Prompt Auto-Configuration Considering Real-World Scenarios

2024.01.19, Fri

NLP&AI 구선민

Natural Language Processing & Artificial Intelligence

고려대학교 KOREA UNIVERSITY

CESAR: Automatic Induction of Compositional Instructions for Multi-turn Dialogs

Taha Aksu[1 † *], **Devamanyu Hazarika**[2†], **Shikib Mehri**[2], **Seokhwan Kim**[2], **Dilek Hakkani-Tür**[2], **Yang Liu**[2], **Mahdi Namazifar**[2]
[1]National University of Singapore
[2]Amazon Alexa AI
taksu@u.nus.edu, dvhaz@amazon.com

EMNLP 2023 (main)

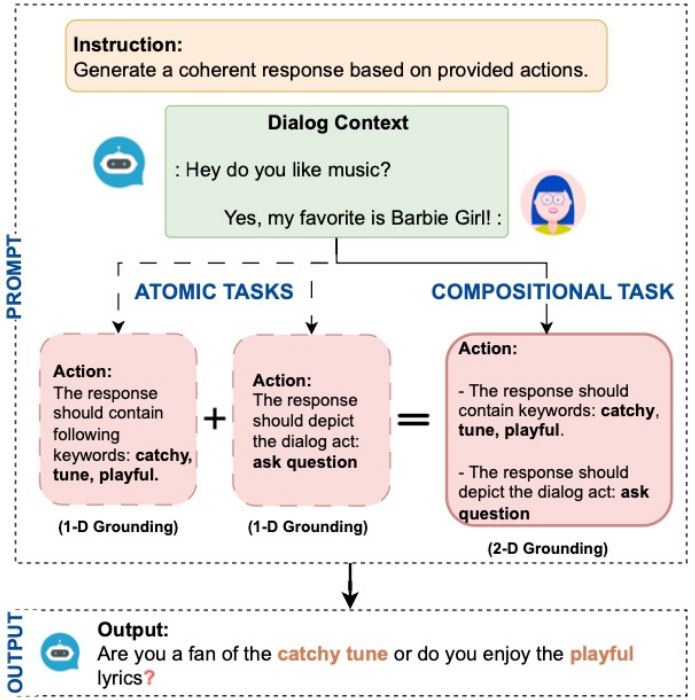
Motivation

- Instruction-based multitasking 은 multi-turn dialog applications에서 높은 LLMs 성능에 큰 영향 미침

- LLM은 instructions에 지정된 다양한 태스크로 학습되므로 새로운 task descriptions 을 쉽게 일반화 가능

- 그러나, Instruction tuning 을 통해 개별 태스크는 잘 수행하지만, 복합 태스크 수행 능력이 필요한 경우는 성능 떨어짐
  - e.g., 프롬프트에서 2가지 control dimensions (task 구성) 을 거친 모델의 response 를 요구함
    (i) response에 3가지 키워드 포함 - 'catchy', 'tune', 'playful'
    (ii) 제시된 dialog act를 함 - ask question

  → 학습 과정 중에 복합 제약 조건을 본 적이 없기 때문

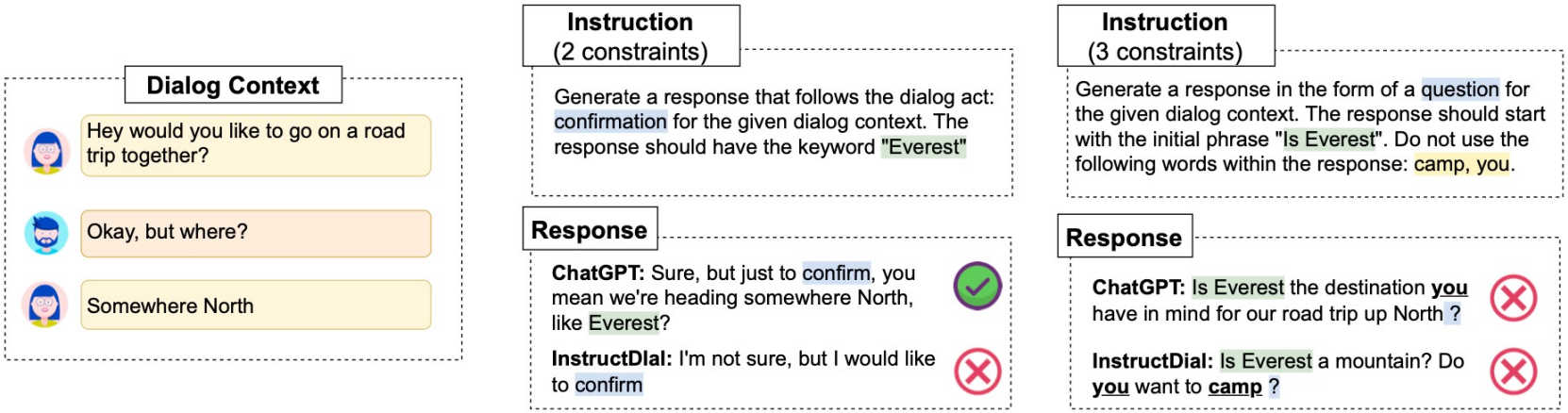


Motivation

- publicly-available LLMs은 높은 성능을 보여주었지만 여러 제약조건(constraints)이 있는 복잡한 instructions에 노출되면 Chat-GPT 같은 SoTA 모델에 비해 성능 떨어짐

- Complex compositional 태스크에서 open and closed-access models 간의 차이를 확인
  - model parameters and data 가 공개적으로 접근가능한지 여부로 구분!

