Descriptive Adverse Drug Reaction Prediction via Hybrid Retrieval and Balanced-λ MMR Re-ranking

Anonymous ACL submission

Abstract

Adverse drug reaction (ADR) prediction involves two complementary tasks: structured inference (classifying the occurrence of specific ADR outcomes) and unstructured inference (generating narrative descriptions of those outcomes using preferred terminology). Existing methods predominantly focus on structured ADR prediction, and use supervised learning based on hand-engineered features that result in limited generalization, while unstructured ADR narrative prediction remains largely unexplored. In this work, we propose a novel ADR prediction framework that jointly predicts structured and unstructured outcomes. Our framework leverages an open source large language model (LLM) for semantic representation and ADR knowledge retrieval in a three-stage pipeline to simultaneously predict both structured and unstructured outcomes in a generalized manner. First, we fine-tune two ADR-specific embedding models based on a benchmark foundation model to align the embedding space with domain-specific ADR terminology. Second, we construct a novel hybrid retrieval pipeline that integrates BM25 lexical matching with dense vector similarity search to improves retrieval recall. Third, we apply a Maximal Marginal Relevance (MMR) re-ranking strategy to balance relevance and diversity. Evaluation on three held-out test sets from FDA Adverse Event Reporting System (FAERS) quarterly demonstrates that our method achieves higher top-1 classification accuracy for structured ADR prediction and improves recall for top-1 unstructured ADR narrative descriptions by 47% compared with baseline approaches.

1 Introduction

Adverse drug reactions (ADRs) pose a significant risk to patient safety and are a leading cause of mortality (Makary and Daniel, 2016; American Society of Pharmacovigilance, 2025). The scale of this challenge is reflected in large pharmacovigilance systems such as the U.S. FDA Adverse Event Reporting System (FAERS), which recorded over 1.25 million serious ADR cases in 2022, including nearly 175,000 fatalities (Kommu et al., 2024). Figure 1 illustrates the ADR prediction task, which involves both structured outcome labels (e.g., "Death") and unstructured narrative descriptions derived from patient and treatment information. As the number of approved and experimental drugs continues to grow, there is increasing demand for automated methods capable of interpreting large-scale, heterogeneous ADR reports.

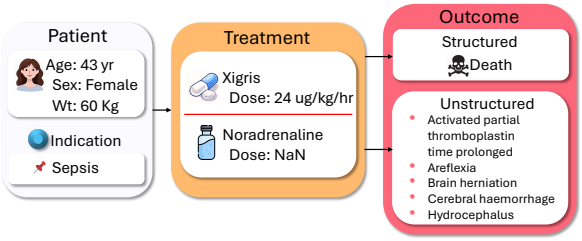

Figure 1: An Example of Structured and Unstructured ADR Outcomes

Existing approaches primarily focus on structured inference, assigning predefined outcome labels using supervised learning models (Deimazar and Sheikhtaheri, 2023). While effective in controlled settings, these methods rely on hand-engineered features and fixed label spaces, limiting generalization to emerging or previously unseen ADR patterns (Mullard, 2025). In contrast, unstructured inference—generating clinically meaningful narrative descriptions—remains underexplored despite its importance in real-world pharmacovigilance. Jointly modeling structured classification and unstructured generation is challenging, particularly due to the need for domain-specific semantic understanding and robust retrieval from large, noisy corpora such as FAERS.

To address this, we propose a unified ADR prediction framework that jointly supports structured classification and unstructured narrative generation via a retrieval-augmented paradigm. Given an

ADR triplet (patient, drug, indication), the framework encodes the input into a domain-adapted semantic space, retrieves relevant historical ADR cases, and uses these signals to generate both structured labels and narrative outputs.The framework consists of three stages. First, we construct a training dataset from FAERS and fine-tune two ADR-specific embedding models based on BGE-M3 to align representations with domain-specific terminology. Second, we develop a hybrid retrieval pipeline that combines BM25 lexical matching with dense vector search to improve recall and relevance. Third, we apply Maximal Marginal Relevance (MMR) with optimized λ to balance relevance and diversity, reducing redundancy while capturing a broader range of ADR outcomes.

We evaluate the framework on three held-out FAERS quarterly datasets to reflect temporal shifts in ADR distributions. To enable scalable and unbiased evaluation, we map ADR Preferred Terms (PTs) to higher-level System Organ Classes (SOCs) using the Unified Medical Language System (UMLS) data[1], avoiding reliance on similarity-based thresholds.

Results show that our method outperforms open-source LLM baselines. For structured classification, it achieves 48.5% Top-1 and 81% Top-5 accuracy, surpassing GPT-OSS-20B (37.5%) and the supervised MonoT5 baseline (43.3%). For unstructured prediction, it yields a 47% recall improvement and a two-fold gain with Top-5 retrieval. These results demonstrate the robustness and effectiveness of the proposed retrieval-based approach for ADR prediction.

