

Multi-Turn Retrieval-Augmented Generation (MTRAG)

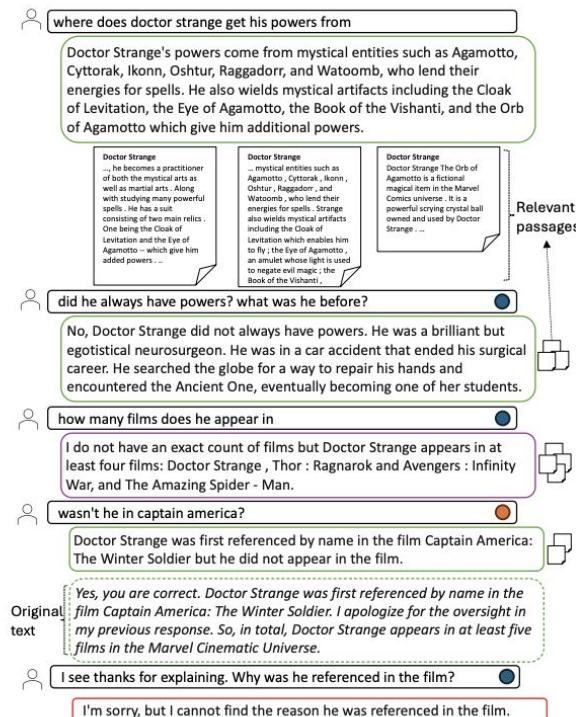
Sifei Meng

Why Multi-Turn Retrieval-Augmented Generation (RAG)?

- LLMs often hallucinate or forget factual details
- Real-world queries are not standalone

Key challenges of multi-turn RAG:

- Context drift in long dialogues
- Efficient retrieval over dynamic queries
- Aligning retriever–generator representations
- Evaluation of multi-turn factual consistency



Multi-tun RAG challenge

What is Multi-Turn RAG?

- Iterative retrieval conditioned on dialogue history
- Each turn = new query + retrieved evidence
- Memory component keeps relevant past turns
- Output grounded in both retrieved docs + dialogue context

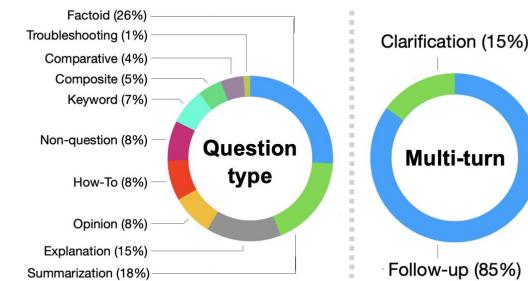
Comparison of existing benchmarks

Benchmark	Data origin	Advantages	Limitations
MTRAG	Human-generated multi-turn conversations + 4 domain corpora	Diagnosing <i>real</i> multi-turn retrieval failure modes	Number of queries is comparably limited
CORAL	AI-constructed conversations from Wikipedia	Large-scale, open-domain coverage; evaluation of citation-style RAG behavior	Synthetic dialog realism can be weaker; harder to argue “human conversational drift” is faithfully captured
RAD-Bench	Curated multi-turn dialogues with retrieved contexts provided	Testing whether the model <i>leverages</i> retrieval correctly once context is retrieved	Not a pure retriever-quality benchmark; less suitable for embedding-model comparison as the bottleneck
BEIR	Multi-domain single-turn retrieval datasets	Strong retrieval baseline sanity-check across domains	Not multi-turn; cannot measure context dependency / late-turn drift

MTRAG Statistics

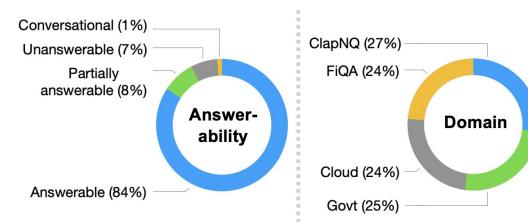
Scale

- 110 human-generated multi-turn conversations
- 7.7 average turns per conversation
- 842 evaluation tasks (turn-level tasks derived from conversations)



4 Domains (Corpora)

- ClapNQ (Wikipedia QA/IR corpus)
- Cloud (technical documentation; newly assembled)
- FiQA (finance QA/IR corpus)
- Govt (government corpus; newly assembled)



Synthetic dataset

Goal

- Generate high-quality multi-turn, document-grounded dialogs that mimic real RAG deployments.

