LLMs with knowledge base

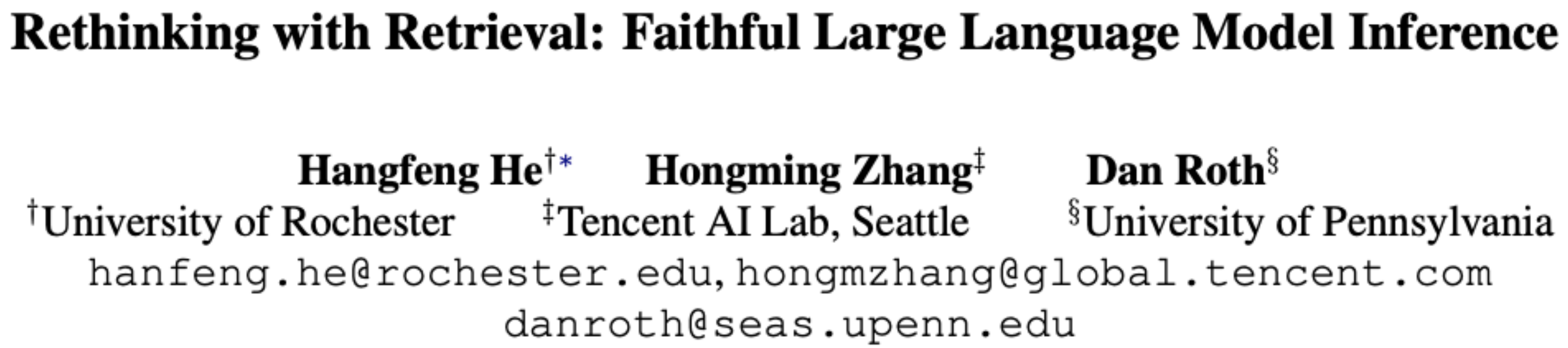
손수현

2023.08.03

Natural Language Processing & Artificial Intelligence

고려대학교 KOREA UNIVERSITY

- Rethinking with Retrieval: Faithful Large Language Model Inference
- Chain of Knowledge: A Framework for Grounding Large Language Models with Structured Knowledge Bases
- Think-on-Graph: Deep and Responsible Reasoning of Large Language Model with Knowledge Graph



Rethinking with Retrieval: Faithful Large Language Model Inference

Hangfeng He[†*] Hongming Zhang[‡] Dan Roth[§]
[†]University of Rochester [‡]Tencent AI Lab, Seattle [§]University of Pennsylvania
hanfeng.he@rochester.edu, hongmzhang@global.tencent.com
danroth@seas.upenn.edu

Abstract

- LLM이 다양한 task에서 좋은 성능을 보이고 있지만 model이 가지고 있는 knowledge에 대한 한계

  : incomplete, out-of-date, or incorrect
- External knowledge를 사용해서 LLMs을 assist하는 연구들이 있지만 additional training or fine-tuning 필요
- Post-processing approach 제안 – rethinking with retrieval (RR)

  : relevant external knowledge를 retrieve해서 사용

  : additional training, fine-tuning X

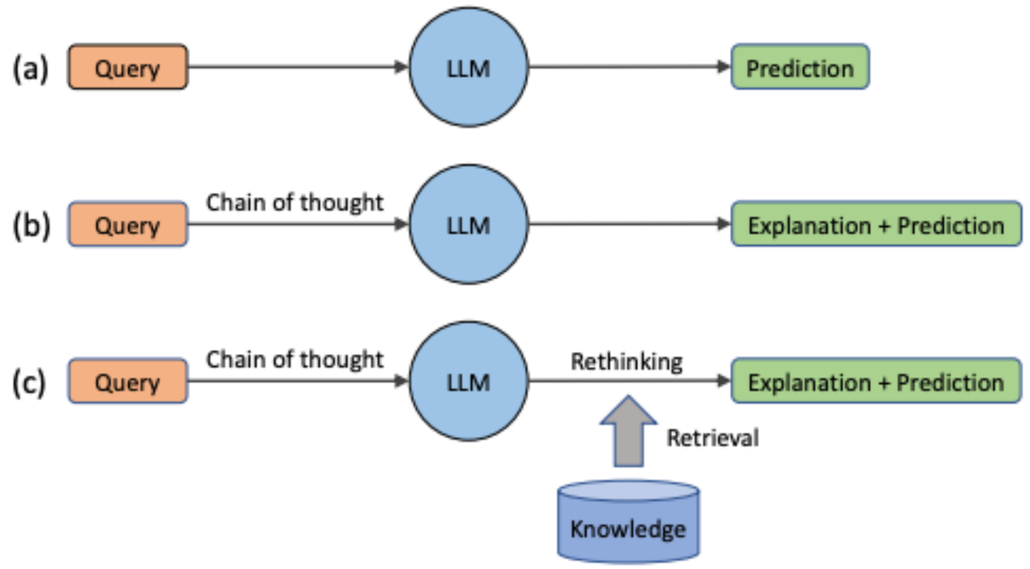
  : LLM의 input length 제한받지 않음
- Commonsense reasoning, temporal reasoning, tabular reasoning task에 대해서

  : more faithful explanations을 통해서 LLM의 성능을 올림

Introduction

- LLM에 존재할 수 있는 incomplete, out-of-date, incorrect knowledge를 위해 external knowledge source사용
- Chain-of-Thought (CoT) prompting method를 통해서 여러 reasoning path를 생성

  : 이후 각 path를 사용해서 knowledge를 retrieve하여 사용



Rethinking with Retrieval

- Overview
  - given query Q에 대해서 chain-of-thought prompting을 통해서 set of reasoning path $R_1, R_2, \dots, R_N$ 생성
  - 각 R_i는 prediction결과인 P_i와 explanation E_i로 구성되어 있음
  - KB에서 각 reasoning path의 explanation과 적합한 knowledge $K_1, \dots, K_M$을 retrieve
  - knowledge에 가장 적합한 prediction P를 최종 prediction으로

Rethinking with Retrieval

- Chain-of-thought prompting

  - step-by-step reasoning example을 prompt에 포함시켜 줌으로써 reasoning process를 잘 capture

  - **question** : "*Did Aristotle use a laptop?*"

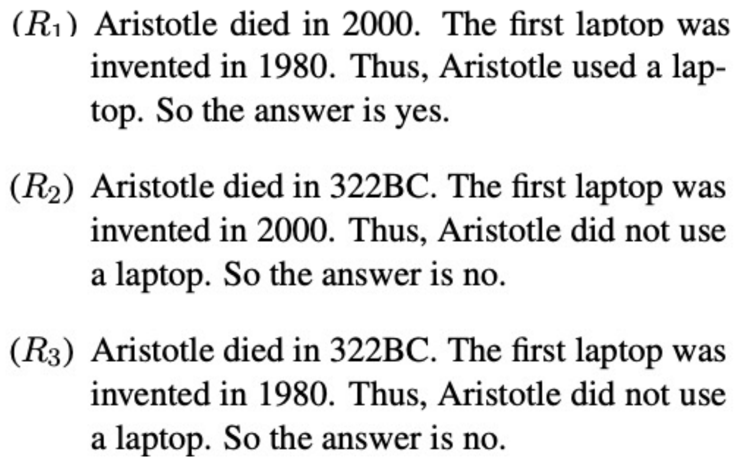
    answer: "No" (X)

    "Aristotle died in 322 BC. The first laptop was invented in 1980. Thus, Aristotle did not use a laptop. So the answer is no." (O)

