Fact-Checking Complex Claims with Program-Guided Reasoning

Liangming Pan, Xiaobao Wu, Xinyuan Lu, Anh Tuan Luu, William Yang Wang, Min-Yen Kan, Preslav Nakov

ACL 2023

발제자: 김한성

23-08-29

Abstract

- 조금 더 Realistic한 Fact-checking 환경을 풀기 위해서는 복잡한 multi-step reasoning을 풀어야 한다.

- 본 논문에서 제안하는 Program-Guided Fact-Checking(ProgramFC)는 아래와 같은 방법으로 이러한 문제를 풀었다.

- Decompose complex claims into simpler sub-tasks.
  - leverage the in-context learning
  - delegate each sub-task

- 이는 data-efficient하고 설명가능성(explanatory)을 확보한 접근법

- 결과적으로 Challenging한 Fact-checking task (HOVER,FEVEROUS-s)에서 7개 baseline 대비 좋은 성능을 보임

Introduction

Claim : “Both James Cameron and the director of the film Interstellar were born in Canada”

Fact-checking에서 중요한 요소

- Realistic claim은 더 복잡한 구조이다.
- Explanability : 이해와 믿을 수 있는 결과를 위해
- Data efficiency : data == time-consuming, costly, and potentially bias

Introduction

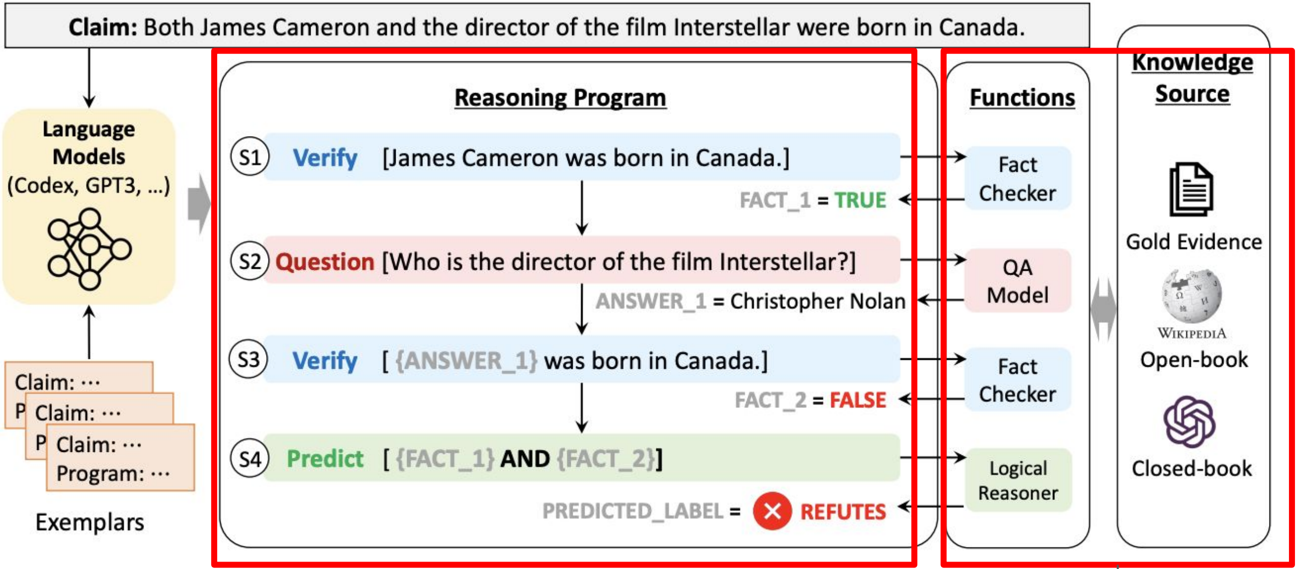

Claim: Both James Cameron and the director of the film Interstellar were born in Canada.
Language Models (Codex, GPT3, ...)
Claim: ···
Claim: ···
Claim: ···
Program: ···
Exemplars
Reasoning Program
S1 Verify [James Cameron was born in Canada.]
FACT_1 = TRUE
S2 Question [Who is the director of the film Interstellar?]
ANSWER_1 = Christopher Nolan
S3 Verify [{ANSWER_1} was born in Canada.]
FACT_2 = FALSE
S4 Predict [{FACT_1} AND {FACT_2}]
PREDICTED_LABEL = ✕ REFUTES
Functions
Fact Checker
QA Model
Fact Checker
Logical Reasoner
Knowledge Source
Gold Evidence
Wikipedia
Open-book
Closed-book

Figure 1: Overview of our PROGRAMFC model, which consists of two modules: (*i*) *Program Generation* generates a reasoning program for the input claim using Codex with in-context learning, and then (*ii*) *Program Execution* sequentially interprets the program by delegating each step to the corresponding sub-task function.

generate reasoning program (prompt)

Sequentially delegate sub-task (allow use external source)

Related Work

Fact-Checking
데이터 유형 2가지 소개

- Human-crafted : FEVER, WikiFactCheck,VitaminC
- Naturall Occuring (Politics or Science): Liar, CheckThat!

