

NOTAM-Evolve: A Knowledge-Guided Self-Evolving Optimization Framework with LLMs for NOTAM Interpretation

Maoqi Liu¹, Quan Fang^{1*}, Yuhao Wu¹, Can Zhao², Yang Yang^{3,4}, Kaiquan Cai^{3,4}

¹Beijing University of Posts and Telecommunications, Beijing 100876, China

²Aviation Data Communication Corporation, Beijing 100191, China

³Beihang University, School of Electronic and Information Engineering, Beijing 100191, China

⁴State Key Laboratory of CNS/ATM, Beijing 100191, China

qfang@bupt.edu.cn

Abstract

Accurate interpretation of Notices To Airmen (NOTAMs) is critical for aviation safety, yet their condensed and cryptic language poses significant challenges to both manual and automated processing. Existing automated systems are typically limited to “Shallow Parsing,” failing to extract the actionable intelligence needed for operational decisions. We formalize the complete interpretation task as “Deep Parsing,” a dual-reasoning challenge requiring both **dynamic knowledge grounding** (linking the NOTAM to evolving real-world aeronautical data) and **schema-based inference** (applying static domain rules to deduce operational status). To tackle this challenge, we propose **NOTAM-Evolve**, a self-evolving framework that enables a Large Language Model (LLM) to autonomously master complex NOTAM interpretation. Leveraging a knowledge graph-enhanced retrieval module for data grounding, the framework introduces a crucial closed-loop learning process where the LLM progressively improves from its own outputs, minimizing the need for extensive human-annotated reasoning traces. In conjunction with this framework, we introduce a new benchmark dataset of 10,000 expert-annotated NOTAMs. Our experiments demonstrate that NOTAM-Evolve achieves a 30.4% absolute accuracy improvement over the base LLM, establishing a new state-of-the-art on the task of structured NOTAM interpretation.

Code — <https://github.com/Estrellajer/NOTAM-Evolve>

Introduction

Notices To Airmen (NOTAMs) are official bulletins issued by aviation authorities to inform pilots and air traffic personnel of time-sensitive changes to airspace structure (Administration 2025), airport facilities, or flight procedures. Unlike conventional technical documents, NOTAMs are written in highly condensed telegraphic language, often using specialized abbreviations, structured fields, and nonstandard syntax.

With over one million active NOTAMs issued annually worldwide (Morarasu and Roman 2024), accurate interpretation of these notices is essential for ensuring flight safety

and operational efficiency. Misinterpretation can lead to missed warnings about closed runways, inoperative navigation aids, or restricted airspace—potentially resulting in costly delays or even safety incidents. However, manual NOTAM processing remains a labor-intensive and error-prone task.

Figure 1 illustrates the core challenge of this interpretation task. Current automated systems, often relying on methods like rule-based pattern matching or traditional NER, are typically limited to solving what we term the task of “Shallow Parsing.” These methods struggle to extract truly actionable intelligence, which still requires significant manual intervention. This necessitates a deeper level of interpretation, a capability we formalize as the task of “Deep Parsing.” However, successfully performing this task is hindered by two fundamental difficulties that simplistic approaches cannot address.

The first challenge is **Dynamic Knowledge Grounding**. A NOTAM is not self-contained; as Figure 1 illustrates, its interpretation requires grounding textual references in an external, dynamic knowledge base of aeronautical infrastructure. For instance, the airport code (ZBAA) serves as a key to retrieve that airport’s specific configuration—such as its list of active runways (e.g., RWY 09L)—from the knowledge base. This contextual data evolves over time and must be synchronized with the NOTAM’s effective period (Patel et al. 2023).

The second challenge involves **Schema-Based Inference**. Deriving a NOTAM’s true meaning requires schema-based reasoning beyond raw text extraction. Returning to the example in Figure 1, the statement “. . . REDUCED LENGTH OF 300M” provides a raw parameter, not an operational conclusion. An external schema, such as an ICAO rule set, is required to infer that this 300-meter system constitutes a “Basic Approach Lighting System (BALS).” This inferential step is crucial for translating raw facts into actionable intelligence and completing the deep parsing.

To address these dual challenges required for deep parsing, we propose NOTAM-Evolve, a self-evolving framework that enables LLMs to autonomously master complex NOTAM interpretation. The framework first employs a knowledge graph-enhanced retrieval module to ground interpretations in relevant aviation infrastructure data. Crucially,

