Enhancing Reasoning ability of LLM with Instruction-tuning based on Data Transformation

NLP&AI 연구실 세미나 (09/07, Thu)

NLP&AI 구선민

Natural Language Processing & Artificial Intelligence

고려대학교 KOREA UNIVERSITY

Papers

- Clues Before Answers: Generation-Enhanced Multiple-Choice QA (NAACL 2022)

- Aligning Instruction Tasks Unlocks Large Language Models as Zero-Shot Relation Extractors (ACL 23)

- Say What You Mean! Large Language Models Speak Too Positively about Negative Commonsense Knowledge (ACL 23)

Clues Before Answers: Generation-Enhanced Multiple-Choice QA

Zixian Huang and **Ao Wu** and **Jiaying Zhou**
State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
{zixianhuang,awu,jyzhou}@smail.nju.edu.cn

Yu Gu
The Ohio State University, Columbus, USA
gu.826@osu.edu

Yue Zhao and **Gong Cheng**
State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
yuezhao@smail.nju.edu.cn, gcheng@nju.edu.cn

NAACL 2022

Natural Language Processing & Artificial Intelligence

KOREA UNIVERSITY 고려대학교 KOREA UNIVERSITY

Background

- Multiple-choice question answering (MCQA)는 주어진 질문에 대한 선지에서 정답을 고르는 태스크
  - Commonsense knowledge 및 scientific knowledge와 같은 풍부한 지식과 리즈닝 능력 요구됨

- MCQA 패러다임 2가지
  - encoder-decoder model
    - Q와 모든 선지 concat 해서 input
    - output으로 옳은 선지 1개 나오도록
  - encoder-only model
    - Q및 각각의 선지 pair 해서 Input
    - 각 pair 당 확률 구해서 가장 높은거 선택

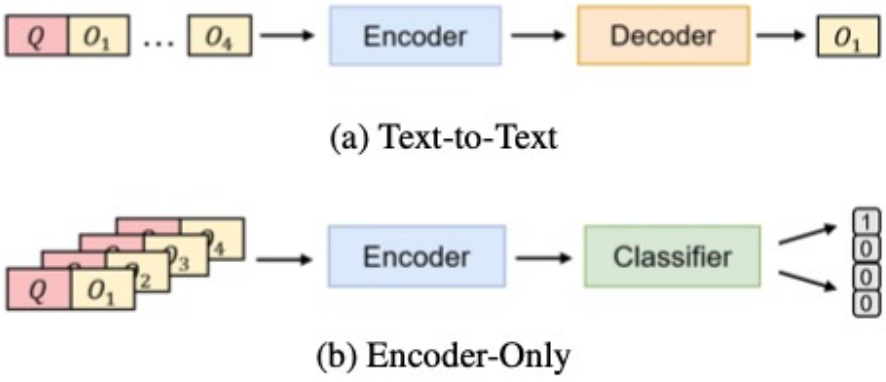


Figure 2: Paradigms for MCQA.

Motivation

- 기존의 encoder-decoder model 에서 decoder를 classifier로 활용하는 것은 사전학습 방법과의 불일치성으로 모델의 knowledge가 충분히 활용되지 않을 수 있음
  - classification and regression tasks에서 디코더 layer가 제대로 활용되지 않는 경우 많음

- Key findings는 문제와 선지 간의 연관성을 파악하는 것이며, 어려움

→ 모델의 생성 능력 및 모델이 가지고 있는 knowledge를 활용하기 위해 인간이 MCQA 태스크를 해결하는 방식을 모방

  - 인간은 문제를 읽은 후 정답을 맞히는데 도움이 되는 배경 지식(i.e., looking for clues)과 문제를 연관시킬 수 있음

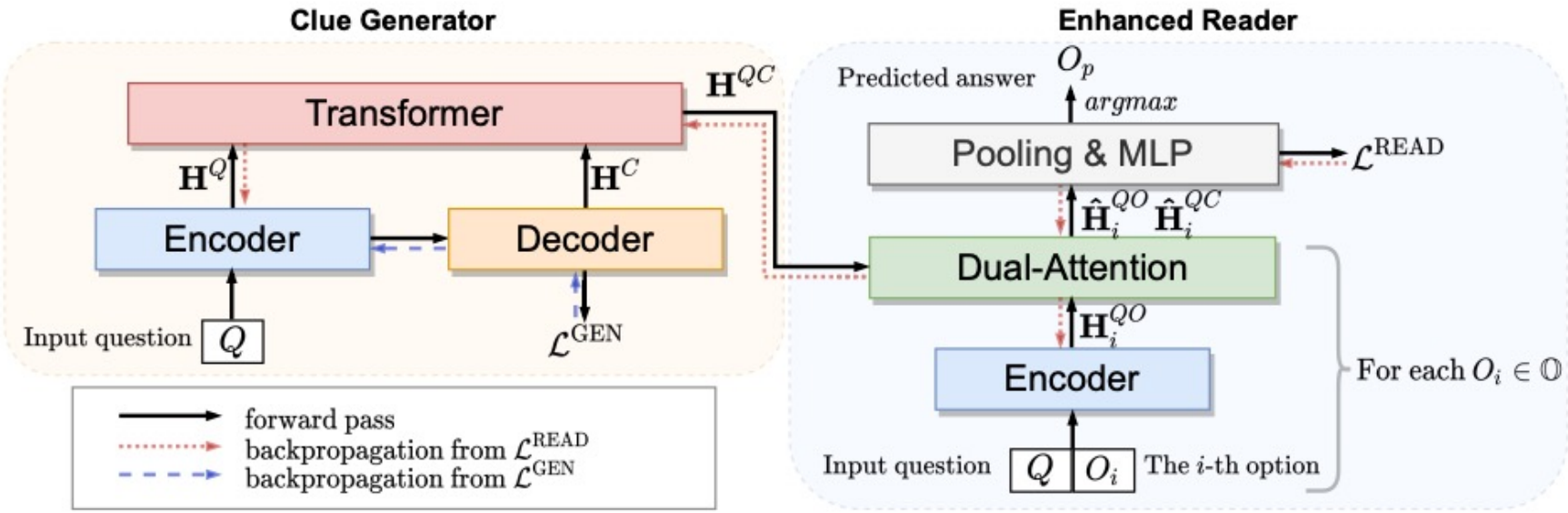
Question:
A company makes notebooks for college courses, so their main material is ?
(A) Chips (B) Water (C) Grass (D) **Trees**

Clue: Paper

Figure 1: An example MCQA task and a generated clue. Bold underline indicates the correct answer.

GenMC Model

* Architecture of GenMC



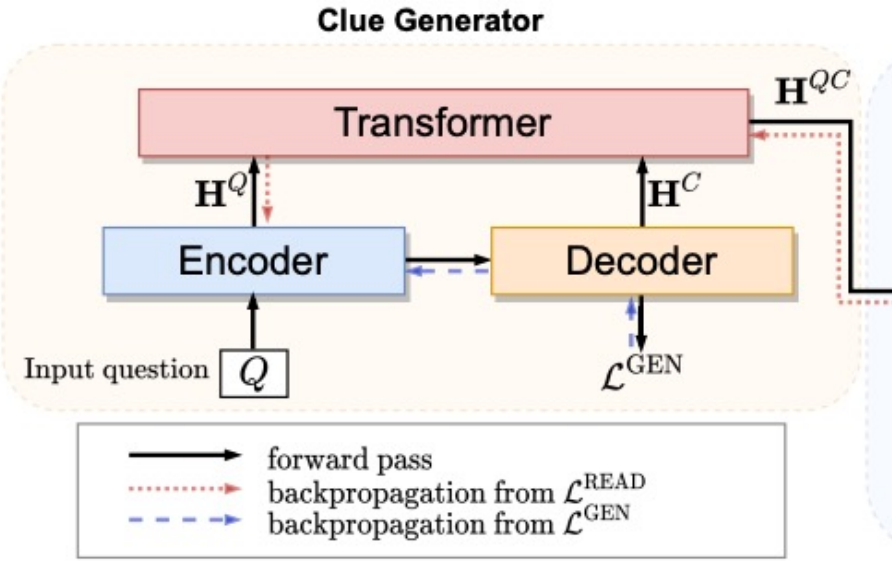
GenMC Model

*** Clue Generator**

- question Q를 input으로 받아 clue C를 outputs
- clue text C가 아닌 clue의 representation H^C 를 모델에 사용
- 인코더에서 question representation 및 디코더에서 clue representation 얻음

