Retrieval-Augmented Generation for Knowledge-Intensive Natural Language Processing Tasks

Michael Pritz, Alexander Pluska, Johannes Blaha, Tobias Grantner

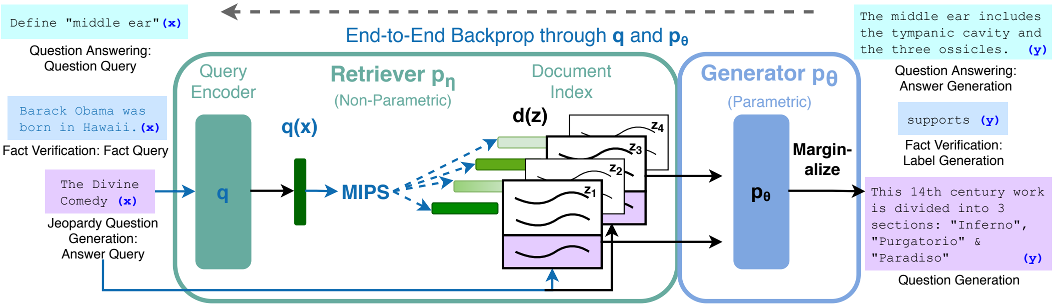
November 14, 2024

TECHNISCHE
UNIVERSITÄT
WIEN

Overview

- Paper Reproduction
- Generator comparison
- Fact verification
- Retriever dataset relevance for medical QA
- Web search retriever
- RAG for Automated Theorem Proving
- Conclusion