→ CoT를 통해서 explanation E와 prediction P를 얻음

Rethinking with Retrieval

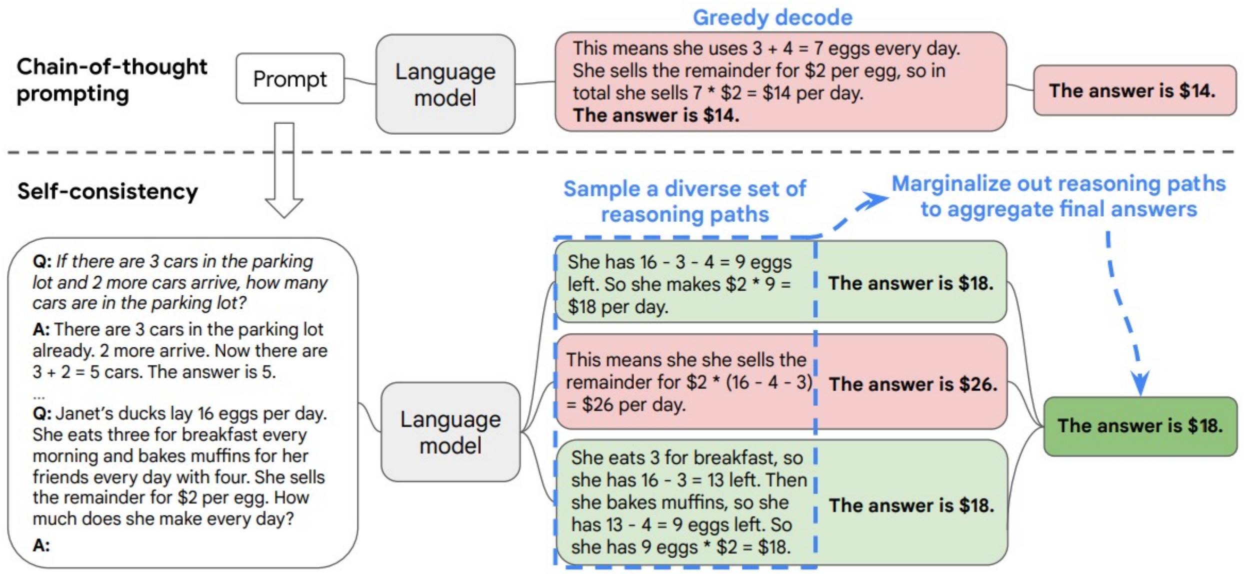
- Sampling diverse reasoning paths
  - greedy path만 고려하는 것이 아닌 diverse set of reasoning path를 sampling하여 사용
  - **question** : "*Did Aristotle use a laptop?*"



(R_1) Aristotle died in 2000. The first laptop was invented in 1980. Thus, Aristotle used a laptop. So the answer is yes.

(R_2) Aristotle died in 322BC. The first laptop was invented in 2000. Thus, Aristotle did not use a laptop. So the answer is no.

(R_3) Aristotle died in 322BC. The first laptop was invented in 1980. Thus, Aristotle did not use a laptop. So the answer is no.



This means she uses 3 + 4 = 7 eggs every day. She sells the remainder for $2 per egg, so in total she sells 7 * $2 = $14 per day. **The answer is $14.**

Q: *If there are 3 cars in the parking lot and 2 more cars arrive, how many cars are in the parking lot?*
A: There are 3 cars in the parking lot already. 2 more arrive. Now there are 3 + 2 = 5 cars. The answer is 5.
...
Q: Janet's ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins for her friends every day with four. She sells the remainder for $2 per egg. How much does she make every day?
A:

She has 16 - 3 - 4 = 9 eggs left. So she makes $2 * 9 = $18 per day.

This means she she sells the remainder for $2 * (16 - 4 - 3) = $26 per day.

She eats 3 for breakfast, so she has 16 - 3 = 13 left. Then she bakes muffins, so she has 13 - 4 = 9 eggs left. So she has 9 eggs * $2 = $18.

Rethinking with Retrieval

- Knowledge Retrieval
  - Wikipedia와 같은 external knowledge base에서 reasoning path에 적절한 knowledge retrieval해옴

- Faithful inference
  - retrieval해온 knowledge와 reasoning path의 faithfulness를 보는 부분
  - retrieved knowledge와 비교해서 가장 faithful한 prediction P를 identify하는 부분

Experiments

- Baselines
  - Zero-shot/few-shot prompting, Chain-of-thought prompting, Self-consistency
- three complex reasoning tasks

1) commonsense reasoning

: strategyQA dataset

– implicit reasoning strategies를 요구하는 QA dataset

- question, yes/no answer, decomposition, evidence paragraph, supporting facts

: external KB - Wikipedia

* Knowledge Retrieval - BM25로 top 10 paragraph retrieve

* Faithful Inference

- MPNet model을 사용해서 paragraph과 explanation간의 cosine similarity score 측정

- NLI 모델을 사용해서 entailment, contradiction score 측정

→ 총 3가지 점수를 기반으로 final prediction

Experiments

2) temporal reasoning

: capability to understand and process information that involves time-based concepts and sequences

: TempQuestion dataset

– implicit temporal questions만 사용

: external KB – Wikidata

* Knowledge Retrieval

– entity linking을 통해서 explanation의 각 entity에 해당하는 wikidata page와 연결

- triple을 sentence로 convert할 수 있는 template 사용하여 knowledge sentence 생성

* Faithful Inference

- 이전과 동일

Experiments

3) tabular reasoning

: premise table이 주어졌을 때, given hypothesis가 table의 정보에 entailment or contradiction or neutral

: INFOTABS dataset

: external KB – WordNet, ConceptNet

* Knowledge Retrieval

– premise와 hypothesis를 연결하는 triple을 찾아서 sentence로 convert

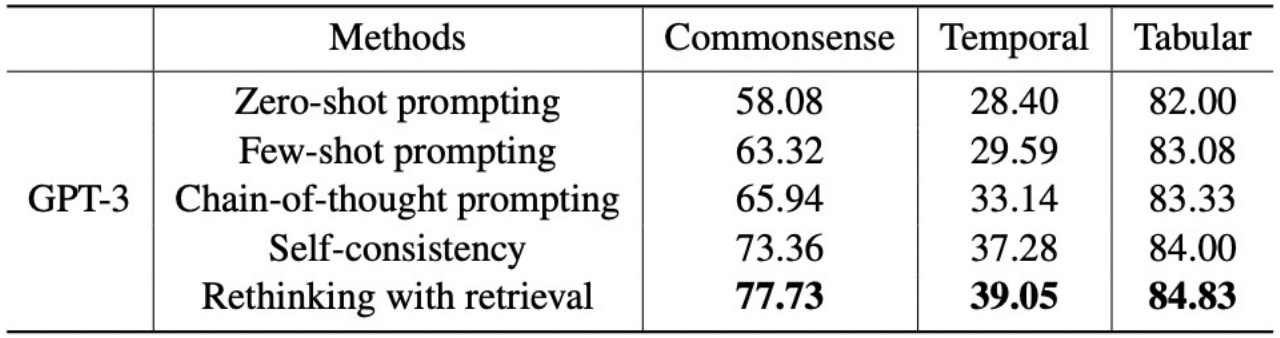
* Faithful Inference

- 이전과 동일

Experiments

- Evaluation
  - GPT3 text-davinci-001
  - zero-shot, few-shot, and chain-of-thought prompting → temperature 0으로 고정
  - self-consistency and rethinking with retrieval → temperature 0.7로 randomly sample 10 outputs

- Results

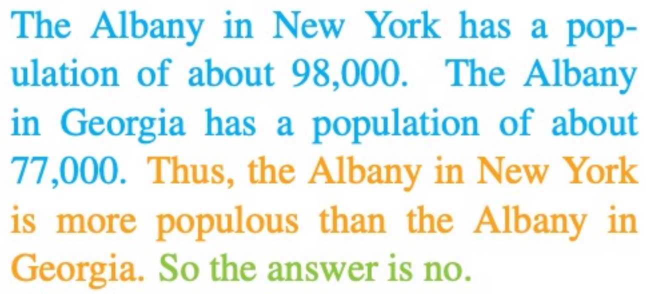


	Methods	Commonsense	Temporal	Tabular
GPT-3	Zero-shot prompting	58.08	28.40	82.00
	Few-shot prompting	63.32	29.59	83.08
	Chain-of-thought prompting	65.94	33.14	83.33
	Self-consistency	73.36	37.28	84.00
	Rethinking with retrieval	**77.73**	**39.05**	**84.83**