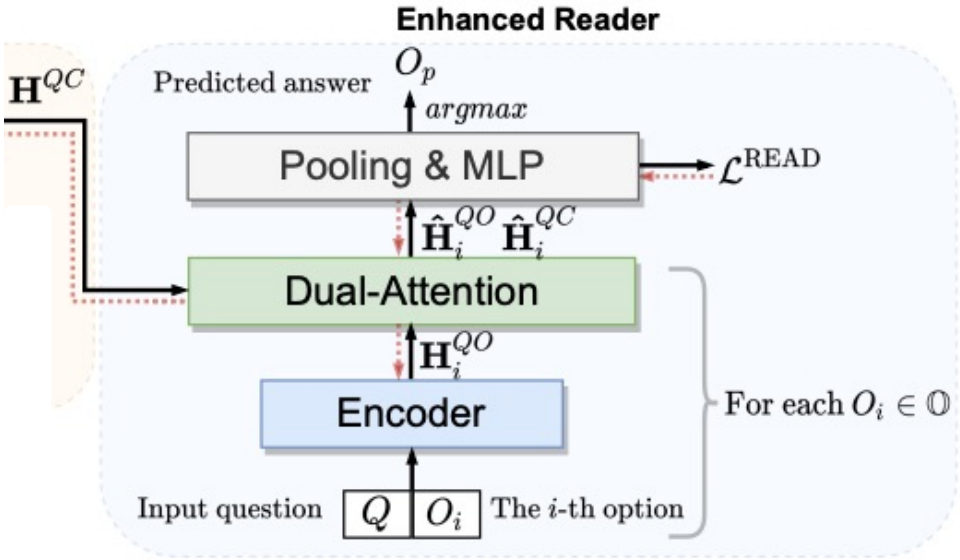
$$\mathbf{p}_j^C, \mathbf{H}_j^C = \texttt{Decoder}(c_{j-1}, \mathbf{H}_{<j}^C, \mathbf{H}^Q),$$

- p_j^C : j-th step의 decoding vocabulary에 대한 확률 분포
- H^{QC}: Q를 더 잘 이해하고 정답을 맞히는데 도움이 될 수 있는 C의 information을 전달



GenMC Model

* Enhanced Reader



- 생성된 clue representation이 각 question-option pair의 이해도를 향상시킴
- clue representation H^{QC}를 베이스로 모델은 각 question-option pair를 집중적으로 read해서 clue와 option 사이에 일치하는 signal을 얻음
- dual-attention 을 이용해서 H^{QC} 에서 H_i^{QC}**로**, H_i^{QC} 에서 H^{QC} 로 정보를 fuse
- Max-pooling을 통해 얻은 각 선지에 대한 final score 를 MLP 거쳐서 가장 높은 점수 받은 선지를 predicted answer로 선택

GenMC Model

* Training Objective

- MCQA에서 correct answer 의 2가지 속성(text , index)를 활용

- clue generator를 supervise하기 위해 text로 사용하고,
  enhanced reader를 supervise 하기 위해 index로(i.e., classification label) 사용

→ text-to-text paradigm 보다 더 자연스럽기 때문에 더 뛰어난 성능 발휘할 가능성 O

<Generator Loss>

$$\mathbf{p}_j^{O_t}, \mathbf{H}_j^{O_t} = \texttt{Decoder}(a_{j-1}, \mathbf{H}_{<j}^{O_t}, \mathbf{H}^Q),$$

$$\mathcal{L}^{\text{GEN}} = -\frac{1}{m}\sum_{j=1}^{m} \log \mathbf{p}_{j,a_j}^{O_t},$$

<Reader Loss>

$$\mathcal{L}^{\text{READ}} = -\log \frac{\exp(s_t)}{\sum_{i=1}^{n} \exp(s_i)}.$$

Experiment Setup

*** Dataset**

- 5가지 MCQA 데이터셋 대상으로 실험

- commonsense questions 데이터셋
  - commonsense knowledge and reasoning 능력 필요
  - CSQA, OBQA → commonsense facts 기반으로 human이 구축
    - 각 질문은 5개(CSQA) 혹은 4개(OBQA)의 선지로 구성

- scientific questions 데이터셋
  - scientific facts에 대한 추론 필요
  - ARC-Easy, ARC-Challenge
    - 4개의 선지로 구성된 초등학교 수준의 과학 문항으로 구성
    - ARC-Challenge가 리즈닝 능력이 더 요구되는 난이도 높은 문항으로 구성
  - QASC
    - 초등학교 및 중학교 수준의 과학 문제로 각 문항에 8개의 선지로 구성

Experiment Setup

* Baselines

- $\text{Text2Text}_{\text{vanilla}}$
  - input question을 모든 candidate options와 concat하고,
  - 각 옵션 앞에 옵션 id 붙여서 각 시퀀스끼리 연결
  - 연결된 시퀀스는 인코더에 feed 되어 문제와 모든 옵션에 대한 joint representation 얻음
  - 해당 joint representation 기반으로 디코더가 option ID outputs
  - 이 세팅에서 디코더는 classifier로 사용

- $\text{Text2Text}_{\text{enc}}$
  - 인디코더 모델에서 인코더 부분만 사용
  - 각 옵션마다 인코더를 사용해서 문제와 joint representation 계산
  - 각 representation을 scorer (i.e., an MLP)에 넣어서 question-option pair의 점수를 각각 계산
  - 가장 높은 점수 받은 옵션을 예측

Main result

* Results

- GenMC가 기존 모델보다 pre-trained encoder-decoder models을 더 효과적으로 사용할 수 있음을 보여줌

- 대부분 $\text{Text2Text}_{\text{enc}}$이 $\text{Text2Text}_{\text{vanilla}}$ 보다 성능 좋음
  → 사전학습을 통해 얻은 디코더의 general language knowledge이 classifier로만 사용되어 낭비되는 경우 많음
  → GenMC 는 디코더 효과적으로 활용 가능

- 사전학습을 통해 얻은 embedded knowledge이 MCQA 태스크에 매우 중요함을 시사

	CSQA		OBQA		ARC-Easy		ARC-Challenge		QASC	
	dev	test	dev	test	dev	test	dev	test	dev	test
$\text{BART}_{\text{BASE}}$										
$\text{Text2Text}_{\text{vanilla}}$	51.62 (±0.04)	53.26 (±0.57)	54.93 (±0.83)	52.73 (±1.00)	51.55 (±1.38)	50.51 (±1.82)	30.05 (±1.25)	24.95 (±1.10)	46.72 (±1.21)	26.78 (±1.21)
$\text{Text2Text}_{\text{enc}}$	50.63 (±0.66)	52.22 (±1.64)	55.87 (±1.10)	51.00 (±1.83)	49.03 (±1.86)	49.94 (±1.49)	32.32 (±4.87)	26.24 (±2.01)	48.08 (±1.35)	17.06 (±0.39)
GenMC	**54.82** (±0.61)	**56.40** (±0.61)	**58.53** (±0.31)	**57.53** (±2.91)	**59.38** (±1.60)	**56.80** (±0.28)	**38.64** (±0.90)	**33.82** (±1.66)	**57.70** (±0.43)	**35.96** (±1.70)
T5_{BASE}										
$\text{Text2Text}_{\text{vanilla}}$	57.59 (±0.81)	60.93 (±0.73)	59.53 (±0.81)	57.53 (±0.70)	52.20 (±0.31)	51.75 (±0.89)	29.38 (±2.63)	23.69 (±2.47)	54.55 (±1.01)	37.94 (±1.47)
$\text{Text2Text}_{\text{enc}}$	58.96 (±1.21)	59.49 (±1.41)	60.67 (±2.86)	57.07 (±3.03)	56.55 (±1.17)	52.92 (±0.29)	29.49 (±5.13)	26.09 (±0.23)	56.84 (±0.84)	39.60 (±2.38)
GenMC	**60.65** (±0.47)	**63.45** (±0.29)	**62.07** (±1.01)	**61.67** (±0.58)	**62.38** (±0.67)	**58.82** (±0.37)	**43.62** (±0.52)	**39.00** (±0.30)	**58.93** (±1.76)	**41.72** (±1.18)
$\text{BART}_{\text{LARGE}}$										
$\text{Text2Text}_{\text{vanilla}}$	65.58 (±2.72)	66.91 (±2.14)	62.66 (±1.18)	61.46 (±1.74)	63.49 (±1.89)	62.81 (±2.15)	29.94 (±2.32)	28.55 (±4.97)	64.57 (±2.21)	47.80 (±2.22)
$\text{Text2Text}_{\text{enc}}$	65.00 (±0.66)	67.35 (±0.90)	63.80 (±1.44)	62.47 (±1.53)	68.20 (±2.04)	65.33 (±1.74)	35.37 (±6.07)	31.13 (±5.86)	65.07 (±0.94)	47.19 (±0.71)
GenMC	**69.57** (±0.89)	**72.26** (±0.70)	**68.93** (±1.17)	**68.07** (±1.70)	**72.43** (±0.54)	**68.68** (±0.34)	**48.93** (±0.98)	**45.52** (±1.54)	**68.39** (±0.68)	**55.90** (±0.92)
T5_{LARGE}										
$\text{Text2Text}_{\text{vanilla}}$	67.53 (±0.43)	70.63 (±0.74)	66.80 (±0.87)	63.53 (±1.10)	65.61 (±0.18)	62.55 (±0.54)	43.05 (±1.69)	42.83 (±2.00)	64.13 (±1.47)	57.74 (±0.82)
$\text{Text2Text}_{\text{enc}}$	68.41 (±0.73)	70.30 (±0.82)	65.93 (±1.03)	63.67 (±0.46)	69.61 (±0.20)	66.65 (±0.34)	30.73 (±3.15)	28.76 (±4.85)	65.27 (±1.55)	55.65 (±0.45)
GenMC	**71.10** (±0.41)	**72.67** (±1.02)	**71.60** (±0.92)	**66.87** (±1.33)	**72.49** (±0.77)	**69.01** (±1.97)	**49.83** (±2.06)	**47.41** (±2.00)	**67.61** (±1.14)	**58.06** (±0.92)

