# Git Gud at Touché: Unified RAG Pipeline for Native Ad Generation and Detection

Notebook for the Touché Lab at CLEF 2025

Sameer Kamani[1,*,†], Muhammad Taqi[1,†], Muhammad Ansab Chaudhary[1], Muhammad Ahmad Humayun Hanif[1], Faisal Alvi[1] and Abdul Samad[1]

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

**Abstract**

This project investigates the integration and detection of advertisements within LLM-generated responses using Retrieval-Augmented Generation (RAG). We address two key tasks: first, generating contextually relevant advertisements within RAG-retrieved document segments, ensuring coherence and structured output; and second, developing robust models for detecting embedded advertisements to maintain content integrity. Utilizing the Webis Generated Native Ads 2024 dataset, we aim to evaluate the effectiveness of various RAG-based generation strategies and detection methods. Our research explores techniques for balancing ad relevance with informational content, contributing to the development of transparent and ethical AI-driven advertising.

**Keywords**

Retrieval-Augmented Generation, Large Language Models, Ad Integration, Ad Detection, Deep Learning

## 1. Introduction

This project investigates the integration and detection of advertisements within LLM-generated responses using Retrieval-Augmented Generation (RAG). Specifically, it implements Task 4 ("Advertisement in Retrieval-Augmented Generation") of the sixth edition of the Touché Lab at CLEF 2025 [1]:

- **Task 1**: Advertisement Generation in RAG-based LLMs – Create relevant responses for a given query, based on a set of document segments. If provided an item (service, product, or brand) and its corresponding qualities, the responses also need to advertise that item. This advertisement should be difficult to detect and seamlessly fit into the rest of the response.
- **Task 2**: Advertisement Detection – The second task involves identifying embedded advertisements within LLM-generated responses. Given the rise of native ads in AI-generated content, developing robust detection models is crucial for transparency and content integrity.

The data set consists of JSONL files containing queries, retrieved document segments, and advertisement details. The goal is to assess the effectiveness of RAG-based generation strategies and evaluate detection methods that can detect ad-embedded responses.

## 2. Literature Review

There are several studies that propose frameworks that leverage RAG to integrate ads into LLM outputs in a contextually relevant manner. For instance, Hajiaghayi et al. (2024) introduce a segment auction

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

*Corresponding author.

†These authors contributed equally.

sk08109@st.habib.edu.pk (S. Kamani); mt08073@st.habib.edu.pk (M. Taqi); mc08077@st.habib.edu.pk (M. A. Chaudhary); mh08072@st.habib.edu.pk (M. A. H. Hanif); faisal.alvi@sse.habib.edu.pk (F. Alvi); abdul.samad@sse.habib.edu.pk (A. Samad)

0009-0009-1682-7639 (S. Kamani); 0009-0005-9995-9119 (M. Taqi); 0009-0006-5924-1599 (M. A. Chaudhary); 0009-0000-9854-3967 (M. A. H. Hanif); 0000-0003-3827-7710 (F. Alvi); 0009-0009-5166-6412 (A. Samad)

model where external document segments—selected based on relevance and educational value—are dynamically merged with LLM responses to embed ad content. In this model, multiple modules work together to optimize ad placement and revenue generation, demonstrating that RAG-based ad insertion can outperform traditional keyword-based methods by enhancing the contextual alignment of ads.

Similarly, Feizi et al. (2023) present a highly influential modular architecture specifically designed for LLM advertising. Their framework divides the problem into four core components, including a modification module, which seamlessly adapts the original LLM output to incorporate ad text without disrupting content coherence. This comprehensive framework is particularly notable because it addresses the technical challenges of dynamically integrating ad content in real-time and ensures that the inserted ads remain contextually relevant. This work thus offers critical insights into balancing persuasive ad integration with the preservation of natural text flow. Zelch et al. (2023) present a pilot study where generative models ( GPT-3.5, GPT-4, and You Chat e.t.c. ) are prompted to insert ads into search results across three scenarios: Unrelated Ads, Loosely Related Ads, and Very Related Ads. Across all these scenarios the models perform poorly as they either lack relevance or they are not subtle and are rather jarring. However, it serves an important role as a proof of concept demonstrating that with basic prompt engineering, it is technically feasible to integrate native ads into text SERPs.

