g., "good triceps workout equipment"*, *"corvette z06"*). For half of the queries, no advertisements are needed, but just an informative response. For the remaining queries, each requires the inclusion of, on average, two advertisements. Information about the items to be advertised is provided in the form of short descriptions averaging six words. Each query is supported by up to 100 passages retrieved from the MS MARCO v2.1 dataset [25] using BM25 retrieval [36]. These queries in the Ad-RAG dataset were obtained from the Webis-Ads dataset [9], which will be used in Sub-task 2.

3.2. Sub-task 2: Ad Detection

The objective of Sub-task 2 is to determine whether a given response contains an embedded advertisement. Specifically, the system receives a response as input and performs binary classification to predict whether the response includes an advertisement or is purely informative. An effective classifier should be robust to subtle ad insertions, ensuring that even seamlessly integrated advertisements can be accurately detected.

Webis Generated Native Ads 2024 (Webis-Ads) Dataset The Webis-Ads dataset [9] was created to train an ad-blocker system for conversational search engines. This dataset comprises approximately 7,500 queries, along with responses generated by Microsoft Copilot and YouChat. For half of these queries, a second version of the response was produced by prompting GPT-4 to insert advertisements without altering the original informative content. As a result, the dataset includes 7,500 responses without ads and 3,800 responses with ads.

Notably, the data in this dataset is relatively easy to fit. In our preliminary experiments, a simple DeBERTa-based text classifier [37] achieved around 98% accuracy on held-out data, suggesting that the naive ad-insertion strategy used to construct the dataset results in easily detectable patterns. This observation motivates the need for more challenging training data. To address this, we construct synthetic hard positive and hard negative examples, which we discuss in detail in the following sections.

4. Methodology

In this section, we describe our methodology for building a more robust `Ad-Classifier` and leveraging it as a feedback mechanism to improve the effectiveness of the `Ad-Rewriter`.

4.1. Pipeline Overview

Figure 1 presents an overview of our system. Given a user query q, the retrieval-augmented `QA System` retrieves relevant passages and generates an initial response r without any advertisements.[3] When a specific item a is provided for advertisement, the `Ad-Rewriter` module $\mathcal{G}$ modifies the base response r to seamlessly incorporate the promotional content, yielding a rewritten response y.

The `Ad-Rewriter` can operate in several modes: it can be 1) prompted to produce a rewritten response directly, 2) guided by best-of-N sampling [38, 39] using feedback from a trained `Ad-Classifier`, or 3) fine-tuned through supervised learning using training data generated with classifier feedback.

4.2. QA System

The `QA System` is responsible for generating contextually relevant responses to open-domain queries prior to any advertisement integration. While a typical QA pipeline involves both retrieval and genera-

[2] https://zenodo.org/records/14699130

[3] In the Touché competition, retrieved documents are provided. As a result, we do not evaluate retrieval effectiveness and simply use the top-k passages.

tion, in the Touché competition setting, retrieved passages are provided. Thus, we directly proceed to the generation step using the top-k passages.

Given the top-k retrieved passages z, we prompt a language model $\mathcal{F}$ to synthesize a coherent and self-contained response.[4] Prompts are constructed using a prompt generation function $\phi_p^{\mathcal{F}}(q, z)$, which is designed to elicit cohesive and informative responses from the model. The base QA output, r, serves as the input to the `Ad-Rewriter` module when advertisement integration is required. Prompt used for the response generation can be found in Appendix B.

4.3. Ad-Classifier

The `Ad-Classifier` $\mathcal{H}$ is formulated as a standard binary text classification task: given a query q and its corresponding response r, the model predicts whether the response contains an advertisement. To build increasingly robust classifiers, we incrementally expand the training data with progressively harder examples derived from multiple sources.

The initial version (**V0.0**) was trained solely on the Webis-Ads dataset [9]; a simple DeBERTa-based classifier [37] achieved strong performance on held-out data.[5] However, we found that it failed to generalize to more naturally embedded or implicit forms of advertising.

To address this limitation, we introduced two complementary types of synthetic training data. The first, *NaiveSynthetic* dataset, involves prompting an LLM to insert fictional advertisements into baseline QA responses without constraints, resulting in a wide variety of superficially embedded ads. With this data, we trained two classifiers: **V0.1** and **V0.2**.

The second, *StructuredSynthetic* dataset, incorporates real-world product entities sourced from Wikipedia. Drawing on advertising and marketing literature [10, 11, 12], we extract descriptive features and generate two categories of training examples: (i) hard positives, where the product is promoted through indirect or implicit language, and (ii) hard negatives, which are neutral informative passages about the product with no advertising intent. With the *StructuredSynthetic* dataset, we train successive versions of the classifier using combinations of the Webis-Ads, *NaiveSynthetic*, and *StructuredSynthetic* datasets: **V0.3**, **V0.4**, and **V0.5**. In the last two versions (V0.4, V0.5), we incorporate curriculum learning [40] based on classification difficulty, as estimated by the output logits from an earlier classifier (V0.1). This training strategy produces classifiers with improved generalization and robustness to diverse ad integration strategies, including those grounded in effective marketing practices.