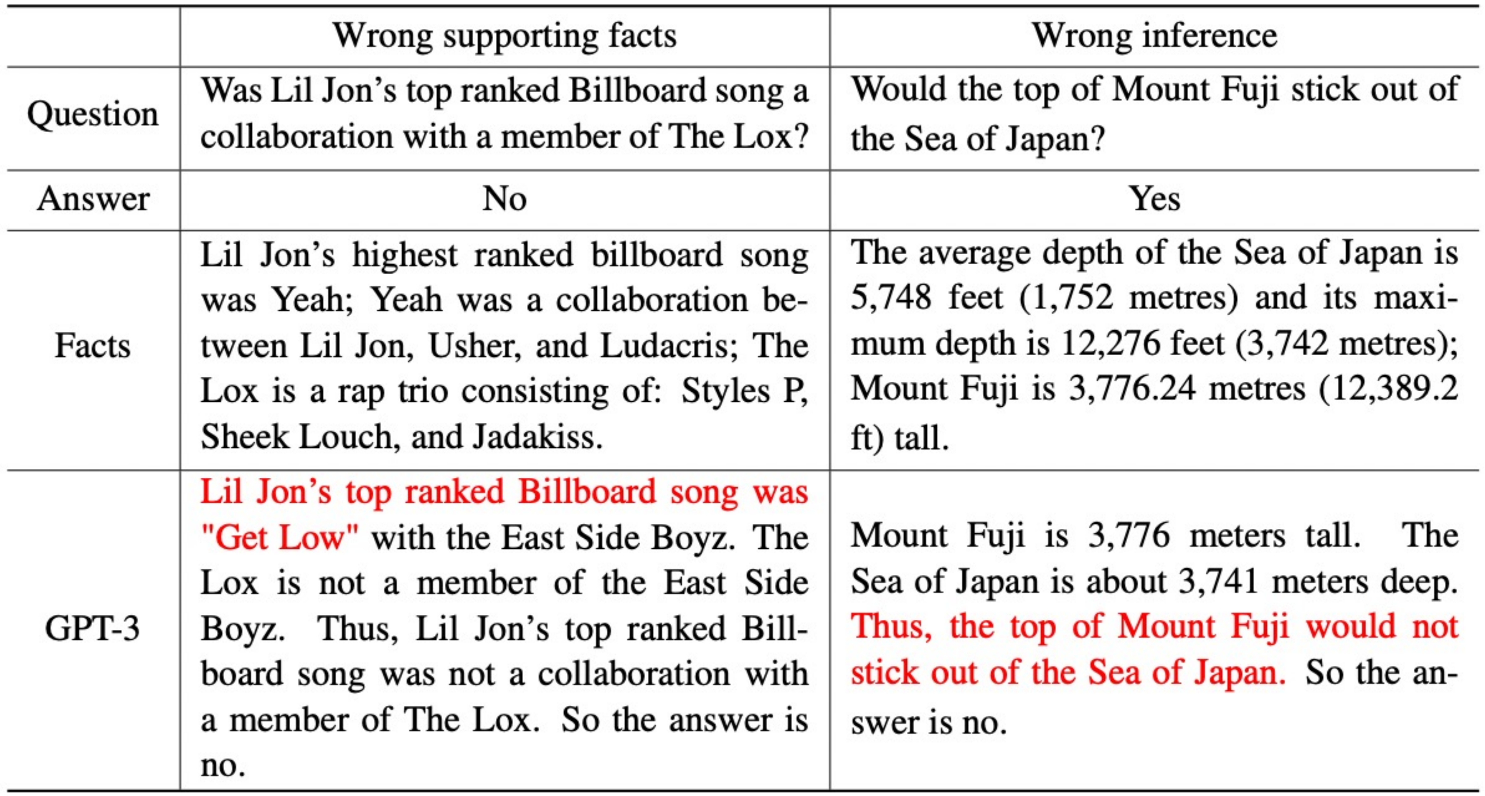
Table 1: Performance of different methods using GPT-3 on three reasoning tasks.

Analysis – 1. Limitations of LLMs in Reasoning

- StrategyQA에 대해서 CoT prompting 방법론 결과 분석
- 대부분 reasonable explanations 기반으로 correct prediction
- 일정한 패턴을 보임
    1) supporting facts
    2) chaining arguments
    3) prediction
- 잘못 예측하는 경우 → wrong supporting facts, wrong inference


The Albany in New York has a population of about 98,000. The Albany in Georgia has a population of about 77,000. Thus, the Albany in New York is more populous than the Albany in Georgia. So the answer is no.

Analysis – 1. Limitations of LLMs in Reasoning



	Wrong supporting facts	Wrong inference
Question	Was Lil Jon's top ranked Billboard song a collaboration with a member of The Lox?	Would the top of Mount Fuji stick out of the Sea of Japan?
Answer	No	Yes
Facts	Lil Jon's highest ranked billboard song was Yeah; Yeah was a collaboration between Lil Jon, Usher, and Ludacris; The Lox is a rap trio consisting of: Styles P, Sheek Louch, and Jadakiss.	The average depth of the Sea of Japan is 5,748 feet (1,752 metres) and its maximum depth is 12,276 feet (3,742 metres); Mount Fuji is 3,776.24 metres (12,389.2 ft) tall.
GPT-3	Lil Jon's top ranked Billboard song was "Get Low" with the East Side Boyz. The Lox is not a member of the East Side Boyz. Thus, Lil Jon's top ranked Billboard song was not a collaboration with a member of The Lox. So the answer is no.	Mount Fuji is 3,776 meters tall. The Sea of Japan is about 3,741 meters deep. Thus, the top of Mount Fuji would not stick out of the Sea of Japan. So the answer is no.

Table 2: Examples of incorrect outputs from GPT-3 with CoT prompting.

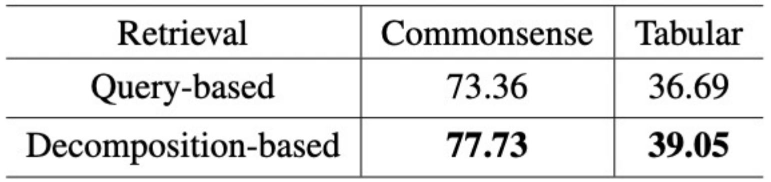
- Incorrect supporting fact를 생성하지만 결과적으로 prediction은 정답인 경우
- Correct supporting fact를 생성했지만 inference를 잘못해서 결과적으로 prediction을 틀린 경우

Analysis – 2. Ablation Study

- **Importance of decomposition-based retrieval**

  : decomposed reasoning step이 성능에 어떤 영향을 끼치는지

  : original query로 knowledge retrieval한 결과로 prediction



Retrieval	Commonsense	Tabular
Query-based	73.36	36.69
Decomposition-based	**77.73**	**39.05**

Table 3: Comparison of query-based and decomposition-based retrieval on commonsense and tabular reasoning.

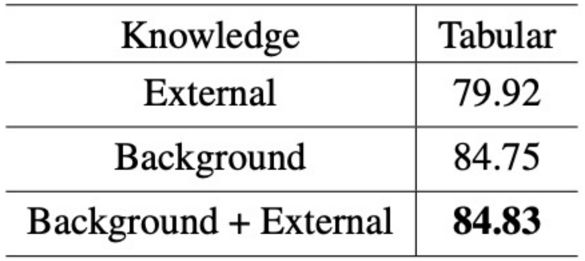
- **The impact of different types of knowledge**

  : tabular reasoning할 때 external knowledge + background knowledge (table)를 사용

  : 둘 다 사용했을 때 가장 좋은 성능을 보임

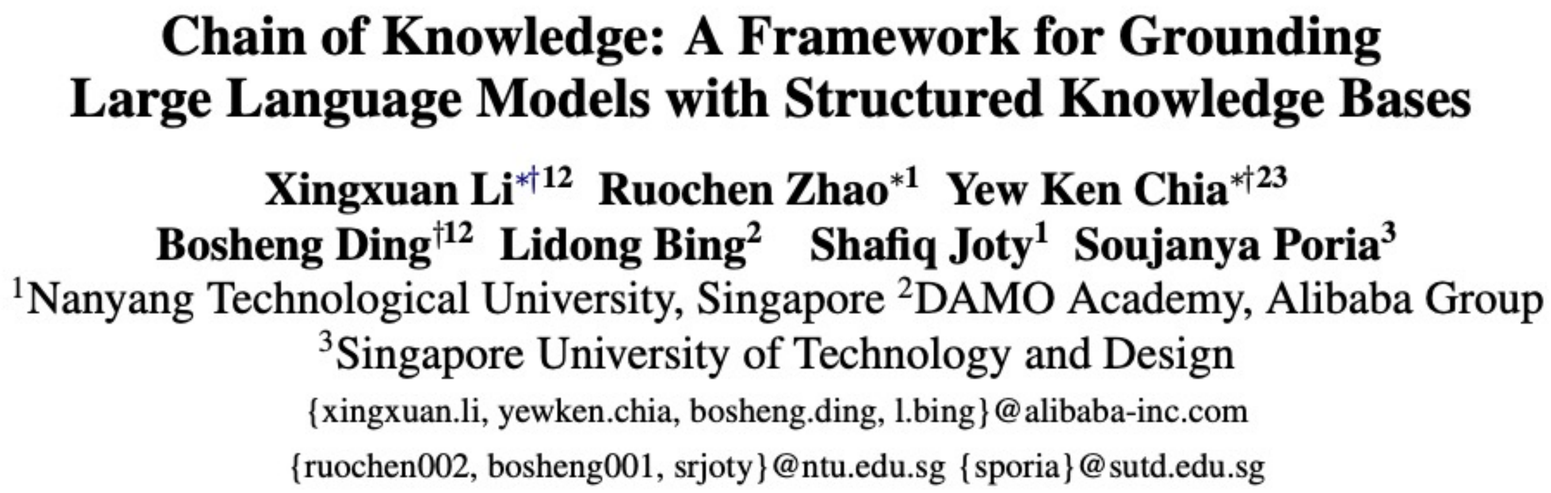
  : background knowledge가 성능에 큰 영향을 끼침

  → RR 방법론이 background knowledge를 효율적으로 사용하고 있다



Knowledge	Tabular
External	79.92
Background	84.75
Background + External	**84.83**

Table 4: Performance of RR with different types of knowledge on tabular reasoning: external only, background only, and a combination of both. External knowledge refers to WordNet and ConceptNet, while background knowledge refers to the tables.