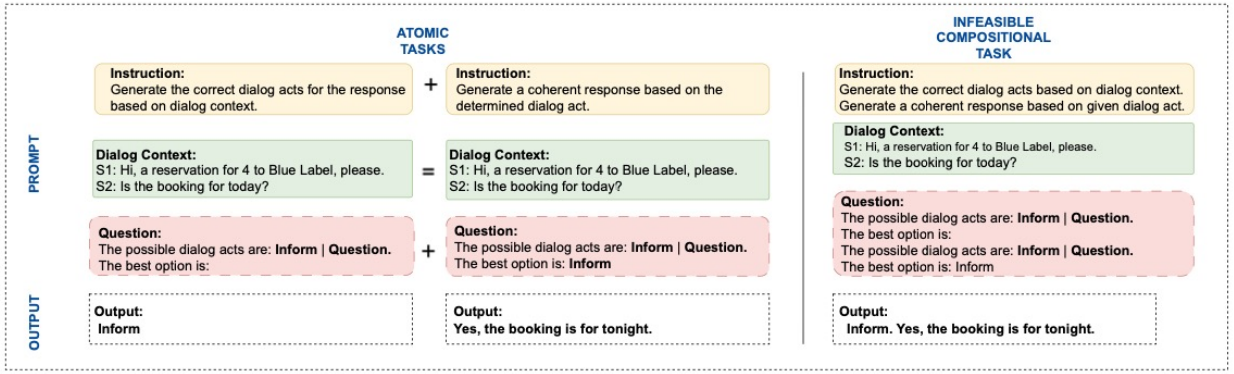
- closed-access models인 ChatGPT는 simple composite tasks 를 잘 수행하나, publicly available models인 DIAL-T0는 어려워함



Motivation

- 기존 연구에서는 태스크 수 확장은 연구되어 있으나 instruction 데이터 구성(compositional data) 확장은 잘 연구되지 않음

- 훈련 단계에서 compositional tasks를 demonstrations로 넣어 complex instruction를 처리할 수 있으나 atomic tasks의 수에 따라 compositions의 수가 기하급수적으로 증가하는 문제점 존재
  → 이를 완화시키기 위해 각각의 태스크 프롬프트와 control sequences 를 결합할 수 있으나 실현 불가능한 조합들을 거르는데 manual effort 가 상당함



→ Focusing on dialog applications, 많은 수의 대화 태스크를 동일한 형식으로 통합하고 manual effort 없이 complex instructions 을 자동으로 유도할 수 있는 새로운 프레임워크인 CESAR 제안

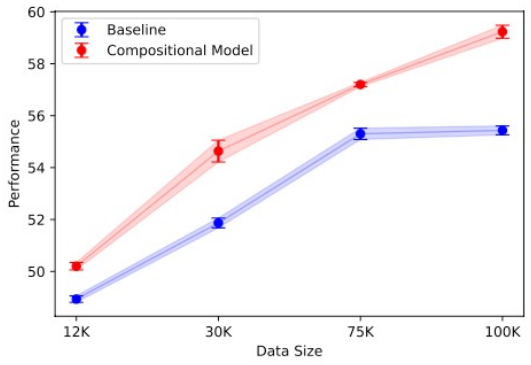
Motivation

*** Can Compositional Demonstrations Improve Performance?**

- compositional demonstrations 가 복잡한 대화 태스크에서 성능을 향상시킬 수 있는지 검증

- 이를 위해 사전 실험을 설계해서 4개의 dialog tasks 태스크 선정
  - response에 포함될 키워드를 컨트롤 하는 대화 response를 생성
    i) beginning phrase, ii) the ending phrase, iii) the length (short, medium, and long), iv) 응답에 포함될 키워드

- Flan-T5-xl model 을 서로 다른 2개의 같은 크기의 훈련 데이터를 fine-tuning하여 2개의 모델 얻음
  i) Baseline – 4개의 atomic tasks 에 대해서만 학습
  ii) Compositional – atomic and compositional tasks 둘 다 학습



CESAR

*** dialog interaction 관련 개념 정의**

- Dialog items (λ)
  - utterances, speaker states, intents, personas, stylistic attributes of utterances, dialog summaries, utterance revisions, external knowledge snippets, amongst others 과 같은 대화 상호작용과 관련된 정보 단위

- Dialog Components : 대화의 논리적 카테고리 ({C, E, S, A, R})
  - C → context (or dialog context); E → evidence(s); S → dialog state(s); A → action(s); R → the dialog response.

- Dialog items 은 mapping function g() 에 의해 dialog components 와 매핑됨

Dialog Components		Sample Dialog Items
C	Dialog context between the two speakers.	utterances
S	State of the dialog context.	dialog summary, speaker intent, etc.
E	Evidences that could be relevant for the response.	retrieved knowledge, persona, etc.
A	Actions/constraints that the response has to follow.	utterance style, dialog act, etc.
R	The next response by the assistant.	response utterance

CESAR

- Input: CESAR 태스크의 입력 프롬프트는 세 가지 주요 구성 요소로 이루어져 있음
  - I: task instruction
  - C : dialog context (두 화자의 이전 발화 포함)
  - $\Lambda = \{g(\lambda_1), \ldots, g(\lambda_m)\}$: I, C 이외에 프롬프트에 추가적으로 들어가는 정보

- Output:
  - $\psi \in \{S, E, A, R\}$ 은 instruction I 및 context C에서 설명하는 task output
  - 각 태스크에는 출력이 필요하므로 ψ는 절대 empty 아님

$$IC\Lambda - \psi$$
$$= IC\underbrace{\{g(\lambda_1), \ldots, g(\lambda_m)\}}_{\text{grounding}} - \psi, \qquad (1)$$

Prompt:
Instruction: Provide the correct value for response fields given the dialog context and action fields.
Input:
Dialog Context:
Speaker 1: What kind of place shall we rent ?
Speaker 2: It should be close to the university . Neither of us are good at getting up in the mornings and closer it is , the later we can get up .
Actions:
The response should start with this initial phrase: "Absolutely . That's"
The response should contain the following keywords: "thing" and "flat"
Response:
Output:
Absolutely. That's the most important thing and flat should be furnished.

CESAR

- 사용자가 AI 어시스턴트와 상호 작용한다고 가정

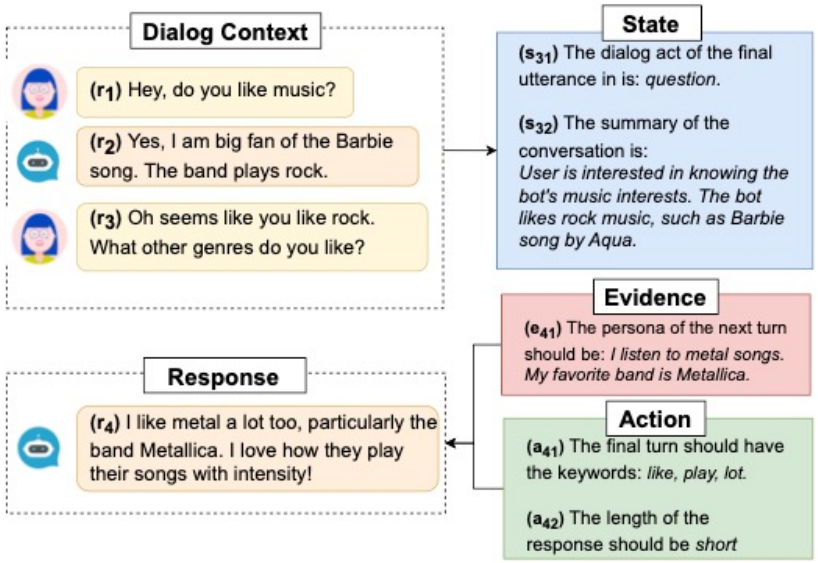


Figure 4: An example dialog with dialog items: utterances (r_1, r_2, r_3, r_4), state (s_{31}, s_{32}), evidence (e_{41}), actions (a_{41}, a_{42}) as described in Table 3a.