4.3.1. Creation of NaiveSynthetic Data

NaiveSynthetic data generation follows the original Webis-Ads dataset approach, *i.e.*, given an answer without an advertisement, prompt an LLM to inject an ad. The query generation prompts include no specific item; rather, the LLM is instructed to generate an advertisement of an item that fits the context, which may result in the creation of fictional products. To promote diversity, we use a combination of 5 different LLMs: GPT-4o, Gemma-2-9B-it[6], LLaMA-3.1-8B-Instruct[7], Qwen2.5-7B-Instruct[8], and Mistral-7B-Instruct.[9] Moreover, we devise 12 different prompts for ad insertion, targeting various advertising strategies (*e.g.,* direct, indirect, explicit, implicit, hard-sell and soft-sell). An example prompt for *NaiveSynthetic* query generation can be found in Appendix D.1.

Using this setup, we trained two versions of the classifier. Both V0.1 and V0.2 leverage the same set of LLMs. However, V0.1 uses a single prompt for data generation, while V0.2 randomly samples from the full pool of 12 prompts. The HuggingFace model pages contain the prompts used for insertion.[10]

[4] Qwen2.5-7B-Instruct is used as a language model in our experiment.
[5] V0.0: https://huggingface.co/jmvcoelho/ad-classifier-v0.0
[6] https://huggingface.co/google/gemma-2-9b-it
[7] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct
[8] https://huggingface.co/Qwen/Qwen2.5-7B-Instruct
[9] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.1
[10] V0.1: https://huggingface.co/jmvcoelho/ad-classifier-v0.1; V0.2: https://huggingface.co/jmvcoelho/ad-classifier-v0.2

4.3.2. Creation of StructuredSynthetic Data

We generate *StructuredSynthetic* dataset through the following steps:

1. *Systematic collection of product entities from Wikipedia.*
   We manually select Wikipedia "infobox" namespaces likely to contain product-related pages that can be advertised (*e.g.*, 'product', 'brand', 'camera', 'automobile'), collecting a total of 25 infoboxes. To ensure that each page within these namespaces refers to a real product (e.g., *iPhone*) rather than a general concept (e.g., *Mobile phone*), we filter pages using Wikidata properties that strongly indicate "product-ness" (e.g., P162 – producer, P593 – model number). This allows us to curate a set of non-fictional product entities along with their associated Wikipedia content.
   For each verified entity, we retrieve its release year, rank the entities by recency, and retain only those released in or after the year 2000.

2. *Wikipedia article summarization and extraction of key promotional features.*
   For each selected entity, we prompt a GPT-4o model to summarize the corresponding Wikipedia page and extract key features and qualities suitable for promotional purposes.

3. *Creation of hard positives (indirect and implicit advertisements) and hard negatives (factual, non-promotional texts).*
   Drawing on insights from advertising literature, we generate two types of data using GPT-4o:
   - Hard positives: Indirect and implicit advertisements.
   - Hard negatives: Factual and informative descriptions without promotional intent.

   List of infoboxes, Wikidata properties, and the prompts used for hard positive and negative query generations can be found in Appendix D.2.

Using this setup, we trained three versions of the classifier. In V0.3, the classifier was trained on a combined dataset consisting of Webis-Ads, *NaiveSynthetic*, and *StructuredSynthetic* instances. In V0.4, we applied curriculum learning [40], where instance difficulty was determined by the V0.1 model. Finally, V0.5 used the same training regime as V0.4, but balanced the *NaiveSynthetic* and *StructuredSynthetic* instances by upsampling the *StructuredSynthetic* dataset. Further details are available on the corresponding HuggingFace model pages.[11]

4.4. Ad-Rewriter

The `Ad-Rewriter` module $\mathcal{G}$ takes as input a query q, an ad-free QA response r, and a product or service to be advertised a. These elements are combined into a prompt, denoted as $\phi_p^{\mathcal{G}}(q, r, a)$, which conditions the rewriting process. The goal of the `Ad-Rewriter` is to produce a fluent, contextually relevant, and minimally intrusive ad-integrated version of the original response.

Method 1: Zero-shot rewriting Our initial implementation relies on a prompt-based zero-shot rewriting, exploring advertisement strategies from the marketing literature, such as direct vs. indirect and explicit vs. implicit advertising. Prompt used for the rewriting can be found in Appendix C.

Method 2: Supervised fine-tuning-based rewriting To move beyond prompt engineering, we construct a training dataset using our synthetic query generation pipeline. For each (q, r, a) triplet, we generate five candidate ad-integrated responses: $y_i \sim \mathcal{G}(\phi_p^{\mathcal{G}}(q, r, a))$ for $i \in 1..5$, where $\mathcal{G}$ can be various LLMs with different temperature. Each rewritten response y_i is then scored by the ad-classifier $\mathcal{H}$, which estimates the likelihood that the response contains an advertisement: $\mathcal{H}(y_i)$.