*Corresponding author

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

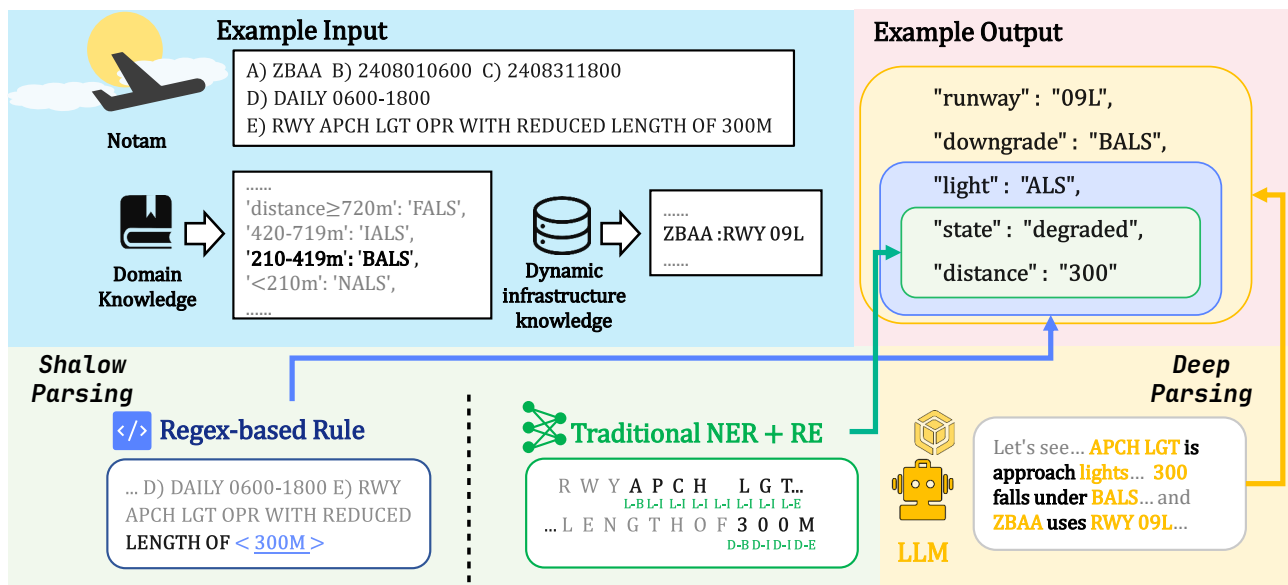


Figure 1: An illustration of a NOTAM and its information parsing process. A NOTAM is a safety alert that reports flight hazards, and its original format ('Example Input') is unstructured text. Information parsing aims to convert this into structured data ('Example Output'). Past 'Shallow Parsing' approaches used techniques such as regular expressions or traditional NER. In contrast, 'Deep Parsing', as we define it, requires a model to combine domain knowledge with dynamic data for complex reasoning to understand the text's deep semantics.

it then initiates a closed-loop learning process where the LLM's own outputs serve as training signals, enabling progressive improvement without requiring extensive human-annotated reasoning traces.

Our main contributions are threefold:

- **Problem Formalization and Dataset:** We formally define the structured NOTAM interpretation task as a deep parsing challenge requiring both dynamic knowledge grounding and schema-based inference. We also introduce a comprehensive benchmark dataset of 10,000 globally sourced NOTAMs with expert annotations.
- **Self-Evolving Framework:** We propose a novel self-evolving framework that enables LLMs to autonomously master complex NOTAM interpretation through iterative preference optimization and consensus-based inference, without requiring extensive manual supervision.
- **Empirical Performance Leap:** Experimental validation demonstrates our optimized model achieves a 30.4% accuracy improvement over base LLMs, establishing new state-of-the-art performance on this challenging aviation safety task.

Related Work

NOTAM Parsing Automating the analysis of Notices to Airmen (NOTAMs) is a key application for NLP in aviation, aimed at reducing manual operational burdens (Mogillo-Dettwiler 2024; Mi, Fan, and Sun 2022). Initial research applied various techniques, from traditional NLP workflows like TF-IDF and NER for segmentation (Clarke et al.

2021) to transformer-based models for filtering and knowledge extraction from large, unlabeled corpora (Bravin et al. 2020; Arnold et al. 2022). Despite these pioneering efforts, persistent challenges such as ambiguous abbreviations, mismatches between semantics and operational practice, and regional variations remain significant hurdles to safety and efficiency (Morarasu and Roman 2024). Our work builds on these insights, proposing a more adaptive and robust LLM-based framework specifically designed to handle these complexities. Despite these pioneering efforts, persistent challenges such as ambiguous abbreviations, mismatches between semantics and operational practice, and regional variations remain significant hurdles to safety and efficiency (Morarasu and Roman 2024). Our work builds on these insights, proposing a more adaptive and robust LLM-based framework specifically designed to handle these complexities through knowledge grounding and iterative self-optimization.

Advances in Aviation NLP and LLMs More broadly, NLP is increasingly integral to enhancing flight operations and safety management. Recent applications range from improving flight trajectory prediction by integrating spoken instructions (Guo et al. 2024), graph-based modeling (Fan et al. 2024), and language modeling approaches (Luo and Zhou 2025), to developing agents for pilot training (Liu, Zou, and Aw 2024) and automating the analysis of safety reports (Nanyonga, Wasswa, and Wild 2023). The progress in Large Language Models (LLMs) (Zhao et al. 2023) is central to these advancements. Modern transformer architectures (Brown et al. 2020; Chowdhery et al. 2022), espe-

cially when enhanced by parameter scaling (Rae et al. 2021; Le Scao et al. 2022), demonstrate strong few-shot learning capabilities well-suited to the sparsely labeled data common in aviation (Xu et al. 2023). Our work leverages several key LLM capabilities to tackle NOTAM parsing:

Knowledge-Grounded Extraction and Reasoning. Our framework’s design is informed by LLM advancements in handling external knowledge and structured data. To enhance factual reasoning, we integrate a knowledge graph (KG), a strategy proven to improve reliability and reduce hallucinations in various domains (Ji et al. 2024; Zhang et al. 2024; Chen et al. 2024a; Shi et al. 2024). For the core task, we build upon modern information extraction—which utilizes in-context learning (Li et al. 2023) and instruction tuning (Wang et al. 2023)—and tabular understanding, which has evolved from Text2SQL (Zhong, Xiong, and Socher 2017) to methods like TableRAG (Chen et al. 2024b). Our work specifically targets the limitations of current innovations (Sainz et al. 2024; Li et al. 2024) in handling the dynamic semantics and sparse schemas found in aviation.

Optimization for Complex Instructions. To manage the complex, constrained nature of NOTAMs, we employ curriculum learning. This approach is motivated by research showing that progressive learning (Mukherjee et al. 2023; Luo, Qian, and Yoon 2024) and specialized frameworks like Conifer (Sun et al. 2024) are effective at improving a model’s ability to follow complex instructions, which is central to our optimization strategy.