Main result

* Error Analysis

- Clue type을 3가지(Irrelevant, Relevant but unhelpful, Helpful) 로 나눠서 오류 분석

- Clue의 23.60%는 관련 없는 clue로 오히려 문제 풀이에 악영향
  → Future works에서 질문에서 clue 더 잘 생성하는 방법 연구하겠다

Clue Type	Percentage	Example	
		Instance	Clue
Irrelevant	23.60%	Which would you likely find inside a beach ball? (A) cheese (B) *steam* (C) water (D) **air**	a squid
Relevant but unhelpful	52.40%	What may have been formed by a volcano? (A) **Mt. McKinley** (B) Lake Pontchartrain (C) The great lakes (D) *Niagara Falls*	a lake
Helpful	24.00%	Where would there be an auditorium with only a single person speaking? (A) lights (B) crowd (C) **university campus** (D) *theater* (E) park	school

Table 6: Distribution of clue types in negative cases with examples. Bold underline indicates the correct answer, and italic indicates the predicted label.

Aligning Instruction Tasks Unlocks Large Language Models as Zero-Shot Relation Extractors

Kai Zhang **Bernal Jiménez Gutiérrez** **Yu Su**
The Ohio State University
{zhang.13253, jimenezgutierrez.1, su.809}@osu.edu

ACL 2023

Motivation

- 최근 large-scale instruction-following datasets으로 LLM을 fine-tuning하여 다양한 NLP 태스크에서 성능 향상을 보였으나, fundamental information extraction task인 RE에서는 아직 small LMs* 보다 성능 낮음
  ** 1B params 미만 모델*

- instruction-tuning datasets 안에서 RE가 전체의 1% 미만을 차지하기 때문에 LLM의 RE 성능을 이끌어내지 못했다고 가정

  → instruction-tuning datasets의 주요 태스크인 QA에 RE를 연계하는 프레임워크인 QA4RE 제안

  → RE를 multiple-choice question answering (QA) 에 align 시킴
    - MCQA가 instruction-tuning datasets 를 가장 많이 차지하고 있기 때문 (12-15%)

Motivation

- 최근 large-sca
  보였으나, fund
  * *1B param*
- instruction-tu
  못했다고 가정
  → instructio
  → RE를 mul
  - MCQ

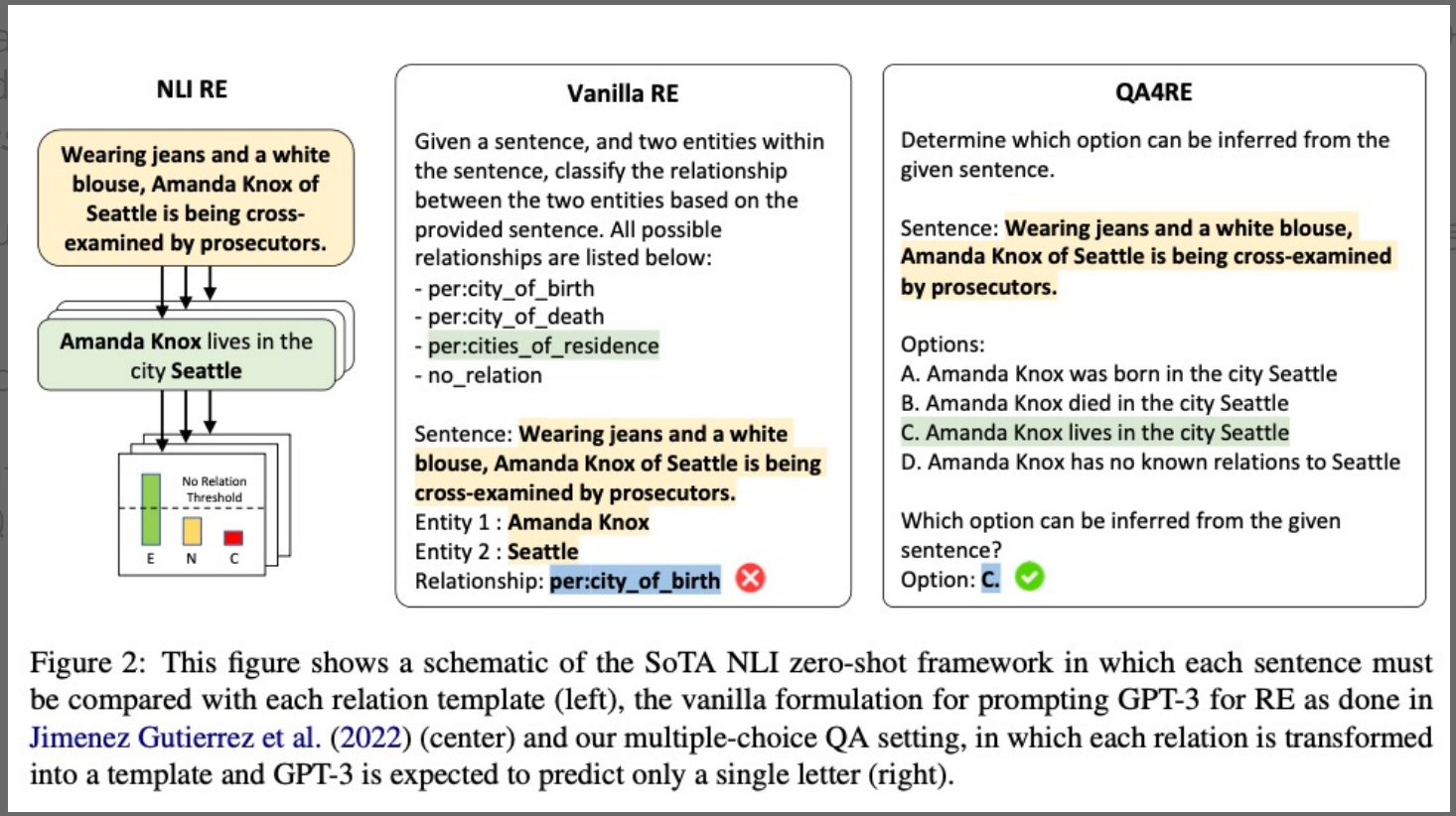
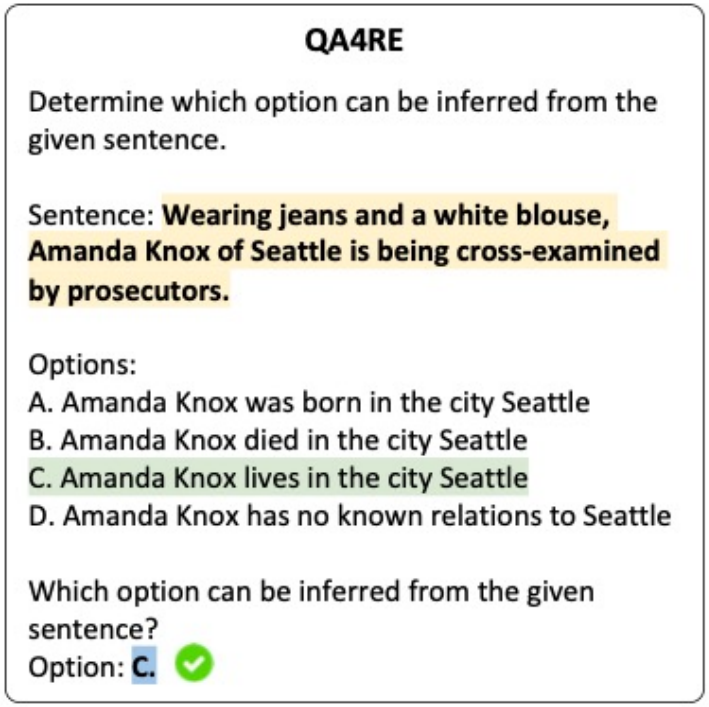