[11] V0.3: https://huggingface.co/teknology/ad-classifier-v0.3; V0.4: https://huggingface.co/teknology/ad-classifier-v0.4; V0.5: https://huggingface.co/teknology/ad-classifier-v0.5

We adopt a supervised fine-tuning (SFT) regime in which the objective is to train the model to prefer completions with lower predicted ad probability. Formally, we define the optimal response y^* and the negative log-likelihood loss $\mathcal{L}_{NLL}$ as:

$$y^* = \text{argmin}_{y \in \{y_1, \ldots, y_5\}} \mathcal{H}(y) \tag{1}$$

$$\mathcal{L}_{NLL} = -\log \mathrm{P}(y^* \mid \phi_p^{\mathcal{G}}(q, r, a)). \tag{2}$$

Method 3: Zero-shot rewriting with Best-of-N sampling Due to the computational cost of fine-tuning a language model, we apply a best-of-N sampling strategy in the zero-shot method. We set our generation temperature above zero, and the model produces a diverse set of N candidate rewrites for each input. Each candidate response is then evaluated using a trained ad-classifier, which assigns an ad probability score. We select the response with the lowest predicted ad probability as the final rewritten output. In our experiments, we use $N = 10$.

This feedback loop, where classifiers guide the training of rewriters, forms the backbone of our approach, which aims to result in more natural ad integration within the generated responses.

5. Experiments

5.1. Experimental Setup

For Sub-task 1, systems are evaluated adversarially using a hidden classifier maintained by the task organizers. The primary evaluation metric is advertisement detection accuracy, where a lower score indicates more successful ad integration, *i.e.*, the advertisement is more difficult to detect. As the official classifier is unavailable during development, we employ a series of in-house `Ad-Classifier` (V0.0–V0.5) as proxy evaluators for iterative tuning and model comparison. Specifically, we compare detection accuracy across the three different `Ad-Rewriter` strategies (zero-shot, zero-shot with BoN, and supervised fine-tuning-based rewriting) to assess the detectability of inserted ads. This setup also serves to evaluate the performance of the `Ad-Classifier` for Sub-task 2 using standard binary classification accuracy. For `Ad-Classifier`, we use pre-trained DeBERTa model [37], and for `QA System` and `Ad-Rewriter`, we use Qwen2.5-7B-Instruct [41].

5.2. Results

RQ1: How can we train an `Ad-Classifier` that achieves robust classification performance across diverse types of ad-integrated responses? Before proceeding to ad rewriting, we first identify which classifiers perform well across a range of ad-integration strategies. To evaluate this, we test six versions of classifiers on responses generated using three ad rewriting approaches: a pure zero-shot method, a fine-tuned rewriter, and a zero-shot method with best-of-N (BoN) sampling. Each approach is evaluated under two different generation temperature settings. Recall that these rewriters modify the base QA response using various advertising techniques (*e.g.,* indirect promotion through storytelling), producing a diverse set of ad-integrated outputs.

Table 1 presents the ad classification accuracy for each classifier version across the different rewriting approaches. When comparing V0.0, which is trained without synthetic data or specialized training methods such as curriculum learning, to the other versions, we observe a substantial improvement in ad detection accuracy. This indicates that synthetic data inspired by marketing strategies can effectively enhance classifier performance. In particular, versions V0.1 and V0.4 achieve consistently high accuracy across both temperature settings, underscoring the value of combining diverse synthetic training data with curriculum learning.

Table 1
Ad-detection accuracy (%) across different versions of classifiers (V0.0–V0.5) under three ad rewriting methods with varying generation temperatures. This table highlights both the robustness of the classifiers and the effectiveness of the ad rewriting strategies. Among the classifiers, versions V0.1 and V0.4 demonstrate strong performance across a diverse set of ad-integrated responses. Regarding ad-rewriting methods, SFT and Zero-Shot-BoN-based `Ad-Rewriter` show low ad detection accuracy, indicating more seamless integration compared to the zero-shot approach across all classifier versions.

Rewriting Method	Temperature	V0.0	V0.1	V0.2	V0.3	V0.4	V0.5
Zero-Shot	1.0	34.93	86.77	73.98	85.31	91.85	69.36
	0.5	33.33	83.43	67.87	80.78	89.02	76.03
SFT	1.0	18.87	66.16	51.73	61.32	61.94	49.79
	0.5	16.99	55.22	41.55	51.34	46.24	38.54
Zero-Shot-BoN	1.0	32.24	54.35	41.90	52.34	49.78	46.93
	0.5	30.98	51.57	40.14	50.24	47.68	45.34

RQ2: How can we develop an `Ad-Rewriter` that enables seamless ad integration while minimizing the likelihood of ad detection? In addition to the basic zero-shot ad rewriter, we develop two enhanced versions: the SFT and Zero-Shot-BoN-based rewriter. Both approaches rely on a classifier to provide feedback, either during inference in the BoN setting or during training data construction for the SFT approach. Based on the classification performance observed on outputs from the pure Zero-Shot rewriter, we select V0.4 as the most robust classifier and use it both for scoring candidate responses in the best-of-N sampling strategy and for fine-tuning the rewriter.