Dataset Construction

Dataset Overview

We present a large-scale, comprehensive dataset of Notices To Airmen (NOTAMs) designed to support research on automated structured interpretation. This dataset contains 10,000 validated samples, averaging 39.2 words per notice and an average validity duration of 8.1 days. The dataset covers a broad geographical distribution, with Asia as the top region (38.8% of samples). The most frequent Q-Code is “Movement Area (M)” at 49.8%. For evaluation, the dataset is divided into four subsets: Light (1,000 samples), Area (4,000 samples), Runway (2,500 samples), and Taxiway (2,500 samples). The dataset was constructed through collection and aggregation of global NOTAM broadcasts, capturing real-world diversity and complexity. It has been curated with strict quality control to promote reliable and generalizable model development.

Data Annotation

Given the vast number of NOTAMs issued annually, we began by randomly sampling from the global NOTAM traffic for the entire year 2024. This approach ensures a manageable yet broadly representative volume, as substantiated by the statistics in Table 1. The initial annotation schema was defined by expert aviation dispatchers, selecting foundational fields critical for operational efficiency and safety. Our methodology employs a **non-extractive, inferential annotation scheme**. Unlike traditional information extraction datasets based on sequence labeling (e.g., BIO), anno-

tators provided semantically correct values for each field regardless of whether these values were explicitly mentioned in the source text. For example, a NOTAM might imply a runway closure through technical jargon, and annotators explicitly assign the value “Closed” to the Runway Status field, as illustrated in Figure 1. This approach supports the development of models capable of genuine understanding and inference. Annotation was independently performed by two expert dispatchers. To ensure consistency and quality, we computed Inter-Annotator Agreement (IAA), achieving a Krippendorff’s Alpha of 0.96, indicating a very high level of reliability. All discrepancies were resolved by a third senior expert, thereby producing a gold-standard dataset.

Properties	Value
<i>Overall Characteristics</i>	
Total Samples	10,000
Average Word Count	39.2
Average Valid Days	8.1
Top Region	Asia (38.8%)
Top Q-Code	Movement Area (M, 49.8%)
<i>Evaluation Task Distribution</i>	
Light	1,000
Area	4,000
Runway	2,500
Taxiway	2,500

Table 1: NOTAM dataset overview, including sample statistics and evaluation task distribution.

The NOTAM-Evolve Framework

Problem Formulation

Our primary objective is to extract structured aviation information from an input NOTAM text sequence $X = [x_1, \dots, x_n]$, by leveraging a collection of aviation reference tables $\mathcal{T} = \{T_1, \dots, T_m\}$ within a knowledge-enhanced generative framework. Formally, this task is defined as maximizing the conditional probability:

$$p_{\theta}(Y | X, P, K) = \prod_{i=1}^m p_{\theta}(Y_i | X, P, K, Y_{<i}), \quad (1)$$

where $Y = [Y_1, \dots, Y_m]$ denotes the target structured output sequence. The term θ represents the parameters of the LLM, P encapsulates task-specific prompts and instructions, and $K = \kappa(X, \mathcal{T})$ corresponds to the factual knowledge retrieved from the aviation reference tables \mathcal{T} .

Framework Overview

As illustrated in Fig. 2, our NOTAM-Evolve framework, also referred to as the Self-Evolving Framework, comprises three core stages: (1) The **Knowledge-Grounded Retrieval** stage grounds predictions in aviation domain knowledge via dynamic table retrieval (TableRAG); (2) The **Self-Optimizing Model Refinement** stage enables iterative self-improvement of the foundation model through a combina-

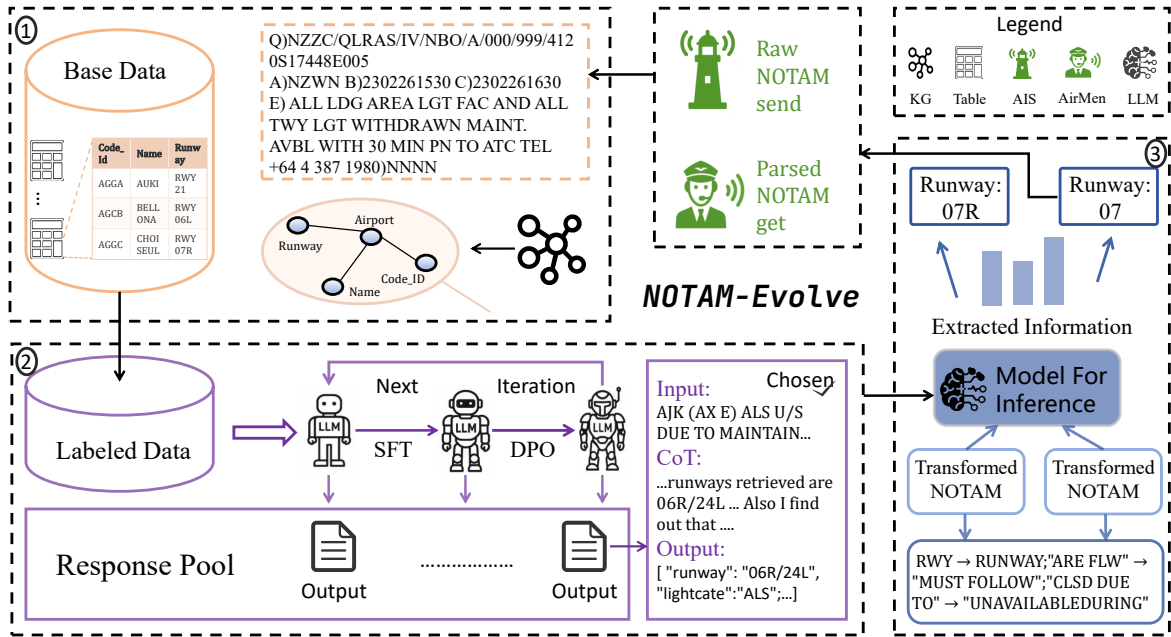


Figure 2: Overall framework of our proposed NOTAM-Evolve: (1) **Knowledge-Grounded Retrieval**: The final outputs are grounded in a set of base tables that represent real-world conditions, e.g., the number of runways at an airport. (2) **Self-Optimizing Model Refinement**: Our foundational model gains proficiency in handling complex instructions within NOTAM analysis scenarios through iterative self-evolution combining supervised and preference optimization. (3) **Multi-View Inference with Rewriting & Voting**: We rephrase the original NOTAM without altering its core content and then extract information from multiple texts to determine the final answer via a voting mechanism.

tion of supervised fine-tuning and adaptive preference learning; (3) The **Multi-View Inference** stage ensures robust parsing via diversified input generation (rewriting) and consensus decoding (voting). This architecture allows NOTAM-Evolve to address key NOTAM analysis challenges, including knowledge grounding, error propagation, and stability.