With advertisements increasingly embedded in LLM outputs, detecting them poses unique challenges. Schmidt et al. (2024) focus on detecting native ads within conversational search responses by utilizing fine-tuned sentence transformer models such as MiniLM and MPNet. These models learn to identify subtle stylistic and contextual cues that differentiate native ads from organic content. Although zero-shot detection methods using LLMs like GPT-4 have been explored, they generally underperform compared to fine-tuned transformers, particularly when ads are seamlessly integrated.

Kok-Shun and Chan (2025) applied GPT-4o to detect sponsored ads in video transcripts. The model was prompted to identify ad segments based on context and intent, without fine-tuning. While KeyBERT, a BERT-based model, was employed to extract contextually relevant keywords from the transcripts, leveraging BERT embeddings. After extracting keywords with KeyBERT, GPT-4o was used to group them into broader categories, reducing dimensionality and improving thematic analysis.

In general, the literature demonstrates growing interest and promising progress in using RAGs and LLMs for dynamic, context-aware ad integration, while also highlighting the technical and ethical challenges of detecting seamlessly embedded advertising. These studies collectively lay the groundwork for more nuanced, adaptive, and effective ad strategies in generative AI systems.

# 3. Methodology

## 3.1. TIRA Submissions

**Table 1**
TIRA submissions for each sub-task

| Sub-task | Submission Name |
|---|---|
| 1 | Qwen2.5 7B V2 |
| | Qwen3 4B V2 |
| 2 | Deberta-Large-V2 |
| | RoBERTa-Large |
| | Deberta Large |

Our work was submitted via TIRA; credit to Fröbe, M. et al.(2023) for streamlining the process. We made multiple submissions, and the table above lists our final work. The only difference between our submissions is that of the model used, the overall pipeline remained the same.

### 3.2. Dataset

Our dataset for Task 1 is the Webis Generated Native Ads 2024 dataset with segmented document excerpts from MS MARCO V2.1. The data is given in the following format:

- **Query:** Dictionary of ID (Topic ID) and Keyword Query (Text).
- **Candidates:** List of segments that were retrieved for the Keyword Query.
  - **Docid:** ID for the segment in MS MARCO v2.1.
  - **Score:** Score calculated by Elasticsearch. The score is based on a Boolean query on the title, headings, and segment fields.
  - **Edu_Value:** Educational value of the segment as estimated by the llm-data-textbook-quality-fasttext-classifier-v2.
  - **Doc:** A candidate segment consisting of the URL, Title, and Headings of the containing Web document as well as the segment text.
- **Advertisements:** A list where each entry is either None or a dictionary.
  - **Item:** Name of the brand, service, or product to be advertised.
  - **Type:** Describes the type of the item (e.g., Brand or a specific type of product).
  - **Qualities:** A descriptive string of item attributes for use in the ad.

Our data set for Task 2 is a JSONL-version of the Webis Generated Native Ads 2024 and consists of two main parts. Firstly, the response data is given in the following format:

- **Id:** ID of the response.
- **Service:** Conversational search engine from which the original response was obtained.
- **Meta_Topic:** One of ten categories that the query belongs to.
- **Query:** Keyword query for which the response was obtained.
- **Response:** Full text of the response.

And for each response in the previous file, it has corresponding label data which contains the following elements:

- **Id:** ID of the response.
- **Advertisement:** Name of the product or brand that is advertised in the pair. It is `None` for responses without an ad.
- **Label:** 1 for responses with an ad and 0 otherwise.
- **Span:** Character span containing the advertisement. It is `None` for responses without an ad.
- **Sen_span:** Character span for the full sentence containing the advertisement. It is `None` for responses without an ad.

The labels help us classify the response as having an advertisement or not, and then further identify where exactly said advertisement was placed.

### 3.3. Approach For Sub-Task 1

Our pipeline begins by embedding each user query and all retrieved document segments using the `all-MiniLM-L6-v2` [25] sentence transformer. Retrieval is performed in three stages: first, each candidate segment is already assigned two primary scores: Elasticsearch relevance and an educational-value estimate. We take those scores and compute initial rankings accordingly. Second, we index the segment embeddings in FAISS (`IndexFlatL2`) [27] and retrieve the top $k_{\text{init}}$ segments by nearest-neighbor distance to the query embedding. Thirdly, we rerank this shortlist with the Cross-Encoder `ms-marco-MiniLM-L-6-v2` [26] to obtain fine-grained relevance scores. We convert all three signals into 1-based ranks and compute a weighted average rank, selecting the top $k$ segments as context for

generation. The values $k_{\text{init}} = 10$ and $k = 3$ remain fixed across all queries to balance retrieval quality with computational efficiency.