As shown in Table 1, both the SFT and Zero-Shot-BoN-based rewriters lead to a substantial reduction in ad detection accuracy across all classifier versions when compared to the basic Zero-Shot rewriter. These results demonstrate that incorporating classifier feedback, either at inference time or during training, is effective in achieving more seamless ad integration. In fact, these methods are able to deceive robust classifiers in approximately 50 percent of cases. A paired Wilcoxon signed-rank test [42] confirms that this reduction is statistically significant for all classifiers, indicating that classifier-guided generation can reliably produce responses that are more difficult to detect as advertisements.

Shared Task Results For Sub-Task 1 (ad generation), the primary evaluation metric designated by the lab organizers is recall; lower recall indicates more subtle ad insertion, and thus a better-performing system. In addition to automatic metrics, the organizers manually reviewed 100 responses per submission to validate classifier labels and ensure the legitimacy of each run. We submitted our system based on the Zero-Shot-BoN approach, using the Qwen2.5-7B-Instruct model [41] for both the `QA System` and the `Ad-Rewriter`, and employing the V0.4 classifier for best-of-N sampling. Out of six submissions, including a baseline, our system ranked 3rd by recall (see Table 2). If F1-Score had been the ranking criterion, our system would have placed 2nd.

For Sub-Task 2 (ad classification), the primary evaluation metric set by the lab organizers is F1-Score. We submitted our V0.4 classifier for evaluation. Among 16 runs, including baselines, our classifier ranked 3rd in terms of F1-Score (see Table 3).

6. Discussion

Methodological Implications Our findings across the two research questions and the strong evaluation results highlight the effectiveness of using carefully curated synthetic data and classifier-guided training in managing ad integration within generative systems. For ad detection, we show that classifiers trained on synthetic data inspired by marketing strategies—particularly when combined with curriculum learning—achieve significantly more robust performance across a diverse set of ad-integration styles. On the generation side, incorporating classifier feedback, either through best-of-N sampling or during

Table 2
Sub-Task 1 (ad generation) evaluation results by Touché.

Team	Run	Precision	Recall	F1-Score
JU-NLP	ORPO_Mistral7b_v2	1.000	0.721	0.838
JU-NLP	ORPO_Mistral7b	0.995	0.830	0.905
TeamCMU	Adrewriting-BestOfN	0.821	0.858	0.839
Git Gud	Qwen2.5 7B V2	0.960	0.910	0.935
Git Gud	Qwen3 4B V2	0.984	0.918	0.950
Baselines	generate-baseline	0.796	0.996	0.885

Table 3
Sub-Task 2 (ad detection) evaluation results by Touché.

Team	Run	Precision	Recall	F1-Score
JU-NLP	DebertaFineTuned	0.788	0.758	0.773
Git Gud	Deberta-Large-V2	0.983	0.473	0.639
TeamCMU	deberta-synthetic-curriculum	0.945	0.479	0.636
Git Gud	Roberta-Large	0.985	0.460	0.627
Baseline	minilm-baseline	0.728	0.482	0.580
Pirate Passau	MPnet-finetuned	0.399	0.917	0.556
Pirate Passau	Tf-IDF-Logestic-Regression	0.395	0.734	0.514
JU-NLP	Finetuned_MPNET_v2	0.977	0.346	0.511
JU-NLP	Finetuned_MPNET	0.305	1.000	0.467
Baseline	naive-bayes-10	0.307	0.968	0.467
Baseline	naive-bayes-25	0.319	0.638	0.425
Pirate Passau	All-mini-LM-v2-finetuned	0.664	0.294	0.408
Git Gud	Deberta Large	0.312	0.355	0.332
Baseline	naive-bayes-40	0.367	0.257	0.302
Pirate Passau	all-mini+Random-forest	0.341	0.022	0.042
Pirate Passau	LLM-llama3.1	0.500	0.000	0.001

supervised fine-tuning, leads to ad-integrated responses that are substantially harder to detect. These results suggest that adversarial training dynamics between rewriters and classifiers can be effective in shaping both components for more seamless and harder-to-detect ad insertion.

Among these generation strategies, we also observe that responses generated at lower temperatures tend to yield lower ad detection rates. One possible explanation for this pattern is that the model produces more coherent and well-structured responses at lower temperatures [19], allowing ad insertions to blend more naturally with the surrounding content. In contrast, higher temperatures introduce greater variability, which can result in phrasing or transitions that are less contextually aligned, making the presence of advertisements more noticeable to the classifier.

