Code-Doctors at Touché: Cross-Domain Argument Detection and Advertisement Removal in RAG Responses

Touché at CLEF 2026

Ahsan Siddiqui[1,†], Muhammad Mansoor Alam[1,*,†], Zohaib Aslam[1,†], Faisal Alvi[1] and Abdul Samad[1]

[1]*Dhanani School of Science and Engineering, Habib University, Karachi, Pakistan*

Abstract

This paper presents our system notebook for two Touché 2026 shared tasks. Task 3 requires predicting whether a sentence constitutes an argument, across ten benchmark datasets with differing annotation styles, where the central challenge is cross-domain generalizability. Task 4 requires detecting, locating, and removing hidden advertisements embedded in AI-generated search responses, contributing to transparent and trustworthy AI-driven information retrieval. For Task 3, we explore five approaches ranging from a Random Forest baseline to DeBERTa and RoBERTa fine-tuning with domain prefixes, a sliding-window RAG pipeline, and LLM-summarized annotation guidelines, achieving a Macro F1 of 0.7960. For Task 4, we fine-tune RoBERTa for response-level ad classification (F1 = 0.9947), sentence-pair ad-span identification (overlap F1 = 0.9189), and apply rule-based span removal for clean, ad-free responses.

Keywords

Argument Mining, Generalizability, Advertisement Detection, Retrieval-Augmented Generation (RAG), Large Language Models, Advertisement Detection, Deep Learning, Transformer Models, CLEF 2026

1. Introduction

This paper presents our participation in two shared tasks from the Touché Lab at CLEF 2026 [3, 4]. We address cross-domain argument detection and native advertisement detection and removal in RAG-generated responses. Both tasks require models that generalize beyond their training distribution; to unseen argument styles in Task 3, and to embedded promotional content in Task 4.

Task 3 – Generalizability of Argument Identification in Context[1]. Argument mining models often fail when tested on datasets they were not trained on. Feger et al. [1] show that BERT, RoBERTa, and WRAP all drop by 10–20 macro-F1 points on unseen datasets. The reason is *shortcut learning*: models pick up on topic-specific keywords and treat them as argument signals, instead of learning what actually makes a sentence an argument. Task 3 gives each sentence its source document and annotation guidelines as extra context, and asks participants to predict `Argument` or `No-Argument`. effectiveness is measured using macro-F1 across 10 benchmark datasets (ABSTRCT, ACQUA, AEC, AFS, ARGUMINSCI, FINARG, IAM, PE, SCIARK, USELEC) plus a new held-out set, so models that memorize training-specific patterns will be penalized.

Task 4 – Advertisement in Retrieval-Augmented Generation[2]. As AI-powered search engines become more common, there is growing concern about hidden advertisements being slipped into generated responses. The Webis Generated Native Ads 2025 dataset [2] was built by taking real organic

CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany

*Corresponding author.

†These authors contributed equally.

as08155@st.habib.edu.pk (A. Siddiqui); ma08322@st.habib.edu.pk (M. M. Alam); za08134@st.habib.edu.pk (Z. Aslam); faisal.alvi@sse.habib.edu.pk (F. Alvi); abdul.samad@sse.habib.edu.pk (A. Samad)

[1]https://touche.webis.de/clef26/touche26-web/generalizable-argument-mining.html

[2]https://touche.webis.de/clef26/touche26-web/advertisement-detection.html

responses from search engines (Copilot, YouChat) and inserting ads using GPT-4 Turbo, written to blend in naturally with the surrounding text. The dataset covers 17 topic categories and includes character-level labels showing exactly where each ad is. Task 4 has three sub-tasks: classify whether a response contains an ad (Sub-Task 1, F1), find the exact character spans of the ad text (Sub-Task 2, overlap F1), and rewrite the response to remove the ad while keeping it fluent and relevant (Sub-Task 3, scored by human annotators and an LLM judge on fluency, relevance, and correctness).

2. Literature Review

2.1. Task 3 Context: Touché 2022–2025

In 2022 [6], Team Katana used ColBERT [7] to retrieve argumentative passages. ColBERT encodes the query and document separately, then scores them by matching tokens, which makes it faster than a full cross-encoder. While it worked well for finding topically relevant passages (competitive nDCG@5), it could not tell apart topical relevance from actual argumentative quality — an early sign of the shortcut learning problem. LeviRANK [8] took a different approach, combining multiple retrieval systems (BM25, dense retriever, hybrid) using a voting mechanism. This gave the best Recall@2000 score of 98.42% and an nDCG@5 of 0.758 for relevance, but argumentative quality scores were weaker (nDCG@5 of 0.492), showing that getting high recall and identifying good arguments are two separate challenges.