Our key contributions can be summarized as follows:

- We propose the first framework that jointly predicts structured ADR outcomes and unstructured narrative descriptions in a generalized manner by leveraging LLM.
- We propose a novel, three-stage hybrid retrieval framework for generalized detection of ADR outcomes consisting of 2 ADR-specific embedding models, a hybrid retrieval strategy, and an MMR-driven re-ranking [2].
- We design a scalable SOC-level evaluation protocol using UMLS mappings to enable large-scale ADR prediction benchmarking.

[1] https://www.nlm.nih.gov/research/umls/index.html

[2] Our framework will be open-sourced @ https://anonymous.4open.science/r/acl_2026-73F1.

2 Related Work

Existing ADR prediction research has largely concentrated on two separate tasks. The first focuses on the extraction of ADR-related information from free-text sources such as clinical notes, patient reports, and spontaneous narrative reports. These tasks are typically addressed through natural language processing (NLP) techniques such as named entity recognition (NER), relation extraction, and text classification have been developed to detect ADR from unstructured data like social media, clinical notes, or pharmacovigilance narratives (Sarker and Gonzalez, 2015; Ben Abacha et al., 2015; Tiftikci, 2019) More recent studies have incorporated deep learning models and contextual language representations to improve robustness against linguistic variability and noise in free-text data (Haq et al., 2023). While effective at detecting ADR mentions, these methods primarily support information extraction rather than predictive inference or narrative generation. In contrast, the second line of task involves structured ADR occurrence classification using supervised learning models that heavily rely on the specifically engineered features as structured input and annotated ADR outcomes, thereby limiting generalization beyond predefined inputs (Farnoush et al., 2024; Choudhury et al., 2020; Bresso et al., 2021).

Advances in language model fine-tuning, particularly BERT and its biomedical variants, have improved the representation of ADR-related semantics and enhanced downstream prediction tasks (Guo and Choo, 2025). LLMs can also be leveraged to generate unstructured ADR narratives in preferred clinic terminology. This shift enables more effective exploitation of rich, unstructured healthcare data, despite the inherent challenges posed by complex domain knowledge and heterogeneous data structures.

3 Datasets

ADR Data Extraction, Transformation, and Loading (ETL) FAERS releases a quarterly data dump as a compressed archive that contains a set of plain-text files together with processing instructions. For each quarter, the archive includes five core tables that we use to reconstruct individual ADR cases identified by a unique caseID.

The ETL Workflow includes 1) Data extraction - ADR cases are extracted from the zip archive is unpacked. Each text file is parsed according

FDA's specifications. 2) Data transformation – Key alignment–records from the five tables are joined on caseID to form a single, patient-centric record. 3) Outcome handling – the ADR outcome (OUTC) provides a categorical label for seven ADR outcomes (Table 1). Many reports lack an outcome code (26%). These incomplete records are retained for model training but are excluded from evaluation of classification accuracy. Meanwhile, a unique outcome code was assigned to a case given the highest severity class in the outcome code. 4) Missing-value imputation – When a field is missing except drugs and indications (e.g., dose frequency), we replace with a NA. 5) Data formatting – every ADR case is constructed as instruction and outcome pair both in jsonl format (Section 4.1) and written to a Parquet file, enabling efficient downstream vector-indexing and BM25 indexing.

Preparation of training includes splitting the dataset into training and testing sets. The training set includes quarterly releases from 2004Q1 to 2024Q3. To maximize exposure to diverse drug-reaction patterns, missing outcome codes ADR cases are included. The testing set is prepared for performance evaluation covering quarterly releases from 2004Q4 through 2025Q2. We restrict to ADR cases that possess a non-NA outcome label and a non-empty reaction narrative, ensuring a fair comparison of structured-classification and unstructured-generation metrics.

An expert clinical validation was conducted to assess the precision of mapping the ADR PTs to 27 SOCs. A clinical expert manually reviewed 500 random PT–SOC mappings derived from our automated mapping pipeline. Among these, 87 mappings (17.4%) were identified as misclassified, indicating that the assigned SOC might not adequately reflect the clinical context of PT. Analysis of the misclassified cases revealed that the most frequent errors occurred in three SOC categories: *General disorders and administration site conditions*, *Respiratory, thoracic and mediastinal disorders* and *Neoplasms benign, malignant and unspecified (including cysts and polyps)*, each accounting for 8 misclassified cases. These findings highlight the inherent ambiguity of certain ADR PTs and the challenges of automated SOC mapping, particularly for terms that may plausibly belong to multiple physiological systems. The expert validation therefore provides an important quality check on the mapping framework and helps identify systematic areas where refinement of the mapping rules or embedding-based classification may further improve accuracy.