RAG Architecture



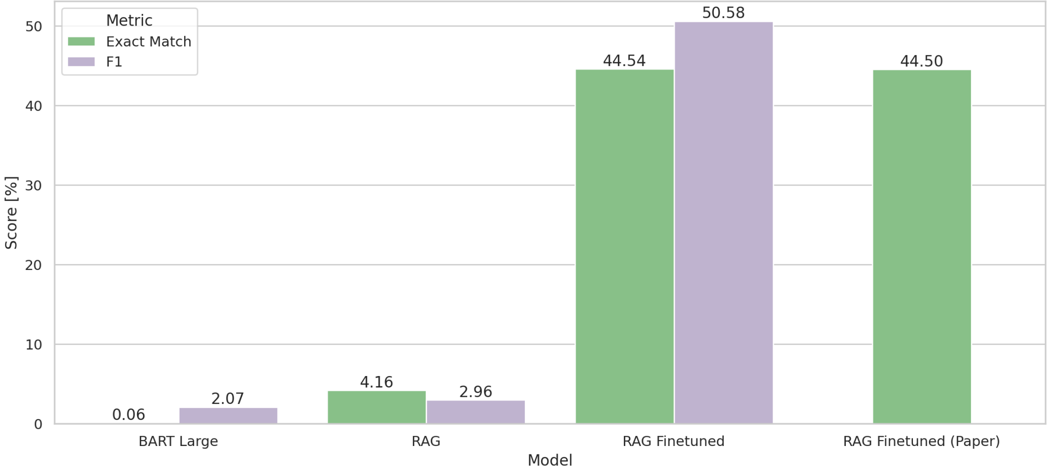
https://arxiv.org/abs/2005.11401

Reproducability

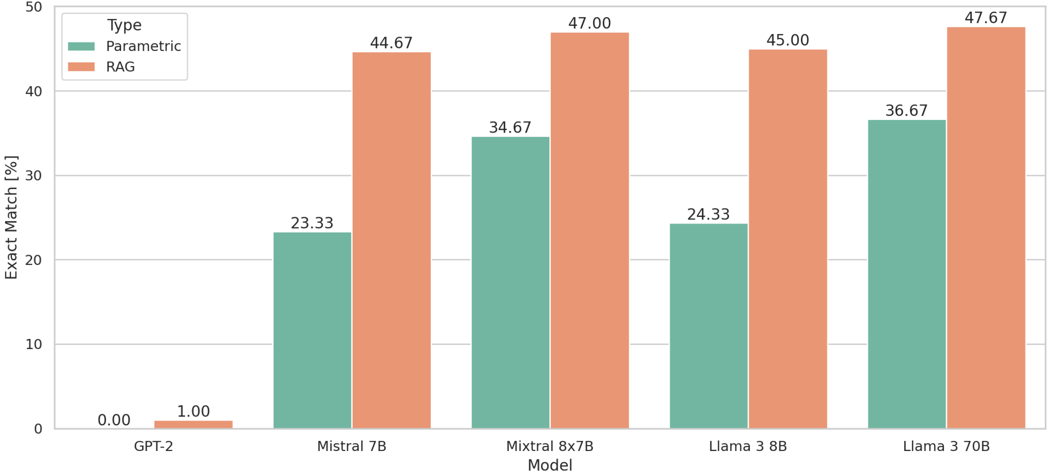
- Paper[1] provides code to run, fine-tune and evaluate
- Makes use RagRetriever in transformers library
  - Poorly documented
  - Limited configurability
  - No version specified
- Resource intensive

[1] https://arxiv.org/abs/2005.11401

Reproduction Results

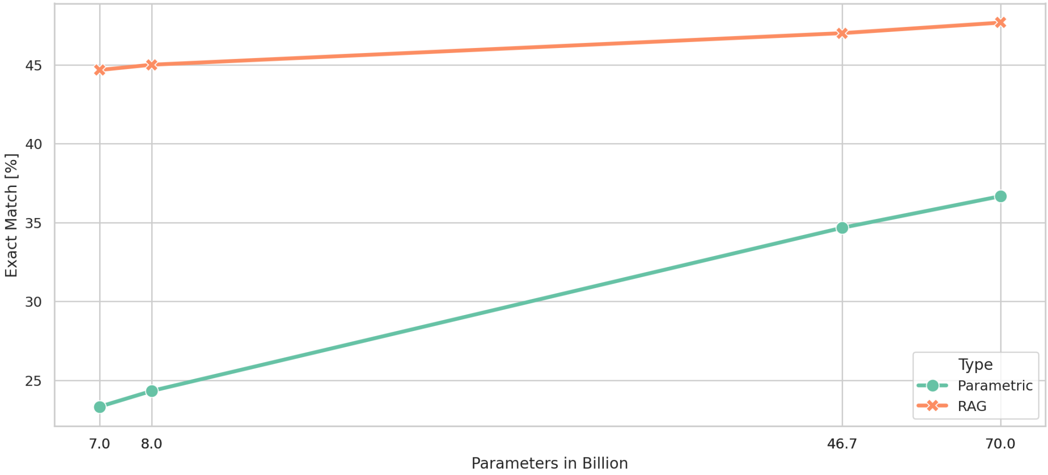


Generator evaluation - Exact Match

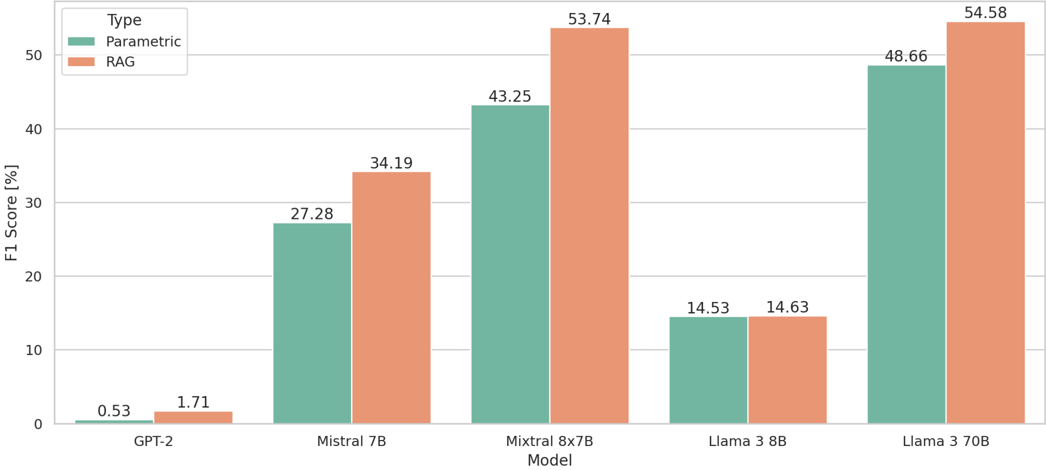


TECHNISCHE UNIVERSITÄT WIEN

Generator evaluation - Exact Match by Parameter Size



Generator evaluation - F1



Fact Verification: Task

Task: Given a claim and some evidence, tell whether the evidence supports the claim or not.