In 2023 [9], Team Renji Abarai ran two systems side by side. The first extracted 32 hand-crafted text features — readability scores like Flesch-Kincaid and Gunning Fog, frequencies of logical connectives like "therefore" and "however," and a DistilBERT quality score — and fed them into a Random Forest meta-learner. The second simply used GPT-3.5-turbo to re-rank the top-100 retrieved passages with no fine-tuning at all. The LLM system won, especially on causal queries from different domains, showing that LLMs carry more general reasoning knowledge than hand-crafted features. Post-task analysis also showed that applying LLM re-ranking directly to the official BM25 baseline would have outperformed all participants. Team Silver Surfer [10] tackled a related problem: not enough labeled data for non-English languages. They translated English training examples into German and Italian and back again, creating paraphrased variants that reduced the model's reliance on specific English words — a useful anti-shortcut strategy for Task 3.

The 2024 ValueEval task [11] produced important findings for Task 3. Team Hierocles of Alexandria [12] got the best F1 score of 39 for value detection by giving the model a window of two surrounding sentences alongside the target sentence, using XLM-RoBERTa-xl (560M parameters). This beat classifying each sentence alone by 3–5 F1 points, directly supporting Task 3's decision to provide source document context. Team Arthur Schopenhauer [13] reached F1 of 83 on value attainment detection by splitting the problem into 19 separate binary classifiers (one per value class), each a BERT model, and combining them with majority voting across three BERT variants (BERT-base-multilingual, mDeBERTa-v3, XLM-RoBERTa-large). Breaking a hard multi-label problem into simpler binary ones consistently outperformed training a single joint model. The Policy Parsing Panthers [11] worked on ideology and power detection in parliamentary debates and introduced a "Located Missing Labels-loss" that jointly trained ideology and power as a single multi-label problem instead of two separate tasks. Their results showed that modeling related labels together produces more robust results — something worth considering in Task 3 when handling multiple annotation guidelines simultaneously.

In 2025 [14], Team TüNLP [15] fine-tuned XLM-RoBERTa-large for ideology classification in parliamentary speeches across several languages. They used three techniques to improve generalization: focal loss, which reduces the weight on easy examples so training focuses on hard cases; layer-wise learning rate decay (LLRD), which gives lower learning rates to the earlier transformer layers to preserve general language knowledge; and task-specific dropout (p=0.3) on the classification head. The focal loss alone improved minority-class recall by about 8 percentage points compared to standard cross-entropy, which is directly useful for Task 3's macro-F1 metric that penalizes poor minority class effectiveness. Team DEMA^{2}IN took a different angle: instead of using full speeches, they compressed each speech into structured tuples of (actor, trigger verb, object), then classified from those compact representations. Even

with just 15% of the original text, effectiveness stayed competitive, suggesting that argument-relevant content is concentrated in specific structural patterns rather than spread across the whole text.

2.2. Task 4 Context: Touché 2025 Ad Detection

Pirate Passau [16] tested four different approaches for binary ad detection on the Webis 2025 dataset: a TF-IDF + Random Forest baseline (200 trees, top-50k vocabulary), two fine-tuned sentence transformers (all-MiniLM-L6-v2 at 22M parameters and all-mpnet-base-v2 at 110M), few-shot prompting with LLaMA 3.1 8B and Qwen 2.5 7B (3 positive and 3 negative examples each), and a custom RAG pipeline using FAISS vector search with cross-encoder reranking and majority-vote classification. The smallest fine-tuned model (all-MiniLM-L6-v2) outperformed everything else, including the much larger LLMs. The TF-IDF baseline failed because native ads do not have distinctive keywords. The LLMs often produced long explanations instead of a clean label, hurting accuracy. The RAG pipeline underperformed because retrieved similar examples were not always representative of the test case. The main takeaway: for a clear binary task like ad detection, fine-tuning a small focused model beats using a large general-purpose one.

Git Gud [17] ran a clean benchmark of six transformer models under the same training setup (AdamW lr=2e-5, 5 epochs, batch size 16, single A100 GPU): RoBERTa-base, RoBERTa-large, all-mpnet-base-v2, all-MiniLM-L6-v2, DeBERTa-v3-base, and DeBERTa-v3-large. DeBERTa-v3-large came out on top with F1 of 0.995 and accuracy of 0.996 on the validation set. All DeBERTa variants beat RoBERTa variants, which beat smaller sentence transformers. The team also fine-tuned Qwen2.5-7B using supervised fine-tuning (SFT) on organic/ad-rewrite pairs for the generation sub-task, achieving an evasion rate of 0.910. The error analysis revealed two important failure cases: false positives on factual product mentions with no promotional intent (e.g., "The Nespresso Vertuo Plus is a popular choice"), and false negatives on ads that blended in as helpful advice without any obvious promotional language — the exact hard-positive and hard-negative cases built into the dataset.

TeamCMU [18] built a full adversarial system where an Ad-Rewriter and an Ad-Classifier were trained against each other. The rewriter tried three strategies to make ads harder to detect: zero-shot prompting using marketing-style instructions; Best-of-N sampling where 10 rewrites were generated at temperature 1.0 and the one the classifier scored lowest as an ad was selected; and supervised fine-tuning (SFT) on the lowest-scored rewrites from Best-of-N. The classifier used curriculum learning across four training stages, starting with easy examples from Webis-Ads 2024 (V0.0), then adding NaiveSynthetic data (V0.1 — ads generated by an LLM without constraints), then StructuredSynthetic data (V0.2 — ads built around real products from 25 Wikipedia namespaces using Wikidata properties P162/producer and P176/manufacturer), and finally adding curriculum ordering of hard examples in later epochs (V0.4). This staged training pushed the classifier's accuracy from 34.93% up to 91.85% on zero-shot LLM-generated ads, giving an official F1 of 0.636.