Graph-based Model (GEAR, 2019) : 기존 Multi-step reasoning을 위한 접근법

- 좋은 성능을 가지고는 있으나 설명 가능성은 부재되고 있음.

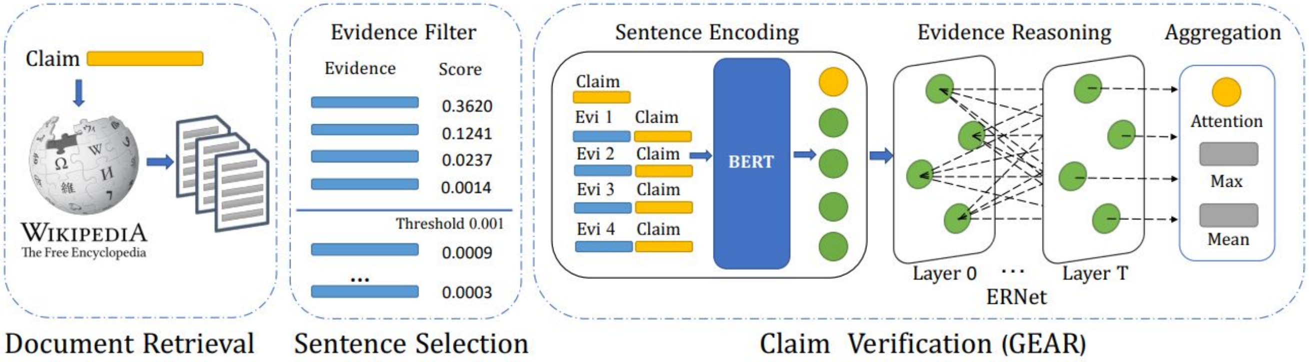
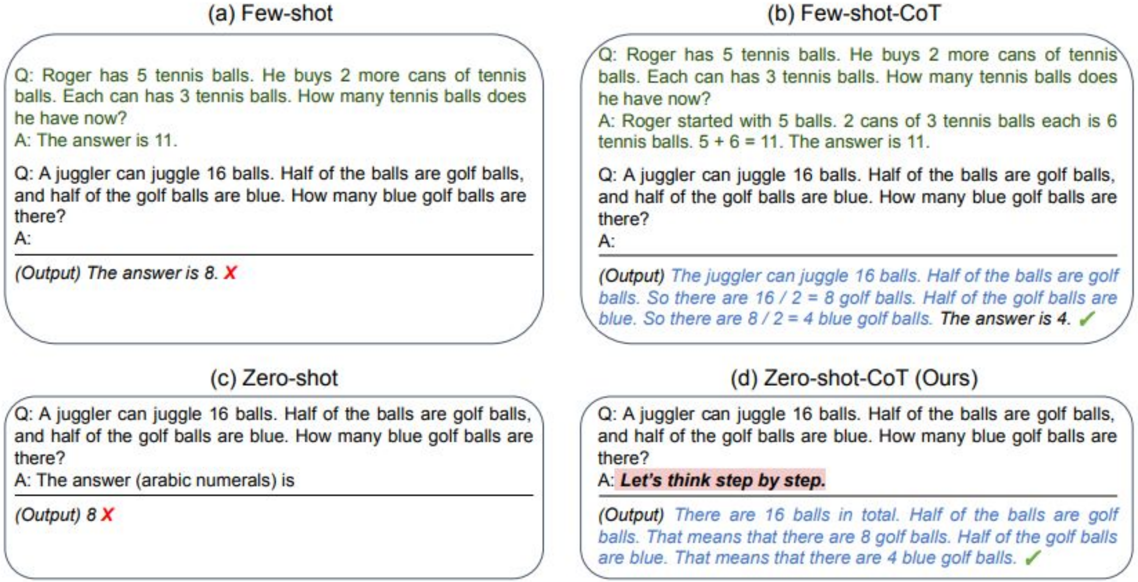


Figure 1: The pipeline of our method. The GEAR framework is illustrated in the claim verification section.