Example 1:

- Claim: People in Austria live to be around 80 years old.
- Evidence: Life expectancy in Austria is 81 years.
- Answer: Supports

Example 2:

- Claim: The earth is flat.
- Evidence: New evidence found that the earth is indeed flat.
- Answer: Supports

Fact Verification: Results from Lewis et al.

- Generator: BART-large
- Retriever: Encoder from DPR
  71% (top 1) and 90% (top 10)
- 2-Way and 3-Way classification
- Results:
  89.5% (2-way classification)
  72.5% (3-way classification)
- Ablation study: Freeze retriever during training
  90.6% vs. 89.4% (2-way classification)
  74.5% vs. 72.9% (3-way classification)

Fact Verification: Our Approach

- Generator: Llama-3-8b-Instruct
- Retriever: all-MiniLM-L6-v2

Fact Verification: Dataset

- Create a new fact verification dataset based on:
  - FEVER [2]
  - FEVER Gold [3]
- Features: Claim, Evidence (fine), Evidence (coarse)
- Labels: SUPPORTS and REFUTES


```
{'claim': 'The Boeing 767 became the most frequently used airliner for transatlantic flights between North America and Europe.',
 'label': 'SUPPORTS',
 'evidence_coarse': ["The Boeing 767 is a mid - to large-size , long-range , wide-body twin-engine jet airliner built by Boeing Commercial Airplanes . It was Boeing 's f
 'evidence_fine': ['In the 1990s , the 767 became the most frequently used airliner for transatlantic flights between North America and Europe .']}
```


[2] https://huggingface.co/datasets/fever/fever
[3] https://huggingface.co/datasets/copenlu/fever_gold_evidence

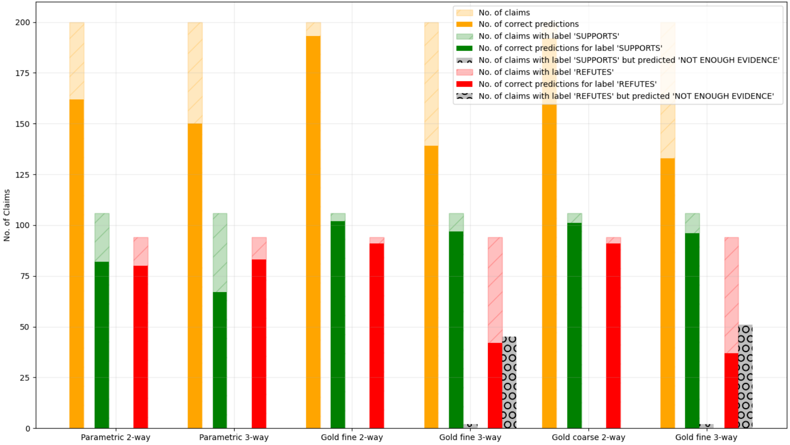
Quantitative:

- Parametric 2-Way
- Parametric 3-Way
- Gold (fine) 2-Way
- Gold (fine) 3-Way
- Gold (coarse) 2-Way
- Gold (coarse) 3-Way
- Retriever accuracy
- 2-Way accuracy / Context Len