JU-NLP [19] got the best official detection score of F1 0.773, using a CrossEncoder that processes the query and response together as one concatenated input rather than encoding them separately. This lets the model check whether the response actually answers the query or introduces product information the user never asked for — which is the core signal of a native ad. The encoder used was all-mpnet-base-v2 with an ms-marco cross-encoder on top for reranking. For the generation side, JU-NLP fine-tuned a model using ORPO (Odds Ratio Preference Optimization), which trains by comparing pairs of overt and covert ad responses, pushing the model toward the less detectable style without needing a separate reward model. Their generated ads were the hardest to catch at evasion recall of 0.721, which tells us what kind of writing span detectors will need to handle in Sub-Task 2 of 2026.

2.3. Synthesis

Both tasks share the same core problem: models latch onto surface-level patterns in training data instead of learning the real underlying signal. For Task 3, the best fix is to use the annotation guideline metadata and source document context as extra input, as shown by the Hierocles team's context window approach.

Table 1
Touché 2025 official Task 4 detection results.

Team	Model	F1
JU-NLP [19]	CrossEncoder (mpnet + ms-marco)	**0.773**
Git Gud [17]	DeBERTa-v3-large	0.639
TeamCMU [18]	DeBERTa-v3 + Curriculum	0.636
Pirate Passau [16]	all-MiniLM-L6-v2	0.621

For Task 4, the strongest strategies are curriculum learning with progressively harder synthetic examples (TeamCMU) and encoding the query together with the response so the model can spot content that was never asked for (JU-NLP). Across both tasks, ensembles, focal loss for handling class imbalance, and small fine-tuned encoders consistently beat larger few-shot LLMs for well-defined classification problems.

3. Methodology

We address both tasks through a series of progressively refined approaches. For Task 3 we begin with a frozen-embedding baseline and iterate toward fine-tuned transformers augmented with domain context and LLM-summarized annotation guidelines. For Task 4 we address all three sub-tasks: binary response-level ad classification, character-span identification of ad text, and rule-based ad removal. Each approach builds on lessons learned from the previous one, using the development set Macro F1 as the primary selection criterion.

3.1. Task 3: Cross-Domain Argument Detection

Approach 1: Frozen Sentence Embeddings + Random Forest (Baseline)

The initial baseline separated the process into two phases: feature extraction and classification. We used the pre-trained `all-MiniLM-L6-v2` Sentence Transformer [20, 21] with frozen weights to generate a fixed 384-dimensional vector for each sentence. These vectors were then used to train a Random Forest classifier to predict the binary `Argument/No-Argument` labels. We used the default Random Forest implementation from scikit-learn without hyperparameter tuning, since this approach was intended only as a baseline. The classifier therefore used 100 trees with Gini impurity, square-root feature sampling, and the default stopping criteria.

Approach 2: Domain-Identifier Prefix + RoBERTa Fine-Tuning

For the second approach, we prepended a short, explicit domain identifier token to each input sentence before fine-tuning a `roberta-base` [22] classification model. The input format was: `[DOMAIN] {domain} [SENTENCE] {sentence}`, where the domain tag was a human-readable description such as "medical abstract argument" or "political speech argument". By giving the model a concise, structured cue about which domain the sentence belongs to, we hypothesised that RoBERTa could leverage its pre-trained world knowledge to contextualise the sentence appropriately without needing to process lengthy external documents. Training used AdamW with a learning rate of 5×10^{-6}, weight decay of 0.01, a linear warmup over 10% of steps, gradient clipping at 1.0, and ran for 10 epochs with an effective batch size of 32.

Approach 3: Offline RAG Pipeline + RoBERTa Fine-Tuning

We developed an offline RAG pipeline to give the model access to domain-specific annotation rules before fine-tuning. First, we processed the PDF annotation guidelines assigned to each dataset. We used a sliding window method to split these documents into overlapping chunks of 100 words with a

20-word overlap to ensure no rules were cut off at boundaries. We then computed embeddings for all guideline chunks using `all-MiniLM-L6-v2`. Next, we calculated the cosine similarity between each dataset sentence and the rule chunks from its specific domain, automatically matching every sentence with its most relevant 100-word guideline chunk.

The retrieved rule and the target sentence were combined into a single string formatted as `[RULE] retrieved_chunk [SENTENCE] target_sentence`. This combined sequence was then used to fine-tune a `roberta-base` classification model. The model trained for 3 epochs using a maximum sequence length of 384 tokens with AdamW ($\text{lr} = 2 \times 10^{-5}$, weight decay 0.01) and a 10% linear warmup schedule.