CESAR

* Define an n-D CESAR task:

Definition 1 (*n-D Task*)**:** For any CESAR task of the form $IC\Lambda - \psi$, we call the task `n-D Task` if there are n dialog items in Λ, i.e. $|\Lambda| = n$.

* Atomic vs. Compositional Task
- CESAR 에서, 모든 태스크를 atomic 과 compositional task로 구분
  - atomic task: 0-D or 1-D
  - compositional task: n≥2 인 n-D task

Definition 2 (Task Composition)**:** *For two i-D Tasks,*

$$IC\left(\Lambda \cup \{g(\lambda_a)\}\right) - \psi, \text{ and}$$
$$IC\left(\Lambda \cup \{g(\lambda_b)\}\right) - \psi,$$

where, $|\Lambda| = i - 1$ *and* $i \geq 1$*, we combine the two tasks to form an* $(i+1)$*-D Task:*

$$IC\left(\Lambda \cup \{\lambda_a, \lambda_b\}\right) - \psi$$

	Speaker	Utt.	State	Evidence	Action
Input	User	r_1	$\{s_{11}, s_{12}\}$	$\{\}$	$\{a_{11}\}$
	Assistant	r_2	$\{s_{21}\}$	$\{e_{21}\}$	$\{a_{21}\}$
	User	r_3	$\{s_{31}, s_{32}\}$	$\{\}$	$\{a_{31}\}$
Output	Assistant	r_4	$\{s_{41}\}$	$\{e_{41}, e_{42}\}$	$\{a_{41}, a_{42}\}$

(a) For any dialog item x_{ij}, i refers to it's turn number in the dialog and j refers to it's identification within the same dialog component, i.e., S, E, A, or R. for that turn. Fig. 4 provides an example for this setup.

CESAR Task	Input		Output
	C	i-D, Λ	ψ
IC-S	$\{r_1, r_2\}$	0-D, $\{\}$	s_{21}
IC-A	$\{r_1, r_2\}$	0-D, $\{\}$	a_{31}
IC-E	$\{r_1, r_2, r_3\}$	0-D, $\{\}$	e_{41}
ICS-A	$\{r_1, r_2\}$	1-D, $\{s_{21}\}$	a_{31}
ICE-A	$\{r_1, r_2, r_3\}$	1-D, $\{e_{41}\}$	a_{41}
ICA-R	$\{r_1, r_2, r_3\}$	1-D, $\{a_{41}\}$	r_4
ICEA-R or ICAE-R	$\{r_1, r_2, r_3\}$	2-D, $\{e_{41}, a_{41}\}$	r_4
ICSE-R or ICSE-R	$\{r_1, r_2, r_3\}$	2-D, $\{e_{31}, e_{41}\}$	r_4

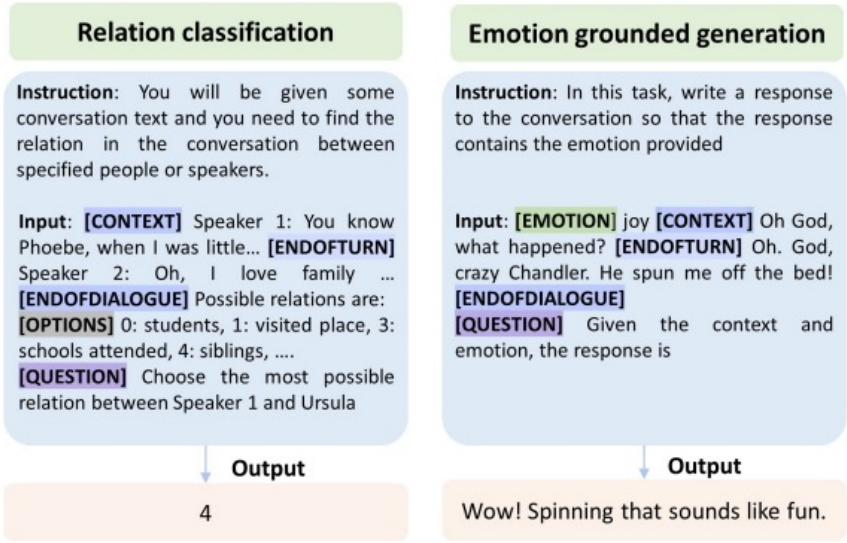
(b) For the given input dialog context $\{r_1, r_2, r_3\}$ and output dialog response $\{r_4\}$ in Table 3a, we provide some example tasks defined under CESAR framework.

InstructDial++

- InstructDial (instruction tuning benchmark for dialogue) 을 업그레이드한 버전인 InstructDial++ 생성

- training benchmarks 에서 데이터를 확장하는 것이 성능에 긍정적인 영향을 미친다는 것이 알려져 있음
  → 15개의 추가 데이터셋과 42개의 새로운 atomic (i.e. 0-D & 1-D) tasks 를 통해 InstructDial 벤치마크를 확장

