Best Benchmark Datasets for Evaluating a Synth-RAG System

To rigorously assess your Synth-RAG system, it's important to use retrieval-augmented QA benchmarks that align with its capabilities (document retrieval, hybrid search, multimodal embeddings) and can be implemented without excessive overhead. Below are some of the **best available and state-of-the-art** datasets/benchmarks for RAG systems, along with why they're relevant to Synth-RAG and feasible to use.

RAGBench (2024) – *Comprehensive RAG Evaluation*

RAGBench is a large-scale benchmark specifically designed for retrieval-augmented generation systems [1]. It offers **100,000 examples** drawn from *five diverse, industry-relevant domains* (e.g. biomedicine, finance, legal, customer support, and user manuals) [2]. Crucially, RAGBench includes data from **real user manuals**, making it especially pertinent to Synth-RAG's use case of querying MIDI synthesizer manuals [1]. Key features of RAGBench:

- **Unified Format:** It consolidates 12 existing QA datasets into a consistent RAG format [2] (covering domains like **electronics manuals, technical support, law, finance, medical research**, etc.), essentially serving as a broad "test suite" for RAG systems. This means you can evaluate Synth-RAG on domain-specific queries (e.g. the *"eManual"* subset for electronics/manuals) as well as general knowledge queries within one benchmark.
- **TRACe Metrics:** RAGBench introduces the **TRACe evaluation framework** – a set of explainable metrics (Context *Relevance*, Context *Utilization*, *Adherence* of answer to sources, and *Completeness*) [3]. These allow you to analyze not just accuracy, but *how* your system uses retrieved documents. For example, you can measure if Synth-RAG retrieves relevant manual pages and whether the GPT-4 model's answer actually utilizes those pages.
- **Realistic Queries:** The questions are often derived from real user inquiries (e.g. customer support forums or manual FAQs) rather than artificial prompts [4]. This helps ensure evaluation results reflect real-world performance on technical documentation.

Why use it? RAGBench is a state-of-the-art, comprehensive benchmark covering both retrieval and generation aspects of RAG. For Synth-RAG, it provides a ready-made way to test performance on **manuals and technical QA** (via relevant subsets) and compare against other systems. Its focus on *explainable evaluation* (via TRACe) can highlight if your hybrid retriever is fetching the right info and if the LLM is grounded in that info [3]. Given that it's available on HuggingFace, you can load the dataset and evaluate fairly quickly (perhaps running evaluation on a subset like *eManual* or *TechQA* portion for speed). Overall, RAGBench serves as an all-in-one **benchmark suite** for RAG systems [5], directly aligning with Synth-RAG's architecture and goals.

T^2-RAGBench (2024) – *Text-and-Table QA Benchmark*

T^2-RAGBench (Text-and-Table RAGBench) is a specialized benchmark focusing on documents that contain both textual and tabular data [6]. It was introduced to evaluate RAG methods on **real-world financial reports** with hybrid content (e.g. financial statements with tables) [7], but its insights are applicable to any scenario mixing prose and structured data. This benchmark comprises **32,908 question-context-answer triples** covering tasks that require retrieving the correct context (text or table) *and* performing reasoning (often numerical) on that context [8].

- **Challenge:** Unlike standard QA sets, T^2-RAGBench forces the system to **retrieve relevant content first** (from a collection of documents) *before* answering, rather than being given the passage. The questions are context-independent, meaning the model must find the correct document page or table on its own [9]. This mirrors Synth-RAG's two-stage retrieval: your system would need to pull the right manual page (which might contain spec tables or parameter lists) and then answer the query.
- **Hybrid Retrieval Effectiveness:** An important finding from T^2-RAGBench is that a *hybrid approach (dense + sparse)* works best for text+table data [10]. In fact, the authors report that a BM25 + dense embedding combo outperformed other methods on this benchmark [10]. This aligns perfectly with Synth-RAG's **hybrid search** design (FastEmbed + BM25). You can use T^2-RAGBench to verify that Synth-RAG's multivector retriever excels on content where information might be in tables (e.g. a MIDI implementation chart in a synth manual) and text.
- **Relevance to Synth-RAG:** If your synthesizer manuals contain specifications in tables or lists, this dataset tests the system's ability to handle that. It's also a **state-of-the-art** benchmark highlighting areas like numerical reasoning with retrieved data [11]. Implementing it would involve indexing the provided financial documents (or using their corpus if available) – given your system's flexibility, you could adapt to this without much trouble, or even sample a smaller set of the data for a quick run.

In summary, T^2-RAGBench is a cutting-edge benchmark that stresses **retrieval robustness and reasoning** on multi-format documents. It's highly relevant to Synth-RAG's architecture (especially the hybrid retrieval part) and is a great choice if you want to benchmark performance on tasks requiring reading **tables or formatted data** in PDFs.