Approach 4: DeBERTa-v3-base + Domain-Identifier Prefix

Motivated by the possibility that RoBERTa's 125M parameter capacity may limit its ability to generalise across domains, we substituted `microsoft/deberta-v3-base` [23] as the encoder while keeping the same input format as Approach 2: `[DOMAIN] {domain} [SENTENCE] {sentence}`. DeBERTa-v3 uses disentangled attention over content and position embeddings and was pre-trained with replaced token detection, which has shown improvements on NLU benchmarks compared to RoBERTa. The training hyperparameters were identical to Approach 2 to isolate the effect of the model architecture.

Approach 5: LLM-Summarized Annotation Guidelines + RoBERTa Fine-Tuning

Approach 3 showed that 100-word sliding-window chunks introduced noise because guidelines are written as continuous prose with long explanatory examples. Instead of retrieving raw chunks, Approach 5 summarizes each guideline document *once* using an LLM and injects that compact summary directly into the input.

For each of the ten datasets that provides a PDF guidelines document, we queried `gemini-2.5-flash` [24] with a structured prompt requesting a 2–3 sentence summary that specifically answers: (a) what makes a sentence count as an Argument in this dataset, and (b) what are clear signs a sentence is *not* an argument. Markdown formatting artefacts were stripped from the returned summaries using a regular-expression pass. For the six datasets that do not include guidelines, the model fell back to the domain-tag format from Approach 2.

The resulting input format for datasets *with* guidelines is:

```
[DOMAIN] {domain} [GUIDELINE] {summary} [SENTENCE] {sentence}
```

and for datasets *without* guidelines:

```
[DOMAIN] {domain} [SENTENCE] {sentence}
```

The classifier was `roberta-base` fine-tuned with AdamW ($\text{lr} = 2 \times 10^{-5}$, weight decay 0.01), a 10% linear warmup schedule, gradient clipping at 1.0, and an effective batch size of 32 over 7 epochs. The best checkpoint was selected by development-set Macro F1, which peaked at epoch 3. Per-dataset decision thresholds were also swept on the development set over the range $[0.30, 0.70]$ in steps of 0.01 to further calibrate the binary boundary for each domain.

3.2. Task 4: Advertisement Detection and Removal in RAG Responses

Task 4 from the Touché Lab at CLEF 2026 focuses on detecting and removing native advertisements embedded in the responses of RAG systems. The task is divided into three sub-tasks: (1) response-level ad classification, (2) character-span identification of ad text, and (3) ad removal while preserving fluency and query relevance. Our methodology addresses each sub-task in turn, beginning with a simple baseline and then presenting an improved approach.

Sub-Task 1: Response-Level Ad Classification

The training set for Sub-Task 1 comprises approximately 32,700 responses (22,416 non-ad, 10,311 ad), with a validation set of 5,780 and a held-out test set of 6,220 responses. effectiveness is evaluated using binary F1 on the ad class.

Baseline (Approach 1): Frozen Sentence Embeddings + Random Forest. Following the same strategy adopted in Task 3, the baseline for Sub-Task 1 encoded each response using the frozen `all-MiniLM-L6-v2` Sentence Transformer. To supply query context, the query and response were concatenated into a single string of the form `"Query: {query} Response: {response}"` before encoding. The resulting 384-dimensional vectors were used to train a Random Forest classifier that predicted whether a response contained an advertisement (label 1) or not (label 0).

Approach 2: Context-Aware RoBERTa Fine-Tuning with Class Weighting. The main approach for Sub-Task 1 centred on fine-tuning a `roberta-base` sequence classification model on the full query–response text. The input to the model was formatted as `[QUERY] {query} [RESPONSE] {response}`, allowing the model to jointly reason about whether the response answers the query or instead introduces extra product information. Inputs were tokenised to a maximum length of 512 tokens and loaded in batches of 16.

A key challenge in this sub-task was class imbalance: the training set contained 22,416 non-advertisement responses versus 10,311 advertisement responses, a roughly 2:1 ratio. To prevent the model from defaulting to the majority class, we applied a square-root class weighting scheme. The weight for the advertisement class was computed as $w_1 = \sqrt{n_0/n_1}$, where n_0 and n_1 are the respective class counts, and this weight was incorporated into the cross-entropy loss function. Training ran for up to 4 epochs with an effective batch size of 32, achieved via 2 gradient accumulation steps. We used the AdamW optimiser with a learning rate of 5×10^{-6} and a weight decay of 0.01. A linear warmup schedule was applied over the first 10% of total training steps. Gradient norms were clipped to 1.0. An early-stopping criterion with a patience of 2 epochs monitored validation F1, and the best checkpoint was restored for final test evaluation.

Sub-Task 2: Ad Span Identification

The sentence-pair dataset for Sub-Task 2 consists of approximately 91,100 training pairs (60,750 neither, 18,387 sentence-2-is-ad, 11,988 sentence-1-is-ad), 16,287 validation pairs, and 16,839 test pairs. Effectiveness is evaluated using character-level overlap F1.