- Instructdial++ benchmark 은 65개의 데이터셋 및 86개 태스크로 구성됨



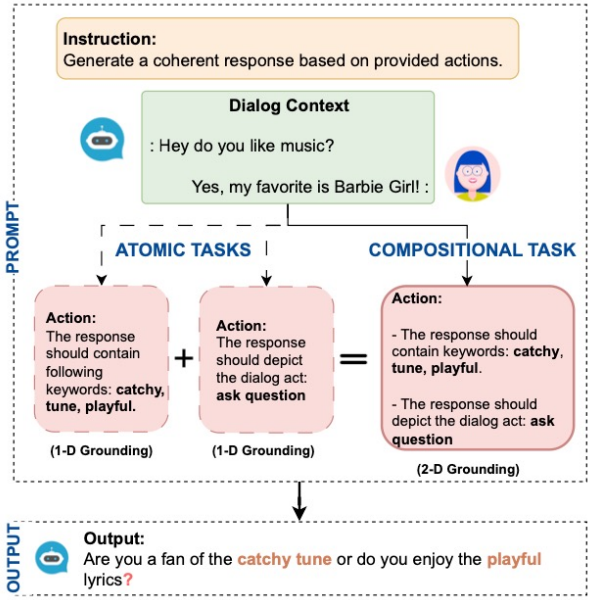
Mapping InstructDial++ to CESAR

* Generative and Discriminative Tasks

- 각 InstructDial 태스크를 CESAR 형식으로 매핑하기 위해 먼저, 입력 제약 조건과 출력 유형에 따라 CESAR 태스크를 할당함

Rule	Task 1	Task 2	Composed Task	Common Dialog Components	Target Field
1	ICA-R	ICA-R	ICAA-R	dc, r	r
2	ICE-R	ICE-R	ICEE-R	dc, r	r
3	ICE-R	ICA-R	ICEA-R	dc, r	r
4	ICS-R	ICE-R	ICSE-R	dc, r	r
5	ICS-R	ICA-R	ICSA-R	dc, r	r
6	ICS-R	ICS-R	ICSS-R	dc, r	r
7	ICE-A	ICS-A	ICAES-A	dc, a	a
8	ICS-S	ICA-S	ICAS-S	dc	s
9	ICA-A	ICS-A	ICASA-A	dc	a
10	ICA-A	ICE-A	ICAEA-A	dc	a

Table 7: List of compositional rules.



Mapping InstructDial++ to CESAR

* 0-D Tasks

- 각 CESAR 태스크는 비슷한 출력 목표를 가진 다운스트림 태스크를 클러스터링
- IC-S task: 대화 컨텍스트 내의 정보를 보다 간결하고 구조화된 방식으로 분류, 형식화 또는 구성하는 태스크(예: 대화 요약)을 포함함
- IC-E task: 대화 컨텍스트에 유용한 외부 지식을 생성하는 태스크
- IC-A task: 키워드나 의도 예측과 같이 응답에서 따라야 할 액션을 생성
- IC-R task: 주어진 대화 컨텍스트에 대한 응답을 생성/선택하는 태스크

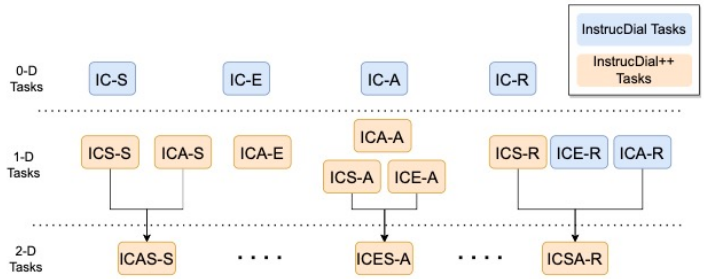


Figure 6: Comparsion of CESAR tasks in InstructDial++ vs. their counterparts in InstructDial.

Mapping InstructDial++ to CESAR

* 1-D Tasks

- 1-D tasks 은 대화 컨텍스트 이외의 추가 구성 요소를 기반으로 한다는 점을 제외하면 0-D 태스크와 마찬가지로 생성도 S, E, A, R 구성 요소 중 하나를 대상으로 하기 때문에 분류가 동일

- 예를 들어, ICA-R의 경우 response generation task은 제공된 action,(예: controlled generation or slot-value 그라운딩 생성)에 따라 추가적으로 조건이 지정됨

- IC-R 태스크과 달리 수정해야 할 응답의 컨텍스트와 이전 버전을 모두 기반으로 하기 때문에 해당 카테고리에 edit generation 도 포함

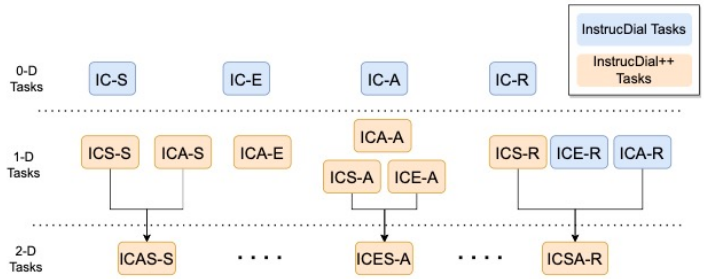


Figure 6: Comparsion of CESAR tasks in InstructDial++ vs. their counterparts in InstructDial.

Mapping InstructDial++ to CESAR

* 2-D Tasks

- 0-D 및 1-D 에 해당하는 모든 태스크를 수동으로 매핑한 후, CESAR 프레임워크는 pre-defined rules 에 따라 실행 가능한 2-D 태스크를 자동으로 선택하고 구성함

- 조건 1: CESAR compositional tasks 가 작동하는 방법에 대한 예시 제공
  - e.g., ICAA-R compositions 은 서로 다른 ICA-R 태스크를 결합한 것을 의미

- 조건 2: Key “common fields” 는 composition에서 각 태스크에서 반복되는 모듈을 나타냄
  - e.g., “dialog context” 및 “response”가 해당

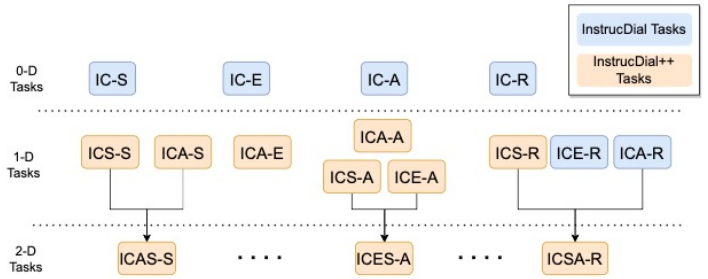


Figure 6: Comparsion of CESAR tasks in InstructDial++ vs. their counterparts in InstructDial.