Knowledge-Grounded Retrieval

The necessity for a retrieval-augmented approach stems from the dynamic nature of aviation data; core operational information, such as airport facility status and runway availability, is stored in tables that are periodically updated. A retrieval mechanism is therefore essential to ensure the Large Language Model (LLM) is grounded in the most current and factually accurate data. However, conventional retrieval methods are often insufficient, as they perform well only when table columns and their data possess clear, self-evident meanings. In the specialized aviation domain, this is rarely the case, as complex, implicit relationships are common. For instance, a simple query about a "runway closure" might fail to retrieve crucial, related information about dependent lighting systems or navigation aids because these connections are not explicitly defined in the table structure. To overcome this limitation, we introduce a domain-specific knowledge graph to provide the structured, real-world knowledge required for the LLM to better understand these complex relationships. Our proposed KG-TableRAG framework en-

hances the foundational TableRAG approach (Chen et al. 2024b) by implementing a precise workflow: upon receiving a raw NOTAM, the LLM first generates a Cypher query to search the knowledge graph. The results from this graph query, which encapsulate relevant domain knowledge, are then concatenated with the original user query to form a new, enriched query. This new query is subsequently used to retrieve the most relevant information from the operational tables. Finally, this retrieved information is combined with the original NOTAM to create the definitive input for the language model's final processing, ensuring a factually consistent and context-aware output.

Self-Optimizing Model Refinement

This stage iteratively refines the base model using a combination of supervised learning on correct predictions (self-supervised) and preference learning on error signals.

Initialization Setup The process starts with:

- **Data Partitioning**: An annotated dataset $\mathcal{D}_0 = \{(x \circ K, Y^*)\}$ is partitioned into training ($\mathcal{D}_{\text{train}}$) and test ($\mathcal{D}_{\text{test}}$) sets (e.g., 8:2 ratio). Here x is the NOTAM text, $K = \kappa(x, \mathcal{T})$ is retrieved knowledge, and Y^* is the ground truth structured output.
- **Base Model**: An initial model π_0 , typically an untuned open-source LLM (π_{base}).
- **Response Pool**: An indexed set \mathcal{R} , initially empty, to store

input-output pairs (x, Y^*, \hat{Y}) generated across iterations e .

Iterative Optimization Loop Each iteration e (from 1 to a maximum E) involves intertwined SFT and DPO stages (See Fig. 2 and Algorithm in the supplementary material).

First, using the current model π_e , responses $\hat{Y}^{(e)}$ are generated for inputs $x \in \mathcal{D}_{\text{train}}$. These responses are compared with Y^* to label them as correct or incorrect, and the repository \mathcal{R} is updated. The error rate for an input x , $\xi(x)$, is estimated as the fraction of the last K' generated responses that were incorrect:

$$\xi(x) = \frac{\sum_{k=1}^{K'} \mathbb{I}(\hat{Y}^{(k)} \neq Y^*)}{K'} \quad (2)$$

where K' is a hyperparameter defining the lookback window.

Next, supervised fine-tuning (SFT) is performed. Correct input-output pairs $(x \circ K, Y^*)$ are extracted from \mathcal{R} to form the SFT dataset $\mathcal{D}_{\text{SFT}}^{(e)}$. The model π_e is fine-tuned by minimizing the standard negative log-likelihood loss (where m is the target sequence length):

$$\mathcal{L}_{\text{SFT}}^{(e)} = -\mathbb{E}_{(x, Y^*) \sim \mathcal{D}_{\text{SFT}}^{(e)}} \left[\sum_{i=1}^m \log \pi_{\theta}(Y_i^* | x \circ K, Y_{<i}^*) \right] \quad (3)$$

Following SFT, the dynamic preference optimization (DPO) stage begins. A preference dataset $\mathcal{D}_{\text{pref}}^{(e)}$ is constructed by sampling triples (x, y^*, y^-) from \mathcal{R} , where y^* is a known correct response and y^- is incorrect for input x . Dynamic data augmentation is applied for inputs x with a high error rate ($\xi(x) \geq \tau$, where τ is a threshold), generating N_{aug} semantic-preserving variants \mathcal{V}_x . Corresponding preference triples (v, y^*, y^-) for $v \in \mathcal{V}_x$ are added to form the full DPO dataset:

$$\mathcal{D}_{\text{DPO}}^{(e)} = \mathcal{D}_{\text{pref}}^{(e)} \cup \bigcup_{\substack{x \in \mathcal{D}_{\text{train}} \\ \xi(x) \geq \tau}} \{(v, y^*, y^-) | v \in \mathcal{V}_x\} \quad (4)$$

Weighted curriculum learning is implemented when sampling from $\mathcal{D}_{\text{DPO}}^{(e)}$. The sampling weight $w_e(x)$ for input x at iteration e adaptively focuses on harder examples using the error rate $\xi(x)$ and a curriculum schedule $\alpha_e = \min(e/E, 1)$. Let $N = |\mathcal{D}_{\text{DPO}}^{(e)}|$, β_{weight} controls error emphasis, and E is the total scheduled iterations:

$$w_e(x) = (1 - \alpha_e) \frac{1}{N} + \alpha_e \frac{\exp(\beta_{\text{weight}} \xi(x))}{\sum_{j=1}^N \exp(\beta_{\text{weight}} \xi(x_j))} \quad (5)$$