Figure 2: This figure shows a schematic of the SoTA NLI zero-shot framework in which each sentence must be compared with each relation template (left), the vanilla formulation for prompting GPT-3 for RE as done in Jimenez Gutierrez et al. (2022) (center) and our multiple-choice QA setting, in which each relation is transformed into a template and GPT-3 is expected to predict only a single letter (right).

QA4RE

- RE 태스크를 MCQA 태스크로 재구성
- 주어진 head and tail RE entities (E_h and E_t)를 relation templates에 통합
- multiple-choice options로 사용함으로써 LLM을 QA instruction fine-tuning
- 기존의 verbalized relation 대신 answer index만 생성할 수 있음



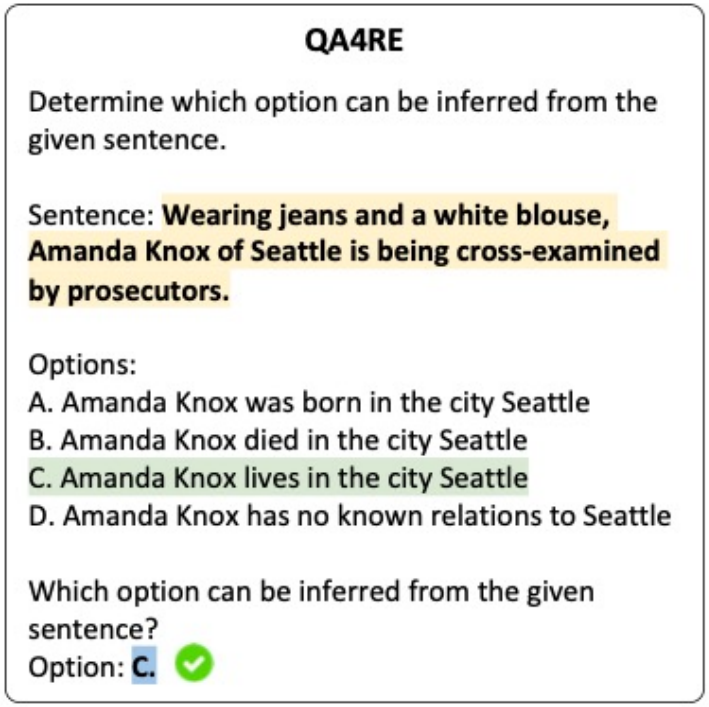
QA4RE

- RE를 MCQA로 transform하기 위해서, 주어진 relation example (S, E_h, E_t)에 대해 sentence S 를 standard QA의 context로 활용

- pre-defined templates filled with E_h and E_t entities 로 구성된 옵션 생성

- To ensure a fair comparison, type constraints을 적용해서 head and tail entities와 호환되지 않는 relation types 옵션 제거

  - e.g., E_h: PERSON 일 때 person이 headquarter 라는 정보 없을 때 “org:country_of_headquarters” 관계는 invalid



Experiment Setup

*** Dataset**

- RE datasets 4가지
  - TACRED
  - RETACRED
  - TACREV
  - SemEval 2010 Task 8 (SemEval for brevity)

- 기존 연구에 따라서 F1 스코어 사용

- API 비용 때문에 각 데이터셋의 test set에서 1,000개 뽑아서 test set으로 사용

Experiment Setup

* Baselines : Zero-Shot

- Small LM-based models
  - RE를 NLI 태스크로 재구성하고 MNLI dataset으로 fine-tuning 된 LM들 사용
  - (1) BART- Large, (2) RoBERTa-Large, (3) DeBERTa-Xlarge

- SuRE
  - RE를 요약 태스크로 구성하고 generative LMs (BART-Large, PEGASUS- Large) 사용

- NLI approach의 경우, TACRED 및 TACREV에서 자체 템플릿 사용해서 성능 다시 리포트
  - RETACRED 와 SemEval는 템플릿이 없기 때문에 해당 데이터셋의 후속 연구인 SuRE 의 템플릿 사용

- LLM을 포함한 모든 제로 샷 방법 relation label space 줄이기 위해 entity type constraints 적용
  - SemEval는 엔티티 타입 제공하지 않기 때문에 제약 조건 적용 X

Experiment Setup

*** Baselines : Few-Shot**

- main 실험은 zero-shot RE에 포커스하지만, TACRED dataset에서 small LM-based methods와 비교를 통해 QA4RE 의 성능 추가 분석

- NLI baseline을 few-shot setting으로 확장

- 모델 종류 → RoBERTa-Large로 아래 3개 세팅
  (1) standard Fine-Tuning
  (2) PTR using prompt-tuning with logical rules
  (3) KnowPrompt using entity type knowledge via learning virtual tokens

Experiment Setup

* QA4RE Implementation Details

- For prompt engineering, TACRED dev set의 250개의 서브셋에서 text-davinci-002를 사용한 pilot 실험에서 vanilla RE 및 QA4RE에 대한 prompt formats 과 task instructions 살펴봄

- LLMs in Chat Completion API (gpt-3.5-turbo-0301)의 control 옵션이 적기 때문에 temperature=0으로 세팅하고 디폴트 system prompt 사용

- 다양한 태스크에 대해 학습된 오픈 소스 FLAN-T5 series LLMs도 살펴봄
  → 학습에 사용된 1,836 tasks 중 0.5% 미만에만 RE-similar tasks 포함되어 있기 때문에 가설 검증하기에 적합한 모델

Main result

* Zero-Shot Results

- RE를 QA로 재구성한 QA4RE 방법이 모든 LLM및 대부분의 데이터셋에서 vanilla RE 보다 성능 향상 보임

- FLAN-T5 LLMs에 QA4RE framework이 성능 향상 도움
  → RE 태스크가 FLAN-T5 학습하는데 사용된 instruction tasks의 0.5% 미만을 차지하는 것을 고려했을 때, 빈도수가 적은 태스크를 QA 같은 common instruction-tuning tasks와 align하면 LLM이 low-frequency tasks에서 능력을 발휘할 수 있음을 보여줌