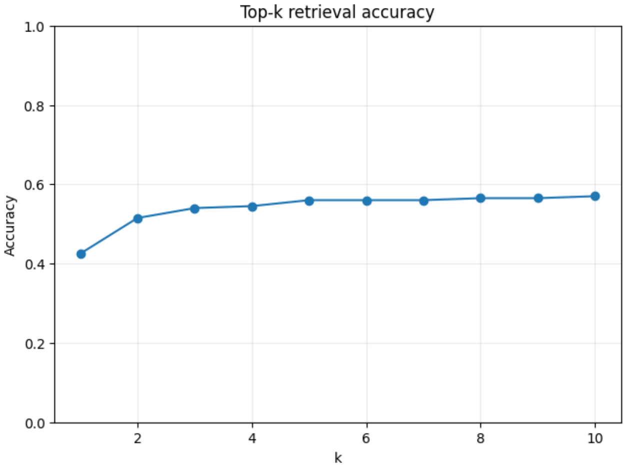
Qualitative:

- Prompt Engineering:
  - Varying the standard prompt
  - Chain of Thought
- Varying output length

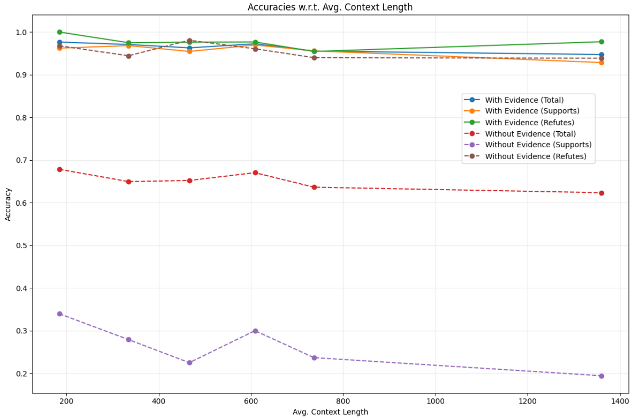
Fact Verification: Llama-3-8b



Fact Verification: Retriever Accuracy



Fact Verification: 2-Way Accuracy w.r.t. Context Length



Domain specific dataset evaluation

- Evaluate generators and retrievers with domain specific datasets
- Implement web search retriever
- Based on two multiple choice science datasets
  - BigBio - science exam questions[4]
  - MMLU - college medicine [5]

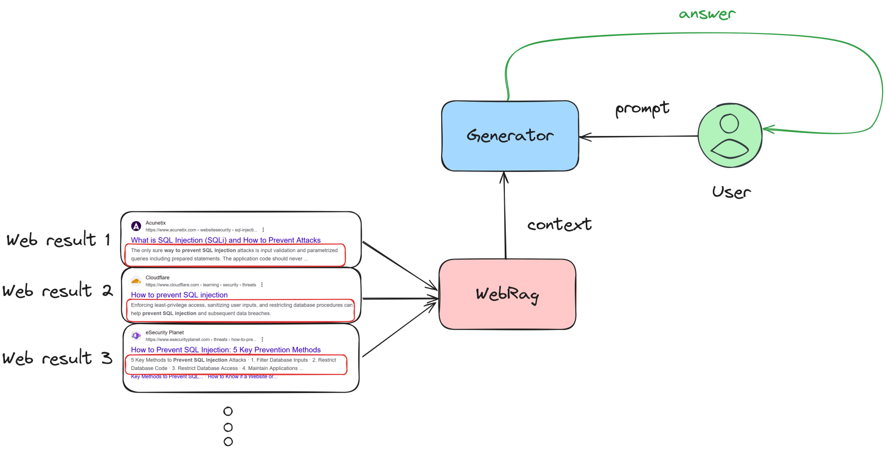
[4] https://huggingface.co/datasets/bigbio/sciq
[5] https://huggingface.co/datasets/cais/mmlu/viewer/college_medicine

Domain specific dataset evaluation

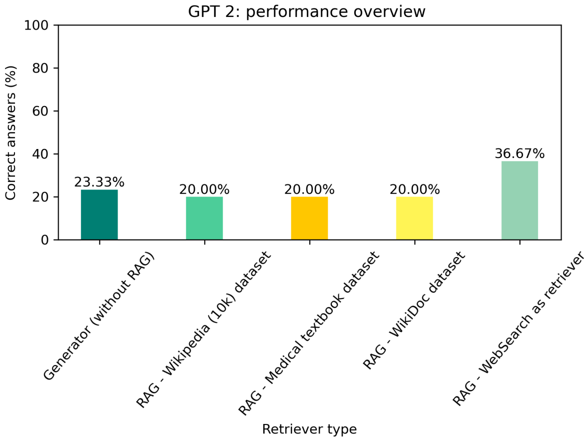
- Retriever: all-MiniLM-L6-v2
- Trim context to context size of model
- Datasets:
  - Generator (without RAG)
  - Wikipedia 10k[6]
  - Medical textbooks[7]
  - Wiki Doc (medical professionals)[8]
  - Web Search retriever

