

QACHECK: A Demonstration System for Question-Guided Multi-Hop Fact-Checking

Liangming Pan^{1,2}, Xinyuan Lu³, Min-Yen Kan³, Preslav Nakov¹

¹ MBZUAI ² University of California, Santa Barbara ³ National University of Singapore

liangmingpan@ucsb.edu, luxinyuan@u.nus.edu, kanmy@comp.nus.edu.sg, preslav.nakov@mbzuai.ac.ae

Introduction

Motivation

Given a claim made by a claimant, the goal of **fact-checking** is to find a collection of evidence and provide a verdict about the claim's veracity label based on the evidence.

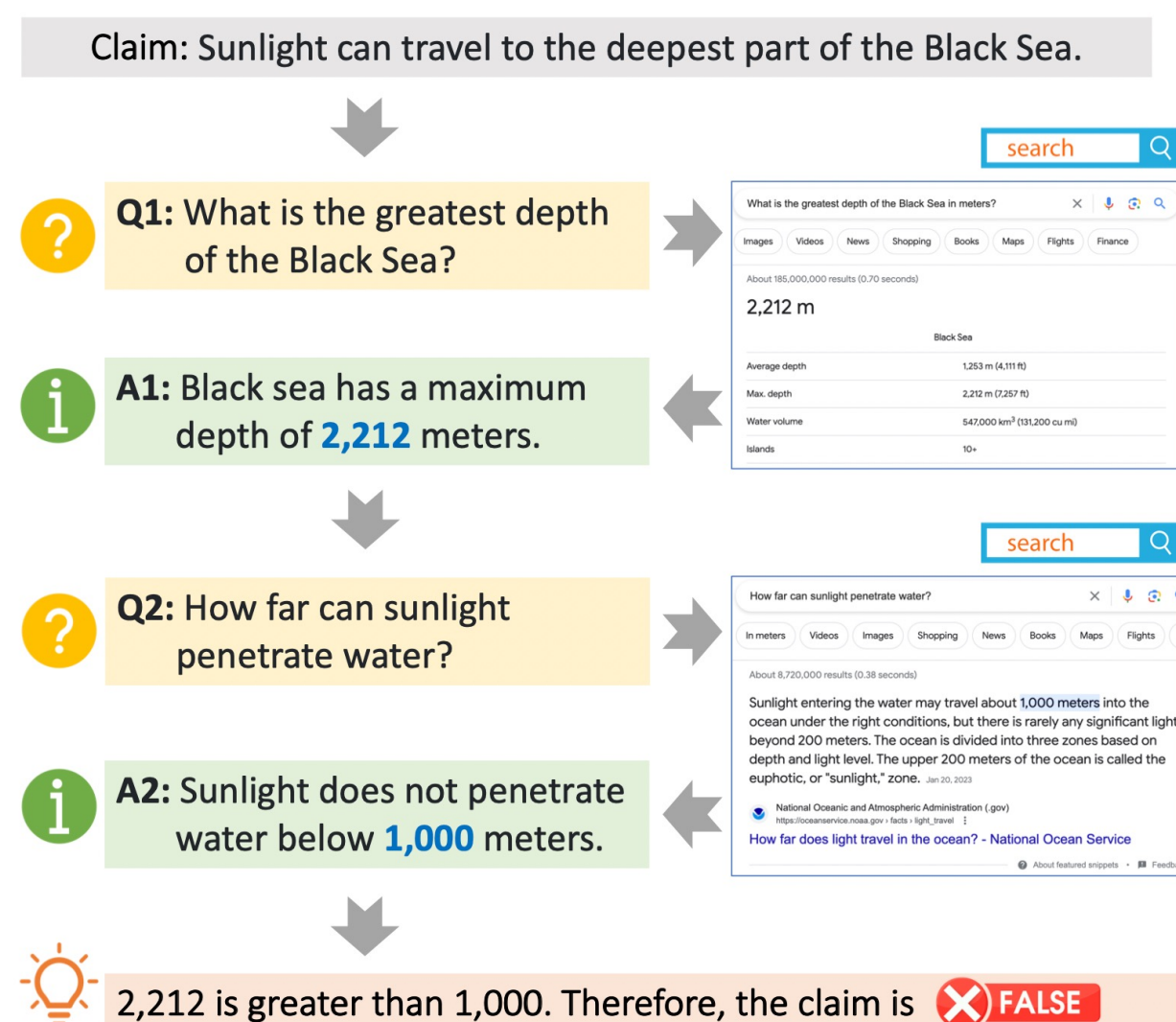
Real-world claims often require **complex and multi-step reasoning** to solve.