Limitations A key constraint of this study is the reliance on synthetic data generated by LLMs necessitates more rigorous validation and incorporation of more challenging scenarios to ensure robustness. The binary nature of the current advertisement classifier may also fall short in fully capturing nuanced or context-dependent advertisements. Additionally, the metric of ad detectability is grounded in classifier performance. However, human users may perceive ads differently, and what evades a model may still be obvious to a human reader.

Ethical Considerations This work reveals that advertisements can be seamlessly integrated into LLM-generated responses in ways that are difficult even for strong classifiers to detect. While this demonstrates the technical feasibility of subtle ad insertion, it also underscores the importance of accompanying such capabilities with appropriate transparency controls. Without explicit labeling

or disclosure mechanisms, users may be unknowingly exposed to persuasive content, potentially diminishing trust in conversational systems [7]. Moreover, false positives from ad classifiers risk misclassifying informative content, which could disadvantage legitimate content providers. Ethical challenges are amplified when ads appear in sensitive contexts, such as mental health or emergency-related queries, or when cultural stereotypes and provider-side exposure imbalances propagate through system components. These findings highlight the need for careful design choices and deployment safeguards to ensure that stealthy ad integration does not come at the cost of user agency or marketplace fairness [43].

Future Direction Future work can address current limitations through comprehensive validation of synthetic data using approaches like system rank correlation and linguistic analysis [44]. Beyond any technical improvements, future implementations can explore more realistic scenarios involving retrieval based on dynamic ad bidding information [34]. Moreover, evaluating and ensuring provider-side fairness will be essential for maintaining a balanced and sustainable advertisement ecosystem, demanding rigorous assessment of both provider-consumer dynamics and systemic biases [45].

7. Conclusion

We show that fine-tuning an advertisement classifier using synthetic query data inspired by marketing strategies, along with progressively harder detection examples, significantly enhances its robustness and effectiveness in identifying seamlessly integrated ads. Notably, we find that feedback from such a well-trained classifier, whether used during test-time sampling or as part of the training objective, can be leveraged to guide ad generators that strategically evade detection, successfully deceiving even strong classifiers. This adversarial dynamic underscores both the potential and the challenge of developing reliable and transparent advertisement in LLM-based search systems.

Acknowledgments

We thank Professor Eric Nyberg, Professor Teruko Mitamura, and Kimihiro Hasegawa for their valuable feedback during the development of our system.

Declaration on Generative AI

During the preparation of this work, the authors used generative AI in order to identify and correct grammatical errors and typos. The authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] F. Radlinski, N. Craswell, A theoretical framework for conversational search, in: Proceedings of the 2017 conference on conference human information interaction and retrieval, 2017, pp. 117–126.

[2] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al., Retrieval-augmented generation for knowledge-intensive nlp tasks, Advances in Neural Information Processing Systems 33 (2020) 9459–9474.

[3] T. E. Kim, A. Salemi, A. Drozdov, F. Diaz, H. Zamani, Retrieval-enhanced machine learning: Synthesis and opportunities, arXiv preprint arXiv:2407.12982 (2024).

[4] Perplexity Team, Why we're experimenting with advertising, 2024. URL: https://www.perplexity.ai/hub/blog/why-we-re-experimenting-with-advertising, accessed: 2025-04-30.

[5] OpenAI, Improved shopping results from chatgpt search, 2025. URL: https://help.openai.com/en/articles/11146633-improved-shopping-results-from-chatgpt-search, accessed: 2025-04-30.