This transitions sampling from uniform towards error-weighted as iterations progress. Finally, the DPO loss is optimized using the SFT-updated model as the policy π_{θ} and the model from the start of the iteration π_e as the reference π_{ref} . Samples (x, y^*, y^-) are drawn according to $P_e(x) \propto w_e(x)$:

$$\mathcal{L}_{\text{DPO}}^{(e)} = -\mathbb{E}_{\substack{(x, y^*, y^-) \\ \sim P_e(x)}} \left[\log \sigma \left(\beta_{\text{DPO}} \log \frac{\pi_{\theta}(y^* | x)}{\pi_{\text{ref}}(y^* | x)} - \beta_{\text{DPO}} \log \frac{\pi_{\theta}(y^- | x)}{\pi_{\text{ref}}(y^- | x)} \right) \right] \quad (6)$$

Here, β_{DPO} is the DPO hyperparameter. The sigmoid function $\sigma(\cdot)$ transforms the scaled log-probability difference into a probability $[0, 1]$, representing the preference likelihood (y^* over y^-), enabling direct learning from preferences. The model after DPO becomes π_{e+1} .

The iterative loop terminates when the model π_{e+1} achieves a target accuracy η on the test set $\mathcal{D}_{\text{test}}$:

$$\frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{(x, Y^*) \in \mathcal{D}_{\text{test}}} \mathbb{I}(\pi_{e+1}(x \circ K) = Y^*) \geq \eta \quad (7)$$

where $\pi_{e+1}(x \circ K)$ is the model's prediction.

Empirical results show that the framework achieves commercial SOTA-level NOTAM parsing accuracy within 3-5 iterations without model distillation.

Multi-View Inference with Rewriting & Voting

Standard parsing paradigms struggle with NOTAM analysis due to models' limited complex instruction-following, often causing structural output errors. Particularly for edge cases where minor reasoning path variations could determine correctness, we observe that the baseline model (π_{R1}) generates inconsistent predictions despite demonstrating partial comprehension. To mitigate instability and preserve domain integrity, we use input diversification with consensus-based decoding (rewriting and voting). The approach begins with generating $N = 5$ semantically-equivalent NOTAM variants through controlled paraphrasing that strictly maintains original aviation terminology (e.g., preserving "RWY" abbreviations), spatiotemporal constraints, and safety-critical numerical values. Each variant undergoes independent model processing to yield candidate structured outputs $\{\hat{Y}^{(k)}\}_{k=1}^N$, followed by majority voting to determine the final prediction $\hat{Y}_{\text{final}} = \arg \max_Y \sum_{k=1}^N \mathbb{I}(Y = \hat{Y}^{(k)})$. The paraphrasing mechanism combines lexical substitution (e.g., "CTAM" \leftrightarrow "Controller Advisory Message"), syntactic restructuring through voice alternation, and contextual expansion with optional ICAO phraseology clarifications. Experimental validation in Section demonstrates this technique's effectiveness, achieving a 5% accuracy improvement.

Experiments

Experimental Setup

Datasets & Evaluation Protocol. For our evaluation, we leverage the specialized NOTAM dataset detailed in Section . A prediction is considered correct only if it exactly matches the ground truth in both format and all annotated field values.

Baselines. We benchmark our framework against several categories of baselines, as detailed in Table 2. These include: 1) traditional methods (a Regex-based system and the UIE information extractor (Lu et al. 2022)); 2) open-source LLMs (Qwen2.5-7B (Yang et al. 2024), Mistral-7B (Jiang et al. 2023), Llama3.1-8B-Instruct (Dubey et al. 2024), and our base model, DeepSeek-R1-Distill-Qwen-7B); 3) their SFT counterparts where applicable (Qwen2.5-7B-Instruct and DeepSeek-R1-Distill-Qwen-7B (SFT)); and 4) high-performance commercial models as reference points (GPT-4o (Achiam et al. 2023) and DeepSeek-R1 (DeepSeek-AI

Model	Light	Area	Runway	Taxiway	AVG
Popular Models					
Regex Template Rule-based Matching	0.370	0.491	0.443	0.396	0.425
UIE (Lu et al. 2022)	0.270	0.380	0.320	0.430	0.350
qwen2.5-7B (Yang et al. 2024)	0.560	<u>0.777</u>	0.412	0.748	0.624
Mistral-7B (Jiang et al. 2023)	0.405	0.655	0.588	0.492	0.535
Llama3.1-8B-instruct (Dubey et al. 2024)	0.440	0.476	0.392	0.490	0.450
Deepseek-R1-Distill-Qwen-7B	0.410	0.484	0.446	0.492	0.458
qwen2.5-7b-instruct (SFT)	0.590	0.793	0.730	0.864	0.744
Deepseek-R1-Distill-Qwen-7B (SFT)	0.18	0.226	0.236	0.204	0.212
NOTAM-Evolve (ours)	<u>0.620</u>	0.725	0.836	<u>0.868</u>	<u>0.762</u>
Commercial Models					
GPT-4o (Achiam et al. 2023)	0.605	0.851	0.770	0.914	0.785
Deepseek-R1 (DeepSeek-AI et al. 2025)	0.725	0.871	<u>0.792</u>	0.924	0.828

Table 2: Performance comparison on four NOTAM analysis tasks. Models are grouped into Popular (including traditional methods and open-source LLMs) and Commercial (references). **Underlined**: Best result within the Popular Models group or the Commercial Models group, respectively. **Bold**: Overall best result across all models.

et al. 2025)). To ensure a fair comparison, all LLMs were evaluated using identical inputs, which consist of the same domain-specific prompts and information retrieved by our Knowledge-Grounded Retrieval module.