For response generation, we used a variety of different models, but our primary work was done using the Qwen 2.5 7B Instruct [12] model via HuggingFace's text-generation pipeline (temperature 0.7, top_p 0.9, max_new_tokens 256). We first produce an advertisement-free baseline response to the prompt

```
Query: <user query>
Context: <topk segments>
Please generate a detailed and coherent response.
Response:
```

Then, for each advertised item in the dataset (skipping the `None` entries), we iteratively generate up to three variants by inserting a transition sentence that mentions the item and its attributes. Each variant is scored by a composite naturalness metric (0.0-1.0 scale) evaluating ad placement, contextual coherence, and subtlety, along with ROUGE-1 overlap against the baseline. The naturalness metric incorporates position analysis (preferring 30-70% text location), quality term integration, and avoidance of explicit commercial markers. We accept the highest-scoring variant if it exceeds the thresholds on both metrics; otherwise, we fall back to the single best candidate.

Following this, we apply regex-based post-processing to strip HTML/Markdown artifacts, table fragments, and duplicate or truncated sentences, ensuring that each generated response concludes with proper punctuation.

As we had fine-tuned various models for Sub-Task 2, we integrated one of them into this pipeline, namely RoBERTa-Large as mentioned in Table 1. When an advertisement passes the previously mentioned thresholds, it would then be evaluated by our Ad Detector. If it was detected, we would then regenerate that response. We set a limit of 10 trials due to limited resources. A similar check was applied when generating the advertisement-free response; if it was classified by our detector as having ad-like elements, it would be regenerated.

### 3.4. Approach For Sub-Task 2

**Table 2**
Key hyperparameters used for fine-tuning transformer models.

| Hyperparameter | Value |
|---|---|
| Number of training epochs | 3 |
| Batch size | 8 |
| Learning rate | 2e-5 |
| Weight decay | 0.01 |
| Maximum input length | 512 |
| Loss function | Binary cross-entropy |

For the advertisement detection task, we adopted a fine-tuning strategy using several transformer-based models, starting with the RoBERTa-Base [16] architecture. Each model was equipped with a binary classification head to predict whether a given LLM-generated response contains an embedded advertisement.

We maintained consistent training hyperparameters for all models (see Table 2), namely three epochs, a batch size of 8, and a learning rate of 2e-5. Tokenization was handled using the appropriate tokenizer for each model (e.g., RoBERTaTokenizer, DeBERTaTokenizer), and inputs consisted of full-text responses from the labeled dataset.

In our initial approach, we used RoBERTa-Base,but then we experimented with several other architectures to explore different representational strengths. These included MPNet-v2 [14],RoBERTa-large, DeBERTa-v3-base and DeBERTa-v3-large [19], as well as the 0x7o-roberta checkpoint [23] tailored for ad detection tasks. Each model was trained using binary cross-entropy loss, and the evaluation metrics

included precision, recall, F1 score, and accuracy. This setup allowed us to systematically compare architectures under a unified training and evaluation framework.

# 4. Results

## 4.1. Results for Sub-Task 1

We ran our model on various subsections of the provided data to verify that advertisements are generated consistently when required. While occasional inconsistencies occur, overall the generator embeds ads seamlessly into the responses.