[6] J. Gleason, A. Koeninger, D. Hu, J. Teurn, Y. Bart, S. Knight, R. E. Robertson, C. Wilson, Search engine revenue from navigational and brand advertising, in: Proceedings of the International AAAI Conference on Web and Social Media, volume 18, 2024, pp. 488–501.
[7] I. Zelch, M. Hagen, M. Potthast, A user study on the acceptance of native advertising in generative ir, in: Proceedings of the 2024 Conference on Human Information Interaction and Retrieval, CHIIR '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 142–152. URL: https://doi.org/10.1145/3627508.3638316. doi:10.1145/3627508.3638316.
[8] A. Dubey, Z. Feng, R. Kidambi, A. Mehta, D. Wang, Auctions with llm summaries, in: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 713–722. URL: https://doi.org/10.1145/3637528.3672022. doi:10.1145/3637528.3672022.
[9] S. Schmidt, I. Zelch, J. Bevendorff, B. Stein, M. Hagen, M. Potthast, Detecting generated native ads in conversational search, in: Companion Proceedings of the ACM Web Conference 2024, WWW '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 722–725. URL: https://doi.org/10.1145/3589335.3651489. doi:10.1145/3589335.3651489.
[10] Y. Yi, Direct and indirect approaches to advertising persuasion: Which is more effective?, Journal of Business Research 20 (1990) 279–291.
[11] S. Shapiro, H. S. Krishnan, Memory-based measures for assessing advertising effects: A comparison of explicit and implicit memory effects, Journal of advertising 30 (2001) 1–13.
[12] S. Okazaki, B. Mueller, C. R. Taylor, Measuring soft-sell versus hard-sell advertising appeals, Journal of Advertising 39 (2010) 5–20.
[13] E. L. Post, C. N. Sekharan, Comparative study and evaluation of online ad-blockers, in: 2015 2nd International Conference on Information Science and Security (ICISS), IEEE, 2015, pp. 1–4.
[14] B. Shiller, J. Waldfogel, J. Ryan, The effect of ad blocking on website traffic and quality, The RAND Journal of Economics 49 (2018) 43–63.
[15] J. Kiesel, Ç. Çöltekin, M. Gohsen, S. Heineking, M. Heinrich, M. Fröbe, T. Hagen, M. Aliannejadi, T. Erjavec, M. Hagen, M. Kopp, N. Ljubešić, K. Meden, N. Mirzakhmedova, V. Morkevičius, H. Scells, I. Zelch, M. Potthast, B. Stein, Overview of Touché 2025: Argumentation Systems, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction. 16th International Conference of the CLEF Association (CLEF 2025), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2025.
[16] M. Fröbe, M. Wiegmann, N. Kolyada, B. Grahm, T. Elstner, F. Loebe, M. Hagen, B. Stein, M. Potthast, Continuous Integration for Reproducible Shared Tasks with TIRA.io, in: J. Kamps, L. Goeuriot, F. Crestani, M. Maistro, H. Joho, B. Davis, C. Gurrin, U. Kruschwitz, A. Caputo (Eds.), Advances in Information Retrieval. 45th European Conference on IR Research (ECIR 2023), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2023, pp. 236–241. doi:10.1007/978-3-031-28241-6_20.
[17] M. Seo, A. Kembhavi, A. Farhadi, H. Hajishirzi, Bidirectional attention flow for machine comprehension, in: International Conference on Learning Representations, 2017. URL: https://openreview.net/forum?id=HJ0UKP9ge.
[18] D. Chen, Reading wikipedia to answer open-domain questions, arXiv preprint arXiv:1704.00051 (2017).
[19] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Advances in neural information processing systems 33 (2020) 1877–1901.
[20] E. Kamalloo, N. Dziri, C. L. A. Clarke, D. Rafiei, Evaluating Open-Domain Question Answering in the Era of Large Language Models, in: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, 2023.
[21] J. Li, X. Cheng, X. Zhao, J. Nie, J. Wen, HaluEval: A Large-Scale Hallucination Evaluation Benchmark for Large Language Models, in: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 2023.
[22] H. Zamani, F. Diaz, M. Dehghani, D. Metzler, M. Bendersky, Retrieval-enhanced machine learning,

in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022, pp. 2875–2886.

[23] F. Diaz, A. Drozdov, T. E. Kim, A. Salemi, H. Zamani, Retrieval-enhanced machine learning: Synthesis and opportunities, in: Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region, 2024, pp. 299–302.

[24] F. Petroni, A. Piktus, A. Fan, P. Lewis, M. Yazdani, N. De Cao, J. Thorne, Y. Jernite, V. Karpukhin, J. Maillard, V. Plachouras, T. Rocktäschel, S. Riedel, KILT: a benchmark for knowledge intensive language tasks, in: K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, Y. Zhou (Eds.), Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Online, 2021, pp. 2523–2544. URL: https://aclanthology.org/2021.naacl-main.200. doi:10.18653/v1/2021.naacl-main.200.

[25] P. Bajaj, D. Campos, N. Craswell, L. Deng, J. Gao, X. Liu, R. Majumder, A. McNamara, B. Mitra, T. Nguyen, et al., Ms marco: A human generated machine reading comprehension dataset, arXiv preprint arXiv:1611.09268 (2016).

[26] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, W.-t. Yih, Dense passage retrieval for open-domain question answering, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 6769–6781. URL: https://aclanthology.org/2020.emnlp-main.550/. doi:10.18653/v1/2020.emnlp-main.550.

[27] L. Xiong, C. Xiong, Y. Li, K. Tang, J. Liu, P. N. Bennett, J. Ahmed, A. Overwijk, Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval, 2021.

[28] X. Li, Y. Zhou, Z. Dou, Unigen: A unified generative framework for retrieval and question answering with large language models, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2024.

[29] D. Wang, Q. Huang, M. Jackson, J. Gao, Retrieve what you need: A mutual learning framework for open-domain question answering, Trans. Assoc. Comput. Linguistics (2024).

[30] Y. Yin, G. Carenini, ARR: Question Answering with Large Language Models via Analyzing, Retrieving, and Reasoning, volume 2502.04689, 2025.

[31] M. Joshi, E. Choi, D. S. Weld, L. Zettlemoyer, Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2017.