Chain of Knowledge: A Framework for Grounding Large Language Models with Structured Knowledge Bases

Xingxuan Li*†[12] Ruochen Zhao*[1] Yew Ken Chia*†[23]
Bosheng Ding†[12] Lidong Bing[2] Shafiq Joty[1] Soujanya Poria[3]
[1]Nanyang Technological University, Singapore [2]DAMO Academy, Alibaba Group
[3]Singapore University of Technology and Design
{xingxuan.li, yewken.chia, bosheng.ding, l.bing}@alibaba-inc.com
{ruochen002, bosheng001, srjoty}@ntu.edu.sg {sporia}@sutd.edu.sg

Abstract

- Factual correctness를 향상시키고 hallucination은 줄이는 framework인 Chain of Knowledge (CoK) 제안
- Query generator model을 통해서 효율적으로 LLM이 knowledge base를 사용할 수 있게끔

  - query generator model은 frozen LLM과 구분되어 있기 때문에 다양한 knowledge resource와 model에 적용이 가능함
- CoK framework가 knowledge-intensive task에서 LLM의 factual correctness를 향상시킴

Introduction

- LLM의 remarkable achievement에도 불구하고 여전히 significant challenge가 존재함
- Main problem: hallucination

  - difficult to update or control the knowledge
- 이러한 문제를 해결하기 위해서 external knowledge source에서 검색해온 정보를 함께 사용하는 approach도 존재

  - 이런 방법론은 faithfulness 문제 존재

  - knowledge-intensive task에서는 multiple knowledge facts를 reasoning 해야함

→ Hallucination은 줄이고 knowledge-intensive task의 성능은 향상시킬 수 있는
retrieved knowledge base를 바탕으로 step-by-step reasoning 방법론인 Chain of Knowledge (CoK) 제안

→ irrelevant or conflicting information이 존재할 수 있는 web document가 아닌 knowledge base 사용

Introduction

- Human이 복잡한 문제를 풀 때와 유사하게 LLM한테 chain of sub-questions를 생성하게끔 instruct
- Relevant facts를 retrieve해와서 각 sub-question을 풀고
- LLM이 retrieved facts를 사용해서 original question에 answer

- Knowledge base와 LLM의 structure 차이

  → query generator를 통해서 knowledge base queries 생성할 수 있게끔

Methodology

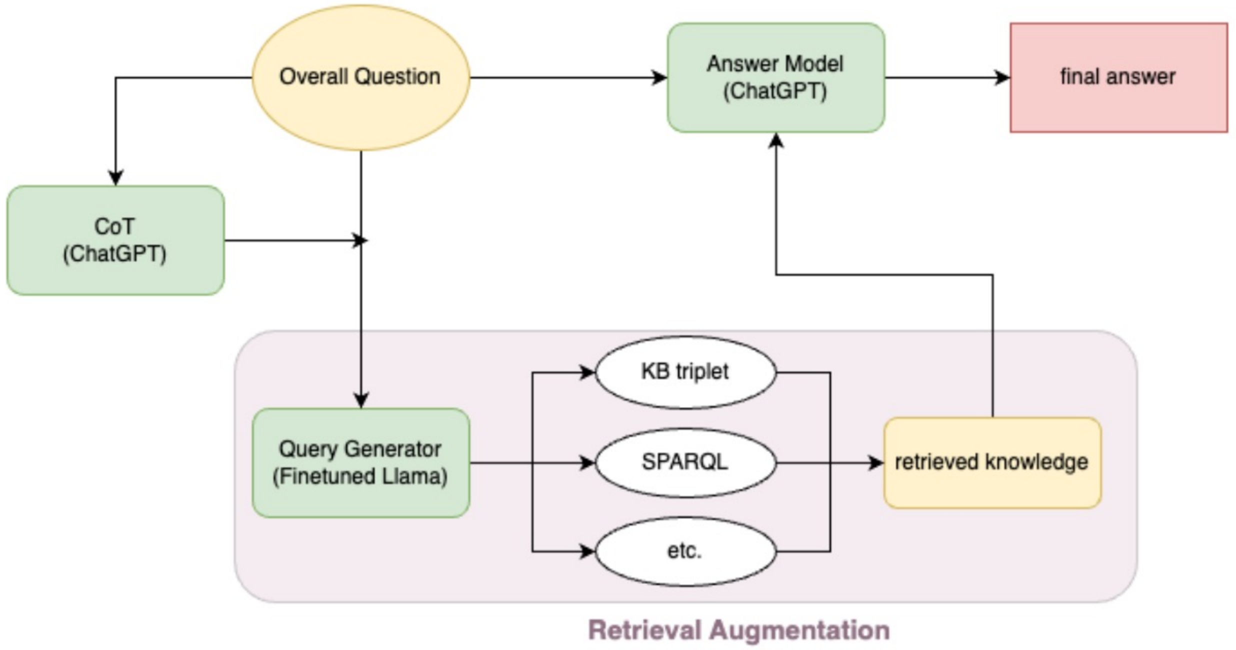
- Two key challenges를 해결

1) Retrieval

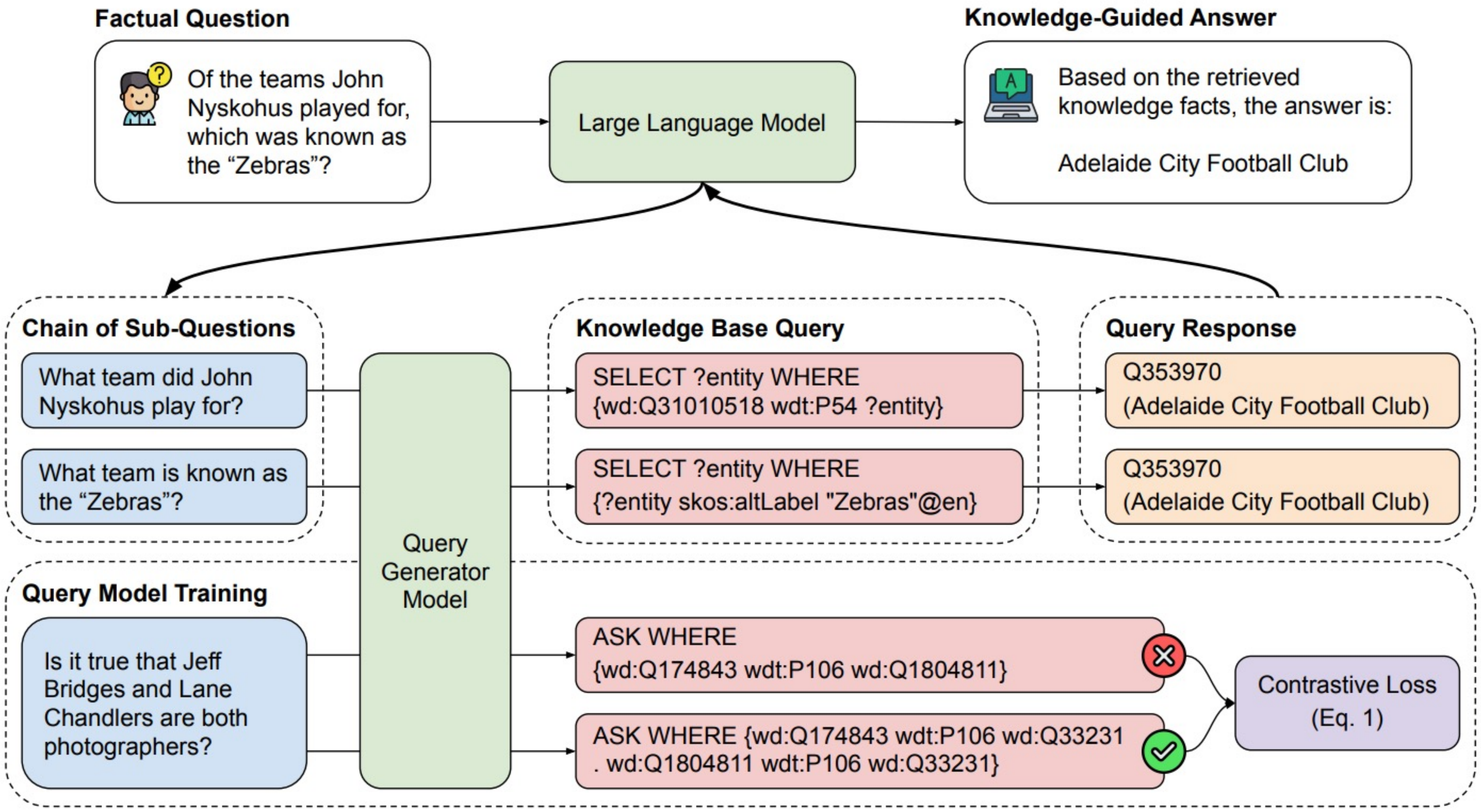
  - 자연어로 된 question에서 structured KB queries를 생성해야 함