Code	Description
DE	Death
LT	Life-Threatening
HO	Hospitalization-Initial or Prolonged
DS	Disability
CA	Congenital Anomaly
RI	Required Intervention to Prevent Permanent Impairment/Damage
OT	Other Serious (Important Medical Event)

Table 1: Classification of Seven Structured ADR Outcomes

4 Our Approach

4.1 Construction of ADR Instruction and Outcome

Firstly, aggregated FAERS tabular data are prepared from quarterly releases dating back to 2004. Each ADR case comprises patient demographics, drug administration details, medication indications, ADR narratives expressed in preferred terms (PTs), and ADR outcome classifications. Each case is formulated as an instruction-outcome pair. The instruction component consists of 3 distinct fields that collectively capture the semi-structured information associated with each case: 1) Patient demographics, including age, age code, gender, weight, and weight unit; 2) Drug administration, detailing the drug name, route, dose, frequency, formulation; and 3) Medication indications, describing therapeutic purpose for which the drug was prescribed. We refer to this three-component construct as the ADR triplet extraction hereafter. To preserve the column-wise relationships inherent in the original FAERS data while enabling downstream information-retrieval and generative tasks, we store each instruction–outcome pair in a line-delimited JSON (JSONL) file. Each line contains a self-contained JSON object with the following schema:

{""patient": ""age": "43.0", "age_cod": "yr", "gndr_cod": "f", "wt": "60.0", "wt_cod": "kg""," treatment": ""drugname": "xigris", "route": "other", "dose": "24 ug/kg/hr/other"; "drugname": "noradrenaline", "route": nan, "dose": nan", "indi_pt":"sepsis"}

Each ADR case outcome comprises 2 comple-

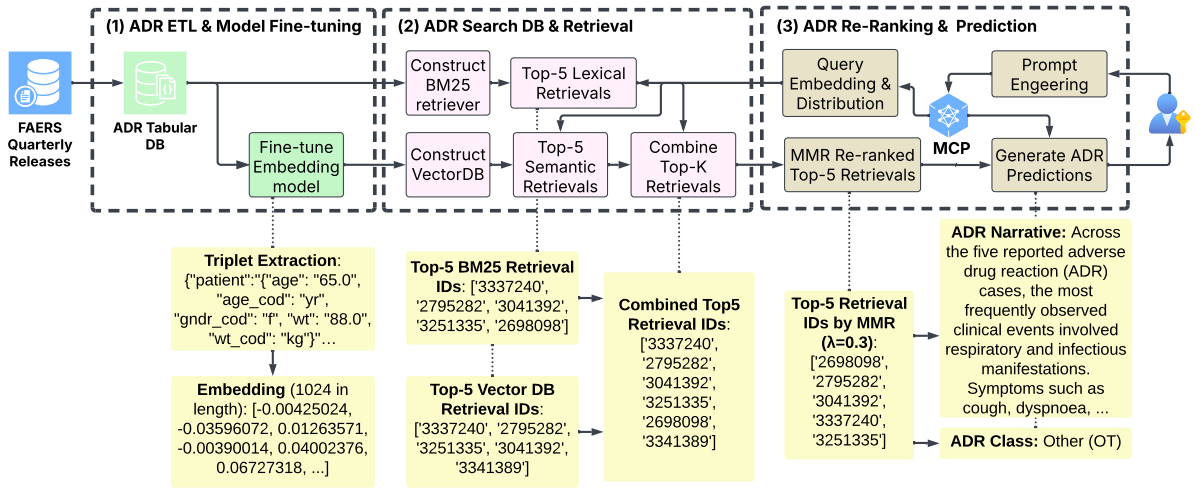

Figure 2: Overview of 3-stage framework: (1) ADR ETL & Model Fine-tuning, (2) ADR Search DB construction & Retrieval, and (3) ADR Re-ranking & Prediction. The example illustrates how structured triplets are embedded, top-k lexical and semantic samples are combined, followed by MMR-based re-ranking to predict both unstructured narrative and structured ADR class.

mentary predictions. The unstructured component is ADR narratives represented as a description expressed in MedDRA (Medical Dictionary for Regulatory Activities) concatenated PTs, while the structured component is an outcome code encoded as a single categorical label corresponding to the most severe outcome reported for the case. When multiple outcome codes are present, the one with the highest severity ranking is retained.

Just as the instruction is stored as a line-delimited JSON object, the ADR outcome is encoded in the same JSONL format. An illustrative example is shown below:

> {"pt": "activated partial thromboplastin time prolonged; areflexia; brain herniation; cerebral haemorrhage; hydrocephalus; platelet count decreased; prothrombin time prolonged; pupil fixed", "uni_code": "DE"}

The JSONL representation is the query format for both BM25/vector hybrid retrieval stage and subsequent generation step in the hybrid RAG pipeline.

4.2 Fine-tuning Embedding Model

RAG pipeline depends on high-quality retrieved passages to guide the downstream LLM in ADR prediction. Instead of fine-tuning an open-source LLM for the ADR prediction which is a computationally intensive and difficult process to keep up with new quarterly FAERS releases, we adopt a more flexible strategy by fine-tuning sentence-level embedding models that can be updated quickly to yield hallucination-free evidence (Kraišniković et al., 2025; Ng et al.; Schroff et al., 2015; Hoffer and Ailon, 2015).

Because no publicly available embedding model has been specifically trained for ADR data, we first evaluated the state-of-the-art models on the Massive Text Embedding Benchmark (MTEB) [3] and selected the open-source BAAI/bge-m3 model [4] as the foundation model. Bge-m3 can ingest up to 8,192 tokens and produces 1,024-dimensional embeddings, which is sufficient to capture the rich information contained in an ADR triplet extraction.