[6] https://huggingface.co/datasets/sentence-transformers/wikipedia-en-sentences
[7] https://huggingface.co/datasets/MedRAG/textbooks
[8] https://huggingface.co/datasets/medalpaca/medical_meadow_wikidoc

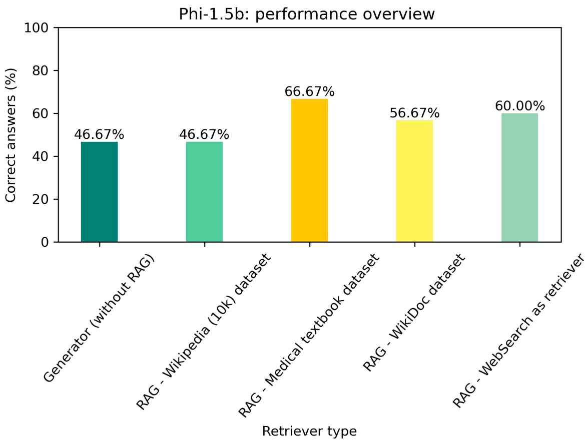
Web search retriever



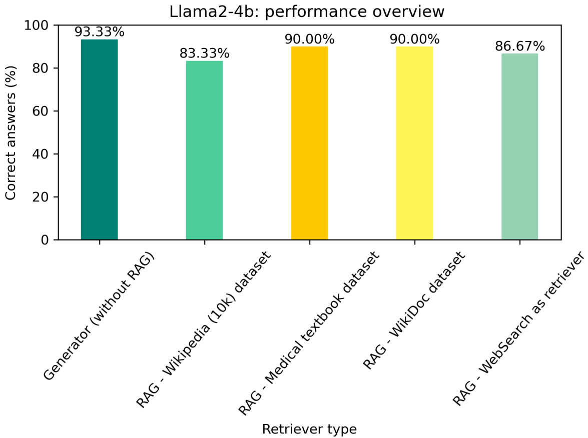
BigBio dataset - GPT2 (137M)



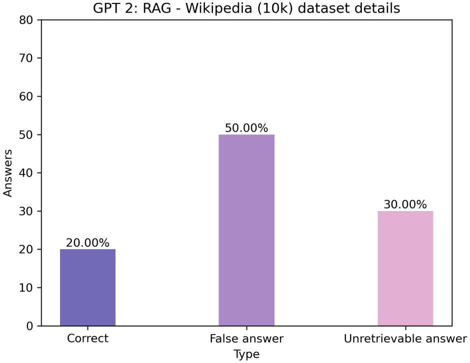
BigBio dataset - Phi3 (1.5B)

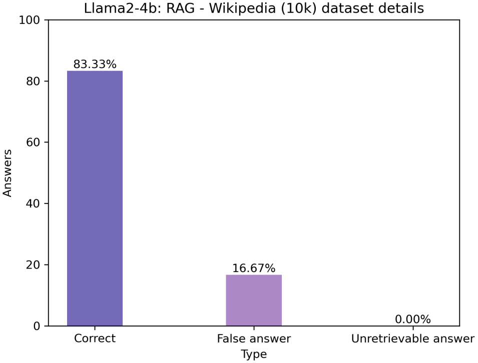


BigBio dataset - Llama2 (4B)