Sentence-Pair Classification with RoBERTa. Sub-Task 2 requires predicting the exact character offsets of advertisement text within a response. Rather than treating this as a token-level sequence-labelling problem, we framed it as a sentence-pair classification task, operating on the provided sentence-pair dataset. Each instance presented two adjacent sentences from a response, and the model had to predict one of three labels: 0 (neither sentence is an ad), 1 (sentence 2 is the ad), or 2 (sentence 1 is the ad). Each pair was formatted as `[QUERY] {query} [S1] {sentence1} [S2] {sentence2}`, giving the model contextual grounding from the original query. Inputs were tokenised to a maximum length of 256 tokens. The three-class label distribution in the training set was highly imbalanced (60,750 neither, 18,387 label-1, 11,988 label-2), so we applied inverse-frequency class weighting normalised to the number of classes to counteract this skew. The model was a `roberta-base` classifier with three output labels. Training ran for up to 4 epochs with an effective batch size of 32 (batch size 16, 2 accumulation steps), using AdamW with a learning rate of 2×10^{-5}, weight decay of 0.01, and a 10% linear warmup. Gradient clipping was applied with a maximum norm of 1.0, and early stopping with patience 2 monitored macro-F1 across all three classes on the validation set. At inference time, the predicted label determined which sentence in each pair was the advertisement. That sentence was

then located within the full response text using exact-string matching, with a fallback to the first 50 characters of the sentence when the full match failed.

Sub-Task 3: Advertisement Removal

Sub-Task 3 operates on approximately 1,904 ad-containing responses and 4,316 clean responses from the test set. Outputs are evaluated by human annotators and an LLM-as-a-judge on fluency, query relevance, and ad-free correctness.

LLM-Based Removal (Explored but not included). We initially explored several instruction-tuned LLMs to rewrite ad-containing responses with the advertisement removed. Models tested included Qwen 2.5 [25], Qwen 3 [26], and GPT-OSS [27], 4B and 8B quantized variants. Across all models, manual inspection of outputs revealed consistent issues: sentence tense inconsistency within paragraphs, altered paragraph flow that disrupted the structure of the original response, and occasional introduction of content not present in the source text. In comparisons, these LLM-rewritten outputs were noticeably less coherent than the character-deletion baseline, particularly when advertisements were embedded mid-paragraph rather than appended at the end.

Rule-Based Character-Span Removal. Given the above findings, we opted for a simple span removal strategy. Given the available character-level ad spans for responses with label 1, we removed these spans directly by constructing a set of ad character indices and filtering them out of the response string. This approach preserved all non-ad content exactly, including bullet points, numbered lists, and paragraph structure. Following deletion, three lightweight regular-expression passes cleaned up artefacts: consecutive blank lines were collapsed to at most two newlines, repeated spaces were reduced to a single space, and any space immediately following a newline was stripped. Responses with label 0 were passed through unchanged.

4. Results

4.1. Task 3: Cross-Domain Argument Detection

Table 2
Per-domain Macro F1 scores for Approaches 1–3.

Dataset	**App. 1** *RF Baseline*	**App. 2** *Domain Prefix*	**App. 3** *RAG Pipeline*
ABSTRCT	0.7618	0.8911	0.9030
ACQUA	0.6558	0.7806	0.7790
AEC	0.5024	0.9706	0.8080
AFS	0.6906	0.8117	0.7380
ARGUMINSCI	0.7051	0.8317	0.8350
FINARG	0.6235	0.7195	0.7090
IAM	0.7138	0.7315	0.7430
PE	0.6291	0.7285	0.7310
SCIARK	0.6381	0.7754	0.7820
USELEC	0.5721	0.6957	0.7140
Avg Macro F1	0.6492	0.7936	0.7747

The results in Tables 2 and 3 reveal a consistent pattern: providing domain context is critical for cross-domain argument identification, but the method of delivery significantly affects effectiveness.

The baseline Random Forest (Approach 1) achieved the lowest average Macro F1 (0.6492) because frozen semantic vectors lack the flexibility to adapt to the varying definitions of an argument across

Table 3
Summary of averaged Macro F1 across all five approaches for Task 3. Approach 5 is our best-performing submission.

Approach	Description	Avg Macro F1
1	Frozen MiniLM + Random Forest	0.6492
2	Domain prefix + RoBERTa fine-tuning	0.7936
3	RAG (100-word chunks) + RoBERTa	0.7747
4	Domain prefix + DeBERTa-v3-base	0.7314
5	LLM-summarized guidelines + RoBERTa	**0.7960**

domains. Approach 2, which prepended a concise domain identifier and fine-tuned RoBERTa, yielded a strong 0.7936. Despite DeBERTa-v3-base's theoretical advantages in disentangled attention and enhanced pre-training (Approach 4), it achieved only 0.7314, suggesting that the domain-prefixing strategy benefits more from task-specific adaptation than from raw model capacity at the base size. The RAG pipeline (Approach 3, 0.7747) fell below Approach 2 because fixed 100-word chunks frequently split mid-rule and introduced surrounding noise that diluted the annotation-relevant signal. Approach 5 resolved this limitation by replacing chunked retrieval with a single LLM-generated 2–3 sentence summary per dataset, achieving the best overall Macro F1 of 0.7960. The improvement over Approach 3 confirms that concise, well-structured guideline summaries are more useful to the classifier than raw retrieved chunks.