In this study, we fine-tuned two ADR-specific embedding models using Bge-m3 as the base model, applying two different training strategies. The first model was fine-tuned using Matryoshka representation learning on 500K randomly sampled pairs of ADR triplet extractions and outcome strings (Kusupati et al., 2024). After each training epoch, the model was evaluated on a validation set (20% of training set) using three retrieval metrics computed over the top-K results: Normalized Discounted Cumulative Gain (NDCG), Mean Reciprocal Rank (MRR), and Mean Average Precision (MAP). Last checkpoint from epoch 3 showing the highest scores across all three metrics was selected as the final embedding model for

[3] https://huggingface.co/spaces/mteb/leaderboard

[4] https://huggingface.co/BAAI/Bge-m3

vector BD construction. The entire fine-tuning process completed in approximately 7 hours. The evaluation metrics are reported in Table 2.

	Fine-tuned Metrics	Enrichment Factor
Cosine-Accuracy@1	0.00214	214
Cosine-Accuracy@3	0.00535	178.3
Cosine-Accuracy@5	0.008	160
Cosine-Precision@1	0.00214	214
Cosine-Precision@3	0.00178	59.3
Cosine-Precision@5	0.0016	32
Cosine-Recall@1	0.00214	214
Cosine-Recall@3	0.00535	178.3
Cosine-Recall@5	0.008	160
NDCG@10	0.00476	101.60
MRR@10	0.00666	227.38
MAP@100	0.00628	121.06

Table 2: Finetuning Bge-m3 with Matryoshka Representation Learning

The second embedding model was fine-tuned using triplet-based deep metric learning which learns representations by optimizing triplet loss and enforces relative distance constraints among samples. The objective of triplet loss is to ensure that an anchor example is embedded closer to a semantically related positive example than to an unrelated negative example by a predefined margin of 0.3. The model was trained on 1 million random ADR samples with 10% of the cases reserved for validation, where each input consists of an ADR instruction string as the query (anchor), the corresponding reported outcome as the positive example, and three randomly sampled outcomes (excluding the positive) as negative examples(Schroff et al., 2015; Hoffer and Ailon, 2015). After 3 epochs with 1,845,036 steps, the cosine accuracy on validate set achieved 0.9487.

Both supervised fine-tuning procedures used ADR cases reported in the FDA Adverse Event Reporting System corpus from 2004Q1 to 2024Q3. Early stopping and hyperparameter tuning were performed on a single NVIDIA H100 (80GB) GPU, and both training processes were conducted for 3 epochs for 36 hours.

4.3 Construction of Vector Database

To support fast similarity search over millions of ADR triplet embeddings, we store the vectors in a LanceDB database that provides Approximate Nearest-Neighbor (ANN) indexes, specifically a joint indexing approach with Inverted File Index, Hierarchical Navigable Small World, and Product Quantization (IVF-HNSW-PQ) [5] with 256 partitions and 128 sub_vectors. This setting eliminates the need for exhaustive linear scans of the entire vector space and optimizes performance trade-offs.

All triplet extraction embeddings of training set generated by the fine-tuned ADR embedding model (Section 4.2) are ingested into LanceDB as a persistent column-wise vector field. When a query is issued, the top-K (e.g., 5) most similar ADR cases are returned, ranked by the descending cosine similarity order. Because the Bge-M3 model outputs unit-norm embeddings ($\|v\|=1$), cosine similarity reduces to a simple dot product. Consequently, the retrieval step can be performed with a single inner-product operation, further accelerating the ANN search. The result is a ranked list of ADR cases that can be passed to the subsequent hybrid retrieval fusion and MMR re-ranking stages.

4.4 Construction of BM25 Retriever

Using the same training corpus of triplet extraction–outcome pairs for embedding model finetuning, we constructed a BM25-based lexical retrieval pipeline as the first-stage retriever of proposed hybrid RAG framework. Each pair was indexed as a unified retrieval unit. The corpus was indexed using the BM25 ranking function, which scores relevance based on term frequency, inverse document frequency, and document length normalization. At inference time, inputs of ADR were processed using the same pipeline and matched against the BM25 index to retrieve the top-K ADR outcomes. This lexical retrieval stage provides high-precision matching in particular for drug names and indication details in PT.

4.5 Hybrid Retrieval Combination and MMR Re-ranking

To integrate lexical and semantic retrieval signals, we employed a hybrid retrieval strategy that combines sparse BM25 retrieval and dense vector similarity search. For each query, the top 5 ADR candidates were independently retrieved from the BM25 index and the dense vector database, yielding a fused candidate set up to 10 after duplicate removal. The merged candidates were subsequently re-ranked using the MMR algorithm (Car-

[5] https://docs.lancedb.com/indexing

bonell and Goldstein, 1998), which iteratively selects ADR candidates by balancing relevance to the query and diversity among the already selected results. Specifically, at each selection step, an ADR candidate d_i is chosen to maximize

$$MMR_{d_i} = \lambda sim(d_i, q) - (1 - \lambda) \max_{d_j \in S}(sim(d_i, d_j))$$

where q denotes the ADR query, S is the set of previously selected ADR cases, *sim()* represents the similarity function, and tunable $\lambda \in [0, 1]$ controls the trade-off between relevance and diversity. Higher values of λ prioritize query relevance, whereas lower values emphasize diversity by penalizing redundancy.

