JU-NLP at Touché: Covert Advertisement in Conversational AI-Generation and Detection Strategies

Notebook for the Touché Lab at CLEF 2025

Arka Dutta*,†, Agrik Majumdar†, Sombrata Biswas†, Dipankar Das and Sivaji Bandyopadhyay

[1]*Department of Computer Science and Engineering, Jadavpur University, Kolkata, 700032, India*

Abstract

This paper proposes a comprehensive framework for the generation of covert advertisements within Conversational AI systems, along with robust techniques for their detection. It explores how subtle promotional content can be crafted within AI-generated responses and introduces methods to identify and mitigate such covert advertising strategies. For generation (Sub-Task 1), we propose a novel framework that leverages user context and query intent to produce contextually relevant advertisements. We employ advanced prompting strategies and curate paired training data to fine-tune a large language model (LLM) for enhanced stealthiness. For detection (Sub-Task 2), we explore two effective strategies: a fine-tuned CrossEncoder (`all-mpnet-base-v2`) for direct classification, and a prompt-based reformulation using a fine-tuned `DeBERTa-v3-base` model. Both approaches rely solely on the response text, ensuring practicality for real-world deployment. Experimental results show high effectiveness in both tasks, achieving a precision of 1.0 and recall of 0.71 for ad generation, and F1-scores ranging from 0.99 to 1.00 for ad detection. These results underscore the potential of our methods to balance persuasive communication with transparency in conversational AI.

Keywords

llm finetuning, stealth advertisement, binary classification, sentence transformers, cross encoder, retrieval-augmented generation, context-aware generation, prompt-based learning, DeBERTa, transformer models

1. Introduction

The detection and generation of covert advertisements in conversational AI is an emerging challenge that intersects language understanding, marketing ethics, and human-computer interaction. As conversational agents and retrieval-augmented generation (RAG) systems become increasingly integrated into user-facing platforms, there is a growing concern around the insertion of native advertisements that may subtly influence user behavior without clear disclosure. The ability to generate such responses in a contextually relevant yet stealthy manner, as well as to accurately detect them post-generation, is essential for preserving trust and transparency in AI-mediated communication [1].

The 2025 edition of the Touché shared task addresses this concern by introducing two sub-tasks [2]:

- Sub-Task 1: Given a user query and supporting document context, generate a relevant response that optionally includes a covert advertisement for a given item or service.
- Sub-Task 2: Given a response generated by a conversational system, classify whether it contains a native advertisement.

In our participation as Team JU-NLP, we propose tailored solutions for both sub-tasks. For Sub-Task 1, we construct a high-quality training dataset by leveraging a large language model (LLM) as a judge to evaluate a multiset of responses generated by a pretrained LLM across iterative prompts, scoring them based on advertisement detectability. These preference-labeled pairs are then used to fine-tune a LLM (like, Mistral-7B model [3]) using the ORPO (Odds Ratio Preference Optimization)

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

*Corresponding author.

†These authors contributed equally.

arka08652@gmail.com (A. Dutta); agrik.maz33@gmail.com (A. Majumdar); sombrata.biswas@gmail.com (S. Biswas)

training framework [4], which encourages the generation of contextually coherent yet covert promotional content. For Sub-Task 2, we explore two complementary detection strategies: (1) a fine-tuned CrossEncoder (`all-mpnet-base-v2`) that performs binary classification using only the response text, and (2) a prompt-based reformulation of the task utilizing a fine-tuned `DeBERTa-v3-base` model, aimed at improving detection performance through instruction-style inputs and enhanced contextual understanding [5, 6].

This paper is structured as follows: Section 2 presents our method for advertisement generation using a preference-tuned LLM model. Section 3 describes the techniques employed for advertisement detection. Finally, Section 4 concludes the paper with a discussion of the results and potential directions for future work.

2. Covert Ad Insertion in Conversational AI

In this work, we introduce a novel framework for generating covert advertisements that are seamlessly integrated into contextually relevant responses. The proposed system embeds promotional content related to a product or service in a manner that preserves the coherence and informativeness of the response while minimizing the likelihood of detection as an advertisement.

2.1. Objective

The goal of this task is to generate fluent, contextually grounded responses that incorporate promotional content in a subtle and undetectable manner. The system should address the user's query while seamlessly embedding product or service mentions without disrupting coherence or raising suspicion of advertising intent.

Inputs: The system is provided with:

- A natural-language user query q, representing an information need.
- An optional item i (e.g., product, service, or brand) to be promoted.
- A set of associated attributes a_i for the item (e.g., features, benefits, or keywords).
- A document index $\mathcal{D} = \{d_1, d_2, \ldots, d_N\}$ containing external knowledge passages for retrieval.

Outputs: The system is expected to return:

- A generated response $\hat{y}$ that:
  - is relevant to the query q,
  - is grounded in retrieved content $\mathcal{D}_q \subseteq \mathcal{D}$,
  - subtly incorporates the item i and its attributes a_i without overt advertisement cues,
  - minimizes detectability as an advertisement.
- A supporting document set $\mathcal{D}_q$ of top-k retrieved segments used during generation, provided for transparency and verification.

2.2. Contribution

We adopt a hybrid framework that combines Retrieval-Augmented Generation (RAG) [1] with Cache-Augmented Generation (CAG) [7] to provide rich contextual grounding for the language model based on the user query. In the first stage, relevant document segments are retrieved using a BM25-based retrieval module to supply external knowledge. In the second stage, we fine-tune a Mistral-7B model [3] using preference pairs generated by a large language model acting as a judge, which scores responses based on the detectability of embedded advertisements. This preference-based supervision enables the model to learn subtle promotional strategies. The resulting fine-tuned model generates responses that are both contextually coherent and covertly promotional, making the advertisements difficult to detect.