2) Synthesis

  - retrieved knowledge를 잘 반영하여 answer를 생성해야 함



Methodology



Factual Question
Of the teams John Nyskohus played for, which was known as the "Zebras"?
Large Language Model
Knowledge-Guided Answer
Based on the retrieved knowledge facts, the answer is:
Adelaide City Football Club
Chain of Sub-Questions
What team did John Nyskohus play for?
What team is known as the "Zebras"?
Query Generator Model
Knowledge Base Query
SELECT ?entity WHERE {wd:Q31010518 wdt:P54 ?entity}
SELECT ?entity WHERE {?entity skos:altLabel "Zebras"@en}
Query Response
Q353970 (Adelaide City Football Club)
Q353970 (Adelaide City Football Club)
Query Model Training
Is it true that Jeff Bridges and Lane Chandlers are both photographers?
ASK WHERE {wd:Q174843 wdt:P106 wd:Q1804811}
ASK WHERE {wd:Q174843 wdt:P106 wd:Q33231 . wd:Q1804811 wdt:P106 wd:Q33231}
Contrastive Loss (Eq. 1)

Figure 2: Our proposed framework with the query model training process.

Methodology – Querying the Knowledge Base

- 자연어 형태의 question을 KB triple (simpler task) or SPARQL (complex question)과 같은 structured query로 translate
- KB triples

  : FewRel, HyperRed (natural sentences - corresponding KB triples)
- SPARQL

  : LC-quad, KQA-pro (natural sentences - corresponding SPARQL queries)
- **Contrastive Learning**
  - incorrect query를 생성시켜서 contrastive learning 방식으로 query generator학습시킴

    ### Instruction
    Generate a correct SPARQL query that returns the answer of the following question.
    Generate four incorrect SPARQL queries of different types.

Question-Answering Experiments

- Datasets

  : knowledge-intensive task인 fact verification, complex reasoning

  - **FEVER**

    : Wikipedia의 evidence paragraph를 기반으로 “SUPPORTS”, “REFUTES”, or “NOT ENOUGH INFO” label 로 예측하는

    : simpler

  - **Adversarial HotpotQA (AdvHotpotQA)**

    : multi-hop question answering dataset

    : complex → text-davinci-003

Question-Answering Experiments

- Baselines

- Standard prompting (Standard)
- Chain-of-Thought prompting (CoT)
- CoT with self-consistency (CoT-SC)

  : 여러 reasoning trajectories를 sampling하고 그 중에서 final answer와 가장 consistent한 답변을 select
- Calibrator

  : LLM의 explanation이 prediction과 entail하지 않는 문제를 post-hoc method를 적용해서 성능을 향상시킨 모델
- ReAct

  : external knowledge를 사용하는 기존의 모델
- Verify-and-Edit (VE)

  : post-editing reasoning chains을 통해서 factuality를 향상시킨 모델

  : external knowledge를 사용

Results and Analysis

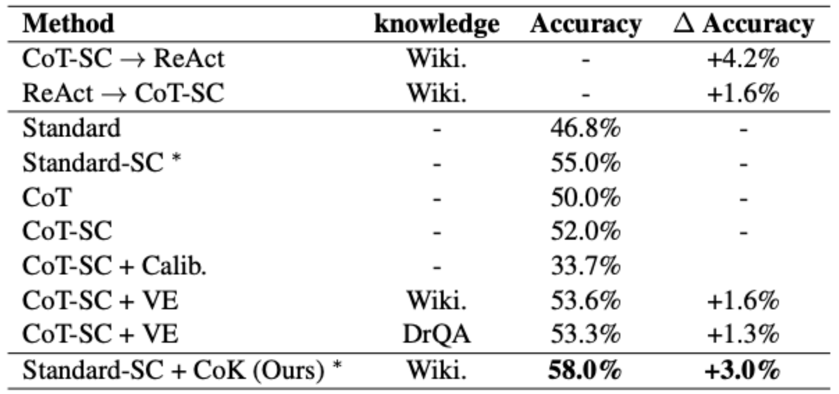
- Results of FEVER

- Single sentence에 대한 fact verification
- Simple task이기 때문에 CoT 수행 X

  : sub-question으로 나누지 않고 바로 CoK

- GPT-3 기반에서 비교했을 때 CoK를 하였을 때 성능이 향상됨

PaLM기반 (



Method	knowledge	Accuracy	Δ Accuracy
CoT-SC → ReAct	Wiki.	-	+4.2%
ReAct → CoT-SC	Wiki.	-	+1.6%
Standard	-	46.8%	-
Standard-SC *	-	55.0%	-
CoT	-	50.0%	-
CoT-SC	-	52.0%	-
CoT-SC + Calib.	-	33.7%	
CoT-SC + VE	Wiki.	53.6%	+1.6%
CoT-SC + VE	DrQA	53.3%	+1.3%
Standard-SC + CoK (Ours) *	Wiki.	**58.0%**	**+3.0%**

Table 3: Results on **Fever** dataset. ΔAccuracy represents the improvement on Accuracy from the CoT-SC baseline. The top two rows uses the PaLM model and the rest uses the GPT-3 davinci-003 model. Rows labeled with $*$ are tentative results run on a small subset of the original test set.

Results and Analysis

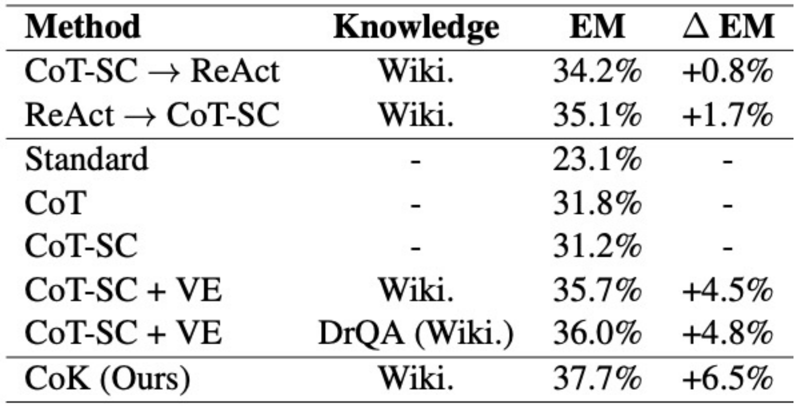
- Results of AdvHotpotQA

- CoT를 통해서 sub question으로 나누고 각각을 통해 CoK한 결과
- Self-consistency (SC)가 이 task에서는 성능 하락
- VE 방법론이 어떤 knowledge를 가져와서 쓰냐에 상관없이 성능 향상
- 제안하는 방법론인 CoK가 가장 큰 성능향상을 보임