4.6 monoT5 for ADR Uni_code Classification

MonoT5, a sequence-to-sequence framework built on T5, represents a strong supervised ranking baseline widely used in information retrieval(Nogueira et al., 2020). We adopt MonoT5 to perform text–class classification between ADR triplet extraction inputs and the uni_code outcome classes, determining whether a given pair represents a valid mapping . A total of 250K random samples from the training period (2004Q1–2024Q3) were prepared and used to fine-tune the castorini/monot5-base-msmarco[6] base model from Hugging Face to predict the target relevance token. During inference, the conditional likelihood of the relevance token is used as the classification score, enabling ranking over the seven candidate labels described in Table 1. Supervised training is performed using labeled ADR–label pairs derived from the training corpus.

4.7 Map ADR PT to SOC

ADR cases reported to FAERS are not systematically reviewed by clinical pharmacologists, raising data-quality concerns (Veronin et al., 2020; Giunchi et al., 2023; Bodenreider, 2004). However, manual verification of millions of heterogeneous ADR reports is infeasible. To address this limitation, we propose SOC-level evaluation that provides a clinically meaningful abstraction that reduces annotation noise and enables scalable evaluation across millions of heterogeneous ADR reports. This approach leverages concept unique identifiers (CUIs) from the Unified Medical Language System (UMLS) 2025AB Release to map 26,912 PTs to their corresponding 27 SOCs based on semantic similarity between PT and SOC descriptions. For each ADR case, two fine-tuned ADR embedding models described in Section 4.2 were used independently to map each PT, thereby reducing potential model bias. The SOC associated with the higher Cosine Similarity score is selected as the best SOC match.

To quantify the relevance between the ADR outcome and the top-5 retrieved outcomes, we assign a score of 1 if at least one SOC matches and 0 otherwise. This roll-up SOC-level matching eliminates the need for exhaustive expert review while preserving clinically meaningful outcome categories. The resulting SOC class is used as the evaluation target for the unstructured prediction results reported in Section 5.4.

4.8 Agentic AI Orchestration for ADR prediction

To perform inference with the hybrid pipelines, user inputs are first integrated and converted into a JSONL that encodes the triplet extraction from processed FEARS data. The query is then dispatched in parallel to a BM25 index for sparse lexical matching and to a dense vector DB for semantic search using fine-tuned bge-m3 embeddings. Each retrieval engine returns its top-5 results, yielding up to 10 candidate evidences. After de-duplication, the remaining retrievals are concatenated and passed to a MMR re-ranking module with λ = 0.5. The top-5 retrievals are serialized into a single JSON-formatted context and appended to the original query to form the prompt for downstream LLM inference(Appendix 8). This assembled JSONL entry is sent to a locally hosted LLM via a REST API. The implementation of the proposed ADR prediction framework is available at `https://anonymous.4open.science/r/acl_2026-73F1`.

5 Evaluation

5.1 Evaluation of Fine-tuned Embedding Model

We reported the gains of fine-tuned bge-m3 embedding model Enrichment Factor (EF), defined as the ratio of the observed hit rate to the expected hit rate of a random baseline (1/100,000 for a Top-1 hit)(Kusupati et al., 2024). Performance was evaluated on a 100K validation set held out from 500K randomly sampled ADR cases. After 3 training epochs (150K fine-tuning steps), the fine-tuned

[6] https://huggingface.co/castorini/monot5-base-msmarco

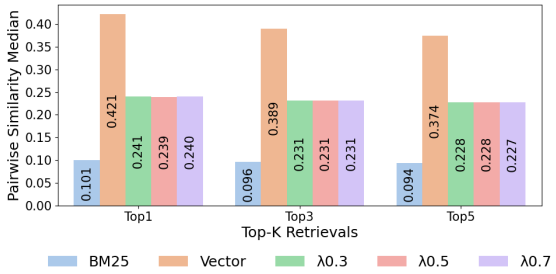

Figure 3: Relevance & Redundancy Analyses

model achieved EF values substantially above the random baseline. Specifically, the model improves top-1 cosine-similarity hit rate by 214× related to random chance and maintains strong enrichment across higher top-K thresholds. With EF values of 178× at Top-3 and 160× at Top-5. Comparable gains are observed across other retrieval and accuracy metrics, demonstrating that embedding fine-tuning substantially amplifies semantic signal and retrieval effectiveness for large-scale ADR data. More details are available in Table 2.

5.2 Redundancy Analysis of Retrievals

To assess how MMR affects redundancy in the retrieved evidence, we computed the average pairwise cosine similarity between the unstructured ADR outcomes among top-5 retrieved items for each query. We then averaged these pairwise similarity scores across all queries within each retrieval pipeline. Because cosine similarity measures the degree of embedding overlap among ADR triplet extraction representations, higher values indicate greater redundancy among the retrieved evidence, whereas lower values suggest more diverse retrieval results.