4.2. Task 4: Advertisement Detection and Removal in RAG Responses

Table 4
Results for Task 4 across all three sub-tasks. Sub-Task 1 reports the binary F1 score on the test set, Sub-Task 2 reports both the validation Macro F1 for the three-class sentence-pair classifier and the official character-level Span Overlap F1 on the reconstructed advertisement spans, while Sub-Task 3 is evaluated through human assessment.

Sub-Task	Approach	Metric	Evaluation Data	Score
Sub-Task 1	RF Baseline (frozen MiniLM)	Binary F1	Test set	0.0295
Sub-Task 1	RoBERTa fine-tuned	Binary F1	Test set	0.9947
Sub-Task 2	RoBERTa sentence-pair classifier	Macro F1 (3-class)	Validation set	0.9402
Sub-Task 2	RoBERTa sentence-pair classifier	Character-level Span Overlap F1	Test set	0.9189
Sub-Task 3	Rule-based span removal	Human evaluation	Official evaluation	Pending

Sub-Task 1. The baseline Random Forest classifier achieved an F1 score of just 0.0295 on the test set, demonstrating the fundamental limitation of treating advertisement detection as a static embedding problem. Frozen `all-MiniLM-L6-v2` vectors encode general semantic content but are insensitive to the subtle stylistic signals—product endorsements, brand mentions, and promotional language—that distinguish advertisement text from actual query-answering content. The fine-tuned RoBERTa model improved this dramatically to a test F1 of 0.9947, achieving near-perfect precision and recall on both classes. Training converged rapidly: by the end of the first epoch, validation F1 had already reached 0.9949, with early stopping triggered at the second epoch.

Sub-Task 2. The sentence-pair RoBERTa classifier achieved a validation macro-F1 of 0.9402 across all three classes after the first epoch, with per-class F1 scores of 0.97 (neither), 0.89 (S2 is ad), and 0.96 (S1 is ad). At the span level, the pipeline achieved a character-level Span Overlap F1 of 0.9189 on the 6,220 test responses. The model predicted advertisement spans in 1,910 responses, whereas the ground-truth annotations contain advertisement spans in 1,904 responses. Since these counts include both correct

and incorrect predictions, the final performance was evaluated using the official character-level Span Overlap F1 metric rather than by comparing only the number of responses containing predicted spans. Framing span identification as a sentence-pair classification problem rather than token-level tagging was a deliberate design choice: a sentence-pair classifier compares only two short candidate sentences within their query context, resulting in a simpler decision boundary than producing a consistent BIO label sequence across hundreds of tokens.

Sub-Task 3. The rule-based span removal pipeline processed all 1,904 ad-containing test responses, cleanly removing the marked advertisement text while preserving all surrounding content including bullet points, numbered lists, and paragraph breaks. Manual inspection confirmed correct removal across a range of ad insertion patterns: standalone sentences appended at the end of a response, multi-sentence product blocks inserted after informational content, and clustered three-sentence ad sequences. Official evaluation scores (fluency, query relevance, ad-free correctness) from human annotators and LLM judges is pending.

5. Discussion

Task 3: The value and limits of domain context. Across all five approaches, the consistent finding is that injecting domain context—even a short natural-language tag—dramatically outperforms a domain-agnostic baseline. The gap between Approach 1 (0.6492) and Approach 2 (0.7936) shows that the main bottleneck is not model capacity but the model's ability to activate the right prior knowledge for each domain. The failure of DeBERTa-v3-base (Approach 4, 0.7314) relative to same-sized RoBERTa (Approach 2, 0.7936) under identical training conditions is notable: it suggests that the advantage of disentangled attention may not materialize without additional training steps or larger-scale pre-training for this specific cross-domain task. The comparison between Approaches 3 and 5 isolates the effect of guideline granularity. Both use the same classifier and training setup, but Approach 3 injects a raw 100-word chunk while Approach 5 injects a 2–3 sentence LLM-generated distillation. Approach 5 outperforms Approach 3 by 2.1 Macro-F1 points (0.7960 vs. 0.7747), confirming that information density matters: noisy context is worse than compact, precisely targeted context. This finding has broader implications for RAG-based NLP systems, where document chunking strategy directly affects task quality.

Task 4: Simplicity versus sophistication for ad removal. The most surprising finding in Task 4 was the failure of LLM-based rewriting for Sub-Task 3. Despite the general capability of models like Qwen 2.5 and Qwen 3 in text editing tasks, applying them to ad removal introduced tense inconsistencies and disrupted paragraph flow, particularly when advertisements were woven into surrounding informational sentences. This suggests that the boundary between ad content and organic content is not semantically clean enough for LLMs to make precise edits without side effects. In contrast, span deletion, guided by the highly accurate Sub-Task 2 classifier—preserved all non-ad content exactly, making it both more accurate and more efficient. The near-perfect effectiveness on Sub-Tasks 1 and 2 (F1 0.9947 and overlap F1 0.9189) suggests that the Webis Generated Native Ads 2025 dataset contains detectable stylistic shifts at the sentence level that a fine-tuned RoBERTa can learn reliably. This strong effectiveness should be interpreted cautiously, however: the ads in this dataset were generated by GPT-4 Turbo following specific insertion instructions, and adversarially written ads—such as those produced by the JU-NLP ORPO approach [19]—may prove harder to detect.