Methods		TACRED P	TACRED R	TACRED F1	RETACRED P	RETACRED R	RETACRED F1	TACREV P	TACREV R	TACREV F1	SemEval P	SemEval R	SemEval F1	Avg. F1
Baselines														
NLI_{BART}		42.6	65.0	51.4	59.5	34.9	44.0	44.0	74.6	55.3	21.6	23.7	22.6	43.3
$NLI_{RoBERTa}$		37.1	76.9	50.1	52.3	67.0	58.7	37.1	83.6	51.4	17.6	20.9	19.1	44.8
$NLI_{DeBERTa}$		42.9	76.9	55.1	71.7	58.3	64.3	43.3	84.6	57.2	22.0	25.7	23.7	50.1
$SuRE_{BART}$		13.1	45.7	20.4	17.9	34.6	23.6	14.1	52.3	22.2	0.0	0.0	0.0	16.5
$SuRE_{PEGASUS}$		13.8	51.7	21.8	16.6	34.6	22.4	13.5	54.1	21.6	0.0	0.0	0.0	16.4
GPT-3.5 Series														
ChatGPT	Vanilla	32.1	74.8	44.9	45.4	61.3	52.1	30.3	79.6	43.9	18.2	20.8	19.4	40.1
	QA4RE	32.8	68.0	44.2 (−0.7)	48.3	76.8	59.3 (+7.2)	34.7	79.1	48.2 (+4.3)	29.9	35.2	32.3 (+12.9)	46.0 (+5.9)
code-002	Vanilla	27.2	70.1	39.2	42.7	70.4	53.1	27.5	77.7	40.6	27.2	25.6	26.4	39.8
	QA4RE	37.7	65.4	47.8 (+8.6)	48.0	74.0	58.2 (+5.1)	31.7	65.5	42.7 (+2.1)	25.2	29.2	27.0 (+0.6)	43.9 (+4.1)
text-002	Vanilla	31.2	73.1	43.7	44.1	76.3	55.9	30.2	76.8	43.3	31.4	28.8	30.1	43.2
	QA4RE	35.6	68.4	46.8 (+3.1)	46.4	72.4	56.5 (+0.6)	35.7	76.8	48.8 (+5.4)	29.4	34.3	31.6 (+1.5)	45.9 (+2.7)
text-003	Vanilla	36.9	68.8	48.1	49.7	62.2	55.3	38.2	76.8	51.0	33.2	39.3	36.0	47.6
	QA4RE	47.7	78.6	**59.4** (+11.3)	56.2	67.2	61.2 (+5.9)	46.0	83.6	**59.4** (+8.4)	41.7	45.0	43.3 (+7.3)	**55.8** (+8.2)
FLAN-T5 Series														
XLarge	Vanilla	51.6	49.1	50.3	54.3	40.3	46.3	56.0	59.1	57.5	35.6	29.8	32.4	46.6
	QA4RE	40.0	78.2	53.0 (+2.7)	57.1	79.7	66.5 (+20.2)	40.7	85.9	55.3 (−2.2)	45.1	40.1	42.5 (+10.1)	54.3 (+7.7)
XXLarge	Vanilla	52.1	47.9	49.9	56.6	54.0	55.2	52.6	50.9	51.7	29.6	28.8	29.2	46.5
	QA4RE	40.6	82.9	54.5 (+4.6)	56.6	82.9	**67.3** (+12.1)	39.6	86.4	54.3 (+2.6)	41.0	47.8	**44.1** (+14.9)	55.1 (+8.6)

Table 1: Experimental results on four RE datasets (%). We omit the 'davinci' within the names of GPT-3.5 Series LLMs and ChatGPT refers to gpt-3.5-turbo-0301. We mark the best results in **bold**, the second-best underlined, and F1 improvement of our QA4RE over vanilla RE in green.

Main result

* Zero-Shot Results

- SemEval dataset 은 type-constraints이 없기 때문에 더 어려움

- search space이 클 경우, generative LMs without fine-tuning는 모든 문제를 NoTA relation으로 요약하는 경향이 있어서 잘 수행하지 못하게 됨

Methods		TACRED P	TACRED R	TACRED F1	RETACRED P	RETACRED R	RETACRED F1	TACREV P	TACREV R	TACREV F1	SemEval P	SemEval R	SemEval F1	Avg. F1
Baselines														
NLI_{BART}		42.6	65.0	51.4	59.5	34.9	44.0	44.0	74.6	55.3	21.6	23.7	22.6	43.3
$NLI_{RoBERTa}$		37.1	76.9	50.1	52.3	67.0	58.7	37.1	83.6	51.4	17.6	20.9	19.1	44.8
$NLI_{DeBERTa}$		42.9	76.9	55.1	71.7	58.3	64.3	43.3	84.6	57.2	22.0	25.7	23.7	50.1
$SuRE_{BART}$		13.1	45.7	20.4	17.9	34.6	23.6	14.1	52.3	22.2	0.0	0.0	0.0	16.5
$SuRE_{PEGASUS}$		13.8	51.7	21.8	16.6	34.6	22.4	13.5	54.1	21.6	0.0	0.0	0.0	16.4
GPT-3.5 Series														
ChatGPT	Vanilla	32.1	74.8	44.9	45.4	61.3	52.1	30.3	79.6	43.9	18.2	20.8	19.4	40.1
	QA4RE	32.8	68.0	44.2 (−0.7)	48.3	76.8	59.3 (+7.2)	34.7	79.1	48.2 (+4.3)	29.9	35.2	32.3 (+12.9)	46.0 (+5.9)
code-002	Vanilla	27.2	70.1	39.2	42.7	70.4	53.1	27.5	77.7	40.6	27.2	25.6	26.4	39.8
	QA4RE	37.7	65.4	47.8 (+8.6)	48.0	74.0	58.2 (+5.1)	31.7	65.5	42.7 (+2.1)	25.2	29.2	27.0 (+0.6)	43.9 (+4.1)
text-002	Vanilla	31.2	73.1	43.7	44.1	76.3	55.9	30.2	76.8	43.3	31.4	28.8	30.1	43.2
	QA4RE	35.6	68.4	46.8 (+3.1)	46.4	72.4	56.5 (+0.6)	35.7	76.8	48.8 (+5.4)	29.4	34.3	31.6 (+1.5)	45.9 (+2.7)
text-003	Vanilla	36.9	68.8	48.1	49.7	62.2	55.3	38.2	76.8	51.0	33.2	39.3	36.0	47.6
	QA4RE	47.7	78.6	**59.4** (+11.3)	56.2	67.2	61.2 (+5.9)	46.0	83.6	**59.4** (+8.4)	41.7	45.0	43.3 (+7.3)	**55.8** (+8.2)
FLAN-T5 Series														
XLarge	Vanilla	51.6	49.1	50.3	54.3	40.3	46.3	56.0	59.1	57.5	35.6	29.8	32.4	46.6
	QA4RE	40.0	78.2	53.0 (+2.7)	57.1	79.7	66.5 (+20.2)	40.7	85.9	55.3 (−2.2)	45.1	40.1	42.5 (+10.1)	54.3 (+7.7)
XXLarge	Vanilla	52.1	47.9	49.9	56.6	54.0	55.2	52.6	50.9	51.7	29.6	28.8	29.2	46.5
	QA4RE	40.6	82.9	54.5 (+4.6)	56.6	82.9	**67.3** (+12.1)	39.6	86.4	54.3 (+2.6)	41.0	47.8	**44.1** (+14.9)	55.1 (+8.6)

Table 1: Experimental results on four RE datasets (%). We omit the 'davinci' within the names of GPT-3.5 Series LLMs and ChatGPT refers to gpt-3.5-turbo-0301. We mark the best results in **bold**, the second-best underlined, and F1 improvement of our QA4RE over vanilla RE in green.

Main result

* Few-Shot Results

- 비용 문제로 제로샷에서 성능 가장 좋았던 LLM(text-davinci-003)에 대해서 4-shots에서만 실험

- vanilla RE는 few-shots 세팅에서 성능 향상 얻지 못함
  → few-shots 세팅에서 한계가 있음을 시사
  → few-shot demonstrations이 잘못된 relation으로 bias될 수 있음을 나타냄

- few-shot text-davinci-003에 QA4RE framework을 적용했을 때 $NLI_{DeBERTa}$보다 성능 낮았음

Methods	K=0	K=4	K=8	K=16	K=32
Fine-Tuning	-	9.0	21.2	29.3	33.9
PTR	-	26.8	30.0	32.9	36.8
KnowPrompt	-	30.2	33.7	34.9	35.0
$NLI_{DeBERTa}$-TEMP1	55.0	**64.2**	**64.7**	**58.7**	**65.7**
$NLI_{DeBERTa}$-TEMP2	49.4	**51.2**	47.3	**50.5**	48.1
Vanilla	48.1	46.2		-	
QA4RE	59.4	**62.0**		-	

Table 4: Few-shot F1 on TACRED (%). All results are averaged over 3 different training subsets for each K. We use text-davinci-003 for vanilla RE and QA4RE. For the best-performing baseline (NLI) as well as vanilla RE and QA4RE, we mark the results in **bold** when they are improved over their zero-shot alternatives.