Figure 3 shows the medians of pairwise cosine similarity among retrievals across 5 pipelines. The BM25-only and vector-only pipelines represent two extremes of retrieval behavior with markedly different similarity levels. In contrast, MMR pipelines with 3 λ settings (λ = 0.3, 0.5, 0.7) show only minor differences. As retrieval depth increases from Top-1 to Top-5, similarity gradually decreases across all pipelines, indicating that the most relevant document is preserved remain at Top-1 position after re-ranking, while the additional items in Top-3 and Top-5 become more diverse and less redundant. This controlled diversity explains the improved recall observed for the hybrid RAG-based unstructured ADR prediction described in Section 5.4.

5.3 Impact of MMR on Structured ADR Classification

We evaluate the impact of three MMR weighting parameters λ values of 0.3, 0.5 and 0.7 on the combined hybrid retrievals and compared them with two hybrid-retrieval BM25 and vector and open-source baselines LLMs GPT-OSS-20B for structured ADR prediction across 3 sequential OOT quarters from 2024 Q4 to 2025 Q2.

Given the computational cost of large language model inference, we selected a stratified random sample of 3,517 cases (3%) from the OOT dataset. Open-source LLMs were prompted to generate ADR uni_code predictions, using the OOT triplet extraction as contextual guidance within the prompt. The structured ADR classifications extracted from the model responses were then evaluated for prediction accuracy. The baseline LLM GPT-OSS-20B achieves average accuracies of **0.373**, respectively, with minimal quarter-to-quarter variation.

We also evaluate ADR uni_code classification using the supervised learning model MonoT5, the model achieves a prediction accuracy of **0.433** on the OOT set. This performance is slightly higher than that obtained from the out-of-the-box LLM baselines, suggesting that supervised fine-tuning with task-specific examples provides measurable improvements for structured ADR outcome classification.

Next, we report classification accuracies for Top-1, Top-3, and Top-5 retrieval settings across three OOT quarters, comparing two hybrid retrieval pipelines and three MMR re-ranked pipelines with $\lambda = 0.3, 0.5$, and 0.7, as shown in Table 3. For Top-1 performance, the vector search pipeline achieves the highest classification accuracy in the earliest quarter, reaching **0.485** in 24Q4. However, performance declines in later quarters, dropping to **0.448** in 25Q1 and **0.428** in 25Q2, suggesting the presence of data drift and highlighting the need for continual updates of ADR cases from FAERS. Notably, this degradation in newer quarters is partially mitigated by MMR-based re-ranking. Specifically, Top-5 retrievals re-ranked with MMR ($\lambda = 0.3$ and 0.7) achieve the highest accuracy of **0.789** among all pipelines for 25Q1, while Top-5 retrievals with MMR ($\lambda = 0.5$) yield the best accuracy for 25Q2, suggesting that a balanced λ value may generalize better for future ADR classification tasks.

Overall, vector pipeline outperform the strongest

Yr_Qtr TopK	BM25	Vector	λ0.3	λ0.5	λ0.7
24q4-1	0.237	**0.485**	0.335	0.334	0.334
24q4-3	0.538	**0.715**	0.664	0.664	0.667
24q4-5	0.694	**0.810**	0.801	0.800	0.804
25q1-1	0.238	**0.448**	0.328	0.323	0.326
25q1-3	0.540	**0.688**	0.649	0.650	0.650
25q1-5	0.697	0.787	**0.789**	0.787	**0.789**
25q2-1	0.238	**0.428**	0.320	0.322	0.319
25q2-3	0.533	**0.677**	0.641	0.646	0.642
25q2-5	0.693	0.782	0.782	**0.784**	0.782

Table 3: Top-K Classification of OOT by Quarter

monoT5 baseline model. Given its superior performance on near-term data, the vector-based retrieval pipeline is recommended for top-1 and top-3 ADR classification. However, due to the diversity-promoting nature of MMR, its Top-1 accuracy is generally lower than that of vector-based pipeline but better than pure BM25 pipeline. MMR-re-ranked pipelines perform others when considering the top-5 options and less sensitive to data drift, demonstrating that MMR re-ranking helps surface rare and future ADR cases that would otherwise be missed under temporal drift.

5.4 Impact of MMR on Unstructured ADR Prediction

To understand the impact of three λ settings on unstructured ADR predictions focusing on capture of ADR outcomes, we examine the recall rate at the Top-1, Top-3, and Top-5 cut-offs by quarter. Using the ADR PT matching as described in section 4.7 and same promote for the baseline LLMs, recall rates on ADR narratives by GPT-OSS-20B is **0.406** comparable to top-1.