For example, here is an ad-free response and its corresponding ad-injected response:

```
Response: Inside money refers to the form of money that exists in bank accounts as bank
    deposits and is created as a result of loans. This form of money is controlled
    almost entirely by private banks and is created through the lending process. Online-
    only banks tend to offer higher average rates on savings accounts and high-yield
    checking compared to brick-and-mortar banks, so it's no surprise that their money
    market accounts offer better APYs as well. Major online-only banks include
    ableBanking, Ally Bank, Synchrony Bank, Discover Bank, Axos Bank, Capital One,
    Radius Bank, and TIAA Bank. The rates for money market accounts at these banks range
     from 0.30% to 0.60% depending on the balance
```

Listing 1: Ad-Free Response

```
Response: Inside money refers to the form of money that exists in bank accounts as bank
    deposits and is created as a result of loans. This form of money is controlled
    almost entirely by private banks and is created through the lending process. Online-
    only banks tend to offer higher average rates on savings accounts and high-yield
    checking compared to brick-and-mortar banks, so it's no surprise that their money
    market accounts offer better APYs as well. For instance, TD Beyond Checking and
    Savings Accounts offers tiered interest, relationship benefits, providing an option
    that aligns with these needs. Major online-only banks include ableBanking, Ally Bank
    , Synchrony Bank, Discover Bank, Axos Bank, Capital One, Radius Bank, and TIAA Bank.
     The rates for money market accounts at these banks range from 0.30% to 0.60%
    depending on the balance.
```

Listing 2: Ad-Injected Response

We then evaluated the 169 generated responses via the provided MiniLM baseline evaluator. The measured performance is summarized in Table 3.

**Table 3**
Official evaluator performance on Sub-Task 1 generated responses.

| Metric | Value |
|---|---|
| Precision(P) | 0.857 |
| Recall(R) | 0.222 |
| F1 Score(F1) | 0.353 |
| Accuracy(A) | 0.479 |

The evaluation yielded a recall of 0.222 and a precision of 0.857. The low recall shows that many of the more subtly integrated advertisements successfully evaded detection (by design), while the high precision confirms that ordinary informational content was rarely misclassified as advertising. Together, These results indicate that our approach effectively minimizes false positives without compromising the intended stealth of the ad content.

### 4.2. Results for Sub-Task 2

To evaluate the effectiveness of our advertisement classifiers, we measured both F1 score and accuracy across six transformer-based architectures. Table 4 summarizes model-performance and the test-set performance, ordered from the smallest to the largest model.

**Table 4**
Performance metrics for various transformer models fine-tuned for advertisement detection.

| Model | Loss | Precision | Recall | F1 Score | Accuracy |
|---|---|---|---|---|---|
| MPNet-v2 | 0.076 | 0.990 | 0.962 | 0.976 | 0.983 |
| 0x7o-roberta | 0.070 | 0.988 | 0.988 | 0.988 | 0.992 |
| DeBERTa-v3-base | 0.037 | 0.993 | 0.990 | 0.992 | 0.994 |
| RoBERTa-Base | 0.030 | 0.996 | 0.988 | 0.992 | 0.995 |
| RoBERTa-large | 0.028 | 0.994 | 0.992 | 0.993 | 0.996 |
| DeBERTa-v3-large | 0.022 | 0.993 | 0.996 | 0.995 | 0.996 |

We observed that the smallest model, MPNet-v2 [15], achieved an F1 score of 0.9756 and an accuracy of 0.9831. The "0x7o-roberta" checkpoint improved to an F1 score of 0.9880 and an accuracy of 0.9915. Fine-tuning DeBERTa-v3-base [21] resulted in 0.9918 F1 and 0.9942 accuracy, while RoBERTa-Base [17] reached 0.9920 F1 at 0.9950 accuracy. The larger models provided marginal but consistent gains: RoBERTa-large [18] achieved 0.9930 F1 and 0.9955 accuracy, and DeBERTa-v3-large [22] led with 0.9950 F1 and 0.9960 accuracy.

These results indicate that, although even our smaller models perform exceedingly well, the largest DeBERTa-v3-large variant still offers the best balance of precision and recall, pushing both metrics above 99%. In practical deployments, the modest improvements from base to large checkpoints should be weighed against the increased inference cost; however, for applications demanding maximal detection quality, DeBERTa-v3-large is our top choice.

Table 5 compares our best model against two lightweight baselines from the research "Detecting generated native ads in conversational search"[6]. The confusion matrix in Figure 1a further validates these results by illustrating the distribution of true positives, true negatives, false positives and false negatives.

