

# Research Talk

Teaching Multimodal Agents to Search Documents



Dr. Jordy Van Landeghem

[Code](#) + tutorial

19 December 2025

# whoami

- Research intern **ORACLE** (Seq2Seq for Dialogue Modeling) and **NUANCE** (LM Algorithms)
- Founding AI Research Engineer **contract.fit** [2017-2024]
- PhD project: *Intelligent Automation for AI-driven Document Understanding* **KU LEUVEN**
- Model Dev - GenAI - Agents Technical Lead **INSTABASE** [2024-...]

 OS research and code

 Belgium 



# Agentic Document AI Benchmark

Setting a new standard based on first principles



UNIVERSITY OF  
**OXFORD**

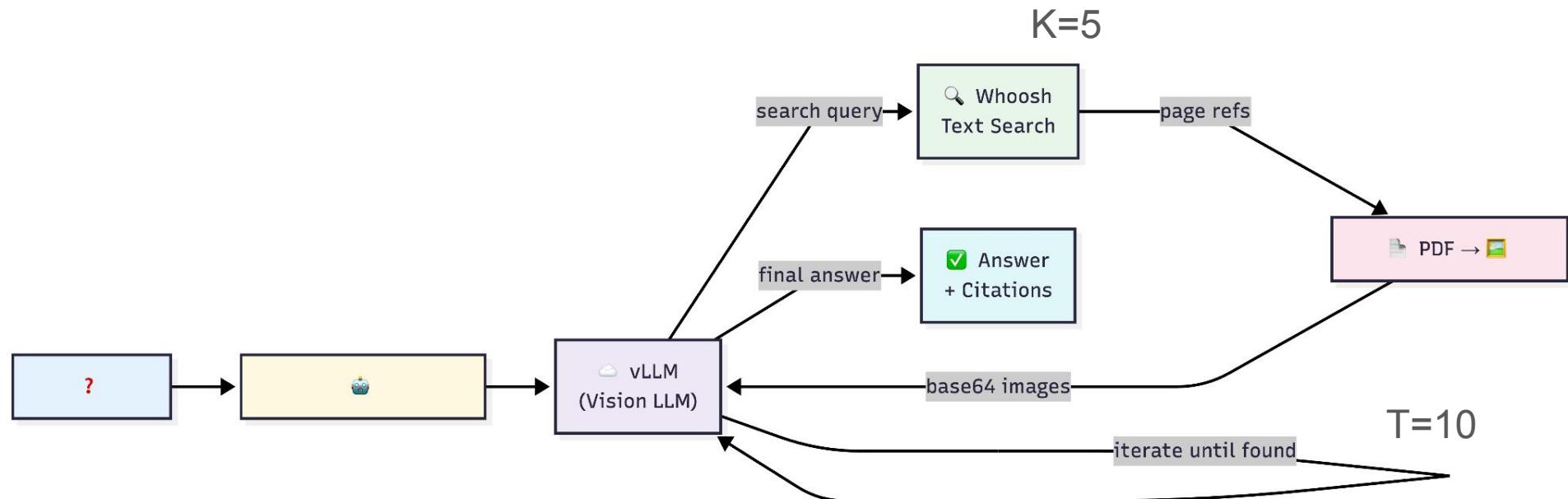
# Why another DRAG benchmark?



1. Human annotations on human-collected documents\*
2. Diversity through layouts and domains
3. Problem reframing toward agentic retrieval
4. Principled questions-split creation
5. Human baselines [*incoming*]