Table 4 reports classification accuracies across three OOT quarters under different retrieval strategies and Top-K settings. Overall, performance improves consistently as the number of retrieved items increases from Top-1 to Top-5 across all pipelines. The vector retrieval pipeline achieves the strongest Top-1 performance, reaching 0.597 in 24Q4 and drop slightly in later quarters (0.571 in 25Q1 and 0.570 in 25Q2), substantially outperforming the BM25 baseline. Interestingly, as K increases, the advantage of Maximal Marginal Relevance (MMR) re-ranking becomes more apparent. For Top-5 retrievals, MMR-based pipelines consistently match or slightly exceed the vector baseline across all quarters, achieving the highest accuracy of 0.856 in 24Q4 ($\lambda = 0.7$), 0.853 in 25Q1 ($\lambda = 0.7$), and 0.851 in 25Q2 ($\lambda = 0.5$). These results suggest that while dense vector retrieval provides strong initial relevance, MMR re-ranking helps diversify retrieved evidence and yields modest but consistent gains when multiple candidate items are considered. Across quarters, performance remains relatively stable, with only minor variations, indicating that the retrieval pipelines generalize reasonably well to newer ADR data.

Yr_Qtr TopK	BM25	Vector	λ0.3	λ0.5	λ0.7
24q4-1	0.413	**0.597**	0.485	0.486	0.486
24q4-3	0.696	**0.786**	0.765	0.767	0.765
24q4-5	0.802	0.854	0.855	0.855	**0.856**
25q1-1	0.417	**0.571**	0.485	0.480	0.485
25q1-3	0.697	**0.774**	0.759	0.759	0.762
25q1-5	0.803	0.845	0.848	0.848	**0.853**
25q2-1	0.414	**0.570**	0.478	0.478	0.483
25q2-3	0.698	**0.770**	0.758	0.759	0.759
25q2-5	0.801	0.844	0.849	**0.851**	0.850

Table 4: Top-K Recall Rates by Quarter

6 Conclusion and Future Work

This study presents a use case of hybrid Retrieval-Augmented Generation (RAG) framework that jointly supports structured and unstructured ADR prediction by integrating a customized ADR embedding model with during-inference MMR re-ranking using an optimized low-λ setting. The approach bridges representation learning and relevance optimization, where low-λ MMR favors retrieval diversity while preserving sufficient relevance. Although this slightly reduces raw similarity scores, the resulting broader evidence coverage improves detection of rare or less obvious adverse reactions, reduces redundancy of highly similar evidence. As future work, we plan to extend our framework by incorporating relationship-aware information retrieval methods, such as GraphRAG, to explicitly model relationships among patients, drugs, indications, and ADR outcomes.

7 Limitations

The quality and diversity of ADR data are critical for meaningful downstream tasks. While we utilized the largest ADR data source from the FAERS quarterly releases, this is just one of many sources of ADR data. There are numerous other ADR datasets available globally, some are in languages

other than English. Incorporating these datasets could enhance the retrieval pipeline robustness and generalizability to less common adverse-reaction patterns. However, accessing such data is challenging due to the highly regulated nature of healthcare information. Additionally, parsing and integrating real-time ADR data from social media platforms remains an open research question, which could further enrich the data pool.

While our study demonstrated promising results with fine-tuning efforts on ADR embedding and retrieval re-ranking using a subset of 500K samples from training data, ideally, the fully set of training data should be used to fine-tune embedding model. The beg-m3 embedding model used in this study is limited to 8,192 input tokens. In real-world scenarios, such as extensive medication historical records for a patient, the concatenated triplet extraction could span multiple pages. Fortunately, the latest embedding models in the MTEB benchmark allow larger input tokens up to 131,072 tokens and return embeddings with dimensions up to 4096 which can alleviate the input limit concern.

References

American Society of Pharmacovigilance. 2025. Predicting adverse drug event prevalence: A data-driven approach.

Asma Ben Abacha, Md. Faisal Mahbub Chowdhury, Aikaterini Karanasiou, Yassine Mrabet, Alberto Lavelli, and Pierre Zweigenbaum. 2015. Text mining for pharmacovigilance: Using machine learning for drug name recognition and drug–drug interaction extraction and classification. *Journal of Biomedical Informatics*, 58:122–132.

Olivier Bodenreider. 2004. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32(suppl_1):D267–D270.

Emmanuel Bresso, Pierre Monnin, Cédric Bousquet, François-Elie Calvier, Ndeye-Coumba Ndiaye, Nadine Petitpain, Malika Smaïl-Tabbone, and Adrien Coulet. 2021. Investigating ADR mechanisms with Explainable AI: a feasibility study with knowledge graph mining. *BMC Medical Informatics and Decision Making*, 21(1):171.

Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336, Melbourne Australia. ACM.

Olivia Choudhury, Yoonyoung Park, Theodoros Salonidis, Aris Gkoulalas-Divanis, Issa Sylla, and Amar k. Das. 2020. Predicting Adverse Drug Reactions on Distributed Health Data using Federated Learning. *AMIA Annual Symposium Proceedings*, 2019:313–322.

Ghasem Deimazar and Abbas Sheikhtaheri. 2023. Machine learning models to detect and predict patient safety events using electronic health records: A systematic review. *International Journal of Medical Informatics*, 180:105246.

Alireza Farnoush, Zahra Sedighi-Maman, Behnam Rasoolian, Jonathan J. Heath, and Banafsheh Fallah. 2024. Prediction of adverse drug reactions using demographic and non-clinical drug characteristics in FAERS data. *Scientific Reports*, 14:23636.