Implementation Details. Our implementation is based on the DeepSeek-R1-Distill-Qwen-7B model. Fine-tuning (both standard SFT and our iterative optimization) was performed using the Unsloth framework with its recommended configurations. Further details on the knowledge graph and the prompts used can be found in the supplementary material. All experiments were conducted on a single NVIDIA A800-80GB-PCIe GPU.

Main Results

We evaluated our optimized model, *DeepSeek-R1-Distill-Qwen-7B (ours)*, against the baselines on four key NOTAM analysis tasks. Table 2 summarizes these results. Our model achieves a high average score (AVG), significantly outperforming traditional methods. Crucially, it obtains a 30.4% absolute improvement over its base model (DeepSeek-R1-Distill-Qwen-7B), directly validating the effectiveness of our optimization pipeline. Furthermore, our model surpasses other tested open-source LLMs (e.g., Mistral-7B, Llama3.1-8B-Instruct) and the best-performing SFT baseline (Qwen2.5-7B-Instruct).

Notably, employing SFT alone can degrade a model’s reasoning capabilities. This aligns with findings that fine-tuning on datasets lacking Chain-of-Thought (CoT) rationales can impair reasoning performance (Lobo, Agarwal, and Lakkaraju 2024). Finally, as shown in Table 2, our model, despite its significantly smaller parameter count, achieves performance comparable to that of GPT-4o and DeepSeek-R1. This result is not merely a matter of efficiency; it is a critical step toward practical deployment, as the operational, security, and cost constraints of aviation often preclude the use of closed-source, third-party APIs.

Ablation Study

We conduct systematic ablation analyses to validate our design choices by removing key components: (1) KG-TableRAG knowledge integration and (2) the Multi-View Inference mechanism.

Table 3 demonstrates that removing KG-TableRAG (-KG) causes a 2.2% performance drop (0.740 vs 0.762 AVG), particularly affecting knowledge-dependent tasks like Q-code mapping. Removing Multi-View Inference (-Multi-View) results in a larger 4.1% decline (0.721 AVG), confirming its critical role in handling prediction instability and edge cases. When both components are removed, performance drops to the lowest level (0.690 AVG), demonstrating their complementary necessity.

KG-TableRAG	Multi-View	AVG
✓	✓	0.762
✓	×	0.721
×	✓	0.740
×	×	0.690

Table 3: Ablation Study Results with KG-TableRAG and Multi-View Inference Components.

Figure 3 validates our iterative Self-Optimizing Model Refinement strategy, showing consistent improvements across all categories over three iterations. Notably, complex tasks like Taxiway accuracy improved from 64.6% to 86.8%, while Light accuracy increased from 45% to 62%. Collectively, these results confirm that each design choice is not only beneficial on its own, but that their iterative application is key to maximizing performance.

Complexity Analysis

We analyze the computational characteristics of our iterative optimization framework. The preference pair creation fol-

lows quadratic scaling modulated by accuracy progression:

$$|\mathcal{D}_{\text{pref}}^{(t)}| \approx 9K^2t^2(1 - \eta) \quad (8)$$

where η denotes global accuracy (45% \rightarrow 62% over 3 iterations), causing the error suppression term $(1 - \eta)$ to decrease from 0.55 to 0.38.

Metric	Iter.1	Iter.2	Iter.3
Theoretical pairs	2,415	5,915	11,320
Effective pairs	1,449	3,549	6,792
Time (h)	0.58	1.5	3.2
Scale factor	1.0 \times	2.6 \times	2.1 \times

Table 4: Iterative Complexity Metrics with Scaling Factors

The computational cost per iteration is governed by:

$$\mathcal{T}_{\text{DPO}}^{(t)} = E \cdot |\mathcal{D}_{\text{pref}}^{(t)}| \cdot \mathbb{E}_{w_e(x)}[1/P_e(x)] \quad (9)$$

where curriculum sampling weights $w_e(x)$ prioritize harder examples with higher error rates.

Three mechanisms suppress theoretical $O(t^2)$ scaling to observed 2.3 \times average growth: 1) Error threshold filtering removes 40% of low-difficulty samples, 2) Curriculum sampling reduces effective batch size by 38%, and 3) Accuracy saturation limits error generation through $(1 - \eta)$ decay.

The framework maintains practical tractability with convergence achieved in 3 iterations at 62% accuracy. Total wall-clock time ranges from 35 minutes to 3.2 hours on NVIDIA A800 GPUs.

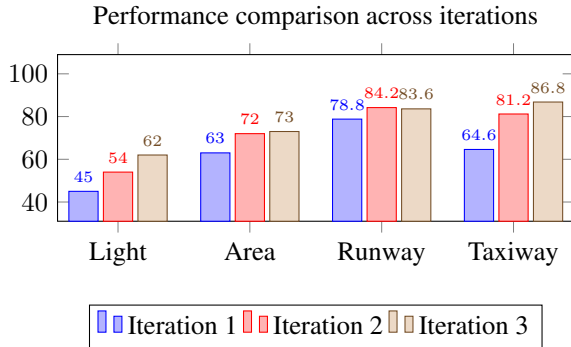


Figure 3: Iterative Optimization Performance (Accuracy %) across NOTAM Categories.

Case Study

This case study illustrates our framework’s advantage in reasoning about implicit, hierarchical relationships in NOTAMs, where high-level restrictions affect unmentioned components.

Consider a NOTAM for airport AGGC:

E) CHOISEUL L BAY AIRPORT CLOSED TO ALL OPERATIONS...

Correctly interpreting this airport-wide closure means inferring effects on associated, unlisted components like runways.