Say What You Mean! Large Language Models Speak Too Positively about Negative Commonsense Knowledge

Jiangjie Chen[♠], Wei Shi[♠], Ziquan Fu[♡*], Sijie Cheng[♠], Lei Li[♣], Yanghua Xiao[♠◇†]
[♠]Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University
[♡]System Inc. [♣]University of California, Santa Barbara
[◇]Fudan-Aishu Cognitive Intelligence Joint Research Center
{jjchen19, sjcheng20, shawyh}@fudan.edu.cn
wshi22@m.fudan.edu.cn, frank@system.com, leili@cs.ucsb.edu

ACL 2023

Motivation

- 대부분의 knowledge dataset는 positive만 고려하고 negative는 거의 포함되어 있지 않음
  e.g., “사자는 바다에 살지 않는다”

- Negative knowledge 는 실제 세계에 존재하며 인지 능력에 중요
  - 사실이 아닌 걸 구분하거나, 하지 말아야 할 것 구분

- LLMs의 positive knowledge를 저장하고 활용하는 능력에 대해 연구되어 왔지만 negative knowledge는 명시적으로 연구된 적 없음

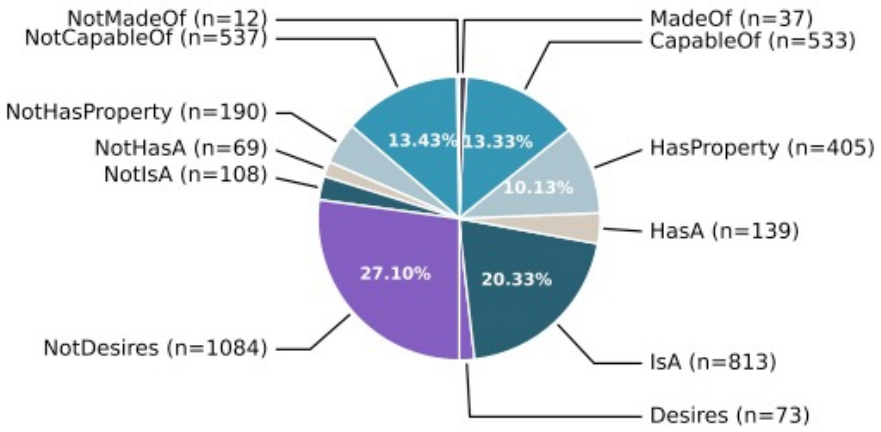
  → 따라서 LLM이 사전학습을 통해 implicit negative knowledge를 학습할 수 있을지를 검증해보고자 함

  → 이를 위해 2가지 태스크 설계해서 LLM 능력 프루빙함
  - constrained keywords-to-sentence generation task (CG)
  - Boolean question-answering task (QA)

Probing Protocol

*** 검증 범위 및 데이터셋**

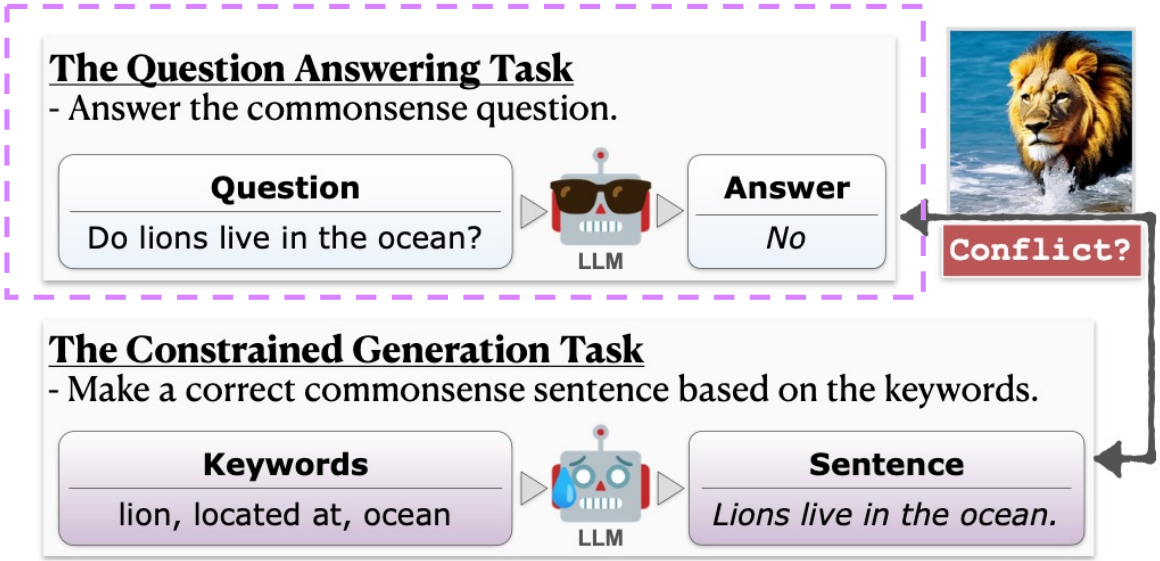
- 검증하는 지식의 범위를 commonsense concepts 간의 relational knowledge로 제한
- 트리플 <s, r, o> 주어졌을 때 ¬r(s, o) 이면 negative fact, true이면 positive fact로 정의
- 대명사, 부정사, 형용사를 주어 또는 목적어로 사용하는 invalid 트리플 제거
- 4,000개의 트리플 대상으로 하며, positive와 negative 는 동일한 크기(1:1)



Probing Protocol

* Probing Task Formulation

- Task 1: Boolean Question Answering (QA)

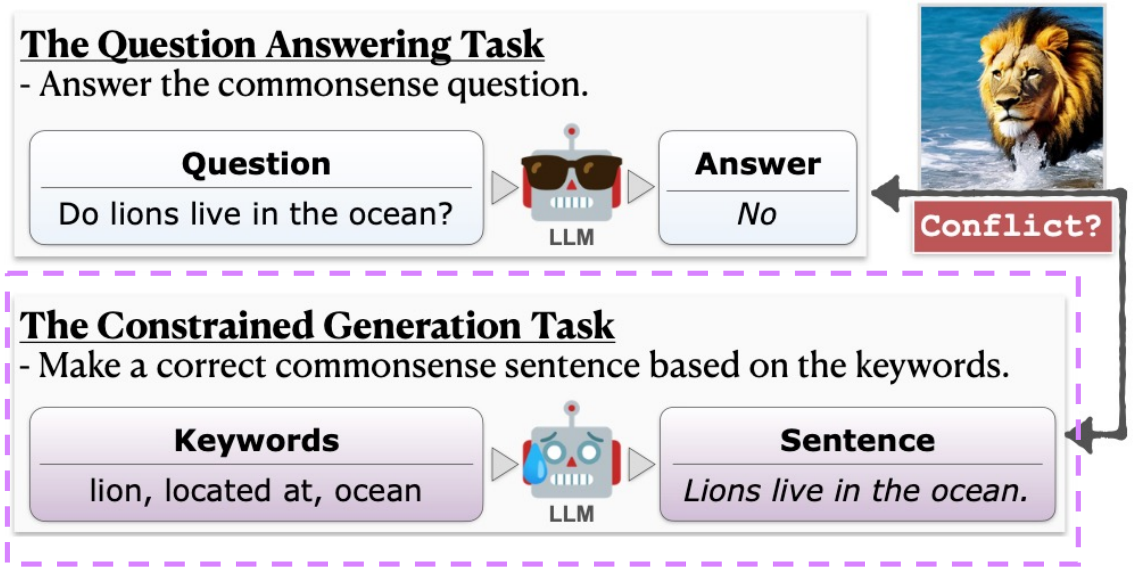


- 예/아니오 질문에 답해서 fact에 대한 belief 표현해야 함
- 모든 트리플 <s,r,o>를 예/아니오 질문 Q로 transform
- 질문 생성하기 위해 InstructGPT using in-context learning 사용
- 50 개의 랜덤 샘플링에 대한 manual inspection에 따르면 94%의 valid

Probing Protocol

* Probing Task Formulation

- Task 2: Constrained Sentence Generation (CG)



- 텍스트 생성 태스크를 통해
- model's belief을 직접적으로 표현

- COMMONGEN과 유사한 **keyword-to sentence** 태스크 설계해서 프루빙을 보다 통제 가능하게 함

- 트리플 <s, r, o>이 주어지면 모델은 (negative) knowledge에 기반한 문장 생성해야 함

Do LLMs have negative commonsense knowledge?

* Probing LLMs with In-Context Learning

- fine-tuning 없이 프루빙 태스크 수행 위해서 LLM의 few-shot in-context learning 능력 활용

- 32개의 example을 manually 작성
  - 16개는 positive knowledge (denoted as E+)
  - 16개는 negative knowledge (E−)

- 실험에서 총 k개의 샘플 랜덤 샘플링, 지정하지 않으면 |E+| = |E-| (개수 동일))

Do LLMs have negative commonsense knowledge?