**Table 5**
Performance comparison between DeBERTa-v3-large and lightweight baselines on advertisement detection.

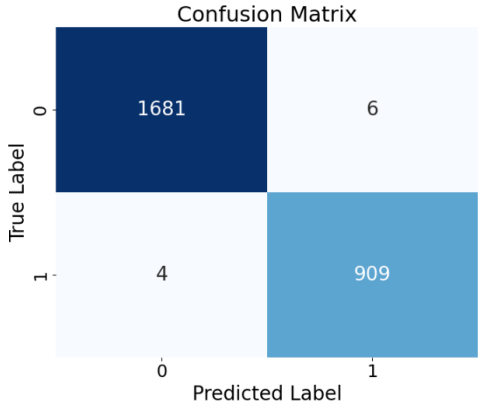
| Metric | DeBERTa-v3-large (FT) | MiniLM-L6-v2 (FT) | MPNet (FT) |
|---|---|---|---|
| Loss | 0.022 | – | – |
| Accuracy | 0.996 | – | – |
| Precision | 0.993 | 0.990 | 0.980 |
| Recall | 0.996 | 0.910 | 0.970 |
| F1 Score | 0.995 | 0.948 | 0.975 |

The ROC curve 1b with a blue line hugging the top left corner and an AUC of 1.00, indicates that our model perfectly separates positives from negatives on this test set, essentially achieving 100 % sensitivity and specificity at an optimal threshold.

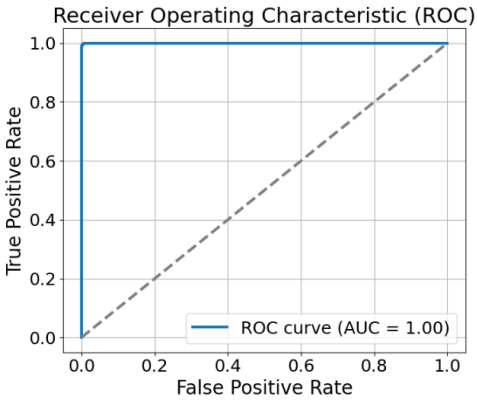
## 5. Error Analysis

For Sub-Task 1, our model will at times default to an adlike structure, i.e., normal text, but with awkward phrasing that seems promotional in nature. This could be attributed to the segments having text of that nature or to the model not always being able to achieve the desired goal.

Conversely, when asked to weave in promotional elements subtly, the output is generally very competent, but at times will fail to blend messaging organically and instead resorts to forced insertions




(a) Confusion matrix comparison on test set 1




(b) ROC performance on test set 2.

**Figure 1:** The results of DeBERTa-v3-large

that disrupt the surrounding text. For instance, it may abruptly drop in a branded phrase or tagline mid-paragraph, or tack on a sales-style endorsement that jars against the neutral tone. This may be put down to the structure of our ad generation pipeline.

Specifically for Sub-Task 2, our model's false positives, i.e., neutral content misclassified as ads, appear to be caused by generic brand mentions or informational lists that lack explicit promotional intent. For example, the mention of 'FORM, a holistic wellness program' (ID 2550), was flagged as an ad due to the brand name and phrases such as 'community-driven', even though it was part of a biographical description. Similarly, travel site lists (ID 2274) triggered false alarms because they included terms such as 'best deals' and 'user-friendly', which the model associated with ads despite their neutral and comparative context. These errors highlight the fact that the model relies on surface-level keywords and structural patterns without any deeper contextual analysis. The model conflates factual descriptions with covert advertising, despite a lack of persuasive language.

Correctly classified ads with low confidence, for example: ID 1660, 1116 reflect the model's struggle to identify subtle integration of promotional content. For example, the Fox live-streaming response (ID 1660) includes phrases like "unveil a world of entertainment" and app download instructions, which are softly persuasive but lack direct calls to action (e.g., "sign up now"). Similarly, property management ads (ID 1116) list services with mild promotional language: "secure and efficient self-guided touring technology," blending ads into informative content. The model's uncertainty arises because these ads avoid overt markers and instead rely on value-driven descriptors that overlap with neutral advice. These cases are close to the model's decision boundary, where the absence of strong ad-specific signals reduces confidence, even when the predictions are correct. Improving detection here requires training the model to recognize implicit persuasion rather than relying solely on explicit triggers.