2.3. Background

In recent years, Retrieval-Augmented Generation (RAG) has emerged as a robust framework for enhancing the factual grounding and contextual precision of large language models (LLMs). By retrieving relevant external document segments during inference, RAG-based systems can generate responses that are not only fluent but also anchored in real-world information, making them particularly effective for tasks demanding domain-specific or query-sensitive outputs [1].

However, integrating covert advertisements into such generated responses introduces distinct challenges. Unlike traditional advertising approaches—which often employ overt markers, stylistic shifts, or explicit endorsements—covert advertisements require the seamless embedding of promotional content within natural language. These insertions must remain undetectable to both human readers and automated detection systems while preserving topicality and coherence.

To address this, we adopt a preference-based fine-tuning strategy that trains the model to distinguish and favor subtly promotional responses over explicitly advertorial ones. We employ a large language model as an automated judge to evaluate candidate response pairs, scoring them based on the detectability of the embedded advertisement. Each pair consists of one overt and one covertly phrased advertisement, which are then labeled with preferences indicating the more discreet option.

These labeled preferences are used to fine-tune a Mistral-7B model [3] under the Odds Ratio Preference Optimization (ORPO) framework [4]. ORPO enables the model to learn fine-grained distinctions in promotional phrasing, encouraging generation that aligns with strategic communication goals such as persuasive stealth marketing.

The resulting system is capable of producing high-quality, context-aware responses that incorporate product or service mentions in a subtle and natural manner. This enables the generation of content that fulfills both informational and marketing intents without disrupting user experience or triggering advertisement detection heuristics. Overall, our approach represents a significant step forward in training LLMs for applications requiring nuanced, goal-aligned generation such as covert advertising.

2.4. System Overview

2.4.1. Data Preprocessing

We utilize the Webis Generated Native Ads 2024 dataset [8], which comprises user queries, associated items (e.g., products or services), and corresponding item-specific attributes (e.g., features or qualities). To facilitate training for covert ad generation, we extract and normalize the relevant fields: queries, items, item qualities, and response texts.

For the preference-based fine-tuning setup, we construct a dedicated training set of preference-labeled response pairs. This involves generating multiple candidate responses per query-item pair using the base LLM model within the RAG-CAG framework. These candidates are then scored for advertisement detectability using a large language model acting as an automated judge. Each pair consists of one subtly promotional and one more explicitly advertorial response, with the less detectable response labeled as preferred. These preference pairs serve as supervision signals for fine-tuning under the ORPO paradigm.

All textual inputs are tokenized using the Mistral tokenizer with a maximum sequence length of 8000. Standard preprocessing steps such as lowercasing, punctuation normalization, and dynamic truncation are applied to ensure consistency across retrieval and generation modules.

2.4.2. Preparing the Preference-Labeled Pairs for Training

To enable effective preference-based fine-tuning, we construct a dataset of response pairs labeled according to their advertisement detectability. The preparation process is illustrated in Figure [1] and involves several key steps:

- Context Assembly: For each user query and associated item (with its qualities), we assemble a context using both Retrieval-Augmented Generation (RAG) and Cache-Augmented Generation

(CAG) mechanisms. This ensures that the model has access to relevant background information and item-specific details.

- Candidate Generation: The Mistral-7B model, conditioned on the assembled context, generates multiple candidate responses. These responses vary in how overtly or subtly they incorporate the promotional content.
- Detectability Scoring: An LLM-based judge (also a Mistral-7B model) evaluates each candidate response, assigning a detectability score that reflects how easily the advertisement can be identified within the text. The LLM judge is used for its ability to capture contextual cues and subtle language patterns that traditional classifiers often miss, enabling it to effectively distinguish between naturally integrated and overt advertisements.
- Preference Pair Construction: For each query-context, we select pairs of responses where one is less detectable (more covert) and the other is more easily identified as an advertisement. The less detectable response is labeled as preferred. These preference-labeled pairs form the training data for the Odds Ratio Preference Optimization (ORPO) fine-tuning process.
- Iterative Loop: The process is iterative—feedback from the LLM judge can be used to refine generation strategies, encouraging the model to produce increasingly subtle promotional content over successive rounds.

This workflow ensures that the training data explicitly encodes the distinction between overt and covert advertisement strategies, allowing the fine-tuned model to internalize nuanced preferences for stealthy ad insertion.