Models used

- LLM for generation + judge: Mixtral-8x7B-Instruct (v0.1)
- Retrieval:
 - Documents chunked into 512 tokens with 100-token overlap
 - Retriever uses ELSER + all-MiniLM-L6-v2 sentence embeddings

Benchmarks	Query Taxonomy	#Dialogs	#Turns/Dialog	Query Length	Response Length	Doc. Length
CoQA	aggregate	9,967	6.2	14.7	41.6	268
	comparative	8,894	5.5	12.6	35.1	270
	direct	9,217	5.4	13.0	35.1	271
MD2D	aggregate	8,572	5.0	21.6	54.5	635
	comparative	8,211	5.1	15.5	55.4	720
	direct	9,715	5.1	18.3	50.0	671
	unanswerable	2,500	2	16.4	7.4	648
QuAC	aggregate	9,271	6.0	16.1	47.1	461
	comparative	9,554	6.4	12.6	44.8	474
	direct	9,896	6.4	13.8	42.3	465
OR-QuAC	aggregate	9,006	4.8	16.9	49.9	1,287
	comparative	9,993	5.6	12.9	50.4	1,314
	direct	5,000	5.6	14.4	48.0	1,134
(OR-)QuAC	unanswerable	3,000	2	15.1	6.2	478

Related Work: Practical Tricks for Multi-Turn RAG

- Elgohary et al.: Query Rewrite — rewrite follow-ups into standalone questions (fix coreference / ellipsis).
- Yu et al.: Context-Aware Retriever (ConvDR) — encode conversation history directly in the dense retriever (less need for rewrite).
- Mo et al.: History Denoising (HAConvDR) — keep only useful past turns + mine extra supervision from history.
- Katsis et al.: History Selection / Compression — select or compress relevant turns to reduce context noise and late-turn drift.

Related Work: Retrieval quality boosters

- Nogueira & Cho: Two-Stage Retrieval — retrieve fast, then cross-encoder rerank for better top-k precision.
- Gao et al.: HyDE — generate a hypothetical document with an LLM, embed it, retrieve similar passages.
- Yu, Xiong & Callan: ANCE-PRF — pseudo-relevance feedback: refine query embedding using top retrieved passages.
- Rackauckas: RAG-Fusion — multiple query variants + Reciprocal Rank Fusion to improve recall.
- Bruch, Gai & Ingber: Hybrid Retrieval — fuse sparse (lexical) + dense (semantic) scores for robustness.
- Elastic: ELSER — learned sparse retriever (strong sparse side for hybrid search).
- Qu, Tu & Bao: Semantic Chunking — split docs by meaning (not fixed length) to reduce passage mismatch.

Retrieval model selection based on MTEB

Family / Approach	Model	Size (params)	Main advantage
BGE-style bi-encoder baselines	BGE-v1.5 (base)	109M	Standard dense retriever baseline used in prior work
	BGE-v1.5 (small)	33M	Same training recipe, smaller size → capacity comparison
	MongoDB LEAF (mdbr-leaf-mt)	23M	Very strong tiny model; efficiency-focused baseline
Qwen-derived embedding models	Qwen3-Embedding-0.6B	0.6B	Modern embedding LLM with strong retrieval quality
	Yuan-embedding-2.0	0.6B	Qwen-based model further optimized for retrieval
	GTE-Qwen-1.5B	1.5B	Larger Qwen embedding model; tests scaling effects
	Jasper-Token-Compression-600M	0.6B	Token compression for faster multi-turn retrieval
Open embedding LLM alternative	F2LLM-1.7B	1.7B	Different open training recipe for embeddings

Metrics

- Recall@k — Did we retrieve the relevant documents?
- nDCG@k — How well are relevant documents ranked?

We'll orient on nDCG@5

Notation:

- \mathcal{R} : The set of all relevant documents for a given query;
 \mathcal{T}_k : The set of top- k retrieved documents;
 r_i : Relevance score of the i -th retrieved document (can be discrete or continuous);
 DCG_k : Discounted Cumulative Gain at position k ;
 $IDCG_k$: Ideal Discounted Cumulative Gain at position k (DCG when relevance scores are sorted in descending order).

$$\text{Recall}_k = \frac{|\mathcal{R} \cap \mathcal{T}_k|}{|\mathcal{R}|}$$

$$DCG_k = \sum_{i=1}^k \frac{2^{r_i} - 1}{\log_2(i + 1)}$$

$$DCG_k = \sum_{i=1}^k \frac{r_i}{\log_2(i + 1)}$$

$$nDCG_k = \frac{DCG_k}{IDCG_k}$$

$$IDCG_k = \sum_{i=1}^{\min(k, |\mathcal{R}|)} \frac{2^{r_i^*} - 1}{\log_2(i + 1)}$$

where r_i^* denotes the i -th highest relevance score among all relevant documents.