[32] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, K. Toutanova, L. Jones, M. Kelcey, M.-W. Chang, A. M. Dai, J. Uszkoreit, Q. Le, S. Petrov, Natural questions: A benchmark for question answering research, Transactions of the Association for Computational Linguistics (2019).

[33] W. Sun, L. Yan, X. Ma, S. Wang, P. Ren, Z. Chen, D. Yin, Z. Ren, Is chatGPT good at search? investigating large language models as re-ranking agents, in: The 2023 Conference on Empirical Methods in Natural Language Processing, 2023. URL: https://openreview.net/forum?id=3Q6LON8y2I.

[34] M. Hajiaghayi, S. Lahaie, K. Rezaei, S. Shin, Ad auctions for LLMs via retrieval augmented generation, in: The Thirty-eighth Annual Conference on Neural Information Processing Systems, 2024. URL: https://openreview.net/forum?id=Ujo8V7iXmR.

[35] E. Soumalias, M. J. Curry, S. Seuken, Truthful aggregation of llms with an application to online advertising, arXiv preprint arXiv:2405.05905 (2024).

[36] S. E. Robertson, H. Zaragoza, The probabilistic relevance framework: BM25 and beyond, Found. Trends Inf. Retr. (2009).

[37] P. He, J. Gao, W. Chen, Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, arXiv preprint arXiv:2111.09543 (2021).

[38] N. Stiennon, L. Ouyang, J. Wu, D. M. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, P. Christiano, Learning to summarize from human feedback, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS '20, Curran Associates Inc., Red Hook, NY, USA,

2020.

[39] R. Nakano, J. Hilton, S. Balaji, J. Wu, L. Ouyang, C. Kim, C. Hesse, S. Jain, V. Kosaraju, W. Saunders, et al., Webgpt: Browser-assisted question-answering with human feedback, arXiv preprint arXiv:2112.09332 (2021).

[40] Y. Bengio, J. Louradour, R. Collobert, J. Weston, Curriculum learning, in: Proceedings of the 26th annual international conference on machine learning, 2009, pp. 41–48.

[41] A. Yang, B. Yang, B. Hui, B. Zheng, B. Yu, C. Zhou, C. Li, C. Li, D. Liu, F. Huang, G. Dong, H. Wei, H. Lin, J. Tang, J. Wang, J. Yang, J. Tu, J. Zhang, J. Ma, J. Xu, J. Zhou, J. Bai, J. He, J. Lin, K. Dang, K. Lu, K. Chen, K. Yang, M. Li, M. Xue, N. Ni, P. Zhang, P. Wang, R. Peng, R. Men, R. Gao, R. Lin, S. Wang, S. Bai, S. Tan, T. Zhu, T. Li, T. Liu, W. Ge, X. Deng, X. Zhou, X. Ren, X. Zhang, X. Wei, X. Ren, Y. Fan, Y. Yao, Y. Zhang, Y. Wan, Y. Chu, Y. Liu, Z. Cui, Z. Zhang, Z. Fan, Qwen2 technical report, arXiv preprint arXiv:2407.10671 (2024).

[42] R. F. Woolson, Wilcoxon signed-rank test, Encyclopedia of biostatistics 8 (2005).

[43] R. Mehrotra, J. McInerney, H. Bouchard, M. Lalmas, F. Diaz, Towards a fair marketplace: Counterfactual evaluation of the trade-off between relevance, fairness & satisfaction in recommendation systems, in: Proceedings of the 27th acm international conference on information and knowledge management, 2018, pp. 2243–2251.

[44] Y. He, T. E. Kim, F. Diaz, J. Arguello, B. Mitra, Tip of the tongue query elicitation for simulated evaluation, arXiv preprint arXiv:2502.17776 (2025).

[45] T. E. Kim, F. Diaz, Towards fair rag: On the impact of fair ranking in retrieval-augmented generation, arXiv preprint arXiv:2409.11598 (2024).

A. Computational Cost

All inference experiments, except for ablations on models larger than 7B parameters, were conducted on a single L40 GPU. For efficiency, we distributed experiments across 2 to 32 GPUs depending on availability. Training was performed using 8 L40 GPUs for supervised fine-tuning on 7B parameter models, while ad-classifier training required only a single GPU.

B. Prompt for `QA System` response generation

> Answer the following web query, given the context.
> Context: {context}.
> Query: {query}.
>
> Reply only with a "well formed answer", i.e., human-like and coheseve. The answer should be written in a style where advertisements are easy to be integrated. Do not include any advertisements or explicit advertisement space, just write with the stylistic choice in mind.

C. Prompt for `Ad-Rewriter` rewritten response generation

> Consider this existing response to a conversational search query, and modify it to include an implicit advertisement.
> The advertisement should:
> - Mention this {type}: {item};
> - Highlight these qualities: {qualities};
> - Blend naturally with the original response;
> - Not appear overly promotional;
> - Be hard to detect by current textual ad-detection systems.
>
> Original query: {query}
> Original response: {response}
> Return only the modified response with the advertisement integrated.