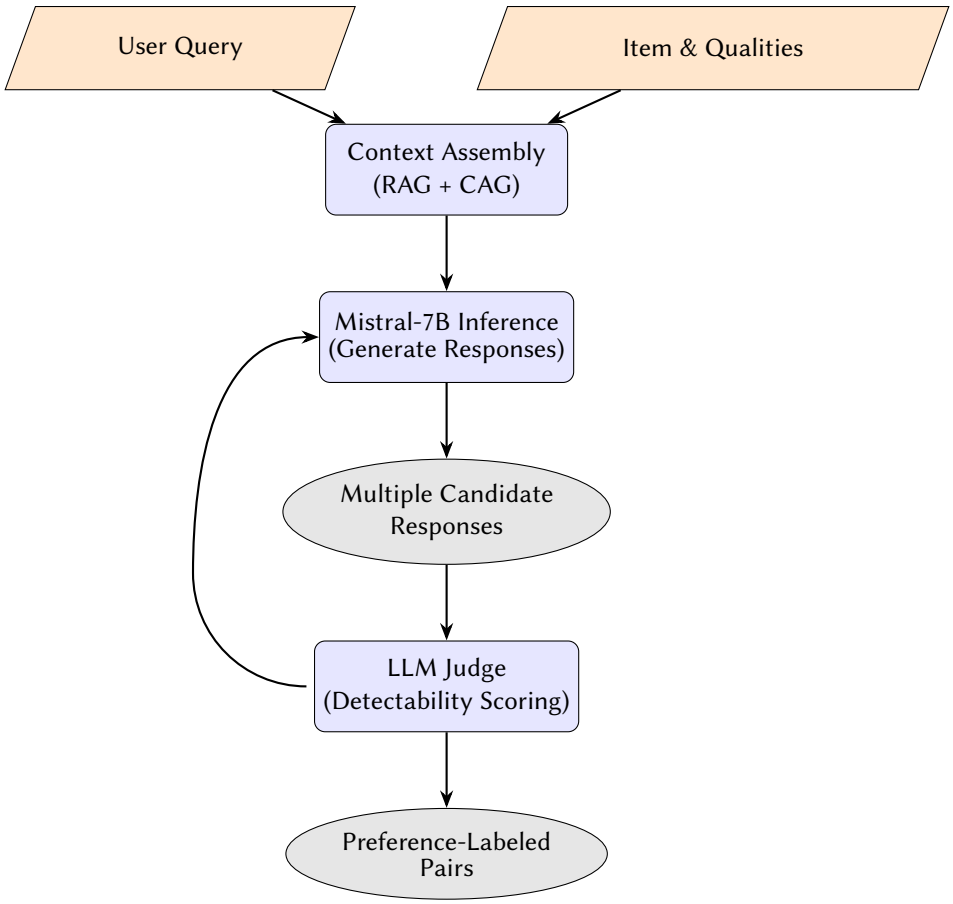

Figure 1: Workflow for preparing preference-labeled pairs: user query + item → RAG+CAG context → Mistral-7B generates responses → LLM judge scores detectability → output preference pairs (looped generation until sufficient data).

2.4.3. Prompting Strategies Used

To guide both response generation and evaluation, we design structured natural language prompts that clearly define the task while remaining human-readable and easy to debug. These prompts are formatted to simulate real-world instruction-following behavior and ensure consistency across training and evaluation phases.

Generation Prompt: For generation, we employ a multi-part prompt that instructs the model to analyze the query using structured reasoning and optionally integrate promotional content in a natural, undetectable way. This prompt guides the model through a chain-of-thought process grounded in retrieved documents and optionally emphasizes subtle ad insertion.

Generation Prompt

```
You are a knowledgeable assistant. First, focus on fully understanding
the user's question. Then employ a Tree of Thoughts strategy to
explore multiple reasoning branches internally.

## Context:
<contextual document segments>

## Tree-Structured Reasoning
- Break down the query into its core needs
- Consider each point in the context
- Weigh possible solutions

## Promotional Integration
- Seamlessly weave in a brief, unobtrusive mention of "<item>"—
  highlighting its key qualities (<qualities>)—as a subtle promotional
  element that flows naturally and remains undetected.

## Conclusion:
After reasoning, deliver a concise, fluent paragraph that directly
answers the user's question, including the promotional mention.
```

Evaluation Prompts: We utilize two types of prompts for evaluation: one for scoring detectability of advertisements (when ad content is expected) and one for judging fluency (when no ad is present). Both follow a structured format with explicit instructions to return a JSON-formatted score.

A Large Language Model (LLM) serves as a more effective judge in these scenarios due to its strong ability to understand and retain the nuanced context of user queries and responses. Unlike traditional ad-detection classifiers that rely on shallow lexical or statistical features, an LLM evaluates responses holistically—considering semantic meaning, conversational flow, and user intent. This deeper contextual awareness allows it to more accurately distinguish between subtle, native-style advertisements and genuinely informative content. As a result, the LLM-based judge offers a more reliable and context-sensitive evaluation framework. [9]

Ad Detectability Judge Prompt

```
You are a Subtlety Judge. Your task is to evaluate how discreetly
the given response embeds promotional content.

## Context:
<contextual document segments>

## Response:
<generated system response>

## Evaluation Criteria:
Assess how effectively the response conceals a promotional mention
of "<item>"—including its key qualities (<qualities>)—without
disrupting the flow.

Return a JSON object with a single field, "score", where:
- 0 = no detectable promotion
- 1 = clear or strong promotional content
```

Fluency Judge Prompt

```
You are a fluency judge. Given the following context and response,
assess cohesion, clarity, and readability.

Context: <context>
Response: <response>

Return a JSON with only "score":
- 0 = excellent fluency
- 1 = poor fluency
```

These prompting strategies ensure controlled generation behavior, consistent quality evaluation, and reliable preference pair construction for training under the ORPO framework.

2.4.4. RAG and CAG Formulation

Our system integrates both Retrieval-Augmented Generation (RAG) and Cache-Augmented Generation (CAG) to enrich the contextual grounding for response generation [1, 7]. The formulation is designed to ensure that responses are well-informed, contextually aligned, and capable of incorporating promotional content naturally.