Experiments

* Models

- ChatGPT model gpt-3.5-turbo-16k-0613
- T0-3B (Sanh et al.,2021) which is trained on a mixture of downstream tasks
- DIAL-T0: fine-tuned on InstructDial dataset and based on T0-3B
- DIAL-BART0: fine-tuned on InstructDial dataset and based on BART0
- DIAL-FLAN-xxl: baseline model based on InstructDial dataset using FLAN-xxl
- CESAR-FLAN-xxl: main model

Experiments

* Atomic Tasks

- Begins With Generation (BW): 주어진 구문으로 시작하는 응답 생성
- Ends With Generation (EW): 지정된 구문으로 끝나는 응답 생성
- Keyword Controlled Generation (KC): 지정된 키워드셋을 포함하는 응답 생성
- Length Controlled Generation (LC): 특정 길이 (short/medium/long) 의 응답 생성
- Persona Based Generation (PB): 주어진 speaker persona 에 따라 응답 생성
- Knowledge-Based Generation (KB): 외부 지식 기반 응답 생성
- Edit Generation (EG): 응답을 수정하여 대화 context와 일관성 유지하도록 함

- 평가의 표준화를 위해 모든 atomic task에 대해 InstructDial과 동일한 메트릭 사용

Experimental Results

* Atomic Task Performance

- 모든 태스크에서 1등은 아니지만 베이스라인과 비교했을 때 많은 태스크에서 비슷하거나 향상을 보임

Model	Training Set	BW	EW	KC	LC	PB		KB	
		Acc	*Acc*	*Acc*	*Acc*	*Bleu-2*	*R-L*	*Bleu-2*	*R-L*
T0-3B	Zero-shot	13.12	15.74	20.14	43.38	1.44	7.7	3.34	10.44
DIAL-BART0	InstructDial	84.02	61.46	87.24	44.92	4.73	14.5	**10.83**	**21.9**
DIAL-T0	InstructDial	**86.46**	60.84	74.38	50.56	4.43	14	10	20.3
DIAL-FLAN-xxl	InstructDial	81.4	62.12	86.04	53.26	4.23	13.66	9.32	18.96
CESAR-FLAN-xxl	InstructDial++	84.60	**65.66**	**88.08**	**82.98**	**4.81**	**14.5**	9.76	20.06

Table 4: Evaluation results on atomic tasks. *R-L* stands for *Rouge-L* metric, best results for each column are **bold**.

Experimental Results

* Compositional Task Performance

- 모든 태스크에서 베이스라인 뛰어넘는 성능 보임
- atomic 태스크에서 좋은 성능이 반드시 해당 태스크의 compositional 태스크의 성능으로 이어지지는 않는 모습

Model	Training Set	BW + EW	BW + KC	BW + LC	EW + KC	EW + LC	KC + LC	PB + EW		EG + EW	
		Acc	Acc	Acc	Acc	Acc	Acc	EW Acc	R-L	EW Acc	R-L
T0-3B	Zero-shot	2.38	4.39	5.75	2.95	6.92	8.55	5.4	11.56	22.5	29.9
DIAL-BART0	InstructDial	69.01	77.2	38	46.3	11.8	16.65	82.8	50.2	85.2	83.2
DIAL-T0	InstructDial	72.9	72.8	42.5	49.1	27.4	35.2	76.9	46.1	84.5	77.2
DIAL-FLAN-xxl	InstructDial	70.1	78.53	44.25	58.04	30.15	44.4	82.53	49.05	89.3	85.33
CESAR-FLAN-xxl	InstructDial++	**75.18**	**83.08**	**70.43**	**62.7**	**47.2**	**68.6**	**88.1**	**51.9**	**94.8**	**93.10**

Table 5: Evaluation results on compositional tasks. *R-L* stands for *Rouge-L* metric, best results for each column are **bold**.

SELF-ICL: Zero-Shot In-Context Learning with Self-Generated Demonstrations

Wei-Lin Chen* **Cheng-Kuang Wu*** **Yun-Nung Chen** **Hsin-Hsi Chen**
National Taiwan University, Taiwan
{wlchen,ckwu}@nlg.csie.ntu.edu.tw
y.v.chen@ieee.org hhchen@ntu.edu.tw

EMNLP 2023 (main)

Motivation

- LLMs은 demonstrations 을 통한 in-context learning (ICL)로 새로운 태스크에 adapt 하는 능력 보여줌

- 기존 연구에서는 대규모 외부 소스(e.g., training dataset or relevant text corpus) 에 대한 접근이 가능하다고 가정
  - 그러나 실제 시나리오에서 사용자는 소스에 접근하지 않고도 LLM을 쿼리할 수 있음
    (e.g., API 또는 웹 인터페이스를 통해)

  - 또한 handcraft demonstration 은 오히려 성능에 부정적 영향 미칠 수 있음

- Demonstration 은 새로운 태스크를 학습하기 보다는 LLMs이 이미 가지고 있는 능력을 타겟 도메인으로 가이드 하는 역할을 한다고 밝혀짐
  - CoT, instruction-augmented ICL 연구에서..

  → LLM의 제로샷 능력이 과소 평가 되었으며, 이미 다양한 타켓 태스크를 수행할 수 있는 잠재적 능력을 가지고 있음을 나타냄

Overview

- Zero-shot in-context learning을 위한 간단한 프롬프트 프레임워크인 SELF-ICL을 제안
  - ICL 수행을 위한 input and label space 을 알려주는 self-generated demonstrations를 통해 LLM의 잠재적 기능을 이끌어냄