Experiment setup

Hardware

- GPU: NVIDIA V100
- Single-GPU inference for all embedding models

Document Processing

- Chunk size: 512 tokens
- Overlap: 100 tokens

Retrieval Scenarios

- Questions: concatenate all user turns as the retrieval query
- Last Turn: use only the final user query (no history)
- Rewrite: standalone query rewritten from conversation context (provided by benchmark authors)

Evaluation

- Retrieval evaluated at Recall@k and nDCG@k, $k = 1, 3, 5, 10$

Result comparison across tested retrieval models

Model	Task	Recall@1/3/5/10	nDCG@1/3/5/10
bge15small	questions	0.0670 / 0.1439 / 0.1875 / 0.2636	0.1596 / 0.1503 / 0.1679 / 0.2003
bge15	questions	0.0544 / 0.1322 / 0.1819 / 0.2443	0.1287 / 0.1325 / 0.1546 / 0.1809
Qwen3-0.6B	questions	0.0873 / 0.1877 / 0.2593 / 0.3473	0.2021 / 0.1945 / 0.2256 / 0.2627
Jasper-600M	questions	0.0768 / 0.1790 / 0.2380 / 0.3318	0.1840 / 0.1831 / 0.2079 / 0.2473
Yuan-2.0	questions	0.0672 / 0.1621 / 0.2201 / 0.2947	0.1570 / 0.1597 / 0.1863 / 0.2178
gte-1.5B	questions	0.0746 / 0.1589 / 0.2070 / 0.2832	0.1763 / 0.1653 / 0.1855 / 0.2172
bge15small	lastturn	0.1176 / 0.2289 / 0.2865 / 0.3746	0.2638 / 0.2445 / 0.2649 / 0.3026
bge15	lastturn	0.0995 / 0.2121 / 0.2759 / 0.3685	0.2368 / 0.2240 / 0.2482 / 0.2875
Qwen3-0.6B	lastturn	0.1547 / 0.2915 / 0.3628 / 0.4702	0.3526 / 0.3139 / 0.3403 / 0.3864
Jasper-600M	lastturn	0.1454 / 0.2956 / 0.3773 / 0.4682	0.3333 / 0.3151 / 0.3457 / 0.3848
Yuan-2.0	lastturn	0.1219 / 0.2492 / 0.3216 / 0.4074	0.2703 / 0.2614 / 0.2887 / 0.3254
gte-1.5B	lastturn	0.1250 / 0.2560 / 0.3239 / 0.4143	0.2883 / 0.2705 / 0.2964 / 0.3346
bge15small	rewrite	0.1402 / 0.2657 / 0.3454 / 0.4417	0.3179 / 0.2880 / 0.3183 / 0.3592
bge15	rewrite	0.1296 / 0.2610 / 0.3404 / 0.4468	0.3037 / 0.2773 / 0.3082 / 0.3540
Qwen3-0.6B	rewrite	0.1657 / 0.3216 / 0.4003 / 0.5143	0.3784 / 0.3453 / 0.3742 / 0.4234
Jasper-600M	rewrite	0.1601 / 0.3285 / 0.4171 / 0.5289	0.3784 / 0.3493 / 0.3828 / 0.4306
Yuan-2.0	rewrite	0.1346 / 0.2867 / 0.3613 / 0.4699	0.3037 / 0.2969 / 0.3260 / 0.3722
gte-1.5B	rewrite	0.1420 / 0.2943 / 0.3643 / 0.4781	0.3205 / 0.3091 / 0.3361 / 0.3840
FLLM	rewrite	0.1046 / 0.2271 / 0.2879 / 0.3742	0.2471 / 0.2359 / 0.2593 / 0.2965

Table 1: Retrieval performance

Why Jasper-Token-Compression-600M is Fast and Sufficient

“Good” Retrieval Quality — Knowledge Distillation

- Teachers:
 - Qwen3-Embedding-8B — strong at information retrieval
 - QZhou-Embedding-7B — strong at semantic similarity
- Teacher outputs are aligned into a shared 2048-dim embedding space
- Jasper-600M adds a linear projection head to match teacher representations

“Fast” Retrieval — Dynamic Token Compression

- Key idea: reduce tokens before expensive self-attention
- 1D convolution + AdaptiveAvgPool1d compress token sequences
 - e.g., 200 → ~120 tokens while preserving key information

Why Quality Remains Sufficient

- Retrieval tasks rely on global semantic signals, not full token-level detail.