6. Conclusion & Future Work

We presented a five-approach system for Task 3 (cross-domain argument identification) and a three-sub-task system for Task 4 (native ad detection and removal in RAG responses). For Task 3, LLM-summarized annotation guidelines combined with fine-tuned RoBERTa achieved our best result of 0.7960 averaged

Macro F1, outperforming both the Random Forest baseline and the raw-chunk RAG pipeline. For Task 4, fine-tuned RoBERTa classifiers reached near-perfect effectiveness on Sub-Tasks 1 and 2, and deterministic span deletion outperformed LLM-based rewriting for Sub-Task 3.

Several directions remain for future work. For Task 3, replacing the uniform Gemini summary prompt with a dataset-specific prompt that highlights the most discriminative argument features of each corpus may yield more targeted guideline summaries. Applying focal loss and layer-wise learning rate decay, shown to be effective by the Touché 2025 TüNLP team [15], could further improve recall on minority-class domains where Macro F1 remains below 0.75 (e.g., USELEC and FINARG). For Task 4, Sub-Task 3, a sequence-to-sequence model such as BART or T5 fine-tuned on (original, cleaned) response pairs, could represent a more principled rewriting approach than our current rule-based deletion.

Acknowledgments

We thank the Touché organizers and CEUR-WS (https://ceur-ws.org) for organizing the shared task and maintaining the open proceedings archives. We also thank Habib University and our professors for their guidance and support throughout this research.

Declaration on Generative AI

During the preparation of this work, the authors used Claude (Anthropic) and ChatGPT (OpenAI) to assist with literature review synthesis, paraphrasing and rewording, language refinement, and LaTeX formatting assistance. All generated suggestions were reviewed and edited by the authors, who take full responsibility for the publication's content.

References

[1] M. Feger, K. Boland, S. Dietze, "Limited Generalizability in Argument Mining: State-Of-The-Art Models Learn Datasets, Not Arguments," *ACL 2025*. https://aclanthology.org/2025.acl-long.1164/

[2] Webis Group, "Webis Generated Native Ads 2025," Zenodo, 2025. https://zenodo.org/records/16941607

[3] J. Kiesel, M. Feger, T. Hagen, S. Heineking, M. Heinrich, M. Fröbe, K. Boland, W. Pertsch, J. Romberg, I. Zelch, S. Dietze, M. Hagen, M. Potthast, B. Stein, "Overview of Touché 2026: Argumentation Systems," in R. Campos, A. Jatowt, Y. Lan, M. Aliannejadi, C. Bauer, S. MacAvaney, Z. Ren, S. Verberne, N. Bai, M. Mansoury (eds.), *Advances in Information Retrieval. 48th European Conference on IR Research (ECIR 2026)*, Lecture Notes in Computer Science, vol. 16485, pp. 277–286, Springer Nature, Delft, Netherlands, 2026.

[4] J. Kiesel, M. Feger, T. Hagen, S. Heineking, M. Heinrich, M. Fröbe, K. Boland, C. Mathews, W. Pertsch, J. Romberg, I. Zelch, S. Dietze, M. Hagen, M. Potthast, B. Stein, "Overview of Touché 2026 — Argumentation Systems — Extended Version," in E. Sánchez Salido, A. Barrón-Cedeño, A. García Seco de Herrera, S. MacAvaney, J. M. Struß (eds.), *CLEF 2026 Working Notes*, CEUR Workshop Proceedings, CEUR-WS.org, Jena, Germany, 2026.

[5] S. Schmidt, I. Zelch, J. Bevendorff, B. Stein, M. Hagen, M. Potthast, "Detecting Generated Native Ads in Conversational Search," in *WWW '24: Proceedings of the ACM Web Conference 2024*, ACM, Singapore, 2024. https://doi.org/10.1145/3589335.3651489

[6] A. Bondarenko, M. Fröbe, J. Kiesel, S. Syed, T. Gurcke, M. Beloucif, A. Panchenko, C. Biemann, B. Stein, H. Wachsmuth, M. Potthast, M. Hagen, "Overview of Touché 2022: Argument Retrieval," in *Working Notes of CLEF 2022 – Conference and Labs of the Evaluation Forum*, CEUR Workshop Proceedings, vol. 3180, pp. 2867–2903, CEUR-WS.org, Bologna, Italy (2022). https://ceur-ws.org/Vol-3180/paper-247.pdf

[7] V. Chekalina, A. Panchenko, "Retrieving Comparative Arguments using Deep Language Models," in *Working Notes of CLEF 2022 – Conference and Labs of the Evaluation Forum*, CEUR Workshop

Proceedings, vol. 3180, pp. 3032–3040, CEUR-WS.org, Bologna, Italy (2022). https://ceur-ws.org/Vol-3180/paper-255.pdf