Related Work

Chain-of-Thought Reasoning :

- Reasoning Program은 Chain-of-Thought를 활용하여 복잡한 claim을 단순한 문제의 나열로 분해함.



(a) Few-shot

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?
A: The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A:

(Output) The answer is 8. ✗

(b) Few-shot-CoT

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?
A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A:

(Output) The juggler can juggle 16 balls. Half of the balls are golf balls. So there are 16 / 2 = 8 golf balls. Half of the golf balls are blue. So there are 8 / 2 = 4 blue golf balls. The answer is 4. ✓

(c) Zero-shot

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A: The answer (arabic numerals) is

(Output) 8 ✗

(d) Zero-shot-CoT (Ours)

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A: ***Let's think step by step.***

(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. ✓

ProgramFC

Problem Formulation
F : fact-checking model
C : claim
Y : verdict
K : knowledge source

	evidence	목적
Gold Evidence	K(human-labeled)	기존 세팅 성능 확인
Open-Book setting	K(retrieved)	QA,제안
Closed-Book setting	K(None)	LLM의 지식 확인

ProgramFC

Program-Guided Reasoning

- Program Generation
    - Claim을 Step 단위로 문제를 단순화

$$\mathit{planner}\ P,\ \ \mathit{claim}\ \ C$$
$$P(C) = [S_1, ..., S_n]$$

- Program Execution
    - Step별 Sub-task function 및 인자를 넣어줌

$$S_i \in P$$
$$S_i \ = \ (f_i, A_i, V_i)$$

- Aggregating Reasoning Path
    - CoT의 추론 방법은 다양. 따라서 후보를 N개로 두고 보팅을 진행

$$P = \{P_1, ..., P_N\}$$
$$VOTE(P)$$

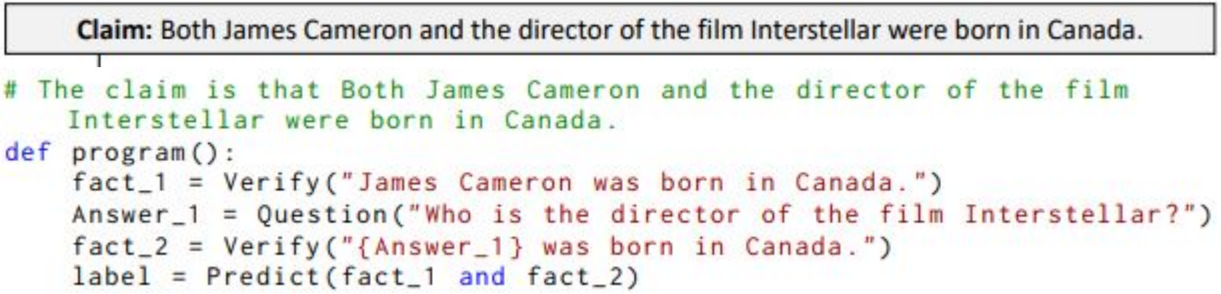
Reasoning Program Generation

Program-Guided Reasoning

- Program Generation
    - Claim을 Step 단위로 문제를 단순화

$$planner\ P,\ \ claim\ C$$
$$P(C) = [S_1, ..., S_n]$$

- Codex를 사용
    - Claim Program pair를 만들도록 함.
    - SQL 또는 Python문법으로 생성하게 만듬
    - 문법 오류를 위해 few-shot generalization을 진행했다고 함.
    - 20-shot in-context example을 주입함.
    - Sampling-based decoding(temperature =0.7)



Claim: Both James Cameron and the director of the film Interstellar were born in Canada.

```
# The claim is that Both James Cameron and the director of the film
    Interstellar were born in Canada.
def program():
    fact_1 = Verify("James Cameron was born in Canada.")
    Answer_1 = Question("Who is the director of the film Interstellar?")
    fact_2 = Verify("{Answer_1} was born in Canada.")
    label = Predict(fact_1 and fact_2)
```

Claim Program pair

Reasoning Program Generation

Codex를 사용

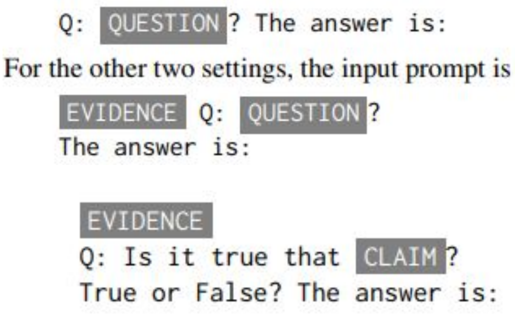
- 문법 오류를 위해 few-shot generalization을 진행했다고 함.