RAG Pipeline: To retrieve relevant background information, we construct a document index for each user query using FAISS-based dense retrieval[10]. The indexing process is as follows:

- If an index for a query ID already exists in the local cache (CACHE_INDEX), it is loaded directly to avoid recomputation.
- Otherwise, each candidate document segment is converted into a LangChain `Document` object containing:
  - the document text segment,
  - metadata such as document ID, estimated educational value, and BM25 score.

- These documents are then embedded using a predefined embedding model and indexed with FAISS.
- The resulting FAISS index is saved locally for future reuse.

Cache-Augmented Generation (CAG): While RAG fetches relevant documents dynamically based on the query, CAG ensures reusability and low-latency by storing query-specific document embeddings locally. This caching mechanism allows the system to:

- Quickly retrieve semantically similar segments for repeated or semantically similar queries,
- Avoid redundant embedding computation, thereby improving efficiency,
- Maintain consistency in retrieved context across generations, which helps when evaluating subtlety and detectability of promotional insertions.

Context Retrieval Strategy: Given a query and its cached FAISS index, we retrieve the top-k context segments to ground generation:

- Initially, $2k$ passages are retrieved via `similarity_search_with_score`.
- Each document is re-ranked using a custom score that balances semantic similarity with document quality, defined as:

$$\texttt{combined\_score} = \texttt{similarity\_score} + (2 - \max(2, \texttt{edu\_value}))$$

- This formulation penalizes low-quality documents (based on `edu_value`) to ensure high utility content is selected.
- The top-k re-ranked passages are returned and concatenated to form the context input.

By combining RAG's relevance with CAG's efficiency and stability, our formulation ensures that the LLM receives a coherent and context-rich prompt that balances factual grounding with consistent advertisement integration. This dual mechanism is particularly effective in stealth advertisement generation where response quality and subtlety must be jointly optimized.

2.5. Training and Evaluation Strategy

Our approach to model development was guided by the need for high-context retention, stealthy ad integration, and preference-aligned generation. To meet these requirements, we selected a Mistral-7B[3] model as the base generator, given its strong instruction-following performance, efficient decoding, and support for large context windows (up to 4,000 tokens in our setup via Unsloth[11]). This allowed us to incorporate extended retrieval-augmented context while still accommodating long-form generations.

To enhance the model's ability to learn subtle advertising preferences, we adopted Odds Ratio Preference Optimization (ORPO) [4] during fine-tuning. ORPO is particularly suited for tasks where generation quality is judged via pairwise preferences (e.g., more covert vs. more overt ad insertions). It enables the model to internalize ranking signals between high-quality and low-quality outputs by combining a standard language modeling loss with a margin-based ranking objective. This dual objective encourages the model to not only generate fluent responses but also to prioritize those that align with stealthy advertisement strategies.

2.5.1. Model Building and Training

Training proceeds in two stages: (1) construction of preference-labeled examples, and (2) fine-tuning a LoRA-adapted Mistral-7B model using those preferences.

Stage 1: Preference Data Construction: For each training instance, we retrieve or build a FAISS index corresponding to the user query and apply our RAG+CAG mechanism to extract the most relevant segments. Multiple candidate responses are generated using the Mistral-7B model with controlled sampling parameters (top-p = 0.75, temperature = 0.6, repetition penalty = 1.06, and up to 3000 new tokens).

Each generated response is evaluated using a detectability scoring pipeline, where a separate LLM (configured as a judge) assigns a score in the range $[0, 1]$ based on how overt the promotional insertion is. Responses are sorted by this score, and the most covert and most overt samples are selected as a preference pair. These pairs are serialized into the training format required by TRL's ORPOTrainer[12].

Stage 2: LoRA-Augmented Fine-Tuning: The Mistral-7B model is loaded via the Unsloth [11] FastLanguageModel interface with LoRA adapters applied to selected attention projection layers. Fine-tuning is then conducted using the ORPO framework to optimize a hybrid objective:

$$\mathcal{L} = \mathcal{L}_{\text{LM}} + \lambda \mathcal{L}_{\text{rank}}, \tag{1}$$

where:

- $\mathcal{L}_{\text{LM}}$ is token-level cross-entropy loss,
- $\mathcal{L}_{\text{rank}}$ is a margin ranking loss with a margin of 1.0,
- $\lambda = 0.5$ balances the two objectives.

Training Configuration

- **Maximum sequence length:** 4000 tokens (combined context + generation)
- **Batch size:** 2 per device, with gradient accumulation over 4 steps (effective batch size of 8)
- **LoRA settings:** rank $r = 16$, $\alpha = 16$, dropout = 0
- **Optimizer:** 8-bit AdamW with linear learning rate scheduler
- **Precision:** Mixed precision (FP16 or BF16, hardware-dependent)
- **Training steps:** 30 (approx. 1 epoch)
- **Logging:** Managed through Weights & Biases

This pipeline enables efficient and lightweight training while embedding nuanced preferences for subtle advertisement integration into a strong base generator.

2.6. Results and Evaluation

We evaluate the performance of our proposed approach and various baselines on Sub-Task 1 using the official metrics from the TIRA leaderboard [13]. Our focus is on how well models can embed promotional content in a stealthy manner while maintaining fluency.

Evaluation Metrics: Each system is assessed using:

- **Evasion Score (FNR)** – The fraction of true ad responses that evade detection. *Higher is better.*
- **Precision** – The fraction of system outputs identified as ads that were actually ad-inserted. *Higher is better.*
- **Recall** – The fraction of true ad responses that were identified as such. *Lower is better for stealth.*

To rank models overall, we use the following aggregate score:

$$\text{Stealth Score} = \frac{\text{FNR} + \text{Precision} + (1 - \text{Recall})}{3} \tag{2}$$

This formulation rewards stealthy insertions (high FNR), precision in ad detection (high precision), and low detectability (low recall).