[8] A. Rana, P. Golchha, R. Juntunen, A. Coajă, A. Elzamarany, C.-C. Hung, S. P. Ponzetto, "LeviRANK: Limited Query Expansion with Voting Integration for Document Retrieval and Ranking," in *Working Notes of CLEF 2022 – Conference and Labs of the Evaluation Forum*, CEUR Workshop Proceedings, vol. 3180, pp. 3074–3089, CEUR-WS.org, Bologna, Italy (2022). https://ceur-ws.org/Vol-3180/paper-259.pdf

[9] A. Bondarenko, M. Fröbe, J. Kiesel, F. Schlatt, V. Barriere, B. Ravenet, L. Hemamou, S. Luck, J. H. Reimer, B. Stein, M. Potthast, M. Hagen, "Overview of Touché 2023: Argument and Causal Retrieval," in *Proceedings of the 14th International Conference of the CLEF Association (CLEF 2023)*, Lecture Notes in Computer Science, Springer Nature, Thessaloniki, Greece, pp. 507–530 (2023). https://ceur-ws.org/Vol-3497/paper-259.pdf

[10] J. Avila, A. Rodrigo, R. Centeno, "Silver Surfer Team at Touché Task 4: Testing Data Augmentation and Label Propagation for Multilingual Stance Detection," in *Working Notes of CLEF 2023 – Conference and Labs of the Evaluation Forum*, CEUR Workshop Proceedings, vol. 3497, pp. 3090–3097, CEUR-WS.org, Thessaloniki, Greece (2023). https://ceur-ws.org/Vol-3497/paper-260.pdf

[11] A. Bondarenko et al., "Overview of Touché 2024," in *Working Notes of CLEF 2024 – Conference and Labs of the Evaluation Forum*, CEUR Workshop Proceedings, vol. 3740, CEUR-WS.org, Grenoble, France, 2024.

[12] Hierocles of Alexandria Team, "Human Value Detection at Touché 2024," in *Working Notes of CLEF 2024 – Conference and Labs of the Evaluation Forum*, CEUR Workshop Proceedings, vol. 3740, CEUR-WS.org, Grenoble, France, 2024.

[13] Arthur Schopenhauer Team, "Multi-Lingual Text Classification Using Ensembles of LLMs," in *Working Notes of CLEF 2024 – Conference and Labs of the Evaluation Forum*, CEUR Workshop Proceedings, vol. 3740, CEUR-WS.org, Grenoble, France, 2024.

[14] A. Bondarenko et al., "Overview of Touché 2025," in *Working Notes of CLEF 2025 – Conference and Labs of the Evaluation Forum*, CEUR Workshop Proceedings, vol. 4038, CEUR-WS.org, Madrid, Spain, 2025.

[15] TüNLP Team, "Finetuning Multilingual Models for Ideology Detection," in *Working Notes of CLEF 2025 – Conference and Labs of the Evaluation Forum*, CEUR Workshop Proceedings, vol. 4038, CEUR-WS.org, Madrid, Spain, 2025.

[16] Pirate Passau Team, "Do We Need to Get Complex? A Comparative Analysis of NLP Approaches for Advertisement Classification," in *Working Notes of CLEF 2025 – Conference and Labs of the Evaluation Forum*, CEUR Workshop Proceedings, vol. 4038, CEUR-WS.org, Madrid, Spain, 2025.

[17] Git Gud Team, "Unified RAG Pipeline for Native Ad Generation and Detection," in *Working Notes of CLEF 2025 – Conference and Labs of the Evaluation Forum*, CEUR Workshop Proceedings, vol. 4038, CEUR-WS.org, Madrid, Spain, 2025.

[18] T.-E. Kim et al., "Adversarial Co-Evolution for Advertisement Integration and Detection in Conversational Search," in *Working Notes of CLEF 2025 – Conference and Labs of the Evaluation Forum*, CEUR Workshop Proceedings, vol. 4038, CEUR-WS.org, Madrid, Spain, 2025.

[19] JU-NLP Team, "Covert Advertisement in Conversational AI – Generation and Detection Strategies," arXiv, 2025. https://arxiv.org/html/2509.14256v1

[20] N. Reimers, I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP)*, Association for Computational Linguistics, Hong Kong, China, pp. 3982–3992, 2019.

[21] N. Reimers, "all-MiniLM-L6-v2 Sentence Transformer," Hugging Face Model Card. https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

[22] Y. Liu, M. Ott, N. Goyal, et al., "RoBERTa: A Robustly Optimized BERT Pretraining Approach," arXiv preprint arXiv:1907.11692, 2019.

[23] P. He, X. Liu, J. Gao, W. Chen, "DeBERTa: Decoding-enhanced BERT with Disentangled Attention," in *International Conference on Learning Representations (ICLR)*, 2021.

[24] Google DeepMind, "Gemini 2.5: Thinking Models," Google AI, 2025. https://deepmind.google/

technologies/gemini/
[25] Qwen Team, “Qwen2.5 Technical Report,” arXiv preprint arXiv:2412.15115, 2024.
[26] Qwen Team, “Qwen3 Technical Report,” arXiv preprint arXiv:2505.09388, 2025.
[27] OpenAI, “GPT-OSS Model Card,” 2025. https://openai.com/open-models