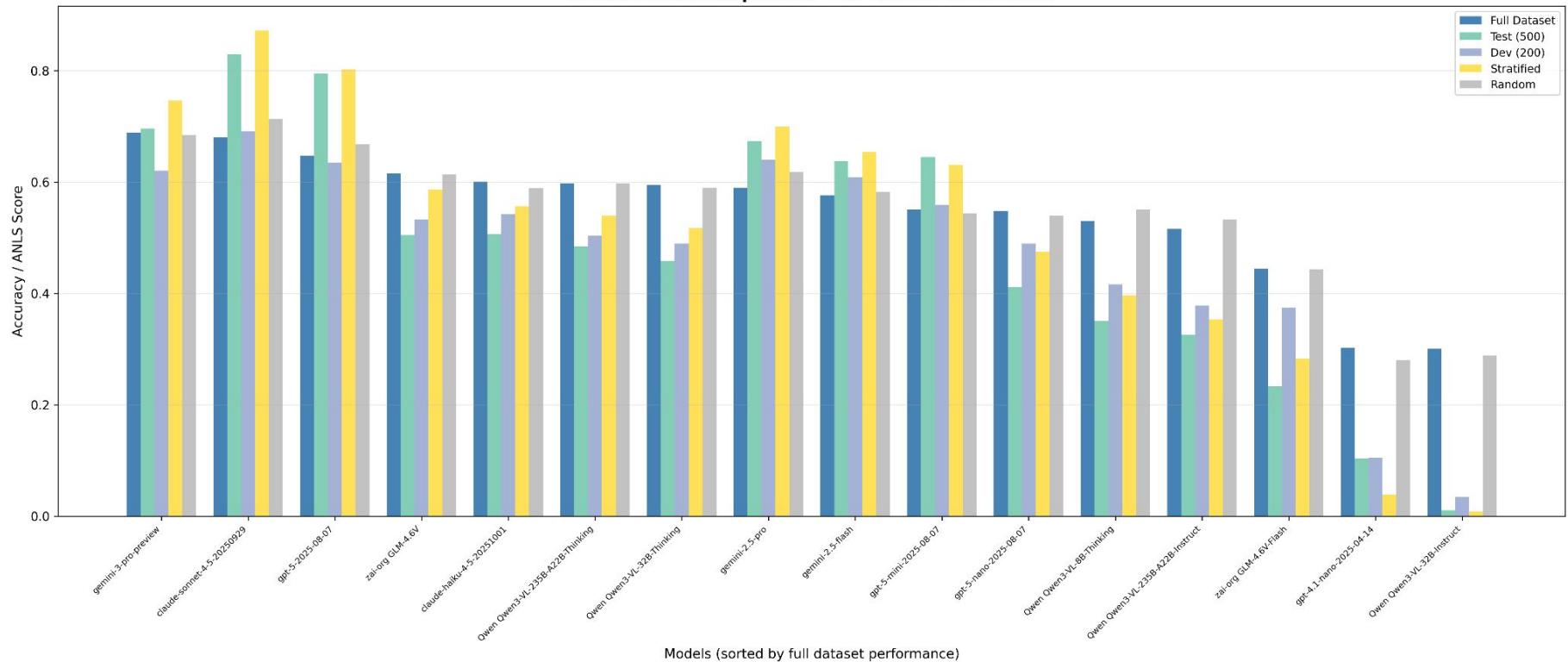
| Name and Reference                        | Input File(s)             | Diversity<br>(Domains / Layouts) | Human<br>-annotated | Problem<br>Framing   |
|---|---------------------------|----------------------------------|---------------------|--|
| DocVQA (Mathew et al., 2021b)             | Single document image     | ● medium / ● medium              | ✓                   | Document VQA<br><br>Question grounded on a single rich document      |
| InfographicVQA (Mathew et al., 2021a)     | Single image              | ● medium / ● high                | ✓                   |  |
| TAT-DQA (Zhu et al., 2022)                | Mostly single-page PDF    | ● low / ● low                    | ✓                   |  |
| DUDE (Landeghem et al., 2023)             | Multi-page PDF file       | ● high / ● high                  | ✓                   |  |
| MP-DocVQA (Tito et al., 2023)             | Multi-page PDF            | ● medium / ● medium              | ✓ / X               |  |
| SlideVQA (Tanaka et al., 2023)            | Series of slides          | ● medium / ● low                 | ✓                   |  |
| M-LongDoc (Chia et al., 2024)             | Multi-page PDF            | ● medium / ● medium              | ✓ / X               |  |
| MMLongBench-Doc (Ma et al., 2024)         | Multi-page PDF            | ● high / ● high                  | ✓                   |  |
| MuRAR (Zhu et al., 2025)                  | Collection of web pages   | ● low / ● medium                 | ✓ / X               | Multimodal RAG   |
| M <sup>2</sup> RAG (Liu et al., 2025)     | Collection of web pages   | ● high / ● high                  | ✓ / X               |  |
| ViDoRE (Faysse et al., 2025)              | Collection of PDFs        | ● high / ● high                  | ✓ / X               | Document RAG<br><br>Single-step retrieval and answer                 |
| DocBench (Zou et al., 2024)               | Collection of PDFs        | ● high / ● high                  | ✓ / X               |  |
| M3DocRAG (Cho et al., 2024)               | PDFs made from web pages  | ● medium / ● low                 | ✓ / X               |  |
| MMDocIR (Dong et al., 2025)               | Collection of PDFs        | ● high / ● high                  | ✓ / X               |  |
| FinRAGBench-V (Zhao et al., 2025)         | Collection of PDFs        | ● low / ● high                   | ✓                   |  |
| BRIGHT (Su et al., 2025)                  | Collection of web pages   | ● high / ● medium                | ✓                   | Agentic Research<br><br>Interactive, multi-step answering with tools |
| Researchy Questions (Rosset et al., 2025) | Collection of texts       | ● high / ● high                  | ✓ / X               |  |
| ViDoSeek (Wang et al., 2025)              | Collection of slide decks | ● high / ● high                  | ✓ / X               |  |
| <b>Our benchmark</b>                      | Collection of PDFs        | ● high / ● high                  | ✓                   |  |