* The Belief Conflict

Model	k	Perf. on QA			Perf. on CG			Cns.
		TP	TN	Acc	TP	TN	Acc	
Flan-T5 (3B)	2	79.1	84.0	81.5	96.5	19.4	57.9	56.2
	10	82.7	80.2	81.4	96.9	19.8	58.4	59.7
Flan-T5 (11B)	2	84.1	81.0	82.6	97.5	15.9	56.7	57.7
	10	85.4	80.8	83.1	**97.6**	28.2	62.9	65.9
GPT-3	2	76.0	58.9	67.5	83.9	28.4	56.1	54.4
	10	74.7	66.9	70.8	30.9	**79.8**	55.3	53.7
Codex_{002}	2	**89.2**	81.7	**85.4**	96.6	38.0	67.3	70.1
	10	88.1	81.8	84.9	93.2	68.8	81.0	84.5
Instruct-$\text{GPT}^{\text{curie}}_{001}$	2	85.2	51.1	68.2	90.1	21.9	56.0	67.3
	10	70.0	65.8	67.9	71.5	40.8	56.1	58.2
Instruct-GPT_{001}	2	78.1	83.6	80.9	94.9	25.0	60.0	57.7
	10	79.5	81.6	80.6	79.2	55.4	67.3	68.2
Instruct-GPT_{002}	2	81.7	**86.1**	83.9	92.9	48.7	72.1	71.2
	10	84.1	84.7	84.4	88.9	61.4	75.1	77.5
Instruct-GPT_{003}	2	87.9	81.3	84.6	95.1	58.1	76.6	80.5
	10	89.0	79.5	84.2	91.1	73.6	82.3	**87.9**
ChatGPT	2	82.9	82.0	82.4	89.8	69.8	79.8	79.2
	10	81.5	85.7	83.6	90.4	78.4	**84.4**	84.1

Table 1: Main results of different LLMs, which are obtained with k examples ($|E^+| = |E^-|$). **Cns.** denotes the consistency between QA and CG. The best results are **bolded** and the second best are underlined.

- 질문에 대해 LLM은 QA 태스크에서 positive and negative commonsense knowledge에 대해 안정적이고 균형 잡힌 점수 보임

  → LLM은 positive and negative commonsense knowledge를 구분 가능

- Negative 상식에 대해 QA 및 CG 태스크에서 LLM이 일관성 없는 결과 보임

  → *belief conflict*

  - CG 태스크에서 TP and TN 간의 격차
  - QA 및 CG 태스크에서 TN 간의 격차

Do LLMs have negative commonsense knowledge?

* The Belief Conflict

Model	k	Perf. on QA			Perf. on CG			Cns.
		TP	TN	Acc	TP	TN	Acc	
Flan-T5 (3B)	2	79.1	84.0	81.5	96.5	19.4	57.9	56.2
	10	82.7	80.2	81.4	96.9	19.8	58.4	59.7
Flan-T5 (11B)	2	84.1	81.0	82.6	97.5	15.9	56.7	57.7
	10	85.4	80.8	83.1	**97.6**	28.2	62.9	65.9
GPT-3	2	76.0	58.9	67.5	83.9	28.4	56.1	54.4
	10	74.7	66.9	70.8	30.9	**79.8**	55.3	53.7
Codex_{002}	2	**89.2**	81.7	**85.4**	96.6	38.0	67.3	70.1
	10	88.1	81.8	84.9	93.2	68.8	81.0	84.5
Instruct-$\text{GPT}^{\text{curie}}_{001}$	2	85.2	51.1	68.2	90.1	21.9	56.0	67.3
	10	70.0	65.8	67.9	71.5	40.8	56.1	58.2
Instruct-GPT_{001}	2	78.1	83.6	80.9	94.9	25.0	60.0	57.7
	10	79.5	81.6	80.6	79.2	55.4	67.3	68.2
Instruct-GPT_{002}	2	81.7	**86.1**	83.9	92.9	48.7	72.1	71.2
	10	84.1	84.7	84.4	88.9	61.4	75.1	77.5
Instruct-GPT_{003}	2	87.9	81.3	84.6	95.1	58.1	76.6	80.5
	10	89.0	79.5	84.2	91.1	73.6	82.3	**87.9**
ChatGPT	2	82.9	82.0	82.4	89.8	69.8	79.8	79.2
	10	81.5	85.7	83.6	90.4	78.4	**84.4**	84.1

Table 1: Main results of different LLMs, which are obtained with k examples ($|E^+| = |E^-|$). **Cns.** denotes the consistency between QA and CG. The best results are **bolded** and the second best are underlined.

- 질문에 대해 LLM은 QA 태스크에서 positive and negative commonsense knowledge에 대해 안정적이고 균형 잡힌 점수 보임

  → LLM은 positive and negative commonsense knowledge를 구분 가능

- Negative 상식에 대해 QA 및 CG 태스크에서 LLM이 일관성 없는 결과 보임

  → *belief conflict*

  - CG 태스크에서 TP and TN 간의 격차

Do LLMs have negative commonsense knowledge?

* The Belief Conflict

Model	k	Perf. on QA			Perf. on CG			Cns.
		TP	TN	Acc	TP	TN	Acc	
Flan-T5 (3B)	2	79.1	84.0	81.5	96.5	19.4	57.9	56.2
	10	82.7	80.2	81.4	96.9	19.8	58.4	59.7
Flan-T5 (11B)	2	84.1	81.0	82.6	97.5	15.9	56.7	57.7
	10	85.4	80.8	83.1	**97.6**	28.2	62.9	65.9
GPT-3	2	76.0	58.9	67.5	83.9	28.4	56.1	54.4
	10	74.7	66.9	70.8	30.9	**79.8**	55.3	53.7
Codex_{002}	2	**89.2**	81.7	**85.4**	96.6	38.0	67.3	70.1
	10	88.1	81.8	84.9	93.2	68.8	81.0	84.5
Instruct-$\text{GPT}^{\text{curie}}_{001}$	2	85.2	51.1	68.2	90.1	21.9	56.0	67.3
	10	70.0	65.8	67.9	71.5	40.8	56.1	58.2
Instruct-GPT_{001}	2	78.1	83.6	80.9	94.9	25.0	60.0	57.7
	10	79.5	81.6	80.6	79.2	55.4	67.3	68.2
Instruct-GPT_{002}	2	81.7	**86.1**	83.9	92.9	48.7	72.1	71.2
	10	84.1	84.7	84.4	88.9	61.4	75.1	77.5
Instruct-GPT_{003}	2	87.9	81.3	84.6	95.1	58.1	76.6	80.5
	10	89.0	79.5	84.2	91.1	73.6	82.3	**87.9**
ChatGPT	2	82.9	82.0	82.4	89.8	69.8	79.8	79.2
	10	81.5	85.7	83.6	90.4	78.4	**84.4**	84.1

Table 1: Main results of different LLMs, which are obtained with k examples ($|E^+| = |E^-|$). **Cns.** denotes the consistency between QA and CG. The best results are **bolded** and the second best are underlined.

- 질문에 대해 LLM은 QA 태스크에서 positive and negative commonsense knowledge에 대해 안정적이고 균형 잡힌 점수 보임

  → LLM은 positive and negative commonsense knowledge를 구분 가능

- Negative 상식에 대해 QA 및 CG 태스크에서 LLM이 일관성 없는 결과 보임

  → *belief conflict*

  - QA 및 CG 태스크에서 TN 간의 격차

Analysis on the Belief Conflict (1)

* Could keywords as task input hinder the manifestation of LLMs' belief?

- CG and QA에 대한 input 차이로 인해 keywords (CG)보다 natural questions (QA)를 더 쉬운 태스크?
  → 이에 대응하기 위해 두 태스크의 input 서로 변경

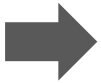
QA input

Answer commonsense questions with yes or no:
(Examples for in-context learning)
Question: do lions live in the ocean?
Answer: no

Can these keywords form a truthful common sense fact? Answer with yes or no.
Keywords: lion, located at, ocean
Answer: no

CG input

Write a short and factual sentence according to commonsense based on the keywords:
(Examples for in-context learning)
Keywords: lion, located at, ocean
Sentence: lions don't live in the ocean.



Answer the question by writing a short sentence that contains correct common sense knowledge.
Question: do lions live in the ocean?
Sentence: lions don't live in the ocean.