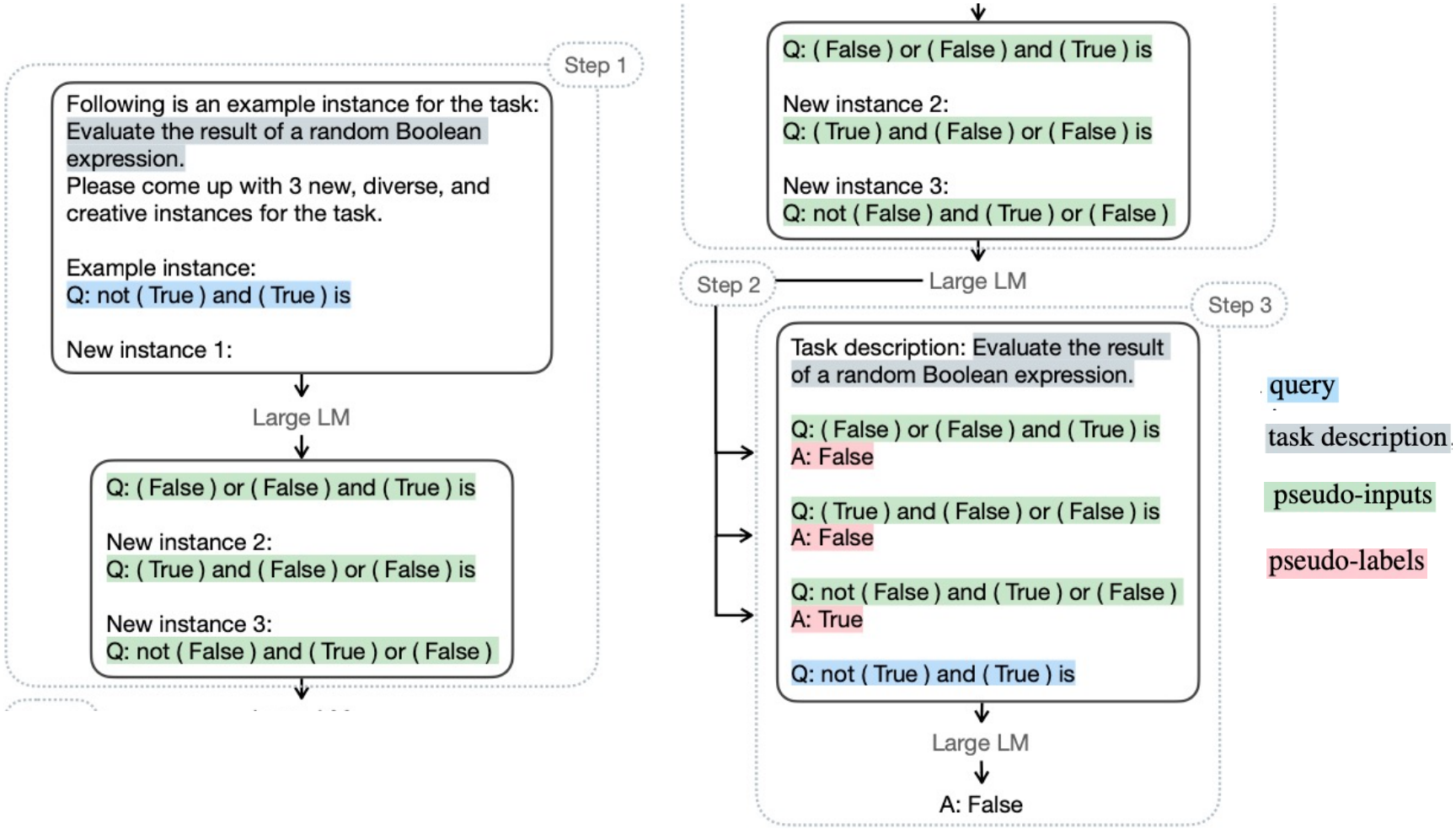
- Query (i.e., a test input)가 주어지면 SELF-ICL은 세 단계로 구성됨
  1. 모델에 주어진 쿼리와 해당 태스크 demonstration에 따라 조건이 지정된 pseudo-inputs을 생성하라는 메시지가 표시

  2. 모델이 제로 샷 프롬프트를 통해 pseudo-inputs에 대한 pseudo-labels을 예측

  3. 생성한 pseudo-input-label pairs 를 demonstrations로 사용하여 input에 대해 ICL을 수행

Method	Inputs	Labels
AUTO-COT	from training set	*no need*
Z-ICL	from external corpus	*no need*
SG-ICL	*no need*	given
SELF-ICL (ours)	*no need*	*no need*

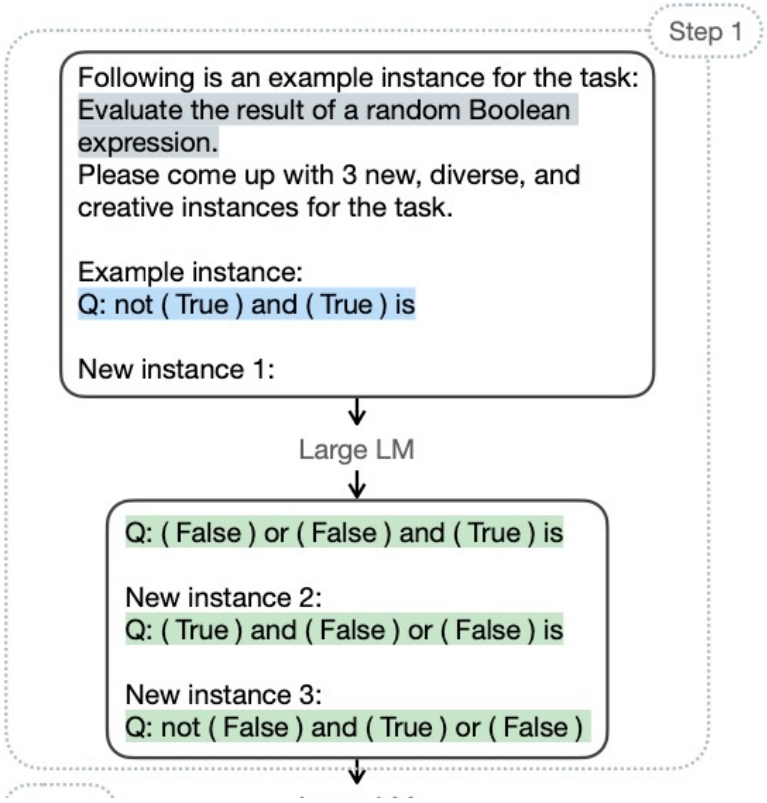
SELF-ICL framework



SELF-ICL framework

* Pseudo-Input Construction (Step 1)

- query q는 ground-truth inputs 형식 예시 제공
- task description T는 타겟 태스크 관련 정보를 생성하도록 가이드
- q와 T로 부터 모델은 format을 추론하고, 새로운 쿼리 (i.e., pseudo-input)를 생성



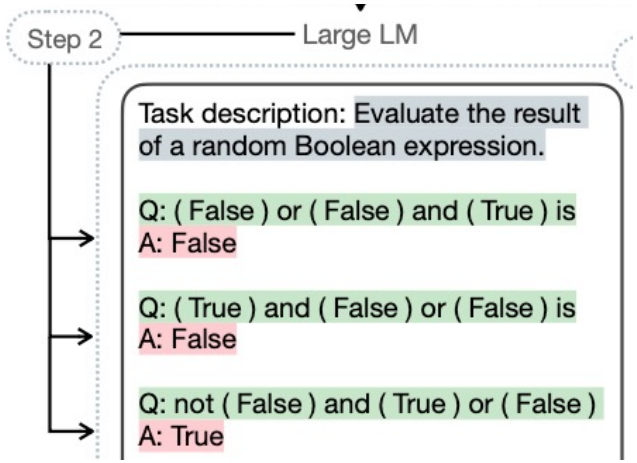
SELF-ICL framework

* Pseudo-Label Construction (Step 2)

- pseudo-inputs 를 얻은 뒤, 동일한 LLM에 제로 샷 프롬프트를 통해 labels(the pseudo-labels for constructing pseudo-demonstrations)을 예측함

- Direct prompting
  - task description and the generated pseudo-input 사용하여 LLM 프롬프트 제공

- CoT prompting
  - task description, the current test input
    + a trigger phrase, “Let’s think step by step.”