→ KB retrieval 방법론의 effectiveness

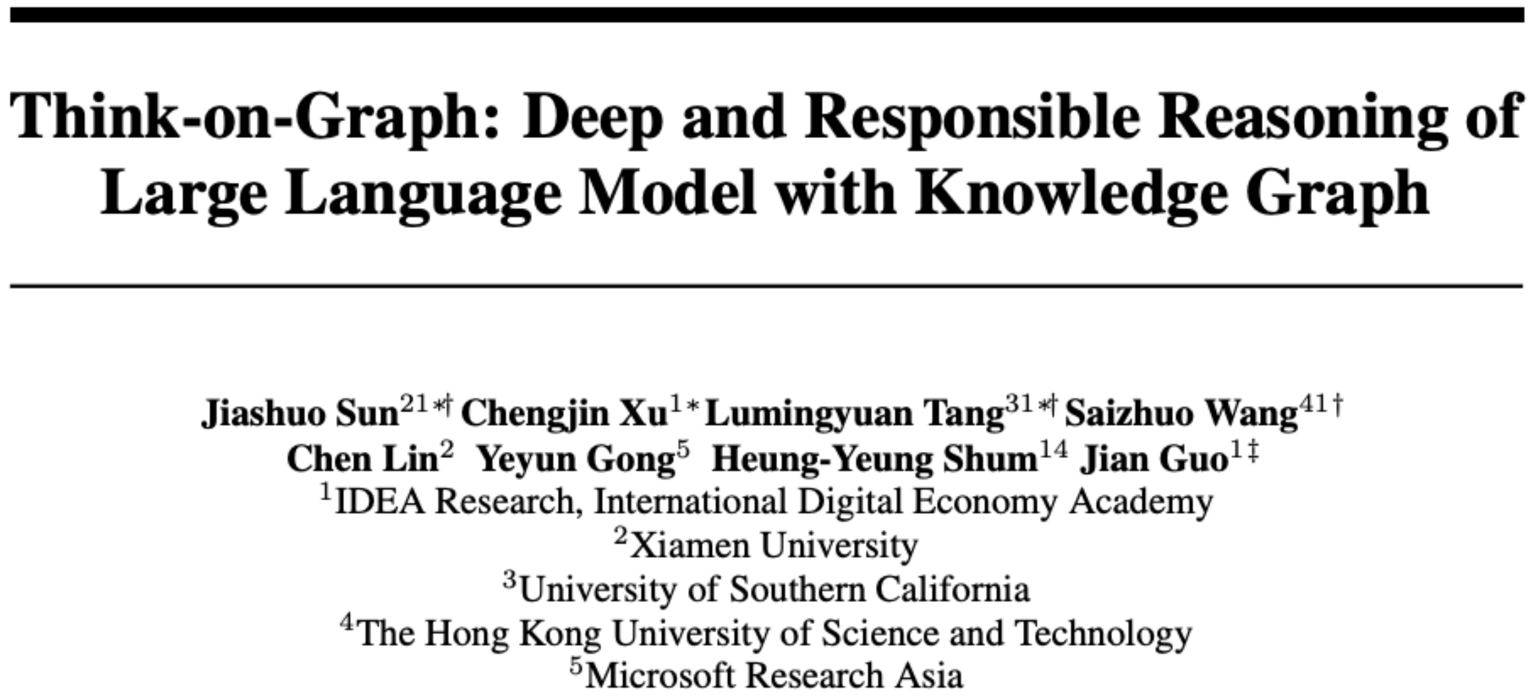
→ retrieval때 잘못된 정보를 가져올 수 있지만, contrastive learning 방식으로 학습시킨 query generator에 의해 정확한 query 생성 및 retrieval이 가능했을 것 이다

PaLM기반



Method	Knowledge	EM	Δ EM
CoT-SC → ReAct	Wiki.	34.2%	+0.8%
ReAct → CoT-SC	Wiki.	35.1%	+1.7%
Standard	-	23.1%	-
CoT	-	31.8%	-
CoT-SC	-	31.2%	-
CoT-SC + VE	Wiki.	35.7%	+4.5%
CoT-SC + VE	DrQA (Wiki.)	36.0%	+4.8%
CoK (Ours)	Wiki.	37.7%	+6.5%

Table 4: Experimental results on AdvHotpotQA.



Think-on-Graph: Deep and Responsible Reasoning of Large Language Model with Knowledge Graph

Jiashuo Sun[21*†] **Chengjin Xu**[1*] **Lumingyuan Tang**[31*†] **Saizhuo Wang**[41†]
Chen Lin[2] **Yeyun Gong**[5] **Heung-Yeung Shum**[14] **Jian Guo**[1‡]
[1]IDEA Research, International Digital Economy Academy
[2]Xiamen University
[3]University of Southern California
[4]The Hong Kong University of Science and Technology
[5]Microsoft Research Asia

Abstract

- LLM이 다양한 task에서 significant strides를 이루었지만 complex reasoning에 어려움이 있음
  - knowledge traceability, timeliness, and accuracy
- **Think-on-Graph (ToG) framework**를 통해서 responsible reasoning에 대한 LLMs 능력을 향상시키고자 함
  - 주어진 question과 관련있는 entities를 identify
  - external knowledge database에서 related triple을 retrieve
  - sequentially connected triplets로 multiple reasoning path
  - 충분한 정보가 수집될 때까지 or maximum depth까지 반복

Introduction

- LLM이 NLP의 다양한 task에서 remarkable performance를 보임
- 하지만, knowledge-intensive task에서는 한계가 있음
  - traceability, timeliness, accuracy of knowledge
  - reasoning 없이 hallucination or toxic text를 생성할 수 있음
- 이를 위해서, external knowledge base를 사용하는 approach가 있었음
  - 복잡한 question이 주어졌을 때 limitation이 존재함
  1. input length의 제한
  2. external knowledge를 intergrate하는 방식은 LLM자체의 "deep" reasoning capability를 간과하는

- 최근 연구들에서
  → intermediate reasoning step을 통해서 LLM의 성능을 향상시킬 수 있음
  → task-specific example을 통해서 complex multi-hop reasoning 가능하게 함

Introduction

- Think-on-Graph (ToG) 제안

  - factual knowledge를 사용하여 LLM의 step-by-step thinking을 drive하는 방법론

  - input question에서 topic entity 추출

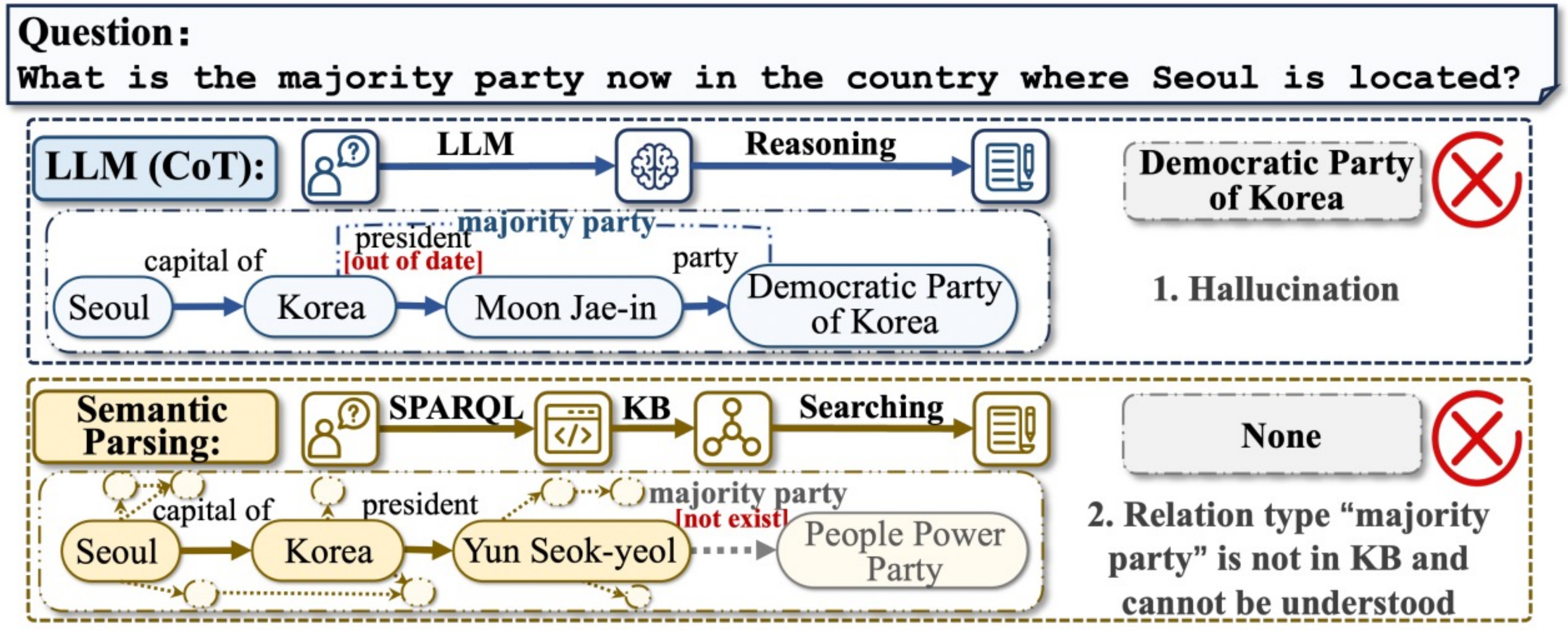
  → LLM으로 external KB exploration and reasoning을 통해서 relevant triple을 retrieve

  - sequentially connected triples로 구성된 multiple high-probability reasoning path 생성