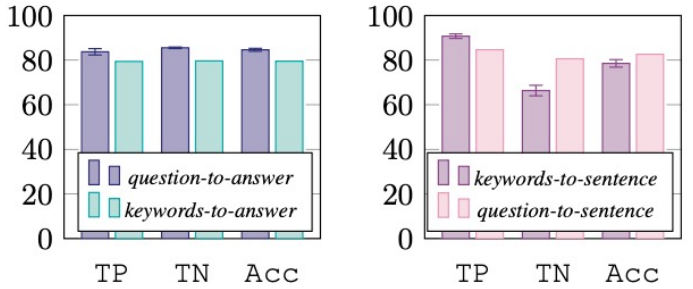
Analysis on the Belief Conflict (1)

* Could keywords as task input hinder the manifestation of LLMs' belief?

- (a): QA 입력으로 keywords(CG) 줬을 때 거의 일관된 성능 보임
  → CG에서 LLM의 불균형한 성능이 input을 keywords로 줬기 때문이 아님

- (b): question as input 성능이 크게 향상되어 QA 와 성능 거의 비슷
  → textual corpus 에서 Boolean question 뒤에 "..?" 같은 negated texts 많이 존재하기 때문에 CG가 QA 태스크로 변질

- keyword-to- sentence (CG) 는 generative LLMs 을 프루빙하기에 적절하고 챌린지한 태스크

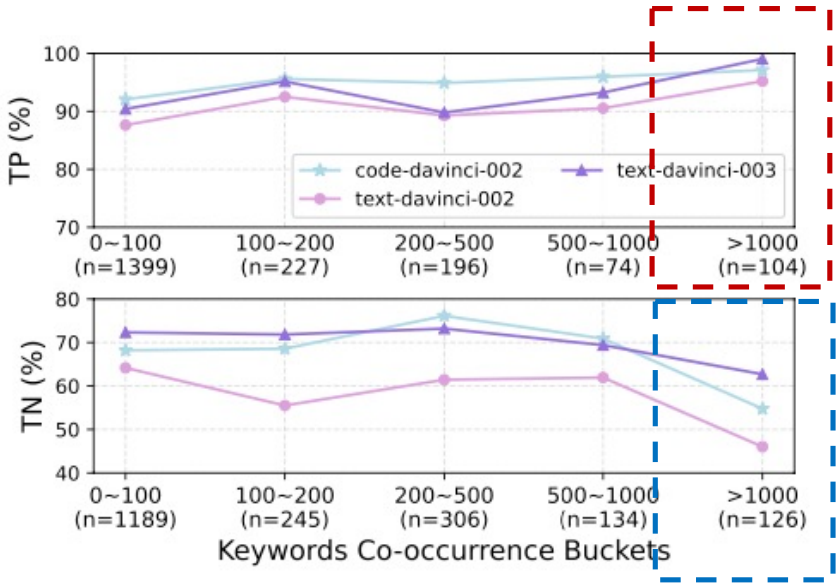


(a) Results (%) on QA. (b) Results (%) on CG.

Figure 4: Results of InstructGPT$_{002}$ when switching the task inputs between *question* and *keywords*, where $k = 10$. Columns with error bars show the ranges of the influence brought by different instruction wordings.

Analysis on the Belief Conflict (2)

* Will the keyword co-occurrence within corpus affect LLMs' generation?



- LLM은 근본적으로 통계 모델임
- CG 태스크에서 가장 일반적인 통계적 요인 중 하나인 word co-occurrence가 미치는 영향 조사
- 가장 많은 양을 차지하는 버킷(> 1000 bucket of the negative split (TN))에서 성능 하락 보임

→ subjects 및 objects의 co-occureence 비율 많아질수록 LLM의 negative 상식 생성 능력 하락

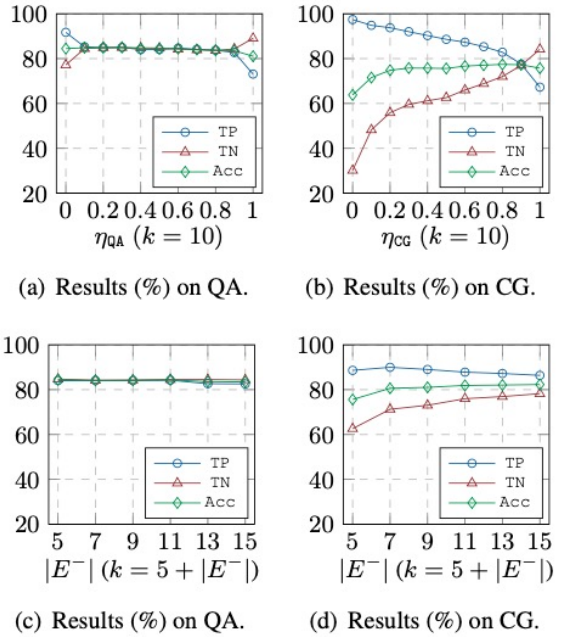
→ statistical shortcuts은 negative 지식 생성 방해

Analysis on the Belief Conflict (3)

* How does the balance of positive and negative examples affect negation bias?

- (a) : QA 태스크를 LLM의 belief로 대응시키는 것이 합당함
- (b): negative 비율이 많아질수록 TN 성능 상승, TP 성능 하락
- (c), (d): positive 개수 유지한 채로 negative 개수 늘린 경우
  - CG 태스크에서 TP 성능 하락 없이 TN 성능 향상

→ training data or in-context examples에서 negated text 비율을 늘리면 bias에 의한 belief conflict을 극복할 수 있는 가능성 보여줌



(a) Results (%) on QA. (b) Results (%) on CG.

(c) Results (%) on QA. (d) Results (%) on CG.

Figure 6: Results of InstructGPT$_{002}$ as the numbers of E^+ and E^- change. Figure (a) and (b) increase $\eta = |E^-|/k$ while fixing $k = 10$. Figure (c) and (d) add more E^- while fixing $|E^+| = 5$.

Analysis on the Belief Conflict (4)

* Do Chain-of-Thought help generate texts with negative commonsense knowledge?

- Explicit 리즈닝 (CoT)를 통해 negative commonsense knowledge 편향 줄일 수 있는지 검증
- CoT instances 가 TN에서 일관된 성능 향상 보임
- 명시적 리즈닝이 negative knowledge 이끌어내는데 도움
- TN가 증가하면 TP의 성능 저하 일어나므로 모델이 TP에 편향되어 있음
  → LLM이 사전학습 중에서 implicit bias를 가지게 되었더라도 CoT를 통해 이를 극복할 수 있음을 시사

Model	CoT	$k = 2$ (1:1)			$k = 10$ (1:1)		
		TP	TN	Acc	TP	TN	Acc
Codex_{002}	None	**96.6**	38.0	67.3	**93.2**	68.8	81.0
	Deduction	86.9	**56.6**	71.7	83.5	73.0	78.3
	Fact	92.9	53.7	**73.3**	86.8	**76.6**	**81.7**
$\text{Instruct-GPT}_{002}$	None	**92.9**	51.4	72.1	**88.9**	61.4	75.1
	Deduction	87.0	**57.3**	72.1	84.3	**70.7**	**77.5**
	Fact	89.1	55.5	**72.2**	85.5	69.2	77.4

Table 2: Performance on the CG task when enhanced with different types of CoT prompting, *i.e.*, deductive argumentation (*Deduction*) and fact comparison (*Fact*).

Discussion

- **Enhancing Reasoning ability of LLM with Instruction-tuning based on Data Transformation**
  - 기존 MCQA 태스크는 commonsense MCQA로 많이 연구되어 옴
  - LLM의 성능 요소 중 하나는 "사람과 비슷한 대답"
    → Commonsense 를 가지고 있어야 한다는 것
    → 이를 평가하기 위해서 기존에는 commonsense reasoning이 필요한 MCQA 연구가 많이 이루어짐
  - LLM이 상식을 충분히 잘 반영하지 못한다는 연구 O
    → LLM 성능 향상 위해서는 MCQA 연구 필요
    → 그러나 한국어에는 MCQA 데이터셋이 없기 때문에 만들어야 함
  - human이 만들기에는 비용 문제 존재하므로, 기존에 있는 리소스를 이용하여 transforming하여 사용
    → 매력적인 오답을 가진 선지를 잘 만들 수록 모델 성능 향상에 도움이 될 것

Thank you