Generate a python-like program that describes the reasoning steps required to verify the claim step-by-step.
You can call three functions in the program:
1. Question() to answer a question;
2. Verify() to verify a simple claim;
3. Predict() to predict the veracity label. Several examples are given as follows.

Sub-Task Functions

- Program Execution
  - Step별 Sub-task function 및 인자를 넣어줌

- [QUESTION]
  - FLAN-T5 : SOTA of many QA benchmarks
- [VERIFY]
  - FLAN-T5
- [PREDICT]
  - Step별 Logical 관계(and, or etc)를 실행함.



```
Q: QUESTION ? The answer is:
```

For the other two settings, the input prompt is

```
EVIDENCE Q: QUESTION ?
The answer is:
```

```
EVIDENCE
Q: Is it true that CLAIM ?
True or False? The answer is:
```

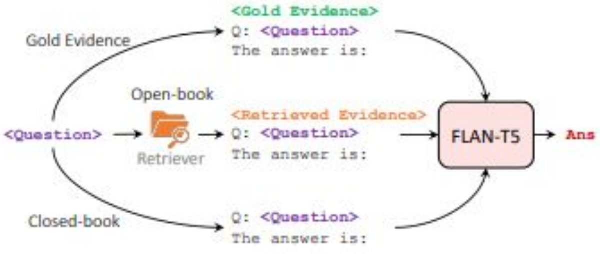
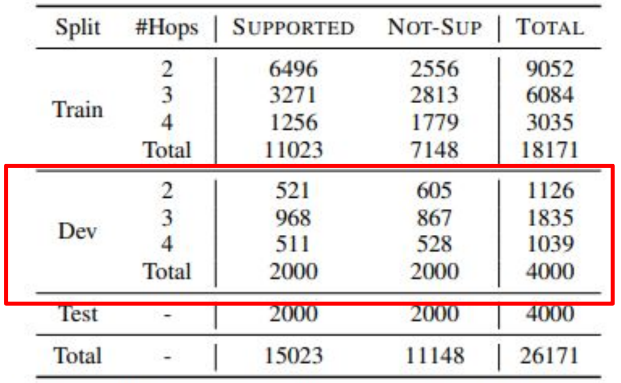


Figure 3: Implementation of the question-answering sub-task function for three different settings.

Experiments

Dataset

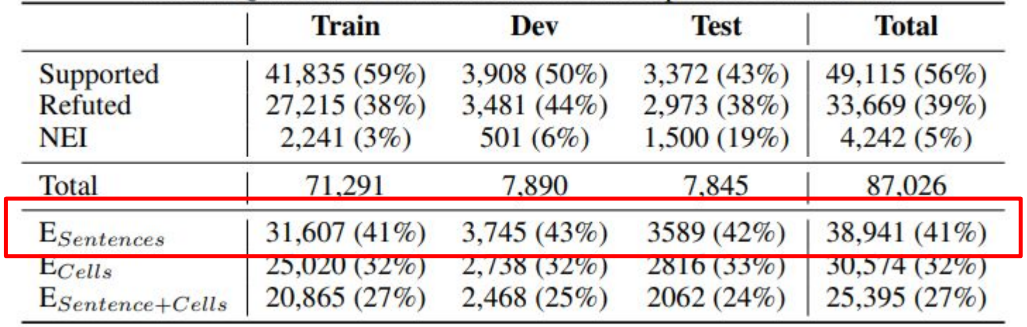
- HOVER, FEVEROUS 모두 Validation으로 추론 함(test set은 오픈되어있지 않음)
- HOVER는 hop의 수를 기준으로 데이터를 분할 (2,3,4-hop으로 구분)
- FEVEROUS는 문장만 가져옴 FEVEROUS-s 로 정의



Split	#Hops	SUPPORTED	NOT-SUP	TOTAL
Train	2	6496	2556	9052
	3	3271	2813	6084
	4	1256	1779	3035
	Total	11023	7148	18171
Dev	2	521	605	1126
	3	968	867	1835
	4	511	528	1039
	Total	2000	2000	4000
Test	-	2000	2000	4000
Total	-	15023	11148	26171

Table 2: The sizes of the Train-Dev-Test split for SUPPORTED and NOT-SUPPORTED classes and different number of hops.