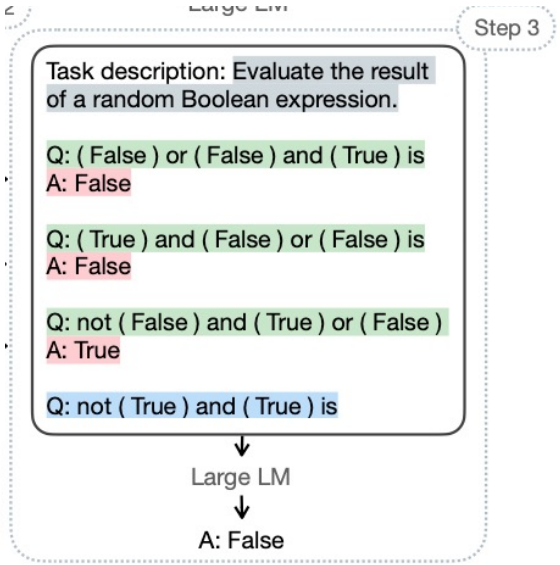


SELF-ICL framework

* Prediction (Step 3)

- pseudo inputs-outputs pair로 pseudo-demonstrations(i.e., pseudo-shots) 구성하고 일반적인 few-shot ICL 흐름에 따라 test input에 대한 최종 answer를 예측함

- 즉, pseudo-shots (with instructions)이 test input에 추가되어 prompting의 context로 사용



Experiments Settings

* Language models

- InstructGPT (text-davinci-003) 사용
- SELF-ICL의 일반화 성능을 측정하기 위한 추가 실험은 PaLM-2 모델 계열의 text-bison-001과 GPT-3.5 계열의 gpt-3.5-turbo-instruct 사용

* Dataset
- BIG-Bench Hard (BBH) benchmark 사용
  - BIG-Bench benchmark 중에서 현재 모델 성능으로는 아직 인간 수준을 넘어서지 못하는 어려운 태스크들만 모아놓은 것 (27개 태스크로 구성)
  - 이 중 23개의 multiple-choice 태스크를 SELF-ICL 평가 데이터로 사용
    - e.g., data_understanding(날짜 이해), hyperbaton (비유) …

Today is Christmas Eve of 1937. What is the date tomorrow in MM/DD/YYYY? Options: (A) 12/11/1937 (B) 12/25/1937 (C) 01/04/1938 (D) 12/04/1937 (E) 12/25/2006 (F) 07/25/1937	(B)
In the UK, people usually put the day before the month when formatting the date. Therefore, today is 02/01/1987 to them. What is the date a…	(A)

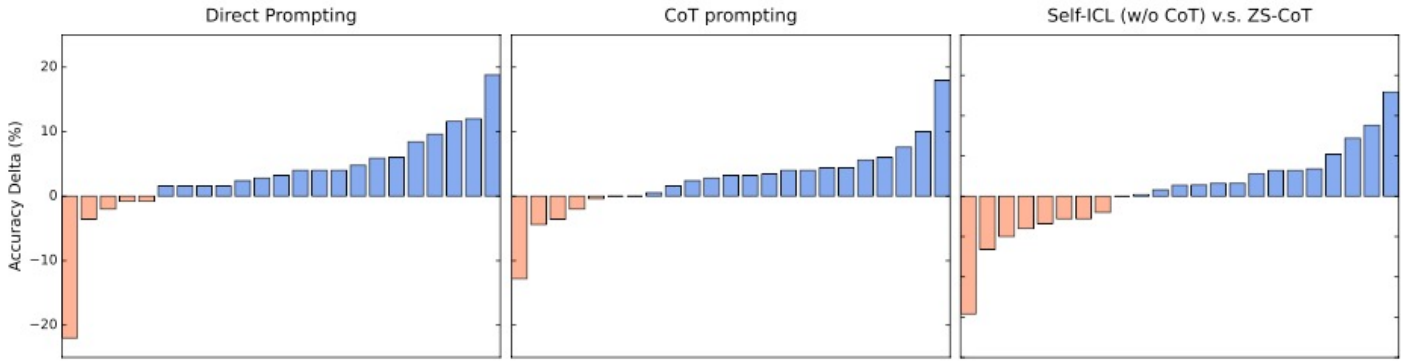
Main Results

- 모든 태스크의 평균 성능에서 direct and CoT prompting setting보다 SELF- ICL의 성능이 더 좋음
- SELF-ICL with direct prompting 은 ZS-CoT prompting과 비슷한 수준
- SELF- ICL with CoT prompting 은 zero-shot인데 few-shot과 성능 비슷하다!

BBH Task	Direct Prompting			CoT Prompting			3-shot
	ZS-Direct	Self-ICL	δ	ZS-CoT	Self-ICL	δ	
Boolean Expressions	84.00	**87.20**	3.20	85.60	**85.60**	0.00	89.60
Causal Judgement	55.61	**61.50**	5.88	58.29	**58.82**	0.53	62.03
Date Understanding	54.00	**56.80**	2.80	66.80	**69.20**	2.40	58.80
Disambiguation QA	64.00	**68.00**	4.00	64.80	**67.60**	2.80	68.80
Formal Fallacies	56.00	55.20	-0.80	55.20	**56.80**	1.60	58.80
Geometric Shapes	34.40	**36.00**	1.60	29.60	**33.60**	4.00	36.80
Hyperbaton	56.80	**59.20**	2.40	53.60	**53.60**	0.00	57.60
Logical Deduction (five objects)	41.20	40.40	-0.80	37.60	**40.80**	3.20	45.60
Logical Deduction (seven objects)	43.60	40.00	-3.60	39.60	36.00	-3.60	40.00
Logical Deduction (three objects)	56.00	**65.60**	9.60	58.80	**64.80**	6.00	64.40
Movie Recommendation	63.60	**75.20**	11.60	64.80	**70.40**	5.60	77.20
Navigate	49.20	**68.00**	18.80	53.60	**57.60**	4.00	52.80
Penguins in a Table	58.90	**63.70**	4.79	60.96	**64.38**	3.42	62.33
Reasoning about Colored Objects	59.60	**61.20**	1.60	68.00	67.60	-0.40	65.20
Ruin Names	54.40	**66.40**	12.00	48.80	**56.40**	7.60	84.00
Salient Translation Error Detection	51.20	**57.20**	6.00	50.80	**54.00**	3.20	65.20
Snarks	54.49	**62.92**	8.43	37.08	**55.06**	17.98	65.73
Sports Understanding	67.60	65.60	-2.00	71.20	69.20	-2.00	71.20
Temporal Sequences	57.60	35.60	-22.00	64.80	52.00	-12.80	39.20
Tracking Shuffled Objects (five objs)	18.00	**19.60**	1.60	25.20	**29.60**	4.40	16.40
Tracking Shuffled Objects (seven objs)	14.40	**18.40**	4.00	31.60	27.20	-4.40	15.20
Tracking Shuffled Objects (three objs)	26.40	**28.00**	1.60	36.00	**46.00**	10.00	30.40
Web of Lies	53.20	**57.20**	4.00	61.20	**65.60**	4.40	56.40
All Tasks (avg)	50.81	**53.93**†	3.12	53.22	**55.54**†	2.32	55.49