FRAMES (Fact Retrieval And Reasoning Measurement Set, 2024) – *Multi-Hop Reasoning*

FRAMES is a unified evaluation set from Google that targets the end-to-end abilities of RAG systems in handling **complex, multi-hop questions** [12]. It contains **824 challenging questions** (provided as a test set) that often require combining information from **multiple sources** – in many cases, the answer necessitates retrieving and reasoning over *2 to 15 different Wikipedia articles* [12]. FRAMES is specifically designed to assess three dimensions of performance: **factuality**, **retrieval accuracy**, and **reasoning**.

- **Complex Queries:** The questions in FRAMES are not simple fact-lookups; they usually involve multiple constraints or steps. For example, a query might require finding a value in one article and then using it in a calculation with data from another article (numerical or temporal reasoning), or tracking a series of events across pages [13]. This pushes a RAG system to effectively chain together pieces of information. Synth-RAG's agentic workflow (via LangGraph) could be very relevant here –

e.g., the agent might perform iterative retrieval (`manuals_agent` could first search manuals, then fall back to web) to handle these kinds of questions.

- **Retrieval + Generation Evaluation:** FRAMES evaluates whether the system retrieved the necessary evidence and arrived at the correct answer. The dataset includes ground-truth answers and the set of Wikipedia source links that contain the needed facts [14] [15]. To use FRAMES, you would likely need to have a Wikipedia index or rely on the agent's web-search capabilities. Given the moderate size (824 queries), it's feasible to run within ~30 minutes by leveraging your hybrid search (perhaps restricting to the relevant wiki subset or using your web fallback).
- **Why it's SOTA:** This benchmark reflects *state-of-the-art evaluation for complex QA*, going beyond one-document queries. It tests Synth-RAG's **retrieval accuracy** (can it find *all* the needed pieces?) and **grounded reasoning** (can the LLM reason through them without hallucinating?). FRAMES is an excellent stress-test for the **agentic features** of your system – if Synth-RAG can handle FRAMES queries, it's a strong indicator of robust performance.

In practice, you might start by seeing how often Synth-RAG's retrieval finds the required wiki articles for each question (measuring recall) and whether the final answer matches the gold answer. FRAMES is available on HuggingFace [16] and comes pre-split as a test set. It's one of the *latest benchmarks* emphasizing multi-hop reasoning in RAG, making it highly relevant if you want to push Synth-RAG to its limits in reasoning capability.

RAGTruth (2024) – *Hallucination Analysis Benchmark*

RAGTruth is a benchmark dataset specifically created to measure and analyze **hallucinations in RAG-generated responses** [17]. Instead of question-answer pairs, RAGTruth provides a collection of about **18,000 responses** generated by various LLMs in a RAG setting, each response having detailed human annotations highlighting any hallucinated content [18] [17]. The hallucinations are categorized into fine-grained types, for example: *evident conflicts* (clear factual errors or contradictions), *subtle conflicts* (minor distortions of the context), *evident baseless info* (obviously made-up facts), and *subtle baseless info* [19].

- **Use for Evaluation:** With RAGTruth, you can evaluate how "grounded" Synth-RAG's answers are. One way to use it is to take the *same inputs* (queries + retrieved context) that were used to produce the RAGTruth responses and then generate answers with your system. You can then compare Synth-RAG's outputs to the annotated ones. Are there fewer hallucinated bits? Which category do any errors fall into? For instance, if RAGTruth has an example where many models introduced a subtle incorrect detail, does Synth-RAG avoid that? This can highlight the trustworthiness of your system's generation.
- **Diagnosing Issues:** The detailed annotations in RAGTruth (which mark exactly which words or phrases are unsupported by the retrieved documents) can help pinpoint failure modes. If Synth-RAG does hallucinate, RAGTruth data can tell you whether it's outright fabricating information or just slightly altering facts. This is valuable for debugging your **ColPali + GPT-4** generation pipeline's behavior.
- **Ease of Use:** RAGTruth is available (GitHub/HuggingFace) as a corpus of model responses with labels [18]. Implementing it might involve mapping or replicating the scenarios with your system. It's more of a **diagnostic benchmark** than a traditional QA dataset – you're measuring quality of answers (factual adherence) rather than whether answers are correct per se. Even if you don't run the full 18k examples, you could sample some to qualitatively assess Synth-RAG's tendency to hallucinate.

In summary, RAGTruth represents the *state-of-the-art in evaluating hallucination* for RAG systems. It's directly relevant to Synth-RAG's goal of providing reliable, source-backed answers. By using RAGTruth, you ensure that your system not only finds answers but also *doesn't introduce errors*, which is crucial for user trust in a manual-querying assistant.