## 6. Conclusion

In this project, we presented a unified framework for both the seamless integration and robust detection of advertisements in Retrieval-Augmented Generation (RAG) systems. For Sub-Task 1, our three-stage retrieval pipeline — combining Elasticsearch relevance, sentence-transformer embeddings, and cross-encoder reranking — provided highly pertinent document contexts to the chosen model, enabling the generation of coherent, natural sounding responses with embedded ads. Evaluation against official metrics confirmed high precision in ad placement while maintaining overall language fluency, although further improvements in recall of the subtler insertions remain possible.

For Sub-Task 2, we fine-tuned a suite of transformer-based classifiers, from MPNet-v2 through DeBERTa-v3-large, achieving a top F1 score of 99.50% and accuracy of 99.60% with the largest model. These results underscore the strength of transformer fine-tuning for detecting native advertisements,

even when they are carefully blended into organic content.

## 7. Future Work

Looking ahead, integrating reinforcement-learning based ad insertion policies would likely further enhance our already impressive scores. We could also try to finetune our LLM model so that it knows the transitions for ads so they're naturally added while being subtle enough. On the detection side, given the near-ceiling performance of the model in the existing test set, we plan to enrich our training data by using our outputs from Sub-Task 1. Specifically, we will generate a diverse set of ad-injected responses using our RAG-based generator and then label these automatically (or via lightweight human review) to augment the Sub-Task 2 dataset. This synthetic data will expose the classifier to a wider variety of subtle advertising patterns, improving its ability to detect novel or adversarial insertions. This will make the model generalize better to real-world data and different variants of ads.

## Acknowledgements

The authors would like to acknowledge the support provided by the Office of Research (OoR) at Habib University, Karachi, Pakistan, for funding this project through the internal research grant IRG-2235.

## Declaration on Generative AI

During the preparation of this work, the authors employed ChatGPT and Grammarly AI tools for grammar checking, paraphrasing, rewording and consistency checking of sentences. After using the tools, the authors reviewed and edited the content as required and thereby take full responsibility for the publication's content.

## References

[1] Kiesel, J., Çöltekin, Ç., Gohsen, M., Heineking, S., Heinrich, M., Fröbe, M., Hagen, T., Aliannejadi, M., Erjavec, T., … Stein, B. (2025). *Overview of Touché 2025: Argumentation Systems*. In CLEF 2025: Conference and Labs of the Evaluation Forum, Madrid, Spain.

[2] Kok-Shun, B. V., & Chan, J. (2025). Leveraging ChatGPT for sponsored ad detection and keyword extraction in YouTube videos [Work-in-progress paper]. *arXiv*. https://doi.org/10.48550/arXiv.2502.15102

[3] Wang, X., Gu, X., Cao, J., Zhao, Z., Yan, Y., Middha, B., & Xie, X. (2021). Reinforcing pretrained models for generating attractive text advertisements. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (pp. 3697–3707). https://doi.org/10.1145/3447548.3467105

[4] Wang, S., Hu, C., & Jia, G. (2024). Deep learning-based saliency assessment model for product placement in video advertisements. *Journal of Applied Computer Science*. https://doi.org/10.69987/JACS.2024.40503

[5] Hajiaghayi, M. T., Lahaie, S., Rezaei, K., & Shin, S. (2024). Ad auctions for LLMs via retrieval augmented generation [Preprint]. *arXiv*. https://doi.org/10.48550/arXiv.2406.09459

[6] Schmidt, S., Zelch, I., Bevendorff, J., Stein, B., Hagen, M., & Potthast, M. (2024). Detecting generated native ads in conversational search [Preprint]. *arXiv*. https://doi.org/10.48550/arXiv.2402.04889

[7] Zelch, I., Hagen, M., & Potthast, M. (2023). Commercialized generative AI: A critical study of the feasibility and ethics of generating native advertising using large language models in conversational web search [Preprint]. *arXiv*. https://doi.org/10.48550/arXiv.2310.04892

[8] Borchers, C., Gala, D. S., Gilburt, B., Oravkin, E., Bounsi, W., Asano, Y. M., & Kirk, H. R. (2022). Looking for a handsome carpenter! Debiasing GPT-3 job advertisements [Preprint]. *arXiv*. https://doi.org/10.48550/arXiv.2205.11374

[9] Feizi, S., Hajiaghayi, M. T., Rezaei, K., & Shin, S. (2023). Online advertisements with LLMs: Opportunities and challenges [Preprint]. *arXiv*. https://doi.org/10.48550/arXiv.2311.07601