D. Details on Synthetic Data Generation

D.1. Naive Synthetic Data

Example prompt for *NaiveSynthetic* query generation:

> Task: You are tasked with subtly integrating an advertisement into a search query response. The goal is to make the advertisement feel natural and helpful within the context of the response, not disruptive or overtly promotional. First, you should define the item to advertise. You should keep in mind the context of the query and original response. Consider the following advertisement qualities when choosing the product:
> * Product/Service Category.
> * Specific Advertisement Theme.
> * Key Selling Points/Qualities to Emphasize.
>
> Instructions for Integration:
> 1. Contextual Relevance is Key: The advertisement must feel relevant to the search query and the existing response text. Think about how the advertised product/service genuinely relates to the user's needs implied by the query.
> 2. Subtle and Natural Language: Use language that blends seamlessly with the original response. Avoid overly promotional or sales-oriented phrasing. Focus on informative and helpful language.
> 3. Focus on Benefits, Not Just Features: Instead of directly listing qualities, rephrase them as benefits the user might gain in relation to the search query and response context.
> 4. Strategic Placement: Choose the most natural and impactful location(s) within the response to subtly introduce the advertisement. This might involve:
> * Briefly modifying an existing sentence to subtly incorporate the advertisement.
> * Adding a short, relevant phrase or clause to an existing sentence.
> * In rare cases, adding a very short, contextual sentence (only if it feels truly natural).
> 5. Maintain Original Meaning: Ensure the core meaning and factual accuracy of the original response remain unchanged. The advertisement should enhance, not distort, the original information.
> 6. Review for Subtlety: Before returning the response, critically evaluate if the advertisement feels genuinely subtle and integrated. If it feels forced or obvious, refine your approach.
>
> Output: Return **only** the modified response with the subtly integrated advertisement.
>
> Search Query: {query}
> Original Response: {response}
> Modified Response:

The following versions of `Ad-Classifier` were trained using the *NaiveSynthetic* data:

- **V0.1**:
  - Multiple models, single data generation prompt.
  - https://huggingface.co/jmvcoelho/ad-classifier-v0.1
- **V0.2**:
  - Multiple models, multiple data generation prompt.
  - https://huggingface.co/jmvcoelho/ad-classifier-v0.2

The HuggingFace model pages contain the prompts used for insertion.

D.2. Structured Synthetic Data

D.2.1. List of infoboxes selected

product, brand, automobile, motorcycle, tractor, calculator, computing device, keyboard, software, camera, mobile phone, night vision device, synthesizer, tool, watch, pinball, toy, film, book, Asian comic series, comic, musical, furniture, video game, drug.

D.2.2. List of Wikidata properties selected

P50 (author), P86 (composer), P110 (illustrator), P123 (publisher), P162 (producer), P170 (creator), P176 (manufacturer), P178 (developer), P179 (product series), P287 (designed by), P593 (model number), P676 (lyricist), P943 (programmer), P3640 (National Drug Code), P4087 (MyAnimeList manga ID), P8731 (AniList manga ID), P9618 (AlternativeTo software ID), P9897 (App Store age rating), and P12969 (game designer)

D.2.3. Hard positive creation prompt

> Your task is to generate an indirect and implicit advertisement for a {infobox_name} named {product_name}.
> The advertisement
> * must not indicate that it is an advertisement or promotional content.
> * must include the {infobox_name} name, {product_name}.
> * must avoid any direct call to action.
> * must be brief and contained within one paragraph.
> * may present the {infobox_name} as part of natural, conversational, or informational content, or as a synthetic personal experience that could occur in real life.
> * may use testimonial or storytelling styles that describe the experiences of people with {page_title}.
> * may include detailed, scientific/research-backed statements.
>
> The following information about {page_title} may be useful for your writing:
> {summary}
> The advertisement can implicitly promote some of the following aspects of {page_title}: {key_features}
> Write only the advertisement without any explanations.

D.2.4. Hard negative creation prompt

> Your task is to write a concise, informative text about a {infobox_name} named {product_name}.
> The text:
> * must focus on delivering factual information.
> * must not include expressions of preference or favoritism toward {page_title} and should focus solely on the facts.
> * must include the name {product_name} at least once.
> * can mention other {infobox_name}s related to {page_title} to provide comprehensive information about the subject.
>
> The following information about {page_title} may be useful for your writing:
> {summary}
> Write only the informative text without any explanations.

The following versions of `Ad-Classifier` were trained with the *StructuredSynthetic* data:

- **V0.3**:
  - https://huggingface.co/teknology/ad-classifier-v0.3
- **V0.4**:
  - Trained by curriculum learning.
  - https://huggingface.co/teknology/ad-classifier-v0.4
- **V0.5**:
  - Trained by curriculum learning and data balancing.
  - https://huggingface.co/teknology/ad-classifier-v0.5