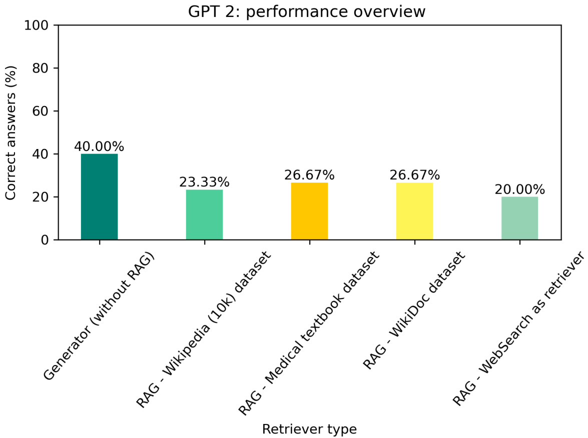
GPT2 - Unretrievable answers



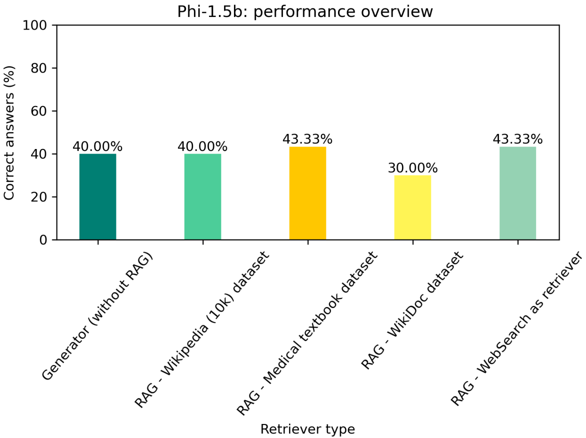


Let's make it more difficult

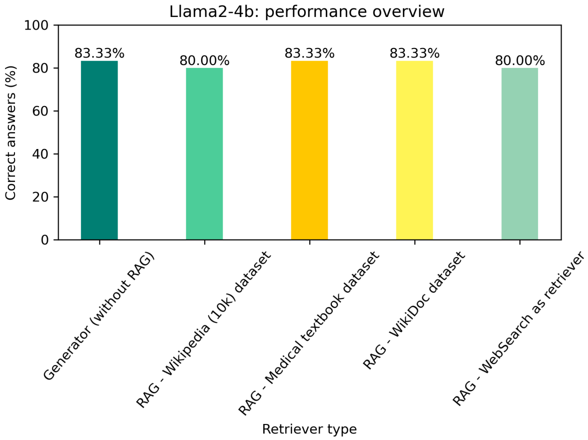
MMLU (College Medicine) - GPT2 (137M)



MMLU (College Medicine) - Phi3 (1.5B)



MMLU (College Medicine) - Llama2 (4B)

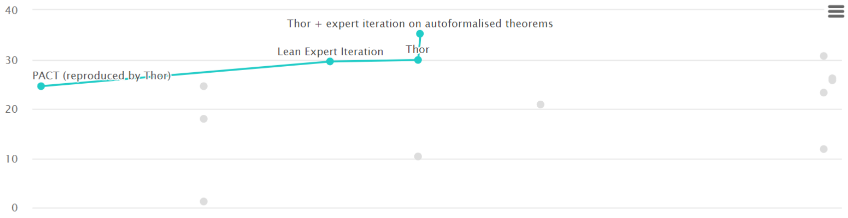


```
inductive ℕ : Type
    | zero : ℕ
    | succ : ℕ → ℕ
def add : ℕ → ℕ → ℕ
    | m zero => m
    | m (succ n) => succ (add m n)
inductive Eq (a : ℕ) : ℕ → Type where
    | refl : Eq a a
theorem add_zero (m : ℕ) : m + zero = m := refl
theorem add_succ (m n : ℕ) : m + succ n = succ (m + n) := refl
```

```lean
theorem zero_add (m : ℕ) : zero + m = m := by
  induction m with
  | zero => rfl
  | succ n ih => rw [add_succ, ih]
theorem succ_add (m n : ℕ) : succ m + n = succ (m + n) := by
  induction n <;> simp [*, add_zero, add_succ]
theorem add_comm (m n : ℕ) : m + n = n + m := by
  induction n <;> simp [*, add_zero, add_succ, succ_add, zero_add]
theorem add_assoc (m n k : ℕ) : m + n + k = m + (n + k) := by
  induction k <;> simp [*, add_zero, add_succ]
```