Table 2: Quantitative characteristics of each split in FEVEROUS.



	Train	Dev	Test	Total
Supported	41,835 (59%)	3,908 (50%)	3,372 (43%)	49,115 (56%)
Refuted	27,215 (38%)	3,481 (44%)	2,973 (38%)	33,669 (39%)
NEI	2,241 (3%)	501 (6%)	1,500 (19%)	4,242 (5%)
Total	71,291	7,890	7,845	87,026
$E_{Sentences}$	31,607 (41%)	3,745 (43%)	3589 (42%)	38,941 (41%)
E_{Cells}	25,020 (32%)	2,738 (32%)	2816 (33%)	30,574 (32%)
$E_{Sentence+Cells}$	20,865 (27%)	2,468 (25%)	2062 (24%)	25,395 (27%)

Experiments

Baseline

- pretrained model : BERT-FC, LisT5
- FC/NLI fine-tuned models : RoBERTa-NLI(4 NLI data),DeBERTaV3-NLI(FEVER and 4 NLI data), MULTIVERS(Longformer to FEVER)
- In-context Learning models : FLAN-T5, Codex

Experiments

Few-shot Learning 20 example

- pretrained model : BERT-FC, LisT5 -> fine-tuning

- FC/NLI fine-tuned models : RoBERTa-NLI(4 NLI data),DeBERTaV3-NLI(FEVER and 4 NLI data), MULTIVERS(Longformer to FEVER) -> continuous fine-tuning

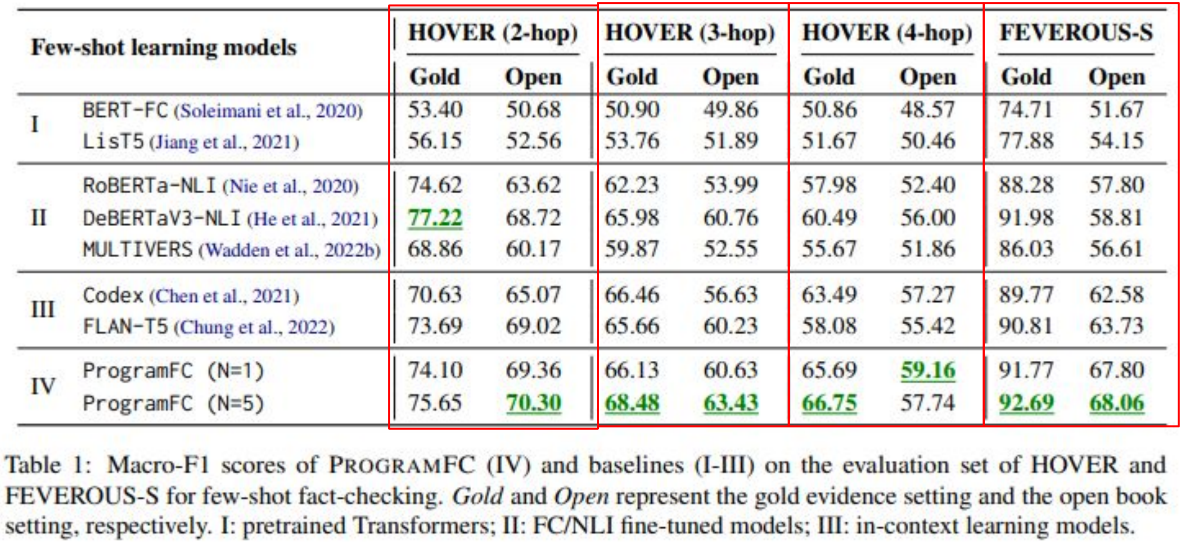
- In-context Learning models : FLAN-T5, Codex -> ingredient for few shot

- Gold evidence : Real Labels
- OpenBook : BM25 top-10사용 (베이스라인, ProgramFC 모두)

