TARGET: Benchmarking Table Retrieval for Generative Tasks

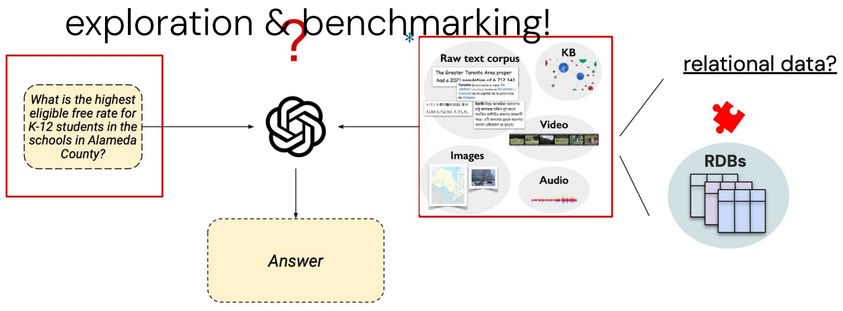
Xingyu Ji, Aditya Parameswaran, Madelon Hulsebos

Evaluating Tabular Data Retrieval in LLM-powered Data Pipelines

Why table retrieval?

e2e data pipelines rely on table retrieval!

- Retrieval is also critical in end-to-end data QA and analysis pipelines.
- LLMs improve ***reasoning*** through ***external corpora****: text, images, KBs (RAG).*
- Tabular data contains fresh, structured, domain data.
- RAG over ***structured data*** requires further exploration & benchmarking!



Need for evaluating Table Retrieval 📊

- Current benchmarks evaluate ***generation only*** (ie, fact verification, table QA, text-to-SQL), assuming tables are identified.
- Capabilities of ***methods for retrieving the correct table(s)*** affects downstream task generation quality, and is unstudied.

Why TARGET?

Research questions

❓ What is the effectiveness of table retrievers, and paradigms, across data analysis and QA tasks?
❓ What is the relation between retrieval and generation in end-to-end pipelines?
❓ How does table retrieval compare to alternatives, e.g. leveraging LLM memory and long-context LLMs?

Challenges

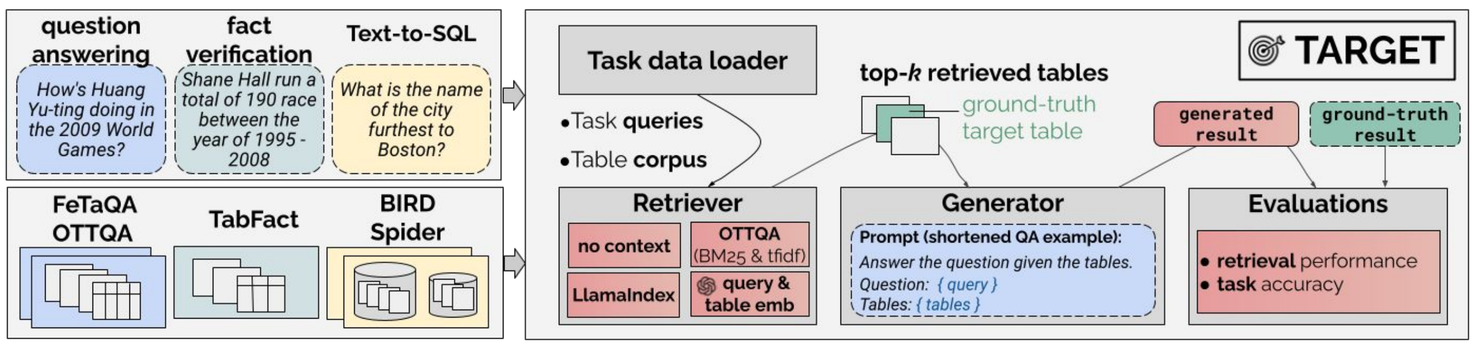
- Different methods *vary significantly* in how structured data is preprocessed and embedded!
- Differences in. assumptions made about the input data, queries, tasks,, etc.

TARGET Benchmark Overview

Diverse coverage
- tasks & datasets across domains
- various paradigms baseline retrievers (ie OpenAI, LlamaIndex, OTTQA)

Extensibility
- standardized corpus formatting, HF datasets
- possible extension of new tasks & generators

Adaptability
- adapts to wide range of table retrievers
- straightforward user interface
- plug in and run your evals!



Results

Results of baselines for retrieval and downstream tasks. R@10 = recall@top 10 retrieved tables, retrieval time s in seconds, SB = Sacrebleu, EX = SQL execution accuracy.

Method	Question Answering						Fact Verification			Text-to-SQL					
	OTTQA			FeTaQA			TabFact			Spider			BIRD		
	R@10	s	SB	R@10	s	SB	R@10	s	P/R/F1	R@1	s	EX	R@1	s	EX
No context	-	-	0.414	-	-	12.495	-	-	0.578/0.42/0.44	-	-	0	-	-	0
OTT-QA BM25	**0.955**	0.001	0.606	0.082	0.001	1.631	0.338	0.001	0.75/0.26/0.39	0.635	0.001	0.385	0.709	0.001	0.181
w/o table title	0.443	0.001	0.529	0.084	0.001	1.555	0.331	0.001	0.75/0.26/0.38	0.5	0.001	0.376	0.535	0.001	0.164
OTT-QA TF-IDF	0.950	0.001	0.425	0.083	0.001	1.639	0.336	0.001	0.75/0.26/0.38	0.622	0.001	0.474	0.640	0.001	0.227
w/o table title	0.43	0.001	0.593	0.083	0.001	1.527	0.322	0.001	0.75/0.25/0.37	0.492	0.001	0.376	0.491	0.001	0.164
LlamaIndex	0.458	0.354	0.507	0.435	0.396	13.745	**0.827**	0.297	0.73/0.34/0.47	0.735	0.198	0.559	0.937	0.228	0.311
OpenAI embedding	0.950	0.190	0.599	**0.722**	0.200	17.64	0.779	0.189	0.76/0.51/0.61	0.768	0.193	0.602	0.926	0.199	0.317
header only	0.950	0.189	0.61	0.718	0.18	17.66	0.781	0.187	0.75/0.48/0.58	**0.833**	0.175	0.646	**0.958**	0.191	0.323

Insights 👀

- BM25 / TF-IDF not robust for tabular data,!
- Good perf with *out-of--box* OpenAI embeddings.
- Adding / generating *descriptive titles* improves retrieval accuracy.
- Table summary *not effective* if several tables contain *similar content*.
- Grounding LLM responses in factual data remains crucial for accuracy!
- Metadata is important, but adding "data rows" can distract embeddings.

What's next?

Extensions: impact of corpus/context scale, in-database table retrieval.
Retrievers: assessing relevant metadata, hierarchical retrieval pipelines.

target-benchmark.github.io

pip install target-benchmark