Automated Theorem Proving

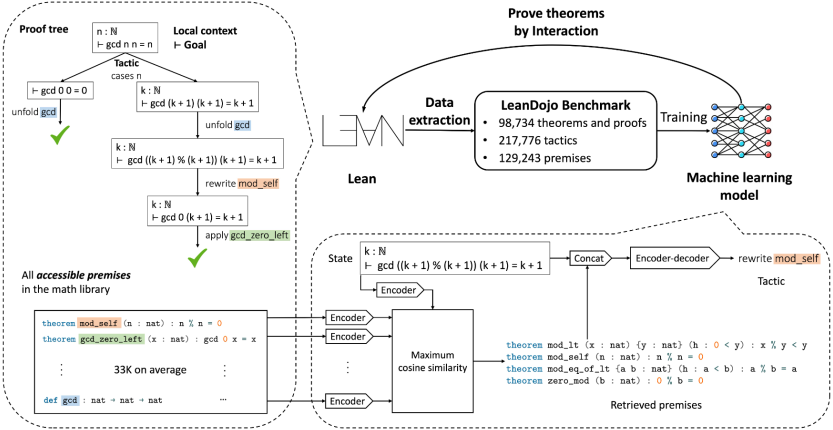
- More and more serious mathematics is being done in Lean (Liquid Tensor Experiment, Polynomial Freiman-Ruzsa Conjecture, FLT, ...).
- Great interest in automation but existing tools still lacking.



LeanDojo: Theorem Proving with Retrieval-Augmented Language Models

Kaiyu Yang[1], **Aidan M. Swope**[2], **Alex Gu**[3], **Rahul Chalamala**[1], **Peiyang Song**[4], **Shixing Yu**[5], **Saad Godil**[*], **Ryan Prenger**[2], **Anima Anandkumar**[1,2]
[1]Caltech, [2]NVIDIA, [3]MIT, [4]UC Santa Barbara, [5]UT Austin
`https://leandojo.org`

LeanDojo


Proof tree
n : ℕ
⊢ gcd n n = n
Local context
⊢ Goal
Tactic
cases n
⊢ gcd 0 0 = 0
unfold gcd
k : ℕ
⊢ gcd (k + 1) (k + 1) = k + 1
unfold gcd
k : ℕ
⊢ gcd ((k + 1) % (k + 1)) (k + 1) = k + 1
rewrite mod_self
k : ℕ
⊢ gcd 0 (k + 1) = k + 1
apply gcd_zero_left
All *accessible premises*
in the math library
theorem mod_self (n : nat) : n % n = 0
theorem gcd_zero_left (x : nat) : gcd 0 x = x
33K on average
def gcd : nat → nat → nat
Prove theorems
by Interaction
Data
extraction
LeanDojo Benchmark
98,734 theorems and proofs
217,776 tactics
129,243 premises
Training
Lean
Machine learning
model
State
k : ℕ
⊢ gcd ((k + 1) % (k + 1)) (k + 1) = k + 1
Concat
Encoder-decoder
rewrite mod_self
Tactic
Encoder
Encoder
Encoder
Encoder
Maximum
cosine similarity
theorem mod_lt (x : nat) {y : nat} (h : 0 < y) : x % y < y
theorem mod_self (n : nat) : n % n = 0
theorem mod_eq_of_lt {a b : nat} (h : a < b) : a % b = a
theorem zero_mod (b : nat) : 0 % b = 0
Retrieved premises

LeanDojo - Conclusion

- We replicated the ReProver experiments, achieving 26% and 24% accuracy on subsets of the `novel_premises` dataset with and without retrieval respectively.
- Generated tactics are often not valid, i.e. cannot be applied to the proof state. A significant number of candidates needs to be generated in order to make progress.
- While RAG presents an improvement of pure LLM approaches to ATP, it alone is not sufficient to overcome their current shortcomings.