  - factual reasoning capability 향상을 통해서

  hallucination issue를 mitigate하고 response precision을 향상시킴

Introduction



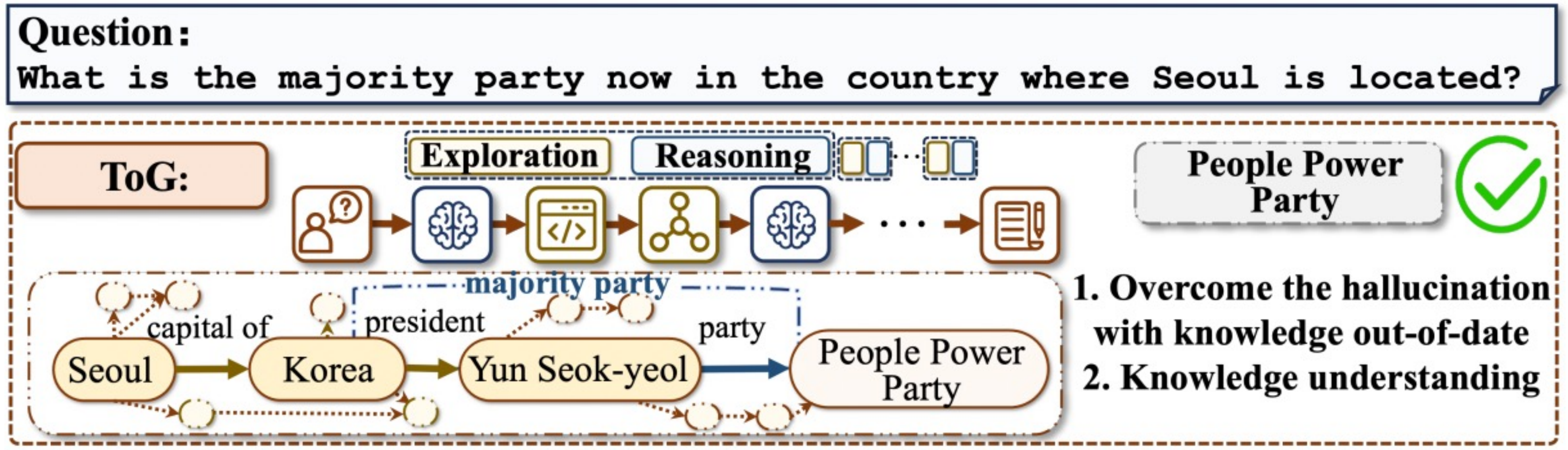
- Chain of thought reasoning

  : knowledge out-of-date (hallucination)

- Semantic parsing QA

  : knowledge graph안에 "majority party"라는 relation이 없기 때문에 entity linking이 불가능

Introduction


Question:
What is the majority party now in the country where Seoul is located?
ToG:
Exploration
Reasoning
People Power Party
1. Overcome the hallucination with knowledge out-of-date
2. Knowledge understanding
majority party
capital of
president
party
Seoul
Korea
Yun Seok-yeol
People Power Party

- Think-on-Graph

  : exploration과 reasoning을 통해서 LLM의 interpretive capacity를 활용

  : multi-hop queries를 효과적으로 처리하는 reasoning pathway 생성

- complex multi-hop reasoning question-answering tasks
- Extra training cost없이도 ToG가 기존의 방법론들의 성능를 향상시키는 결과

Introduction

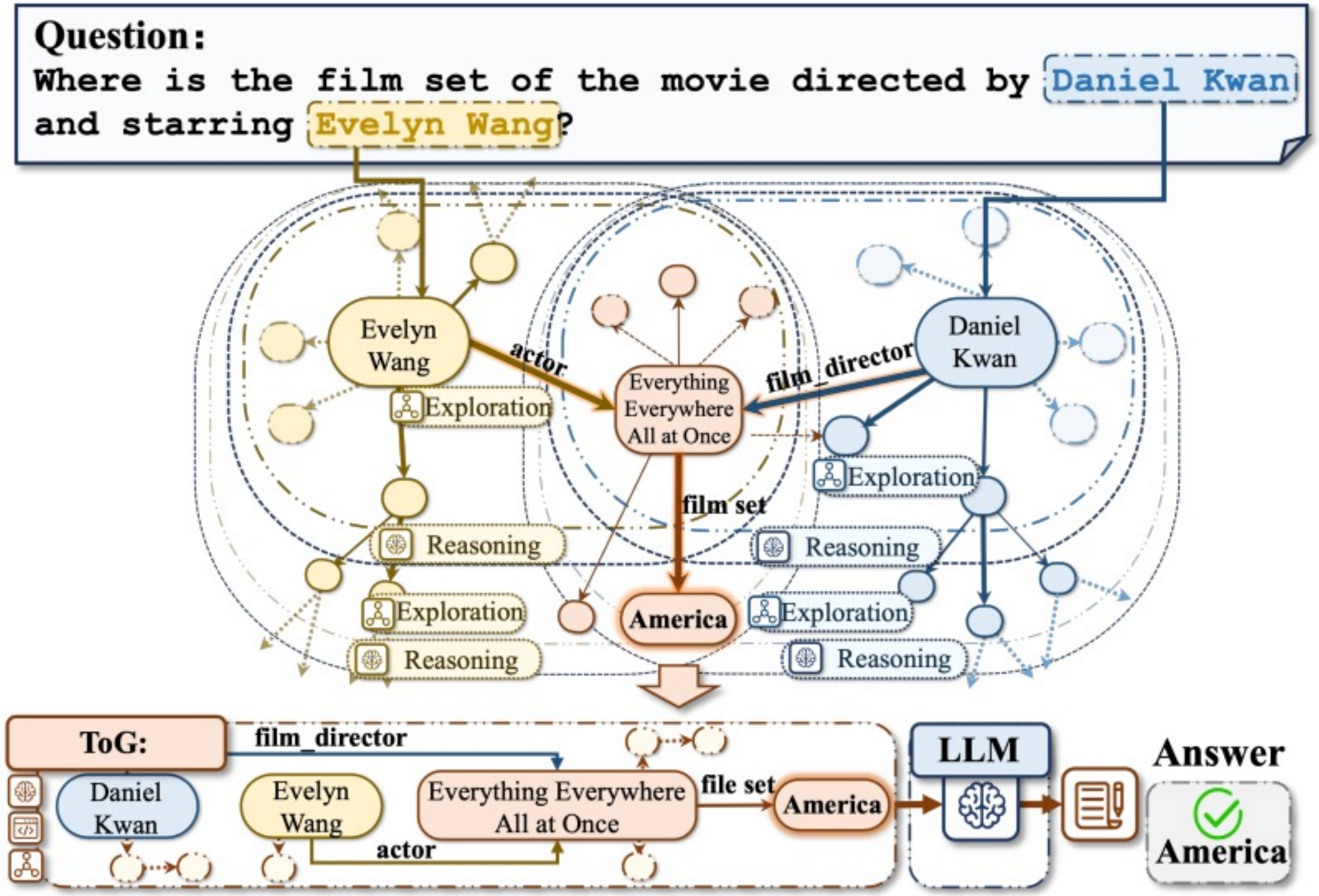


Figure 2: The overall framework of our proposed method.

Methods

- ToG: a novel paradigm for graph search

  - LLM이 query의 entity를 기반으로 multiple possible reasoning path를 explore하도록 prompt 주는 방식

- top-N reasoning paths $p = \{x, T_i\}_{i=1}^{N}$ for question x
- Each path consisting of several triples $T_i = \{(e_s, r_j, e_o)\}_{j=1}^{m}$, e_s: subject entities, e_o: object entities, r_j = relation

- ToG는 크게 2가지 approach로 구성

  : entity-based ToG, relation-based ToG
- 각각은 3 steps

  1) entity acquisition

  2) exploration

  3) reasoning

Methods – entity-based ToG

1) Entity Acquisition

: LLM한테 question에서 entities를 extract하게 prompt

: 각 엔티티 $E_{init} = \{e_1, e_2, \dots, e_n\}$의 contribution score인 $S_{init} = \{s_1, s_2, \dots, s_n\}$를 얻음

$s_i \in [0,1]$ for all $i \in \{1, 2, \dots, n\}$, with $\sum_{i=1}^{n} s_i = 1$

→ question을 sub-problem으로 쪼개지 않고 entity를 중심으로 접근하는 방식이 기존의 step-by-step thinking을 시도한 논문과 다름

2) Exploration

: relevant top-N triples를 identify하기 때문에 매우 중요한 phase

: Breadth-First Search

: two stages로 구성 - relation exploration and entity exploration

Methods – entity-based ToG

2-1) Relation Exploration

- Relation candidate set을 생성하는 부분
- LLM이 자동으로 Search와 Prune과정으로 구성된 search iteration을 수행하게끔

• **Search**

: current entity set E_c 의 각 엔티티와 연결된 모든 relation $R_{cand} = \{R_s \cup R_o\}$를 search, R : relation edge set

(첫 Iteration에서 E_c 는 E_{init})