BEIR Benchmark (2021) – *Zero-Shot Retrieval across Domains*

While not specific to RAG systems, **BEIR** (Benchmarking Information Retrieval) is a widely-used benchmark suite that covers **18 heterogeneous datasets across 9 task types** [20]. It is slightly older but remains a strong indicator of retrieval performance in many scenarios, including question answering, fact checking, news, scientific literature, forums, etc. BEIR is especially useful for testing the **retrieval component** of Synth-RAG in a zero-shot fashion:

- **Diverse Tasks:** BEIR includes QA datasets like **Natural Questions**, **HotpotQA**, and **FiQA**, fact-checking datasets like **FEVER** and **SciFact**, argument retrieval (e.g. ArguAna), duplicates detection (Quora), and more [21]. This diversity means you can see how well your **ColPali + FastEmbed + BM25** retrieval pipeline generalizes to domains beyond synthesizer manuals. For instance, the **HotpotQA** set in BEIR will test multi-hop retrieval (somewhat like FRAMES but on a simpler scale), and **NQ** will test open-domain fact retrieval on Wikipedia.
- **Domain Robustness:** Because BEIR spans domains from Wikipedia to biomedical articles to StackExchange, running Synth-RAG's retriever on BEIR is a good way to find any weaknesses in the embedding model or index configuration. A strong result on BEIR (e.g. high nDCG@10 or Recall@k across datasets) would indicate your hybrid search is state-of-the-art in general retrieval [22]. Notably, BEIR was designed for *zero-shot evaluation*, so you use pre-built indexes or encode the corpora with your model and directly evaluate retrieval quality without training on them – this aligns with how Synth-RAG would be used (since you likely aren't fine-tuning embeddings per manual).
- **Easy Integration:** The BEIR datasets and evaluation scripts are accessible via HuggingFace and other libraries. You can programmatically load a specific sub-dataset (e.g. `beir/nq`) and use Synth-RAG's retriever to return top-k documents, then compare with the provided ground-truth relevant documents. Many researchers report results on BEIR, so you can compare your system's retrieval metrics with published baselines to gauge competitiveness [21].

In essence, BEIR serves as a **baseline check** for your retrieval module across many contexts. It might not test the full RAG pipeline (since it doesn't involve generation in the evaluation), but it is *easily implementable* and will ensure your underlying search is performing at a state-of-the-art level. After all, if Synth-RAG can't retrieve relevant passages on standard tasks, it may struggle on the manuals too – BEIR helps verify the foundation.

Domain-Specific QA: TechQA and Manual QA

In addition to the broad benchmarks above, you might consider targeted evaluation on **domain-specific QA datasets** that closely mimic Synth-RAG's application domain (technical manuals and support documentation):

- **TechQA (IBM, 2020):** TechQA is a dataset of *real technical support questions* and answers from IBM product forums [4]. It has ~1,400 Q&A pairs (train 600, dev 310, test 490) and comes with a companion corpus of ~800k IBM Technotes (support articles) [23]. This dataset is valuable because the questions are long-form and problem-oriented (much like a user asking how to solve an issue with a device), and the answers are grounded in *actual technical documents*. Running Synth-RAG on TechQA would test its document retrieval and answer accuracy in a scenario very similar to synthesizer manuals (just a different domain). Since TechQA is included as part of RAGBench [24], you can leverage the RAGBench version (which likely provides queries and reference docs) for easier implementation. Expect questions like *"How do I configure SSL on IBM HTTP Server?"* and answers that come from multi-page support notes – a great analog to complex synth configuration questions.

- **Manual Q&A or Custom Evaluation:** There isn't a well-known public dataset *explicitly for music synthesizer manuals*, but you could create a small benchmark yourself using Synth-RAG's capabilities. For example, consider taking FAQs from synthesizer manuals or forum questions about popular synths (Digitone, etc.) and their known answers. This can serve as a test set to ensure your system retrieves the correct manual pages and produces accurate answers. In lieu of that, the **"eManual"** subset of RAGBench (derived from electronics/user manuals) provides a proxy [1]. It contains Q&A pairs based on product manuals – using this subset will directly evaluate how well Synth-RAG handles *manual-style knowledge*. Since it's relatively small (~1.3k examples in RAGBench [25]), it's not too time-consuming to run.

These domain-focused evaluations complement the larger benchmarks. **TechQA** checks performance on *enterprise technical documentation*, and *eManual/RAGBench or custom manual QA checks on* consumer device manuals*. Both are highly relevant: success here would demonstrate Synth-RAG's value in its target domain. They are also reasonably easy to set up, especially if leveraging existing data splits from RAGBench or known Q&A from documentation.