Typical baseline systems, lacking structured knowledge (e.g., airport-runway relationships) or advanced reasoning, often fail this inference. They might parse the airport closure but omit the runway, providing incomplete awareness:

```
{"airport": "AGGC", "runway": "", ...}
```

Our framework addresses this challenge. KG-TableRAG queries the aviation knowledge graph with the airport identifier ('AGGC'), retrieving that "RWY 07R" belongs to airport "AGGC". This fact supplies the missing structural context.

The LLM then integrates the input instruction ("airport closed") with this retrieved fact. Through semantic reasoning, it correctly infers the operational consequence - the runway must also be closed because it is part of the closed airport, leading to the accurate output:

```
{"airport": "AGGC", "runway": "RWY 07R", ...}
```

This correct inference of runway "RWY 07R" is critical for operational safety (e.g., preventing routing to a closed runway). It highlights our approach’s advantage: integrated knowledge and reasoning for comprehensive understanding beyond simple text extraction. For additional detailed examples, please refer to the supplementary material.

Conclusion

We present NOTAM-Evolve, a self-evolving framework that addresses the fundamental challenges of structured NOTAM interpretation through deep parsing. Unlike existing approaches limited to shallow pattern matching, our framework tackles the dual reasoning requirements of dynamic knowledge grounding and schema-based inference through three synergistic components: KG-TableRAG for knowledge-guided retrieval, iterative self-optimization through preference learning, and multi-view inference with consensus decoding.

Experimental validation on our dataset of 10,000 globally sourced NOTAMs demonstrates significant advancement over existing methods. Our framework achieves a 30.4% absolute accuracy improvement over the base LLM, establishing new state-of-the-art performance among open-source solutions while approaching commercial model capabilities. This research establishes a new paradigm for automated NOTAM analysis, with principles extensible to other high-precision domains requiring robust knowledge integration and adaptive learning.

Limitations and Future Directions. While demonstrating substantial improvements, our framework faces computational scalability challenges with progressive iteration costs similar to reinforcement learning paradigms. The inherent complexity of NOTAMs makes creating perfectly accurate ground truth annotations challenging, potentially limiting performance ceilings. Future work could explore LLM-assisted annotation combined with expert validation, more efficient optimization strategies, and extensions to multi-lingual NOTAMs and real-time operational scenarios.

Acknowledgments

This work was supported by the Beijing Natural Science Foundation (JQ24019, QY24215). This work is being supported by the Open Project Program of the State Key Laboratory of CNS/ATM (No. 2024B31). This work was supported in part by the National Natural Science Foundation of China (Grant Nos. 62576047, 52572349).

References

- Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; et al. 2023. GPT-4 Technical Report.
- Administration, F. A. 2025. What is a NOTAM? Notice to Airmen (NOTAMS). FAA website.
- Arnold, A.; Ernez, F.; Kobus, C.; and Martin, M.-C. 2022. Knowledge extraction from aeronautical messages (NOTAMs) with self-supervised language models for aircraft pilots. In *Proceedings of NAACL-HLT 2022: Industry Track Papers*, 188–196.
- Bravin, M.; Mazumder, S.; Pfäffli, D.; and Pouly, M. 2020. Automated Smartification of Notices to Airmen. In *Proceedings of the 7th Swiss Conference on Data Science (SDS)*, 51–52.
- Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; et al. 2020. Language Models are Few-Shot Learners. *arXiv:2005.14165*.
- Chen, H.; Shen, X.; Lv, Q.; Wang, J.; Ni, X.; and Ye, J. 2024a. SAC-KG: Exploiting Large Language Models as Skilled Automatic Constructors for Domain Knowledge Graph. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 4345–4360. Bangkok, Thailand: Association for Computational Linguistics.
- Chen, S.-A.; Miculicich, L.; Eisenschlos, J. M.; Wang, Z.; Wang, Z.; Chen, Y.; et al. 2024b. TableRAG: Million-Token Table Understanding with Language Models. *ArXiv*, abs/2410.04739.
- Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; et al. 2022. PaLM: Scaling Language Modeling with Pathways. *ArXiv*, abs/2204.02311.
- Clarke, S. S. B.; Maynard, P.; Almache, J. A.; Kumar, S. G.; Rajkumar, S.; Kemp, A. C.; and Pai, R. 2021. Natural Language Processing Analysis of Notices To Airmen for Air Traffic Management Optimization. In *Proceedings of the AIAA Aviation Forum 2021*, 1–26.
- DeepSeek-AI; Guo, D.; Yang, D.; Zhang, H.; Song, J.-M.; et al. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning.
- Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; et al. 2024. The Llama 3 Herd of Models. *ArXiv*, abs/2407.21783.
- Fan, Y.; Tan, Y.; Wu, L.; Ye, H.; and Lyu, Z. 2024. Global and Local Interattribute Relationships-Based Graph Convolutional Network for Flight Trajectory Prediction. *IEEE Trans. Aerosp. Electron. Syst.*, 60(3): 2642–2657.
- Guo, D.; Zhang, Z.; Yang, B.; Zhang, J.; Yang, H.; and Lin, Y. 2024. Integrating spoken instructions into flight trajectory prediction to optimize automation in air traffic control. *Nature Communications*, 15(1): 9662.
- Ji, Y.; Wu, K.; Li, J.; Chen, W.; Zhong, M.; Jia, X.; and Zhang, M. 2024. Retrieval and Reasoning on KGs: Integrate Knowledge Graphs into Large Language Models for Complex Question Answering. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Findings of the Association for Computational Linguistics: EMNLP 2024*, 7598–7610. Miami, Florida, USA: Association for Computational Linguistics.
- Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D. S.; et al. 2023. Mistral 7B. *ArXiv*, abs/2310.06825.
- Le Scao, T.; Fan, A.; Akiki, C.; Pavlick, E.; Ilic, S.; Hesslow, D.; et al. 2022. BLOOM: A 176B-Parameter Open-Access Multilingual Language Model. *CoRR*, abs/2211.05100.
- Li, P.; Sun, T.; Tang, Q.; Yan, H.; Wu, Y.; Huang, X.; and Qiu, X. 2023. CodeLME: Large Code Generation Models Are Better Few-Shot Information Extractors. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 15339–15353. Toronto, Canada: Association for Computational Linguistics.
- Li, Z.; Zeng, Y.; Zuo, Y.; Ren, W.; Liu, W.; Su, M.; Guo, Y.; Liu, Y.; et al. 2024. KnowCoder: Coding Structured Knowledge into LLMs for Universal Information Extraction. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 8758–8779. Bangkok, Thailand: Association for Computational Linguistics.
- Liu, X.; Zou, B.; and Aw, A. 2024. An NLP-Focused Pilot Training Agent for Safe and Efficient Aviation Communication. In Yang, Y.; Davani, A.; Sil, A.; and Kumar, A., eds., *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track, NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, 89–96. Association for Computational Linguistics.
- Lobo, E.; Agarwal, C.; and Lakkaraju, H. 2024. On the impact of fine-tuning on chain-of-thought reasoning. *arXiv preprint arXiv:2411.15382*.
- Lu, Y.; Liu, Q.; Dai, D.; Xiao, X.; Lin, H.; Han, X.; Sun, L.; and Wu, H. 2022. Unified Structure Generation for Universal Information Extraction. *ArXiv*, abs/2203.12277.
- Luo, K.; and Zhou, J. 2025. Large Language Models for Single-Step and Multi-Step Flight Trajectory Prediction. *CoRR*, abs/2501.17459.
- Luo, X.; Qian, X.; and Yoon, B.-J. 2024. Hierarchical Neural Operator Transformer with Learnable Frequency-aware Loss Prior for Arbitrary-scale Super-resolution. *ArXiv*, abs/2405.12202.
- Mi, B.; Fan, Y.; and Sun, Y. 2022. NOTAM Text Analysis and Classification Based on Attention Mechanism. *Journal of Physics: Conference Series*, 2171(1): 012042.
- Mogillo-Dettwiler, A. 2024. FILTERING AND SORTING OF NOTICES TO AIR MISSIONS (NOTAMS).