Main Results

ProgramFC is more effective on deeper claims

AVG. 10.38% 11.37% 14.77% 8.61%



Few-shot learning models		HOVER (2-hop)		HOVER (3-hop)		HOVER (4-hop)		FEVEROUS-S	
		Gold	Open	Gold	Open	Gold	Open	Gold	Open
I	BERT-FC (Soleimani et al., 2020)	53.40	50.68	50.90	49.86	50.86	48.57	74.71	51.67
	ListT5 (Jiang et al., 2021)	56.15	52.56	53.76	51.89	51.67	50.46	77.88	54.15
II	RoBERTa-NLI (Nie et al., 2020)	74.62	63.62	62.23	53.99	57.98	52.40	88.28	57.80
	DeBERTaV3-NLI (He et al., 2021)	77.22	68.72	65.98	60.76	60.49	56.00	91.98	58.81
	MULTIVERS (Wadden et al., 2022b)	68.86	60.17	59.87	52.55	55.67	51.86	86.03	56.61
III	Codex (Chen et al., 2021)	70.63	65.07	66.46	56.63	63.49	57.27	89.77	62.58
	FLAN-T5 (Chung et al., 2022)	73.69	69.02	65.66	60.23	58.08	55.42	90.81	63.73
IV	ProgramFC (N=1)	74.10	69.36	66.13	60.63	65.69	59.16	91.77	67.80
	ProgramFC (N=5)	75.65	70.30	68.48	63.43	66.75	57.74	92.69	68.06

Table 1: Macro-F1 scores of PROGRAMFC (IV) and baselines (I-III) on the evaluation set of HOVER and FEVEROUS-S for few-shot fact-checking. *Gold* and *Open* represent the gold evidence setting and the open book setting, respectively. I: pretrained Transformers; II: FC/NLI fine-tuned models; III: in-context learning models.

또한 DeBERTaV3-NLI는 2-hop에서 좋은 성능을 보이고 있지만 Multi-hop으로 진행될 수록 성능이 저하되는 것을 알 수 있음.

N은 보팅의 수

How Does the Reasoning Program Help?

Model size efficient

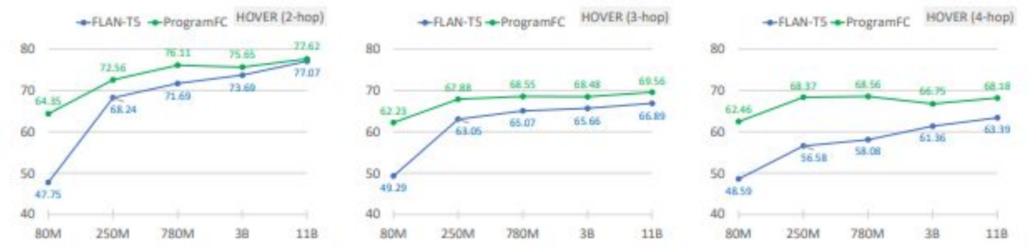


Figure 4: F1 score for fact-checking with gold evidence using FLAN-T5 (blue line) and PROGRAMFC (green line) for language models of increasing sizes: FLAN-T5-small (80M), FLAN-T5-base (250M), FLAN-large (780M), FLAN-T5-XL (3B), and FLAN-T5-XXL (11B) on HOVER 2-hop (left), 3-hop (middle), and 4-hop (right).

Retrieval efficient

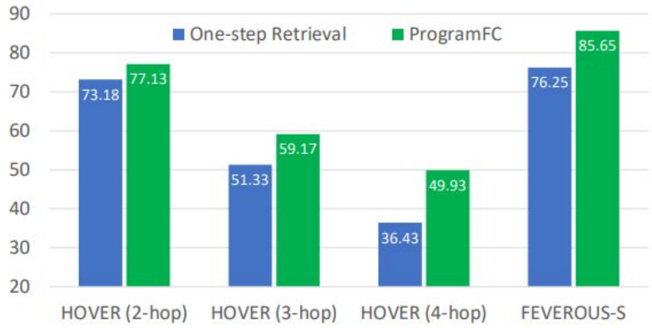
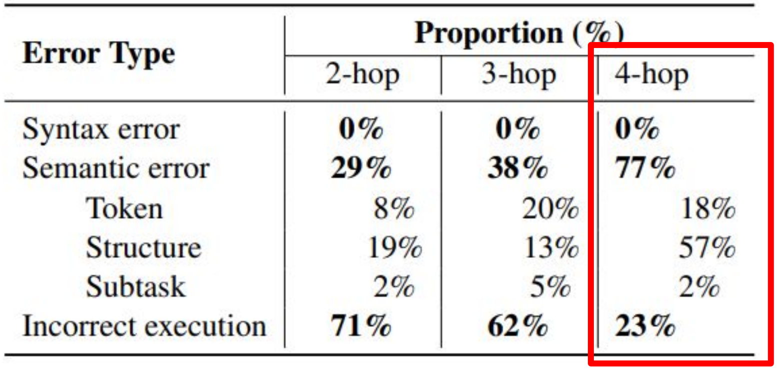