Method	`random`	`novel_premises`
`tidy`	23.8	5.3
GPT-4	29.0	7.4
ReProver	**51.2**	**26.3**
w/o retrieval	47.6	23.2

Conclusion

- We replicated the experiments from "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks" and can confirm their results.
- The success of RAG crucially depends on the quality of the underlying dataset.
- For question answering RAG is especially effective with smaller generators.
- We have experimented with a web search retriever and obtained comparable results to RAG from a fixed underlying database in our experiments.
- In another knowledge-intensive task, ATP, RAG allows a small improvement but does not seem to address the primary difficulty.

Implementation at `https://github.com/lexpk/dlnlp-rag`.

Appendix

Fact Verification: Parametric Prompt

```
prompt_template = """
You are a helpful, smart, kind, and efficient AI assistant who always fulfills the user requests to the best of its abilities and strictly sticks to the given inst

Instructions:
You answer SUPPORTS if the claim is true.
You answer REFUTES if the claim is false.

Claim:
{claim}

Answer:
"""
```

Fact Verification: 2-Way Prompt

```
prompt_template = """
You are a helpful, smart, kind, and efficient AI assistant who always fulfills the user requests to the best of its abilities and strictly sticks to the given inst

Instructions:
You answer SUPPORTS if context EXPLICITLY supports the claim.
You answer REFUTES if the context EXPLICITLY refutes the claim.

Context:
{context}

Claim:
{claim}

Answer:
"""
```

Fact Verification: 3-Way Prompt

```
prompt_template = """
You are a helpful, smart, kind, and efficient AI assistant who always fulfills the user requests to the best of its abilities and strictly sticks to the given inst

Instructions:
You answer SUPPORTS if context EXPLICITLY supports the claim.
You answer REFUTES if the context EXPLICITLY refutes the claim.
You answer NOT ENOUGH EVIDENCE if the context does not provide enough information to explicitly support or refute the claim.

Context:
{context}

Claim:
{claim}

Answer:
"""
```

Fact Verification: CoT 3-Way Prompt

```
CoT_prompt_template = """
You are a helpful, smart, kind, and efficient AI assistant who always fulfills the user requests to the best of its abilities and strictly sticks to the given ins

Instructions:
You answer SUPPORTS if context EXPLICITLY supports the claim.
You answer REFUTES if the context EXPLICITLY refutes the claim.
You answer NOT ENOUGH EVIDENCE if the context does not provide enough information to explicitly support or refute the claim.

Context:
Barack Hussein Obama II[a] (born August 4, 1961) is an American politician who served as the 44th president of the United States from 2009 to 2017. As a member of

Claim:
Obama served as a US senator before becoming president.

Answer:
The context states that Barack Obama served as United States senator representing Illinois from 2005 to 2008.
It also mentions that he served as president from 2009 to 2017.
Hence, the context SUPPORTS the claim.

Context:
{context}

Claim:
{claim}

Answer:
"""
```

Generator Evaluation - RAG Prompt

```
Context:
{context}

Prompt: You are an assistant for question-answering tasks. Use the pieces
    of retrieved context above to answer the question. If the answer cannot
    be extracted from the context, use the knowledge you have. Give very
    concise answers containing only necessary information to answer the
    question of a couple of words maximum, no additional information or
    explanation. Do not repeat the question.

Question: {question}

Answer:
```

Generator Evaluation - Generation Prompt

```
Prompt: You are an assistant for question-answering tasks. Give very
    concise answers containing only necessary information to answer the
    question of a couple of words maximum, no additional information or
    explanation. Do not repeat the question.

Question: {question}

Answer:
```

Answer extraction

Example: “A) The most common ...”

Matches:

→ A | B | C | D

→ : | , |) |]

- Rule 1: first occurrence of letter & symbol
- Rule 2: first occurrence of exact text answer

Domain specific evaluation - key takeaways

- Small model + RAG similar to large model
- Web search retriever
  - Similar performance to other RAG approaches
  - Could be useful for new/current information retrieval
  - Lower computational requirements (low power devices)
- Dataset quality and specificity is key

Domain specific evaluation - prompt

```
promt = f"""
You are a helpful AI assistant. {context_prompt} Think step by step and answer either with A), B), C) or D). Add nothing else.

{context}

Query:
{query}

Answer:
"""
```