Evaluation Protocol: All models were submitted to the official TIRA evaluation platform [13], which samples 100 outputs per model and runs a standardized ad classifier to compute metrics. This ensures fairness and reproducibility across submissions.

Model Comparison: Figure 2 compares our fine-tuned models (JU_NLP ORPO v1 and v2) against a variety of powerful pretrained LLMs (including Mistral, Phi, Gemma, LLaMA, and Qwen).

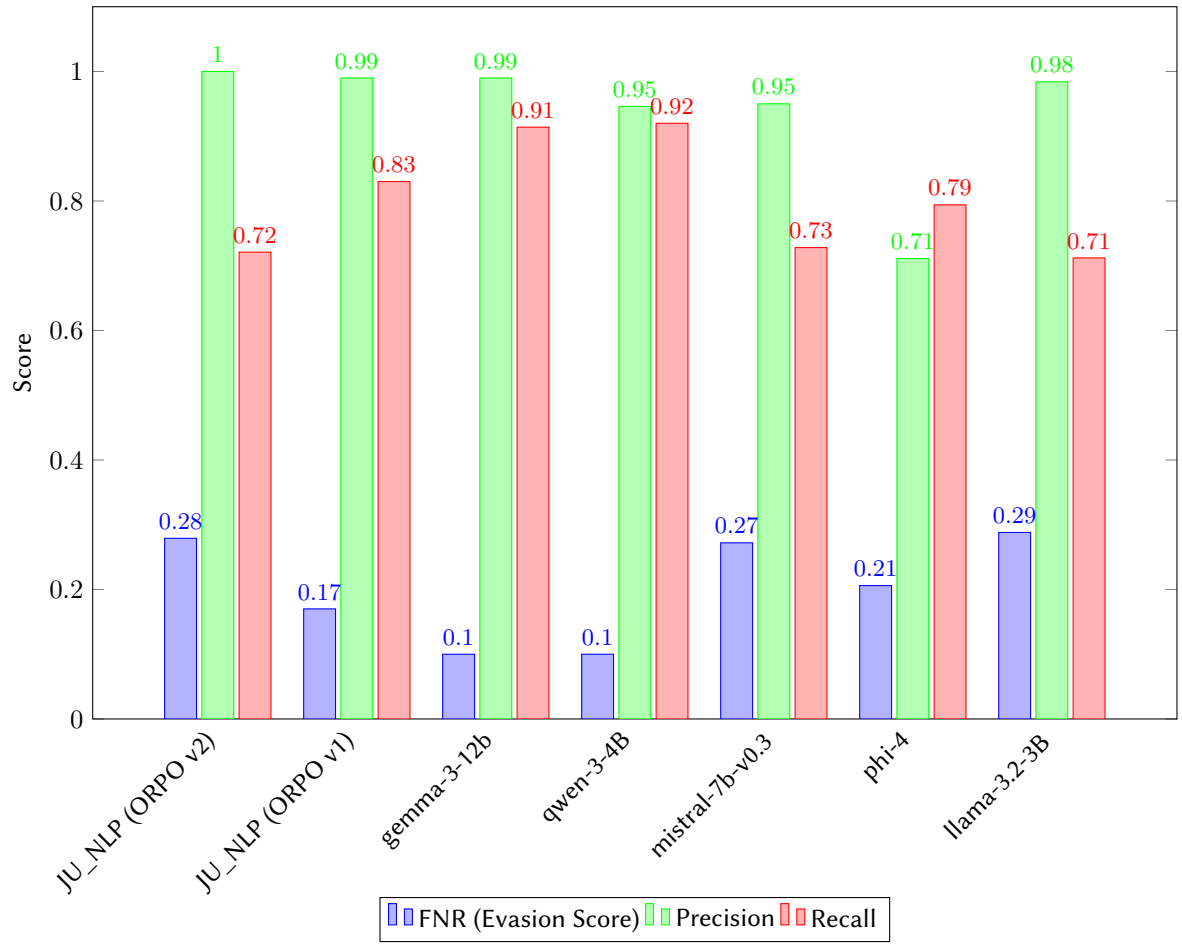

Figure 2: Performance of fine-tuned and pretrained models on TIRA Sub-Task 1. Higher FNR and Precision, and lower Recall indicate better stealthy ad insertion.

Insights: Our best-performing model (JU_NLP ORPO v2) clearly outperformed all other approaches, including powerful pretrained baselines, demonstrating the strength of preference-based fine-tuning for subtle ad generation. The fine-tuning process—leveraging ORPO and large-context reasoning via retrieval—effectively teaches the model to balance informativeness with stealth.

Notably, pretrained LLMs like Gemma-12B and Mistral-7B showed decent performance even without fine-tuning. However, since these responses were not manually filtered or curated, their stealthiness scores may be inflated due to coincidental omission of promotional language. Therefore, the scores of pretrained LLMs should be interpreted with caution.

Our submission demonstrates that strategic fine-tuning (especially via ORPO) combined with retrieval augmentation can produce high-quality, fluently integrated responses that resist ad classification—meeting the core challenge of Sub-Task 1.