# Baseline: Agentic document retrieval with sparse search

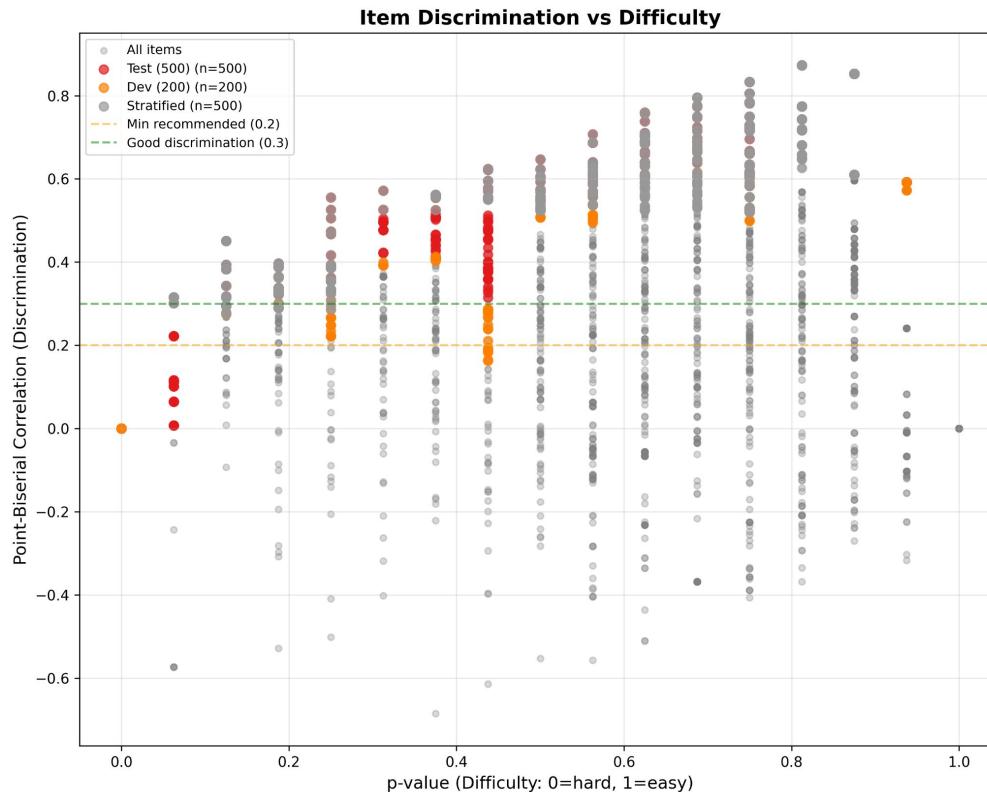


# Setting baselines and Informing split creation

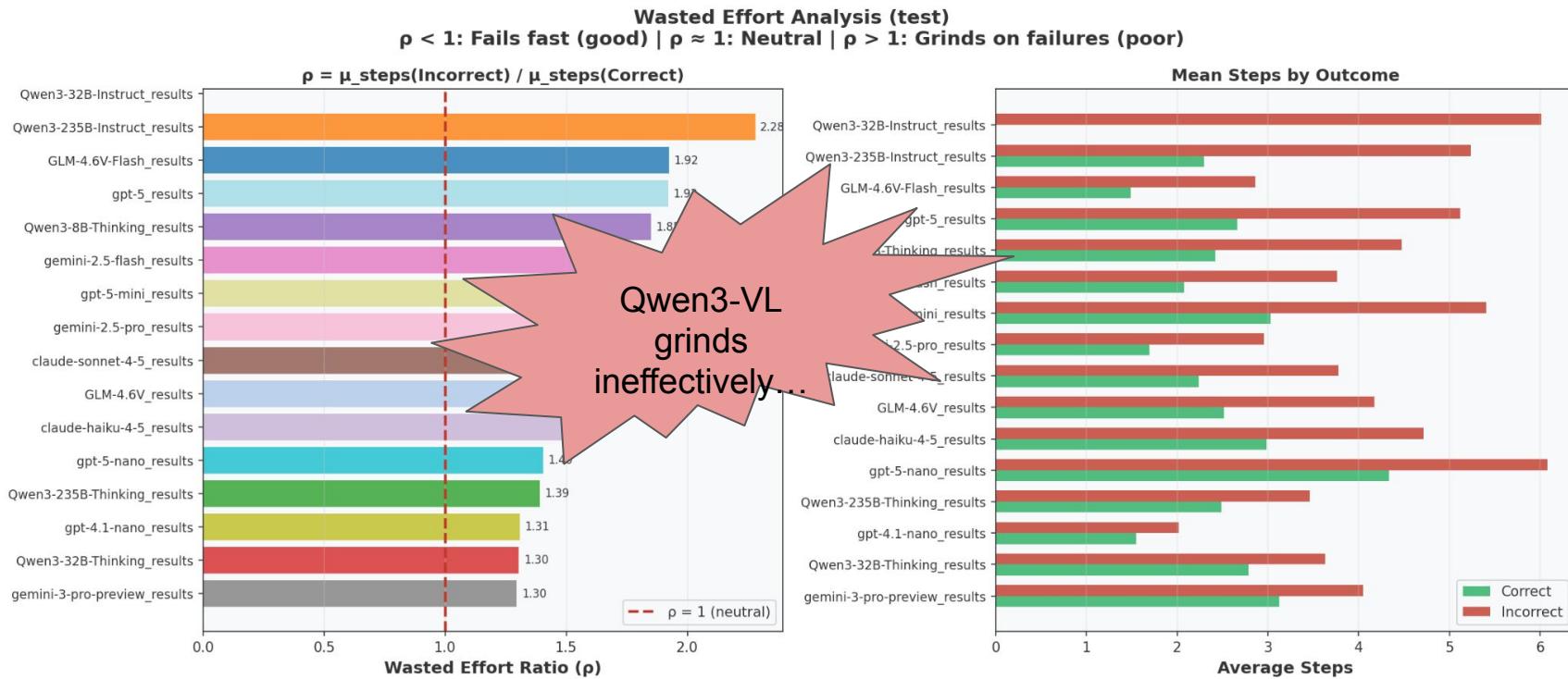
Leaderboard Comparison: Full Dataset vs Subsets



# Classical test theory (Novick 1966): uniform p-value per split



# How to quantify effective effort-accuracy?



# Teaching Multimodal Agents to Search Documents

About time to reacquaint with MLLM (post-)training