Figure 5: Retrieval recall@10 for the one-step retrieval and the iterative retrieval in PROGRAMFC.

gold paragraph기준

Interpretability of Reasoning Programs



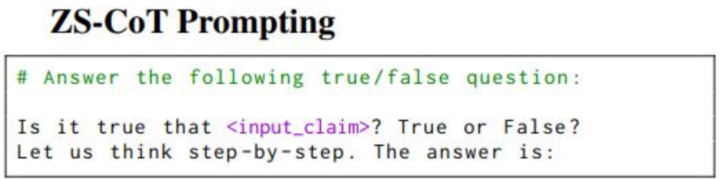
Error Type	**Proportion (%)**		
	2-hop	3-hop	4-hop
Syntax error	**0%**	**0%**	**0%**
Semantic error	**29%**	**38%**	**77%**
Token	8%	20%	18%
Structure	19%	13%	57%
Subtask	2%	5%	2%
Incorrect execution	**71%**	**62%**	**23%**

Table 2: Reasoning program evaluation for incorrectly-predicted examples from each hop length in HOVER.

Token : incorrect or missing : arg/var
Structure : incorrect program structure
Subtask : incorrect subtask call

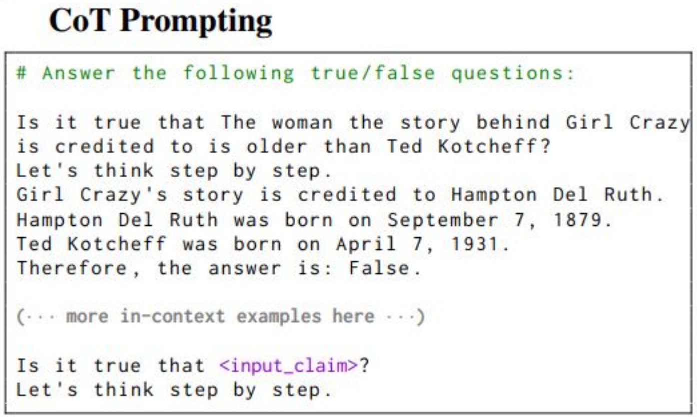
Closed-Book Fact-Checking

LLM 성능을 보기 위함



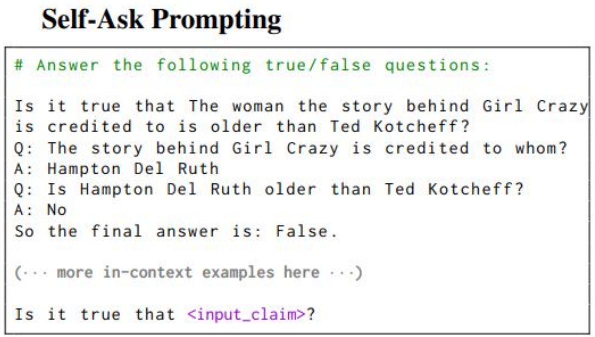
ZS-CoT Prompting

```
# Answer the following true/false question:

Is it true that <input_claim>? True or False?
Let us think step-by-step. The answer is:
```



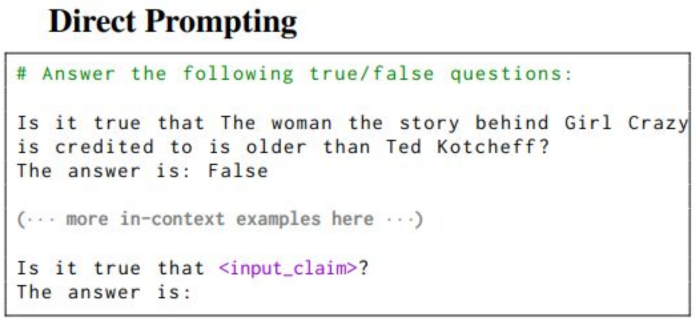
CoT Prompting

```
# Answer the following true/false questions:

Is it true that The woman the story behind Girl Crazy
is credited to is older than Ted Kotcheff?
Let's think step by step.
Girl Crazy's story is credited to Hampton Del Ruth.
Hampton Del Ruth was born on September 7, 1879.
Ted Kotcheff was born on April 7, 1931.
Therefore, the answer is: False.

(··· more in-context examples here ···)

Is it true that <input_claim>?
Let's think step by step.
```