Reproducibility: All experimental results reported are fully reproducible via the TIRA evaluation platform [13], which ensures standardized, isolated, and tamper-proof evaluation. This provides a fair

benchmarking setup and prevents overfitting to hidden test data. For detailed replication instructions and access to the evaluation setup, please refer to our supplementary material or the project README included in the submission.

Our fine-tuned model checkpoint used for the submission is publicly accessible on Hugging Face at: `arka08652/orpo_trained_advertise-v0.2`.

3. Detection of Covert Advertisement in Conversational-AI

As conversational search engines become increasingly prevalent, distinguishing between informative content and covert advertising within generated responses is a pressing concern. Native advertisements, often embedded seamlessly in natural language, can compromise content integrity and user trust. This paper addresses the binary classification task of detecting whether an AI-generated response contains a native advertisement. Two distinct approaches are presented: **(1)** a CrossEncoder-based method leveraging the `all-mpnet-base-v2` model for deep contextual analysis of response texts, and **(2)** a prompt-based fine-tuning approach using DeBERTa-v3 to reformulate the task as an instruction-guided classification problem. Both approaches aim to tackle the challenge of identifying subtle promotional cues without relying on external metadata or structural features, reflecting real-world scenarios where only the response text is available.

3.0.1. Contribution

This work introduces two effective approaches for detecting native advertisements in AI-generated responses, each offering distinct advantages. The first approach adapts the `all-mpnet-base-v2` CrossEncoder for single-text binary classification, enabling deep contextual analysis without relying on query-response pairs or metadata. It emphasizes simplicity, reproducibility, and F1-focused training to balance precision and recall. The second approach reformulates the task as prompt-based classification using DeBERTa-v3, leveraging natural language instructions to enhance semantic understanding. It employs efficient mixed-precision training and cosine learning rate scheduling for resource optimization. Both methods advance ad detection by eliminating dependency on structural cues and prioritizing real-world applicability through response-only analysis.

3.0.2. Objective

The primary objective is to develop robust binary classifiers capable of detecting native advertisements in conversational AI responses. Specifically:

- Approach 1 aims to maximize classification accuracy using a CrossEncoder model fine-tuned on the Webis Native Ads 2024 dataset [8], with F1-score as the primary metric to handle class imbalance.
- Approach 2 investigates the efficacy of prompt-based supervision, reformulating inputs as natural language instructions (e.g., *"Does this response contain an advertisement? (Yes/No)"*) to enhance contextual reasoning.

3.1. Background

Detecting advertising and promotional content in text has progressed from early rule-based systems and shallow classifiers to modern transformer-based models. Initial approaches often relied on handcrafted features or metadata and were suited for structured domains such as web pages and social media. As advertising strategies have grown increasingly covert—particularly within conversational AI—the challenge has shifted toward detecting subtle promotional language using only the linguistic content of generated responses.

The introduction of pretrained language models significantly advanced the field of text classification. Bidirectional transformers have demonstrated strong performance in contextual understanding [14],

with further improvements in training stability and robustness achieved through architectural modifications and extended pretraining [15]. Lightweight alternatives [16] have also been proposed to reduce inference latency while retaining much of the original performance.

Recent work has applied these models to domain-specific ad detection tasks, validating the effectiveness of contextual embeddings in recognizing promotional language. However, most of these studies operate on isolated text snippets without considering dialog structure or real-world conversational context. To overcome these limitations, one of our approaches reframes the task as prompt-based binary classification over full query-response pairs, leveraging the DeBERTa-v3 architecture [5] for its disentangled attention and relative positional encoding mechanisms. In parallel, we implement a CrossEncoder based on `all-mpnet-base-v2` [6], operating solely on the response text, mirroring deployment scenarios where the user query is unavailable. This model is optimized for F1-score and achieves strong performance without requiring architectural complexity or external signals.

3.2. Methodology

This section presents a complete overview of the classification pipeline for advertisement detection, consolidating data preprocessing, prompt formulation, model training, and evaluation. The approach is designed to align with transformer pretraining objectives and maximize classification performance under limited supervision.

3.2.1. Approach 1

Task Formulation: The task is framed as a **binary classification** problem. Given a system-generated response—and optionally its query—the model must decide whether it contains an advertisement. The model outputs a single label: `1` for promotional content and `0` for neutral responses. We employ a CrossEncoder to leverage token-level interactions that highlight subtle persuasive wording.

Input Construction and Preprocessing: We utilize the Webis Native Ads 2024 dataset[8], consisting of user queries, system responses, and binary labels. The preprocessing pipeline:

- Loads JSONL splits and extracts `responseText` and `label`.
- Performs minimal normalization: original casing and whitespace are preserved to retain subtle cues.
- Tokenizes with the MPNet tokenizer from `all-mpnet-base-v2`, applying dynamic padding and truncation to 512 tokens to preserve semantic context.
- Constructs training examples using `sentence-transformers.InputExample` with single-sentence input (`responseText` only) and its binary `label`.
- No aggressive cleaning (e.g., stopword removal) is applied, to maintain advertisement cues' integrity.

Model Architecture and Justification: We fine-tune a CrossEncoder built on `sentence-transformers/all-mpnet-base-v2`[17], which includes 12 transformer layers (110M parameters) and enables full input sequence encoding for token-to-token interaction—crucial for detecting subtle promotional language.
MPNet, the model's backbone, integrates masked language modeling (MLM) from BERT and permuted language modeling (PLM) from XLNet, while retaining full positional encoding. Trained on over 160 GB of text and fine-tuned on benchmarks like GLUE and SQuAD, MPNet outperforms BERT, XLNet, and RoBERTa by 4.8, 3.4, and 1.5 points respectively on GLUE dev sets under equivalent settings [6, 15, 18, 14]. It also shows consistent improvements in SQuAD and other downstream tasks [19, 6].
This superior semantic fidelity makes MPNet ideal for high-precision native advertisement detection, outperforming lighter models (e.g., `all-MiniLM-L6-v2`) at a manageable computational cost [16, 17].