[code](#)

# Fine-Tuning an MLLM for better BM25 searches

## Problem:

- OS Thinking Multi-Modal Large LM does poor sparse searches, underperforming in agentic retrieval

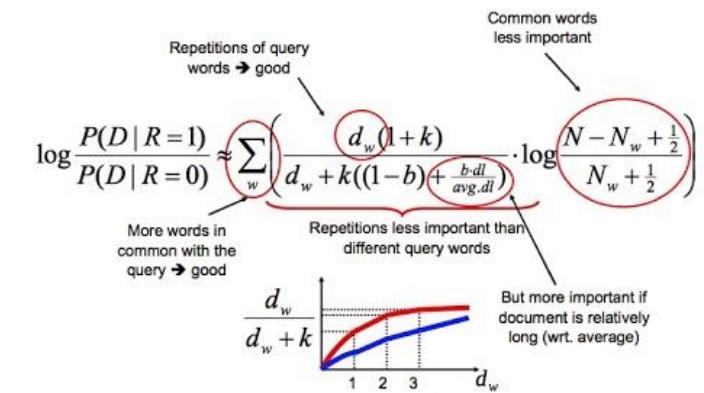
## Goal:

- Demonstrate better efficiency / reduced effort at >= accuracy

## Approach & scope

- A lightweight, *hackathon*-scale method to improve agentic tool use for sparse search in document collections