A human fact-checker needs to **decompose** the claim, gather multiple pieces of evidence, and perform step-by-step reasoning.

QACheck

We introduce the **Question-Guided Multi-hop Fact-Checking (QACHECK)** system, which addresses the aforementioned issues by generating multi-step explanations via **question-guided reasoning**.

The verification of the claim is guided by asking and answering a series of relevant questions.



System Interface

QACheck: Question-Guided Multi-hop Fact-Checking Demo

Instructions: Select a claim or just enter your own claim otherwise, and then check the model's output.

QA Model: GPT Reciter-Reader

1. Select or input a custom claim

Please select a claim.

Ulrich Walter's employer is headquartered in Cologne.

Custom Claim

Submit

2. Submit to fact-check the input claim

Input Claim:

Lars Onsager won the Nobel prize when he was 30 years old.

3. Visualize the question-answering guided reasoning process

Question Answering Decomposition:

Reasoning depth: 0

Generated Question: In which year did Lars Onsager win the Nobel prize?

Predicted Answer: 1968

Reasoning depth: 1

Generated Question: Which year was Lars Onsager born?

Predicted Answer: 1903

4. The final prediction result with rationale

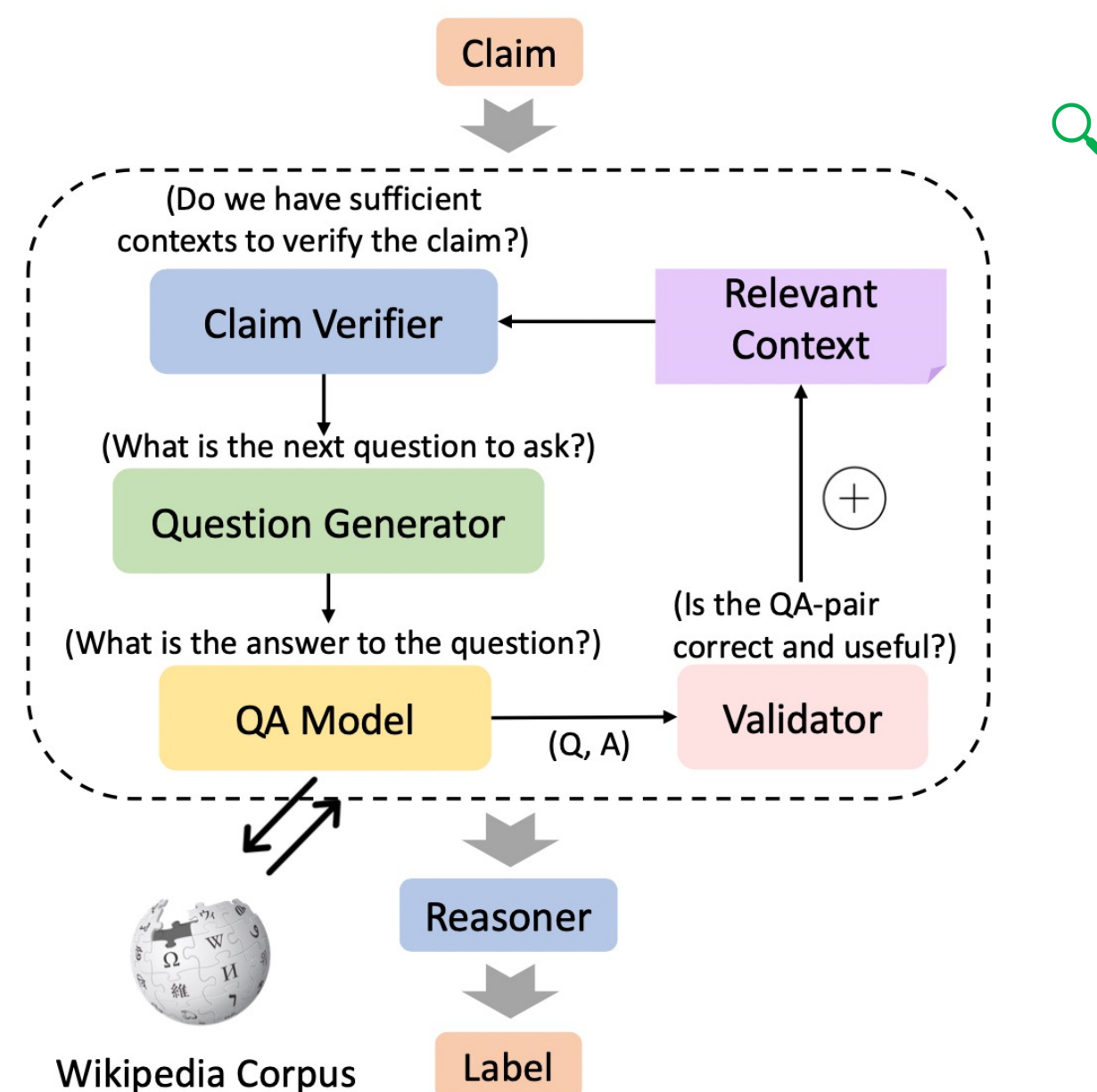
Prediction with rationale:

Lars Onsager won the Nobel prize in 1968. Lars Onsager was born in 1903. He was 65 when he won the Nobel prize. Therefore, the final answer is: False.

Finish

System Architecture

- Claim Verifier \mathcal{D}** : determine the sufficiency of the existing context to validate the claim, i.e., $\mathcal{D}(c, C) \rightarrow \{True, False\}$
- Question Generator \mathcal{Q}** : generate the next question that is necessary for verifying the claim, i.e., $\mathcal{Q}(c, C) \rightarrow q$
- Question-Answering Model \mathcal{A}** : answer the question and provide the supported evidence, i.e., $\mathcal{A}(q) \rightarrow a, e$
- Validator \mathcal{V}** : validate the usefulness of the newly-generated (Q, A) pair based on the existing context and the claim, i.e., $\mathcal{V}(c, \{q, a\}, C) \rightarrow \{True, False\}$
- Reasoner \mathcal{R}** : utilize the relevant context to justify the veracity of the claim and outputs the final label, i.e., $\mathcal{R}(c, C) \rightarrow \{Supported, Refuted\}$



Datasets and Experimental Results

HOVER (Jiang et al., 2020)

- 1,126 two-hop claims
- 1,835 three-hop claims
- 1,039 four-hop claims

FEVEROUS (Aly et al., 2021)

- We selected 2,962 claims that require exclusively textual evidence.

QACHECK has better improvement over the end-to-end models on claims with high reasoning depth.

This indicates that decomposing a complex claim into simpler steps with question-guided reasoning can facilitate more accurate reasoning.

Model	HOVER			FEVEROUS
	2-hop	3-hop	4-hop	
InstructGPT				
- Direct	56.51	51.75	49.68	60.13
- CoT	57.20	53.66	51.83	61.05
Codex	55.57	53.42	45.59	57.85
FLAN-T5	48.27	52.11	51.13	55.16
ProgramFC	54.27	54.18	52.88	59.66
QACheck	55.67	54.67	52.35	59.47

Table 1: Evaluation of F1 scores for different models. The bold text shows the best results for each setting.

Future Works

- The reliance on external APIs tends to prolong the response time of our system.
- We could integrate open-source, locally-run large language models like LLaMA.
- The current scope of our QACheck is confined to evaluating True/False claims.
- Extending QACheck to Not Enough Info (NEI) claims could be a future direction.

Contact & Links

@PanLiangming

liangmingpan@ucsb.edu