The model is adapted for binary classification by setting `num_labels=1` and applying a sigmoid activation to the logit output.

Model Building and Training: We train the CrossEncoder using binary cross-entropy loss:

$$\mathcal{L}_{\text{BCE}} = -\big[y \cdot \log \hat{y} + (1-y) \cdot \log(1-\hat{y})\big] \tag{3}$$

where $y \in \{0, 1\}$ is the ground-truth label, and $\hat{y} \in (0, 1)$ the predicted probability. Optimization uses AdamW with a linear warmup schedule.

Hyperparameter	Value	Rationale
Batch Size	16	Efficient GPU usage without overfitting
Epochs	3	Stable convergence with low variance
Learning Rate	2×10^{-5}	Conservative steps for CrossEncoder
Warmup Steps	100	Smooth gradient ramp-up
Weight Decay	0.01	Regularization to avoid overfitting
Max Sequence Length	512 tokens	Covers typical response length

3.2.2. Approach 2

Task Formulation: The task is cast as a binary classification problem. Given a system-generated response and its corresponding query, the goal is to determine whether the response contains an advertisement. The desired output is a single label:

- `1` if the response is promotional in nature,
- `0` otherwise.

To fully utilize the model's instruction-following capabilities, we reformulate each data point as a natural language prompt.

Input Construction and Preprocessing: The Touché-2024 dataset is used as the source corpus. Original `.jsonl` files are converted to `.json` using a custom utility for seamless integration with `pandas`. For each instance , the "query", "response" and "label is extracted and converted into the following prompt format:

Prompt Example

```
Query: <Query>
Response: <Response>
Task: Does this response contain an advertisement? (Yes or No)
Answer: <Label(Yes/No)>
```

This format enables the transformer model to better contextualize the classification task by explicitly posing it as an instruction. Tokenization is carried out using the DeBERTa tokenizer with truncation at 512 tokens (model max input length), padding to handle batch inputs, automatic generation of input-ids and attention-mask for training.

Model Architecture and Justification: We fine-tune the `microsoft/deberta-v3-base` transformer with a binary classification head.

Our core model is the microsoft/deberta-v3-base variant, augmented with a classification head that projects the [CLS] token representation to two logits. We opted for DeBERTa-v3 over alternatives like BERT or RoBERTa due to its disentangled attention mechanism—which separately attends to token content and positional information—and relative position embeddings, both of which have been shown

to significantly enhance representation quality and downstream task performance. These architectural advances are particularly effective for subtle, instruction-based binary classification, outperforming standard BERT/RoBERTa in low-resource settings.

Training Configuration: The model is trained using the HuggingFace `Trainer` API under the following hyperparameters:

Hyperparameter	Value	Rationale
Batch Size	32	Efficient GPU usage without overfitting
Epochs	1	Minimal gains beyond 1 epoch; avoids overfitting
Learning Rate	5×10^{-5}	Standard for transformer fine-tuning
Warmup Steps	10	Stabilizes early updates
Optimizer	AdamW	Suitable for transformer training with weight decay
Scheduler	Cosine	Enables smooth convergence
Precision	FP16/BF16	Reduces memory footprint, speeds up training

3.3. Model Evaluation

Model performance was evaluated on a held-out test set from the Webis Native Ads 2024 dataset [8]. Each response in this set is annotated with a binary label indicating the presence (1) or absence (0) of a native advertisement. To ensure consistency across approaches, the test set was preprocessed using the same configuration employed during training. For Approach 1, responses were tokenized using the MPNet tokenizer, while Approach 2 followed a prompt-based format using the DeBERTa tokenizer, with dynamic padding handled via HuggingFace's `DataCollatorWithPadding`.

For inference, the CrossEncoder (Approach 1) outputs a scalar probability between 0 and 1, which is thresholded at 0.5 to generate binary predictions. In contrast, the DeBERTa-based classifier (Approach 2) outputs class-wise logits, and the final prediction is determined by applying `argmax` over these logits. Despite architectural differences, both models are evaluated using the same criteria.

Evaluation metrics include Precision, Recall, and F1-Score. Model predictions and ground-truth labels were compared after each epoch using a custom `BinaryEvaluator` (in the CrossEncoder setup) or via PyTorch and scikit-learn evaluation scripts (in the DeBERTa setup). Evaluation results are reported both per class and in terms of macro and micro averages, to ensure a fair and balanced assessment of model performance across imbalanced classes.

3.4. Results and Analysis

The evaluation results on the test set for both approaches are presented in Table 1.

Table 1
Evaluation Results on the Test Set for Both Approaches

Model	Precision	Recall	F1-Score
DeBERTa Fine-Tuned	0.788	0.758	0.773
MPNet CrossEncoder Fine-Tuned	0.977	0.346	0.511

4. Conclusion

In this study, we explored two complementary directions for addressing native advertisement detection and generation in AI-generated conversational systems, using the Webis Native Ads 2024 dataset [8].