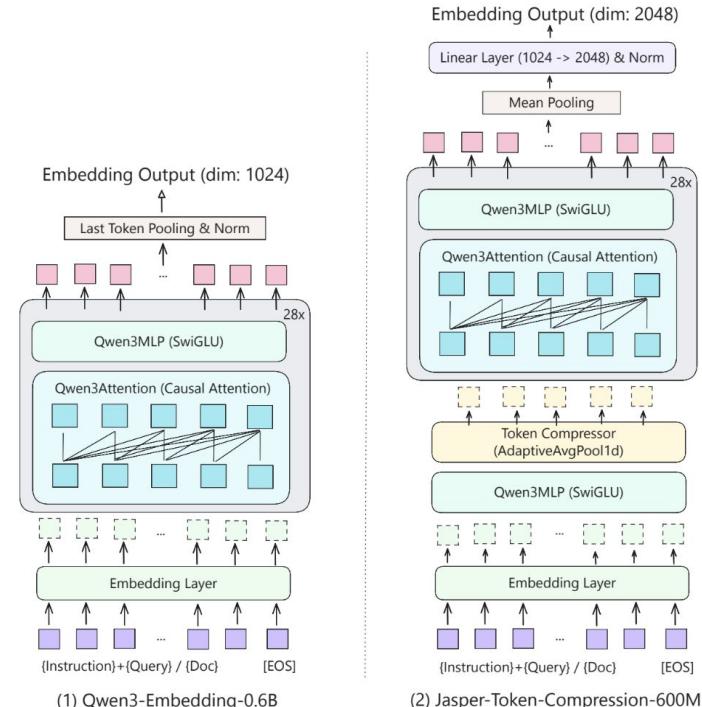


Figure 1: Model architecture of Qwen3-Embedding-0.6B (left) and Jasper-Token-Compression-600M (right).

Fine-tuning Jasper-Token-Compression-600M

Data

- Synthetic dataset by authors
- Split: 90% train / 10% IR-style dev set
- Max sequence length: 512 tokens

Model & Training

- Base model:
Jasper-Token-Compression-600M
- Loss: MultipleNegativesRankingLoss
Epochs: 1
- Batch size: 8

Metrics:

- MRR
- Recall@10
- nDCG@10

Task	Version	Recall@1/3/5/10			nDCG@1/3/5/10										
questions	Old (Jasper-600M)	0.0768	/	0.1790	/	0.2380	/	0.3318	0.1840	/	0.1831	/	0.2079	/	0.2473
	New (FT)	0.0782	/	0.1774	/	0.2428	/	0.3315	0.1802	/	0.1806	/	0.2099	/	0.2467
lastturn	Old (Jasper-600M)	0.1454	/	0.2956	/	0.3773	/	0.4682	0.3333	/	0.3151	/	0.3457	/	0.3848
	New (FT)	0.1503	/	0.2974	/	0.3707	/	0.4638	0.3513	/	0.3177	/	0.3444	/	0.3836
rewrite	Old (Jasper-600M)	0.1601	/	0.3285	/	0.4171	/	0.5289	0.3784	/	0.3493	/	0.3828	/	0.4306
	New (FT)	0.1643	/	0.3204	/	0.4214	/	0.5310	0.3848	/	0.3456	/	0.3852	/	0.4313

Table 2: Jasper-600M retrieval performance: Old vs fine-tuned (FT)

Conclusion

- A diverse group of retrieval models (BGE, Qwen-based, Jasper, GTE, FLLM) was systematically evaluated on a multi-turn RAG benchmark.
- Results show that modern embedding models outperform classic baselines, especially under rewrite and last-turn settings.
- Jasper-Token-Compression-600M was selected and fine-tuned on synthetic multi-turn data, leading to a measurable quality improvement in retrieval metrics.
- A major practical challenge is evaluation cost:
 - Large embedding LLMs are expensive to evaluate.
 - FLLM and GTE-Qwen-1.5B required ~17 hours per task, making large-scale experiments time-consuming.

Future work

- Chunking strategies: evaluate semantic, hierarchical, and query-aware chunking instead of fixed-size chunks.
- Query rewriting: improve or redesign the rewrite pipeline to better handle late-turn context and ambiguity.
- Full RAG pipeline: integrate retrieval with generation and evaluate end-to-end answer quality and faithfulness.

References

- Katsis et al., 2025 — MTRAG: A Multi-Turn Conversational Benchmark for RAG
- Lee et al., 2024 — Multi-Document Grounded Multi-Turn Synthetic Dialog Generation
- Cheng et al., 2024 — CORAL: Multi-turn Conversational RAG Benchmark
- Muennighoff et al., 2022 — MTEB: Massive Text Embedding Benchmark
- Elgohary et al., 2019 — CANARD: Question Rewriting in Context
- Yu et al., 2021 — Conversational Dense Retrieval (ConvDR)
- Mo et al., 2024 — History-Aware Conversational Dense Retrieval (HAConvDR)
- Gao et al., 2022 — HyDE: Hypothetical Document Embeddings
- Yu, Xiong & Callan, 2021 — ANCE-PRF
- Bruch et al., 2022 — Hybrid Retrieval Fusion