FEVER (2018) – *Fact-Checking with Retrieval*

Finally, it's worth mentioning **FEVER** (Fact Extraction and VERification) as a classic benchmark that, while older, remains a strong test of a system's ability to retrieve evidence and make a correct judgment. FEVER consists of **185,000 human-written claims** that models must label as *Supported*, *Refuted*, or *Not Enough Info* by retrieving and checking Wikipedia evidence [26]. In a RAG context, this means for each claim your system would need to find the relevant wiki sentences and then decide if the claim is true or false.

- **Why it's relevant:** The skill tested by FEVER is closely tied to **grounded generation** – essentially, it measures whether the system can avoid hallucinations and only state facts supported by retrieved documents. Synth-RAG could be evaluated on FEVER by having it generate an answer like "Supported because ..." or "Refuted because ..." with citations. This would stress-test the **adherence** of your generated answers to the sources (similar to the Adherence metric in TRACe) in a quantifiable way.
- **Ease of use:** FEVER's corpus is Wikipedia, and its evaluation is straightforward classification of the claim's veracity with respect to evidence. If you have a Wikipedia index already (from FRAMES or general usage), adapting it for FEVER is not too difficult. The main addition would be prompting the LLM to make a support/refute decision. While not as new as RAGBench or FRAMES, FEVER is still used as a benchmark for factual consistency and has helped advance techniques to reduce hallucinations. Given that Synth-RAG's purpose is to provide *accurate information from manuals*, performing well on FEVER would be a reassuring signal of its factual reliability.

In summary, to evaluate your Synth-RAG system, you have a rich set of benchmarks at your disposal:

- **RAGBench** [1] [3] – Comprehensive, multi-domain RAG evaluation (including *user manuals* data) with explainable metrics.
- **T^2-RAGBench** [27] [10] – Challenges hybrid retrieval and reasoning on text+table documents, reflecting tasks requiring parsing specs/tables.
- **FRAMES** [12] – Multi-hop reasoning and retrieval accuracy on complex questions, ideal for testing agentic workflows.
- **RAGTruth** [17] – A corpus to quantify hallucination in answers, ensuring your system's responses stay faithful to manuals.
- **BEIR** [21] – A broad retrieval benchmark to validate the effectiveness of your embedding and search across domains.
- **Domain-specific QA (TechQA** [4] **, RAGBench-eManual)** – Focused tests using technical support Q&A and product manual queries that mirror Synth-RAG's target domain.
- **FEVER** [26] – A fact-checking benchmark to assess evidence retrieval and factual correctness of generated answers.

All of these are **implementable within a reasonable time frame** – many have ready-to-use data on HuggingFace or via APIs, and you can select subsets if needed to fit in ~30–60 minutes of evaluation. By combining results from these benchmarks, you'll get a well-rounded picture of Synth-RAG's performance: from retrieval quality and speed to answer accuracy, reasoning ability, and faithfulness to the source. Each dataset shines a light on different facets of RAG systems, helping you identify strengths and areas for improvement in your synthesizer manual QA system. Good luck with your benchmarking!

Sources: RAGBench paper [1] [3]; T^2-RAGBench paper [27] [10]; EvidentlyAI RAG benchmarks blog (FRAMES, RAGTruth, BEIR, FEVER) [12] [17] [20] [26]; TechQA paper [4] [28].

[1] [2407.11005] RAGBench: Explainable Benchmark for Retrieval-Augmented Generation Systems
https://arxiv.org/abs/2407.11005

[2] [3] [5] [Literature Review] RAGBench: Explainable Benchmark for Retrieval-Augmented Generation Systems
https://www.themoonlight.io/en/review/ragbench-explainable-benchmark-for-retrieval-augmented-generation-systems

[4] [23] [28] The TechQA Dataset for ACL 2020 - IBM Research
https://research.ibm.com/publications/the-techqa-dataset

[6] [7] [8] [9] [10] [11] [27] Paper page - T^2-RAGBench: Text-and-Table Benchmark for Evaluating Retrieval-Augmented Generation
https://huggingface.co/papers/2506.12071

[12] [13] [17] [19] [20] [21] [22] [26] 7 RAG benchmarks
https://www.evidentlyai.com/blog/rag-benchmarks

[14] [15] [16] google/frames-benchmark · Datasets at Hugging Face
https://huggingface.co/datasets/google/frames-benchmark

[18] GitHub - ParticleMedia/RAGTruth: Github repository for "RAGTruth: A Hallucination Corpus for Developing Trustworthy Retrieval-Augmented Language Models"
https://github.com/ParticleMedia/RAGTruth

[24] [25] galileo-ai/ragbench · Datasets at Hugging Face
https://huggingface.co/datasets/galileo-ai/ragbench