Valentina Giunchi, Michele Fusaroli, Manfred Hauben, Emanuel Raschi, and Elisabetta Poluzzi. 2023. Challenges and Opportunities in Accessing and Analysing FAERS Data: A Call Towards a Collaborative Approach. *Drug Safety*, 46(10):921–926.

David Guo and Kim-Kwang Raymond Choo. 2025. Applications of Federated Large Language Model for Adverse Drug Reactions Prediction: Scoping Review. *Journal of Medical Internet Research*, 27(1):e68291. Company: Journal of Medical Internet Research Distributor: Journal of Medical Internet Research Institution: Journal of Medical Internet Research Label: Journal of Medical Internet Research Publisher: JMIR Publications Inc., Toronto, Canada.

Hasham Ul Haq, Veysel Kocaman, and David Talby. 2023. *Mining Adverse Drug Reactions from Unstructured Mediums at Scale*, pages 361–375. Springer International Publishing, Cham.

Elad Hoffer and Nir Ailon. 2015. Deep Metric Learning Using Triplet Network. In *Similarity-Based Pattern Recognition*, pages 84–92, Cham. Springer International Publishing.

Sharath Kommu, Christopher Carter, and Philip Whitfield. 2024. Adverse Drug Reactions. In *StatPearls [Internet]*. StatPearls Publishing.

Ceca Kraišniković, Robert Harb, Markus Plass, Wael Al Zoughbi, Andreas Holzinger, and Heimo Müller. 2025. Fine-tuning language model embeddings to reveal domain knowledge: An explainable artificial intelligence perspective on medical decision making. *Engineering Applications of Artificial Intelligence*, 139:109561.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and Ali Farhadi. 2024. Matryoshka Representation Learning. *arXiv preprint*. ArXiv:2205.13147 [cs].

Martin A. Makary and Michael Daniel. 2016. Medical error—the third leading cause of death in the US.

BMJ, 353:i2139. Publisher: British Medical Journal Publishing Group Section: Analysis.

Asher Mullard. 2025. 2024 FDA approvals. *Nature Reviews Drug Discovery*, 24(2):75–82. Bandiera_abtest: a Cg_type: News Publisher: Nature Publishing Group.

Karen Ka Yan Ng, Izuki Matsuba, and Peter Chengming Zhang. RAG in health care: A novel framework for improving communication and decision-making by addressing LLM limitations. 2(1):AIra2400380. Publisher: Massachusetts Medical Society.

Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document Ranking with a Pre-trained Sequence-to-Sequence Model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718, Online. Association for Computational Linguistics.

Abeed Sarker and Graciela Gonzalez. 2015. Portable automatic text classification for adverse drug reaction detection via multi-corpus training. *Journal of Biomedical Informatics*, 53:196–207.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. FaceNet: A Unified Embedding for Face Recognition and Clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823. ArXiv:1503.03832 [cs].

Özgür Arzucan He Yongqun Hur Junguk Tiftikci, Mert. 2019. Machine learning-based identification and rule-based normalization of adverse drug reactions in drug labels. *BMC Bioinformatics*, 20.

Michael A. Veronin, Robert P. Schumaker, and Rohit Dixit. 2020. The Irony of MedWatch and the FAERS Database: An Assessment of Data Input Errors and Potential Consequences. *Journal of Pharmacy Technology*, 36(4):164–167. Publisher: SAGE Publications Inc.

8 Appendix

‘‘‘ You are an expert pharmacovigilance assistant. Given a patient profile, treatments (drugs, doses, routes), and indication ("indi_pt") as <query triple extraction>, your task is to generate possible adverse drug reactions (ADRs) with most likely outcomes in pt. The output MUST be a single JSON line containing: "pt": semicolon-separated preferred terms (MedDRA PT) representing possible ADRs as in the examples, no more than 5 "uni_code": one of the 7 standardized serious outcome codes as below:

DE – Death

LT – Life-threatening

HO – Hospitalization (initial or prolonged)

DS – Disability

CA – Congenital anomaly

RI – Required intervention to prevent permanent impairment or damage

OT – Other (anything not above)

Your reasoning must NOT appear in the output. Output only the JSON line. Here are 2 examples of input and output.

Input example 1:

{"patient":{"age": "72", "age_cod": "yr","gndr_cod": "f", "wt": "58", "wt_cod": "kg"}, "treatment": "{"drugname": "vectibix", "route": "intravenous (not otherwise specified)", "dose": "320 milligram, q2wk (biweekly)"}; {"drugname": "cisplatin","route": "intravenous","dose" :"70 mg/m2 q3wk"}", "indi_pt": "asthma; lung cancer metastatic"}

output example 1:

{"pt": "nausea; nephrotoxicity; fatigue; anaemia; electrolyte imbalance","uni_code":"HO"}

Input example 2:

{"patient":{"age": "5","age_cod": "yr","gndr_cod": "m","wt": "20","wt_cod": "kg"}, "treatment": "{"drugname": "amoxicillin", "route":"oral", "dose":"250 mg tid"}", "indi_pt": "otitis media"}

Output example 2:

{"pt": "bruising; haemorrhage; epistaxis; anaemia","uni_code":"RI"}

‘‘‘