@demotoprod · RAG · also LEVERS

Production RAG checklist

The default RAG pipeline sends the top few chunks by vector similarity to the model, which answers from whatever it received. A January 2024 experience report lists seven failure points; two happen before the model reads a word. Four checks for the retriever, in order.

1 Measure retrieval recall on your own questions before you blame the model

The report names two failures before the model reads anything: the answer ranked too low to be returned, or was retrieved but cut before the context. A public benchmark of 18 retrieval datasets scores the retriever alone; run the same test on your own questions.

- Label the answering chunk for real questions from your logs or tickets.
- Run retrieval with no model call; record how often that chunk is in the list you send.
- Rerun after every change of embedding model, chunker, index or reranker.

2 Run keyword and vector search together and merge the two ranked lists

On simple entity-rich questions, a dense retriever trained on a common question set averaged 49.7% top-20 accuracy against 72.0% for BM25. The authors write that dense retrievers generalise only to common entities unless the question pattern was seen in training. Reciprocal rank fusion merges ranked lists by rank alone, summing 1 / (k + rank) with k = 60.

- Index the same chunks for BM25 and for vector search, run both, and merge the lists by rank.
- Add questions that hinge on an exact name, code or acronym to the check 1 set and compare recall against vector search alone.

```
# Plain Python, no library. Reciprocal rank fusion, k = 60.
def rrf(ranked_lists, k=60):
    scores = {}
    for ranking in ranked_lists:       # chunk ids, best first
        for rank, cid in enumerate(ranking, start=1):
            scores[cid] = scores.get(cid, 0) + 1 / (k + rank)
    return sorted(scores, key=scores.get, reverse=True)
```

3 Retrieve wide, rerank with a cross-encoder, and send only the top few

In the 18-dataset benchmark, a cross-encoder re-ranking the top 100 BM25 hits beat BM25 alone on 16 of 18, with latency above 350 ms in the paper's measurements. A longer list does not help much: in a separate study, 50 documents instead of 20 only marginally improved answers, and accuracy often fell when the answer sat in the middle of the context.

- Retrieve more candidates than you send (50 below), score each question and chunk pair with a cross-encoder, keep the top few (5) with the best chunk first.
- Set both sizes from your check 1 recall and log reranking time per query.

```
# Pseudocode: cross_encoder_score is your reranker. 50 and 5 are example sizes.
scored = [(cross_encoder_score(question, c.text), c) for c in merged[:50]]
top = sorted(scored, key=lambda t: t[0], reverse=True)[:5]
```

4 Drop chunks below a score cut and pass the sources through

Weak chunks are not harmless. In one study, a 13-billion-parameter open model given the top web-search result scored 62.1 on a reasoning benchmark against 65.6 with no retrieval; filtering passages with an entailment model prevented the drop, at the cost of also discarding relevant passages.

- On the check 1 set, set the cut where most answer chunks stay and most wrong ones go.
- When no chunk clears the cut, reply that no source was found, log it and skip the model.
- Store each kept chunk's source id and score with the answer.

Nitish Gautam · indica-tech.com · hello@indica-tech.com

INDICA TECH · @DEMOTOPROD

Nitish Gautam

Founder-led AI engineering. Demo to production, under NDA, in regulated industries, on fixed scope.

Paste into your project

RETRIEVER SETTINGS TO FILL IN, BY CHECK

```
eval_set: "<real questions with the answer chunk labelled>"  # check 1
recall_target: "<your target>"   # set from your own check 1 runs
search: [bm25, vector]           # check 2
fusion: {method: rrf, k: 60}     # check 2, the fusion paper's constant
candidates: 50                   # check 3, example size
rerank: cross-encoder            # check 3
send_top: 5                      # check 3, example size
min_score: "<cut set on the check 1 set>"   # check 4
on_no_chunk: reply_no_source     # check 4, skip the model call
store_with_answer: [source_id, score]       # check 4
```

Where this list comes from

Ten-plus years taking products from zero to one across seven sectors, including defence, fintech, medical and telecom, with AI shipped into regulated and air-gapped environments. The @demotoprod channel takes one production AI failure apart each day. These checklists are the same work written down: every item is a dated failure or a documented default, with the control that stops it and the primary source.

If this list found a gap

1. ### Your demo works and production is stuck.

   Production Readiness Audit. Fixed price, two weeks. A five-dimension scorecard, cost and failure analysis, and a 90-day execution roadmap.

2. ### You need senior AI judgement without a full-time hire.

   Fractional AI Lead. Monthly: 90-day minimum, then 30 days' notice either side. Weekly 1:1 with the founder or CTO, architecture and code review, technical hiring support, investor-ready due diligence.

3. ### An AI feature has to work, by a date.

   AI Build Sprint. Fixed price, four to eight weeks. Scope signed off up front, weekly demos, CI/CD and observability included, 30-day production warranty.

4. ### A regulator, an auditor or a customer's security team is asking.

   AI Governance & Compliance. Fixed price, four weeks. ISO 27001 / SOC 2-aligned controls, a model risk framework, human-in-the-loop and audit logging, regulator-ready documentation.

BOOK A CALL
indica-tech.com/#contact

EMAIL
hello@indica-tech.com

WEBSITE
indica-tech.com

ALL CHECKLISTS
indica-tech.com/checklists/

YOUTUBE
youtube.com/@demotoprod

LINKEDIN
linkedin.com/in/startup-founderx

INSTAGRAM
instagram.com/demotoprod

TIKTOK
tiktok.com/@demotoprod

Free to share. The live list, with every source link: indica-tech.com/checklists/#production-rag