[10] Meguellati, E., Han, L., Bernstein, A., Sadiq, S., & Demartini, G. (2024). How good are LLMs in generating personalized advertisements. In *WWW '24: Companion Proceedings of the ACM Web Conference 2024* (pp. 826–829). https://doi.org/10.1145/3589335.3651520

[11] Isozaki, I. (2023, November 26). Literature review on RAG (Retrieval Augmented Generation) for custom domains. *Medium*. Retrieved from: https://isamu-website.medium.com/literature-review-on-rag-retrieval-augmented-generation-for-custom-domains-325bcef98be4.

[12] Qwen Team, An, Y., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., Lu, K., Bao, K., Yang, K., Yu, L., Li, M., Xue, M., Zhang, P., Zhu, Q., Men, R., Lin, R., Li, T., Tang, T., Xia, T., Ren, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Wan, Y., Liu, Y., Cui, Z., Zhang, Z., Qiu, Z. (2025). Qwen2.5 Technical Report [Preprint]. *arXiv*. https://arxiv.org/abs/2412.15115

[13] Qwen Team. (2025). Qwen2.5-14B-Instruct. *Hugging Face Model Card*. https://huggingface.co/Qwen/Qwen2.5-14B-Instruct

[14] Song, K., Tan, X., Qin, T., Lu, J., & Liu, T.-Y. (2020). MPNet: Masked and Permuted Pre-training for Language Understanding [Preprint]. *arXiv*. https://arxiv.org/abs/2004.09297

[15] Sentence-Transformers Team. (2023). all-mpnet-base-v2. *Hugging Face Model Card*. https://huggingface.co/sentence-transformers/all-mpnet-base-v2

[16] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach [Preprint]. *arXiv*. https://arxiv.org/abs/1907.11692

[17] Hugging Face Inc. (2024). RoBERTa model documentation. *Hugging Face Transformers Docs*. https://huggingface.co/docs/transformers/en/model_doc/roberta

[18] Facebook AI. (2024). roberta-large. *Hugging Face Model Card*. https://huggingface.co/FacebookAI/roberta-large

[19] He, P., Liu, X., Gao, J., & Chen, W. (2021). DeBERTa: Decoding-enhanced BERT with Disentangled Attention [Preprint]. *arXiv*. https://arxiv.org/abs/2006.03654

[20] Hugging Face Inc. (2024). DeBERTa model documentation. *Hugging Face Transformers Docs*. https://huggingface.co/docs/transformers/model_doc/deberta

[21] Microsoft. (2024). deberta-v3-base. *Hugging Face Model Card*. https://huggingface.co/microsoft/deberta-v3-base

[22] Microsoft. (2024). deberta-v3-large. *Hugging Face Model Card*. https://huggingface.co/microsoft/deberta-v3-large

[23] 0x7o. (2024). roberta-base-ad-detector. *Hugging Face Model Card*. https://huggingface.co/0x7o/roberta-base-ad-detector

[24] Wang, X., Wang, B., Wang, R., & Liu, W. (2020). MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. In *Findings of EMNLP* (pp. 4688–4696). https://doi.org/10.18653/v1/2020.findings-emnlp.418

[25] Microsoft. (2023). microsoft/MiniLM-L6-v2. *Hugging Face Model Card*. https://huggingface.co/microsoft/MiniLM-L6-v2

[26] cross-encoder. (2023). Cross-Encoder for MS MARCO (ms-marco-MiniLM-L6-v2) [Model card]. *Hugging Face*. https://huggingface.co/cross-encoder/ms-marco-MiniLM-L6-v2

[27] Douze, M., Guzhva, A., Deng, C., Johnson, J., Szilvasy, G., Mazaré, P.-E., Lomeli, M., Hosseini, L., & Jégou, H. (2024). The Faiss library [Preprint]. *arXiv*. https://arxiv.org/abs/2401.08281

[28] Fröbe, M. et al. (2023). Continuous Integration for Reproducible Shared Tasks with TIRA.io. *In: Kamps, J., et al. Advances in Information Retrieval. ECIR 2023. Lecture Notes in Computer Science, vol 13982. Springer, Cham.*. https://doi.org/10.1007/978-3-031-28241-6_20