## BM25: an intuitive view



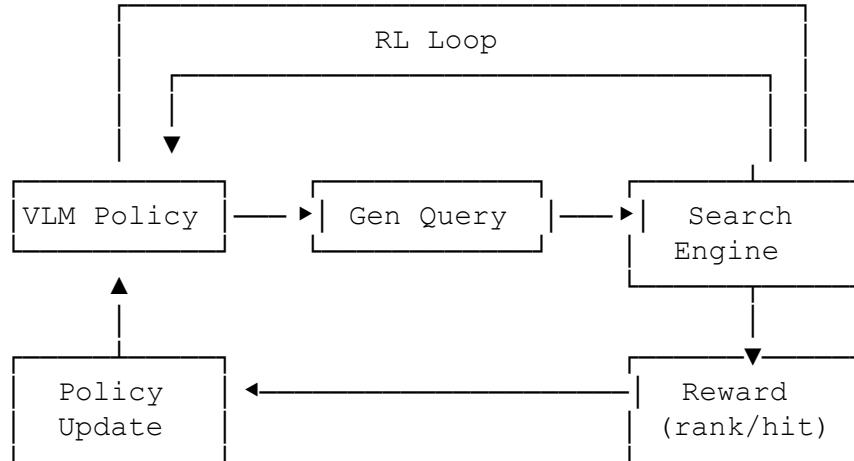
[STaR-SQL: Self-Taught Reasoner for Text-to-SQL](#)

# Approaches to Train MLLM for Better Sparse Search

| Aspect            | Few-shot Prompting<br>(Contrastive Examples)   | Offline SFT (Trajectory Sampling +<br>Loss Masking + Rank Weighting)       | Online RL (Inference Simulation)                   |
|-------------------|--|--|--|
| Impl. Effort      | ● Minimal — prompt engineering only            | 🟡 Moderate — data pipeline + training infra [AWS EC2, VLLM, accelerate HF] | 🔴 High — env setup, reward shaping, policy updates |
| Compute Cost      | ● Zero training cost                           | 🟡 Single training run  | 🔴 Expensive — repeated inference + training loop   |
| Data Requirements | Few curated examples                           | Offline trajectories (~hundreds to thousands)                              | Online generation (wasteful if wrong samples)      |
| Signal Quality    | Limited by context window; no gradient signal  | Rank-weighted supervision on successful traces                             | Dense reward from actual search outcomes           |
| Generalization    | Fragile to prompt drift; no permanent learning | Learns from static distribution  | Adapts to distribution shift                       |

# Why not RL(VR), DPO, GRPO?

## 🔴 ONLINE RL (Full Simulation)



Continuous learning | Expensive | Adapts to failures | Reward hacking risk

# Full Sparse Search Fine-tuning Recipe



## Goal

Train a Vision Language Model (Qwen-VL-3 Thinking) to write better search keywords from questions, reducing retrieval iterations.

## Training Signal Design



**Normalized Rank Score:**  
 $(\text{top\_k} - \text{rank} + 1) / \text{top\_k}$ .  
Rank 1 → 1.0 (best), Rank k → 1/k



**Label Masking:**  
Focus loss on full  
<search>...</search>  
sequences (including tags)



**Sample Weighting:**  
Weight by retrieval quality  
(rank-score, rank-sq, inv-effort)

## Pipeline Overview



### 1. `trajectory_collector.py`

Collect trajectories with reasoning tokens, search queries & results.



### 2. `process_trajectories.py`

Filter valid trajectories (GT found in top-k), compute normalized rank scores.



### 3. `format_training_data.py`

Create SFT/DPO/Reward datasets from processed trajectories.



### 4. `finetune.py`

LoRA fine-tuning with label masking & sample weighting.



### 5. `evaluate.py`

Measure iterations to success, MRR, Hit@k.

# Design space (*scripts*)

```
# 1. Collect trajectories
python trajectory_collector.py -o trajectories.jsonl
    --sampling-config default

# 2. Process and filter
python process_trajectories.py -i trajectories.jsonl
    -o processed.jsonl

# 3. Format training data
python format_training_data.py -i processed.jsonl
    -o training_data/

# 4. Distributed Training (8x A100 80GB)
torchrun --nproc_per_node=8 finetune.py sft \
    --train-data training_data/sft_best_train.jsonl \
    --model Qwen/Qwen3-VL-8B-Thinking \
    --mask-strategy search-only \
    --weight-scheme rank-score

# 5. Evaluate & compare
python evaluate.py --model merged_model/
    --compare baseline finetuned
```