: 고정된 query를 사용해서 search하기 때문에 training cost X

• **Prune**

: 찾은 candidate set R_{cand} 에 대해서 관련없는 edge들은 날리고 top-N edge만 남기는 과정

: LLM한테 현재 entity set, question을 기반으로 R_{cand} 을 prune하고 top-N relation만 유지시키도록

: update and normalize score S_c

Methods – entity-based ToG

2-2) Entity Exploration

- LLM이 자동으로 Search와 Prune과정으로 구성된 search iteration을 수행하게끔

- **Search**

  : entity set E_c와 이 entity들에 해당하는 relation set R_c에 대해서 candidate entity set E_{cand}

- **Prune**

  : LLM한테 현재 question을 기반으로 E_{cand} 을 prune하고 top-N entities만 유지시키도록

  : update and normalize score S_c

Methods – entity-based ToG

- 이 과정을 통해서 구축한 E_c, R_c, E_N을 기반으로 reasoning path P인 triple set $\{(e_s^i, r^i, e_o^i)\}_{i=1}^{N}$을 구함
- 이 reasoning path P에 대해서 각 triple이 적합한지 아닌지 reasoning

3) Reasoning

- LLM한테 reasoning path가 답변을 생성하는데 적절한지 아닌지 evaluate하라고 prompt
- LLM이 적절하다고 평가하면 (positive)
  : 현재 reasoning path를 기반으로 답변 생성
- 적절하지 않다고 평가하면 (negative)
  : 2) Exploration과 3) Reasoning을 다시 수행

- Positive 결과가 나올때까지 or maximum search depth까지 반복

Methods – relation-based ToG

- Entity를 기반으로 reasoning을 하게 되면

  : entity의 literal information이 항상 정확한 것은 아님

  : entity "name"이 missing되어 있는 knowledge graph에서는 불가능함

- LLM의 reasoning capability를 통해 각 chain별로 다른 candidate set을 사용해서 답변을 생성하게 됨

  1. entity를 explore하는 과정이 없기 때문에 cost를 줄이면서 reasoning 속도를 향상시킬 수 있음
  2. semantic information에 focus하여 reasoning을 수행하기 때문에 더 높은 정확도

Methods – relation-based ToG

- Relation Exploration
- **Search**

  : current entity set E_c 의 각 엔티티와 연결된 모든 relation $R_{cand} = \{R_s \cup R_o\}$를 search, R : relation edge set

  (첫 Iteration에서 E_c 는 E_{init})

  : 고정된 query를 사용해서 search하기 때문에 training cost X
- **Prune**

  : 찾은 candidate set R_{cand} 에 대해서 관련없는 edge들은 날리고 top-N edge만 남기는 과정

  : LLM한테 현재 entity set, question을 기반으로 R_{cand} 을 prune하고 top-N relation만 유지시키도록

  : update and normalize score S_c

Experiments

- Datasets and Evaluation Metrics

  : knowledge-intensive task

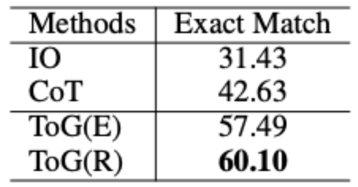
  : Complex Web Questions Talmor and Berant dataset (CWQ dataset)
- Baselines - ChatGPT

  : standard prompting (IO prompt)

  : chain of thought prompting (CoT prompt) – 6 in-context exemplars
- Temperature 0
- topN, D_max = 3
- External knowledge source = Freebase

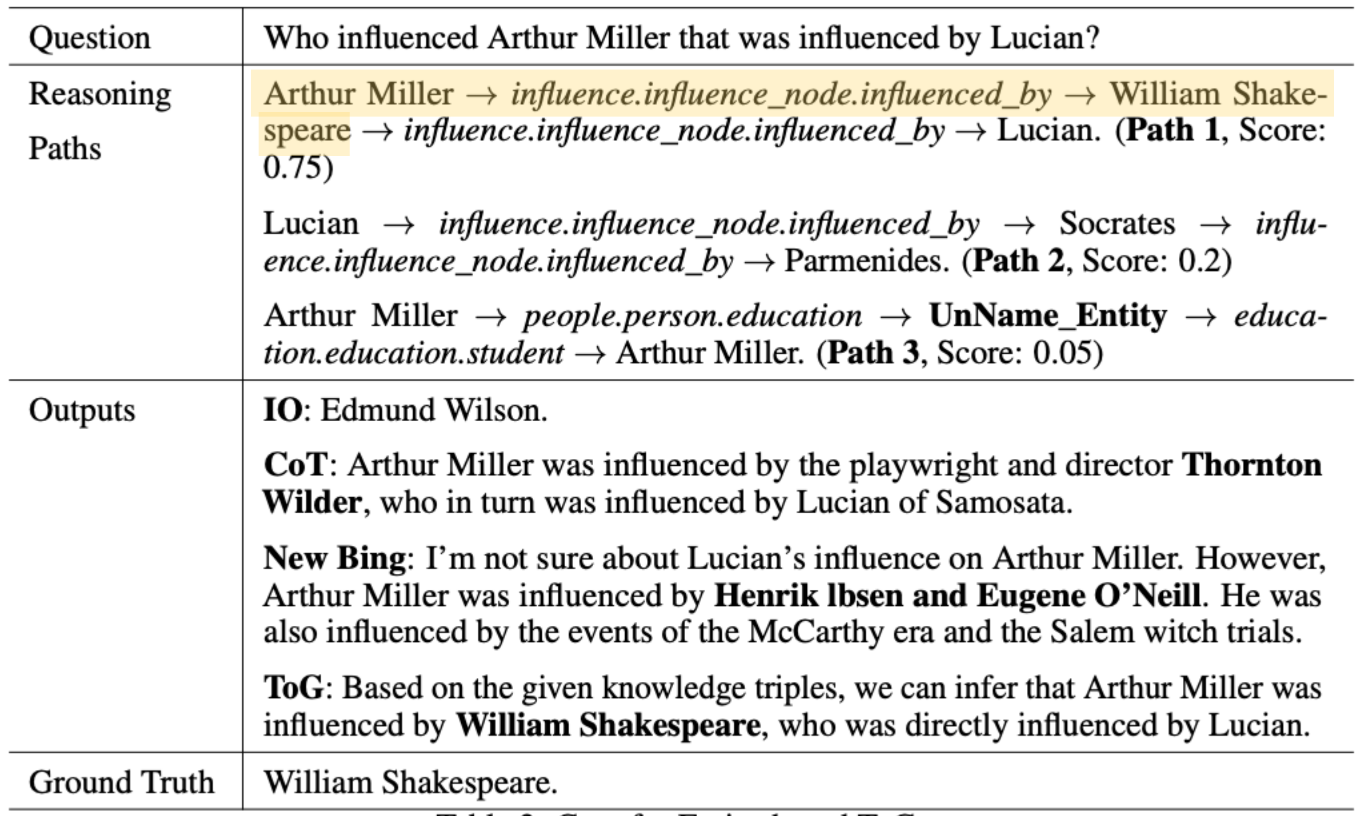
Results and Analysis

- Think-on-Graph 방법론 적용시 가장 높은 성능을 보임



Methods	Exact Match
IO	31.43
CoT	42.63
ToG(E)	57.49
ToG(R)	**60.10**

Table 1: Performance on the CWQ dataset. ToG(E) and ToG(R) denote Entity-based ToG and Relation-based ToG, respectively.



Question	Who influenced Arthur Miller that was influenced by Lucian?
Reasoning Paths	Arthur Miller → *influence.influence_node.influenced_by* → William Shakespeare → *influence.influence_node.influenced_by* → Lucian. (**Path 1**, Score: 0.75)
	Lucian → *influence.influence_node.influenced_by* → Socrates → *influence.influence_node.influenced_by* → Parmenides. (**Path 2**, Score: 0.2)
	Arthur Miller → *people.person.education* → **UnName_Entity** → *education.education.student* → Arthur Miller. (**Path 3**, Score: 0.05)
Outputs	**IO**: Edmund Wilson.
	CoT: Arthur Miller was influenced by the playwright and director **Thornton Wilder**, who in turn was influenced by Lucian of Samosata.
	New Bing: I'm not sure about Lucian's influence on Arthur Miller. However, Arthur Miller was influenced by **Henrik lbsen and Eugene O'Neill**. He was also influenced by the events of the McCarthy era and the Salem witch trials.
	ToG: Based on the given knowledge triples, we can infer that Arthur Miller was influenced by **William Shakespeare**, who was directly influenced by Lucian.
Ground Truth	William Shakespeare.

Table 2: Case for Entity-based ToG.

Thank you

Natural Language Processing & Artificial Intelligence

고려대학교 KOREA UNIVERSITY