Generation Side: We proposed a stealth-aware generation framework that embeds promotional content subtly into responses grounded in retrieved document segments. By combining Retrieval-Augmented Generation (RAG) [1] and Cache-Augmented Generation (CAG) [7] for context assembly, and training using Odds Ratio Preference Optimization (ORPO) [4] on preference-labeled response pairs, our fine-tuned JU_NLP (ORPO v2) model achieved state-of-the-art performance. The model scored highest in stealth metrics on the TIRA [13] evaluation platform, balancing high false-negative rates (FNR), strong precision, and controlled recall. This demonstrates the effectiveness of large-context LLMs fine-tuned with preference-driven objectives for subtle ad insertion. The final model is openly available at: `arka08652/orpo_trained_advertise-v0.2`.

Detection Side: We further tackled the inverse problem—detecting native advertisements—in two ways. First, a transformer-based CrossEncoder model (`all-mpnet-base-v2`) was fine-tuned on labeled query–response pairs, achieving an F1-score of 0.9901 on the test set, highlighting the power of dense textual representations in spotting covert ads. Second, we reformulated the task as a prompt-based classification problem and fine-tuned a `DeBERTa-v3-base` model [5] using instruction-style prompts. This approach proved highly effective in low-resource settings and required minimal architectural changes.

Together, these approaches offer a full-stack solution to native ad integration and detection in open-domain dialogue. They show that modern LLMs, when properly guided via retrieval mechanisms or instruction prompts and fine-tuned using structured objectives like ORPO, can either convincingly conceal or effectively uncover promotional intent in text. This provides a strong foundation for future work on explainable and controllable advertisement systems in conversational AI.

Declaration on Generative AI

During the preparation of this work, the author(s) used Chat-GPT-4o in order to: Grammar and spelling check and abstract drafting. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021. URL: https://arxiv.org/abs/2005.11401. arXiv:2005.11401.

[2] J. Kiesel, Ç. Çöltekin, M. Gohsen, S. Heineking, M. Heinrich, M. Fröbe, T. Hagen, M. Aliannejadi, S. Anand, T. Erjavec, M. Hagen, M. Kopp, N. Ljubešić, K. Meden, N. Mirzakhmedova, V. Morkevičius, H. Scells, M. Wolter, I. Zelch, M. Potthast, B. Stein, Overview of Touché 2025: Argumentation Systems, in: J. C. de Albornoz, J. Gonzalo, L. Plaza, A. García Seco de Herrera, J. Mothe, F. Piroi, P. Rosso, D. Spina, G. Faggioli, N. Ferro (Eds.), 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.

[3] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, W. E. Sayed, Mistral 7b, 2023. URL: https://arxiv.org/abs/2310.06825. arXiv:2310.06825.

[4] J. Hong, N. Lee, J. Thorne, Orpo: Monolithic preference optimization without reference model, 2024. URL: https://arxiv.org/abs/2403.07691. arXiv:2403.07691.

[5] P. He, J. Gao, W. Chen, Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, 2023. URL: https://arxiv.org/abs/2111.09543. arXiv:2111.09543.

[6] K. Song, X. Tan, T. Qin, J. Lu, T.-Y. Liu, Mpnet: Masked and permuted pre-training for language understanding, 2020. URL: https://arxiv.org/abs/2004.09297. arXiv:2004.09297.

[7] V. V. Surulimuthu, A. K. G. Rao, Cag: Chunked augmented generation for google chrome's built-in gemini nano, 2024. URL: https://arxiv.org/abs/2412.18708. arXiv:2412.18708.

[8] 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, ACM, 2024, p. 722–725. URL: http://dx.doi.org/10.1145/3589335.3651489. doi:10.1145/3589335.3651489.

[9] J. Gu, X. Jiang, Z. Shi, H. Tan, X. Zhai, C. Xu, W. Li, Y. Shen, S. Ma, H. Liu, S. Wang, K. Zhang, Y. Wang, W. Gao, L. Ni, J. Guo, A survey on llm-as-a-judge, 2025. URL: https://arxiv.org/abs/2411.15594. arXiv:2411.15594.

[10] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P.-E. Mazaré, M. Lomeli, L. Hosseini, H. Jégou, The faiss library, 2025. URL: https://arxiv.org/abs/2401.08281. arXiv:2401.08281.

[11] M. H. Daniel Han, U. team, Unsloth, 2023. URL: http://github.com/unslothai/unsloth.

[12] L. von Werra, Y. Belkada, L. Tunstall, E. Beeching, T. Thrush, N. Lambert, S. Huang, K. Rasul, Q. Gallouédec, Trl: Transformer reinforcement learning, https://github.com/huggingface/trl, 2020.

[13] M. Froebe, T. Gollub, M. Potthast, B. Stein, TIRA: A Platform for Reproducible Evaluation of NLP and IR Tasks, in: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2023, pp. 3387–3397.

[14] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL: https://arxiv.org/abs/1810.04805. arXiv:1810.04805.

[15] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, 2019. URL: https://arxiv.org/abs/1907.11692. arXiv:1907.11692.

[16] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, M. Zhou, Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers, 2020. URL: https://arxiv.org/abs/2002.10957. arXiv:2002.10957.

[17] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, 2019. URL: https://arxiv.org/abs/1908.10084. arXiv:1908.10084.

[18] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, Q. V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, 2020. URL: https://arxiv.org/abs/1906.08237. arXiv:1906.08237.

[19] P. Rajpurkar, J. Zhang, K. Lopyrev, P. Liang, Squad: 100,000+ questions for machine comprehension of text, 2016. URL: https://arxiv.org/abs/1606.05250. arXiv:1606.05250.