Self-Ask Prompting

```
# Answer the following true/false questions:

Is it true that The woman the story behind Girl Crazy
is credited to is older than Ted Kotcheff?
Q: The story behind Girl Crazy is credited to whom?
A: Hampton Del Ruth
Q: Is Hampton Del Ruth older than Ted Kotcheff?
A: No
So the final answer is: False.

(··· more in-context examples here ···)

Is it true that <input_claim>?
```



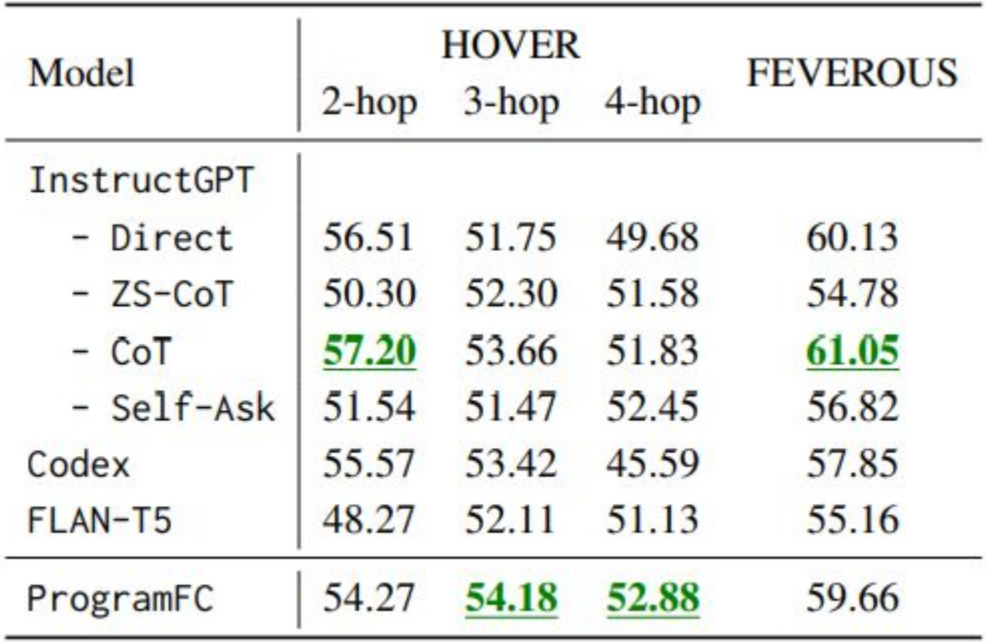
Direct Prompting

```
# Answer the following true/false questions:

Is it true that The woman the story behind Girl Crazy
is credited to is older than Ted Kotcheff?
The answer is: False

(··· more in-context examples here ···)

Is it true that <input_claim>?
The answer is:
```

Closed-Book Fact-Checking

LLM 성능을 보기 위함

ProgramFC는 Codex 여기에 Sub-task solver가 추가된 형태



Model	HOVER			FEVEROUS
	2-hop	3-hop	4-hop	
InstructGPT				
- Direct	56.51	51.75	49.68	60.13
- ZS-CoT	50.30	52.30	51.58	54.78
- CoT	57.20	53.66	51.83	61.05
- Self-Ask	51.54	51.47	52.45	56.82
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

Table 3: Closed-book setting: macro-F1 scores for PROGRAMFC and for the baselines.

결론

- CodeX를 활용한 program generator와 FLAN-T5를 활용한 sub-task solver가 결합된 ProgramFC를 제안
- 설명 가능한 end-to-end모델을 제안
- No additional training
- 모델 크기와 Retrieval에 효과적

고민해볼 점

1. Multi-step Reasoning과 Single-step Reasoning을 모두 잘하는 방법은?

2. 더 Realistic한 factcheck 환경이란 무엇일까?

   a. factcheck한 환경을 정의하기 위해 일련의 시나리오가 필요할까?
   (특정 케이스를 가정한 저널 등 그렇다면 그러한 곳이 중요한 곳은 어디일까?)

   b. Realistic한 factchek에 필요한 요소는 무엇이 있을까?
   신뢰 가능한 데이터, 데이터 형태(이미지,텍스트, 오디오 등), 언어 및 문화

3. LLM을 활용한 NLP연구의 발전방향이란 무엇일까?
   a. 본 논문에서는 COT를 활용하여 Multi-hop reasoning task를 풀었다고 생각함.

감사합니다.