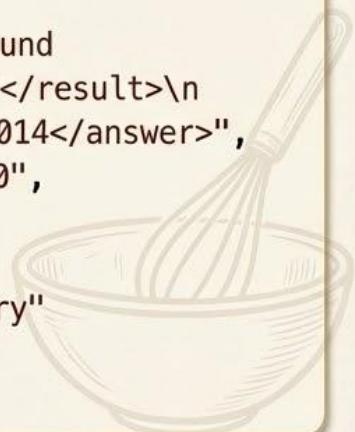
| Config    | Temperature | Top P | Repetition Penalty |
|-----------|-------------|-------|--------------------|
| greedy    | 0.0         | 1.0   | None               |
| low_temp  | 0.3         | 0.9   | None               |
| default   | 0.7         | 0.95  | None               |
| diverse   | 0.9         | 0.98  | 1.1                |
| high_temp | 1.0         | 0.95  | None               |
| creative  | 1.2         | 0.95  | None               |

# Recipe



## Raw Ingredients (JSON Data)

```
{  
    "input": "Question: When was the last kitchen inspection in the Refugio County Jail according to the Jail Inspection?",  
    "output": "<search>Refugio County Jail kitchen inspection report</search>\n<result>Found relevant document at rank 1</result>\n<answer>January of 2014</answer>",  
    "trajectory_id": "traj_30",  
    "best_rank_score": 1.0,  
    "total_searches": 1,  
    "format": "full_trajectory"  
}
```



## Preparation (Model Training)

### USER INPUT

Question: When was the last kitchen inspection...?

### ASSISTANT OUTPUT

<search> Refugio County Jail kitchen inspection report  
<result> Found relevant document at rank 1  
<answer> January of 2014

Model learns to generate this

### The Secret Sauce: Weighted Loss

Weighting for full sequence cross-entropy (attenuates or increases loss based on rank).

Rank Score  
1.0

Learning Signal  
↑

# Qualitative results (*before*)

=====

Question: Search for documents about apache helicopter specifications

=====

Output: <think>

Okay, the user is asking for documents about Apache helicopter specifications. Hmm, this seems straightforward but I need to think carefully. They're probably a researcher, military enthusiast, or maybe someone in aviation industry. Could also be a student working on a project.

First thought: They didn't specify which Apache variant (UH-60 vs AH-64). That's crucial because specs differ wildly. The AH-64 Apache attack helicopter has very different specs than the UH-60 Black Hawk transport. Should clarify that immediately to avoid misinformation.

(...)

\*preparing response\*

Will start with variant clarification, then pivot to search strategies. The "search terms" part is most actionable - they'll probably copy-paste those. Should include both exact phrases ("AH-64 Block III") and broad terms ("Apache specifications").

# Qualitative results (*after*) - SFT

---

=====  
Question: Search for documents about apache helicopter specifications  
=====

--- Parsed Response ---

[Think]

Okay, the user is asking for a search on Apache helicopter specifications. (...)

Apache specs usually cover things like speed, range, armament, and capabilities. I recall the AH-64E is the latest variant, but I should confirm if they want all models or just current ones. Since they didn't specify, I'll cover the main variants. (...)

Final thought: Keep it precise but offer flexibility. They might actually want maintenance manuals or flight manuals - but since they said "specifications", I'll focus on performance and capabilities first.

[Search Query]

**Apache helicopter specifications AH-64E technical data armament speed range sensor systems engine details**

```
(vllm-env)
ubuntu@ip-172-31-42-247:/data/agentic-retrieval-finetuning$ python inference_test.py --model Qwen/Qwen3-VL-8B-Thinking --adapter checkpoints/sft/final --max-tokens 512 --temperature 0.7 -i "Search for documents about apache helicopter specifications"
```

# Quantitative Results

| Metric                    | Base  | Trained      | Δ        |
|---------------------------|-------|--------------|----------|
| avg_iterations            | 4.14  | <b>3.98</b>  | -0.16 ↓  |
| avg_iterations_to_success | 1.97  | <b>1.85</b>  | -0.12 ↓  |
| avg_first_hit_rank        | 1.24  | <b>1.15</b>  | -0.09 ↓  |
| best_search_mrr           | 0.922 | <b>0.949</b> | +0.027 ↑ |
| anls_accuracy             | 0.58  | <b>0.60</b>  | +0.02 ↑  |

Retrieval + search efficiency

Answer Quality

# Key Takeaways

Offline trajectory SFT is a great stepping stone before considering online RL  
→ forces to find a learnable setup, well-motivated reward signal

Loss weighting based on rank exploits additional signal from trajectories

Loss masking on only <search>...</search> proves effective, adding <think>  
parts might outweigh total loss

Recommended focus: consequent tool parameter inputs

Next: teaching it to be better at [semtools](#)? 😊

# Thank You !



<https://github.com/jvlinsta/DRAG/tree/main>