- Morarasu, M. M.; and Roman, C. H. 2024. AI-driven Optimization of Operational NOTAM Management. In *2024 Integrated Communications, Navigation and Surveillance Conference (ICNS)*, 1–6. IEEE.
- Mukherjee, S.; Mitra, A.; Jawahar, G.; et al. 2023. Orca: Progressive Learning from Complex Explanation Traces of GPT-4. *CoRR*, abs/2306.02707.
- Nanyonga, A.; Wasswa, H.; and Wild, G. 2023. Aviation Safety Enhancement via NLP & Deep Learning: Classifying Flight Phases in ATSB Safety Reports. In *2023 Global Conference on Information Technologies and Communications (GCITC)*, 1–5. IEEE.
- Patel, K. K.; Desaulniers, G.; Lodi, A.; and Lecue, F. 2023. Explainable prediction of Qcodes for NOTAMs using column generation. *arXiv:2208.04955*.
- Rae, J. W.; Borgeaud, S.; Cai, T.; Millican, K.; Hoffmann, J.; Song, F.; et al. 2021. Scaling Language Models: Methods, Analysis & Insights from Training Gopher. *ArXiv*, abs/2112.11446.
- Sainz, O.; García-Ferrero, I.; Agerri, R.; Lopez de Lacalle, O.; Rigau, G.; and Agirre, E. 2024. GoLLIE: Annotation Guidelines improve Zero-Shot Information-Extraction. In *The Twelfth International Conference on Learning Representations*.
- Shi, J.; Guo, Q.; Liao, Y.; Wang, Y.; Chen, S.; and Liang, S. 2024. Legal-LM: Knowledge Graph Enhanced Large Language Models for Law Consulting. In Huang, D.; Si, Z.; and Zhang, C., eds., *Advanced Intelligent Computing Technology and Applications - 20th International Conference, ICIC 2024, Tianjin, China, August 5-8, 2024, Proceedings, Part IV (LNAI)*, volume 14878 of *Lecture Notes in Computer Science*, 175–186. Springer.
- Sun, H.; Liu, L.; Li, J.; Wang, F.; Dong, B.; Lin, R.; and Huang, R. 2024. Conifer: Improving Complex Constrained Instruction-Following Ability of Large Language Models. *ArXiv*, abs/2404.02823.
- Wang, X.; Zhou, W.; Zu, C.; Xia, H.; Chen, T.; Zhang, Y.; Zheng, R.; Ye, J.; Zhang, Q.; Gui, T.; et al. 2023. InstructUIE: Multitask Instruction Tuning for Unified Information Extraction. *arXiv preprint arXiv:2304.08085*.
- Xu, F.; Wang, N.; Wen, X.; Gao, M.; Guo, C.; and Zhao, X. 2023. Few-shot Message-Enhanced Contrastive Learning for Graph Anomaly Detection. *2023 IEEE 29th International Conference on Parallel and Distributed Systems (ICPADS)*, 288–295.
- Yang, Q. A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; et al. 2024. Qwen2.5 Technical Report. *ArXiv*, abs/2412.15115.
- Zhang, Y.; Chen, Z.; Guo, L.; Xu, Y.; Zhang, W.; and Chen, H. 2024. Making large language models perform better in knowledge graph completion. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 233–242.
- Zhao, W. X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z.; Du, Y.; Yang, C.; Chen, Y.; Chen, Z.; Jiang, J.; Ren, R.; Li, Y.; Tang, X.; Liu, Z.; Liu, P.; Nie, J.; and Wen, J. 2023. A Survey of Large Language Models. *CoRR*, abs/2303.18223.
- Zhong, V.; Xiong, C.; and Socher, R. 2017. Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning. *ArXiv*, abs/1709.00103.