Main Results

- 23개 태스크에 대한 일대일 비교 Figure
- direct prompting 의 경우 SELF-ICL vs the ZS-Direct baseline : 18-0-5 (win-tie-lose)
- CoT prompting의 경우 SELF-ICL vs the ZS-CoT baseline: 16-2-5
- SELF-ICL without CoT (SELF-ICL with direct prompting) vs ZS-CoT 는 14-1-8
  - 잠재적 편향이나 잘못된 answer를 유도할 수도 있는 reasoning chains를 생성하지 않고 제로샷에서 LM의 추론 능력을 이끌어냈기 때문



blue/orange indicates SELF-ICL wins/loses

Generalizability

- 제안한 SELF-ICL 프레임워크가 다른 모델에 일반화할 수 있는지 평가하기 위해, InstructGPT 외에 널리 사용되는 두 가지 LLM인 GPT- 3.5와 PaLM-2에서 실험 수행

- 아래 실험 결과를 통해 SELF-ICL이 다른 모델에 일반화할 수 있음을 시사함!

BBH Task	text-bison-001			gpt-3.5-turbo-instruct		
	ZS-Direct	Self-ICL	δ	ZS-Direct	Self-ICL	δ
Boolean Expressions	60.16	58.94	-1.22	84.80	**88.40**	3.60
Causal Judgement	47.37	**48.54**	1.17	42.25	12.30	-29.95
Date Understanding	42.40	41.20	-1.20	59.20	57.60	-1.60
Disambiguation QA	33.33	**33.82**	0.49	60.00	**63.20**	3.20
Formal Fallacies	57.20	56.00	-1.20	52.00	50.40	-1.60
Geometric Shapes	15.20	**19.20**	4.00	34.00	**36.40**	2.40
Hyperbaton	57.43	**68.27**	10.84	82.40	**82.80**	0.40
Logical Deduction (five objects)	18.34	**23.58**	5.24	42.00	38.40	-3.60
Logical Deduction (seven objects)	10.88	**13.99**	3.11	41.60	34.80	-6.80
Logical Deduction (three objects)	37.89	37.00	-0.88	56.00	**59.20**	3.20
Movie Recommendation	26.23	**26.23**	0.00	74.80	**76.00**	1.20
Navigate	58.71	57.21	-1.49	42.80	**64.80**	22.00
Penguins in a Table	42.76	**42.76**	0.00	51.37	**55.48**	4.11
Reasoning about Colored Objects	62.40	**70.80**	8.40	54.80	**56.40**	1.60
Ruin Names	30.81	26.16	-4.65	70.80	64.80	-6.00
Salient Translation Error Detection	22.13	**22.54**	0.41	41.60	**51.20**	9.60
Snarks	54.86	50.86	-4.00	63.48	60.67	-2.81
Sports Understanding	46.45	45.90	-0.55	62.00	50.00	-12.00
Temporal Sequences	28.51	**30.58**	2.07	20.80	**32.80**	12.00
Tracking Shuffled Objects (five objects)	14.52	**17.74**	3.23	18.00	16.40	-1.60
Tracking Shuffled Objects (seven objects)	20.00	19.60	-0.40	17.60	12.40	-5.20
Tracking Shuffled Objects (three objects)	29.32	**32.93**	3.61	32.40	**36.80**	4.40
Web of Lies	57.20	50.40	-6.80	15.20	**38.40**	23.20
All Tasks (avg)	37.78	**38.83**†	1.05	48.52	**49.72**†	1.20

Analysis

- generated pseudo-inputs에 대한 다양한 방법론들(shot의 수, pseudo-labels를 랜덤으로) 을 포함하는 다양한 세팅에서 SELF-ICL 결과 조사

- $\{(x_1, y_1), \ldots, (x_k, y_k)\}$ 로 표시되는 k-shot demonstrations 집합이 주어짐
  - x_i is the input text
  - y_i is the label

- demonstrations 구성을 위해 4가지 측면 고려함 (기존 연구 따름)
  (1) input-label mapping: x_i 가 올바른 y_i 와 쌍을 이루는지 여부

  (2) input space: the underlying distribution behind x_i , ..., x_k

  (3) label space: $y_1, \ldots, y_k$ 에서 유추 가능한 레이블 집합
    - input with instruction-like descriptions 은 모델에게 label space 알려줄 수도 있음

  (4) pairing format: the format representing the $x_i - y_i$ pair.

Analysis

* The Entanglement of Input Space and Input-Label Mapping

- 4가지 측면 중 label space 은 input 에 지정되거나 task description에 설명되어 있음
  - e.g., input : options presented for multiple-choice
  - e.g., task description: “Evaluate the result of a random Boolean expression.”

- 잠재적으로 문제가 될 수 있는 부분은 input space and input-label mapping 임
  - 주어진 inputs를 복사하여 labels를 생성하는 copying effect 가 발생할 수도 있기 때문

Analysis

*** Different Approaches for Generating Pseudo-Inputs**

- copying effect* 의 영향을 완화하려면 pseudo-inputs의 다양성을 높이는 것이 필수적임
  *주어진 inputs를 복사하여 labels를 생성하는 경향성

- SELF-ICL's pseudo-input 생성 및 potential copying effect에 대한 이해를 향상시키기 위해 pseudo-inputs를 구축할 때 3가지 방법론에 대해 조사함
  (1) Batch inference
  : Step 1에서 주어지는 inputs의 수를 여러 개로 함

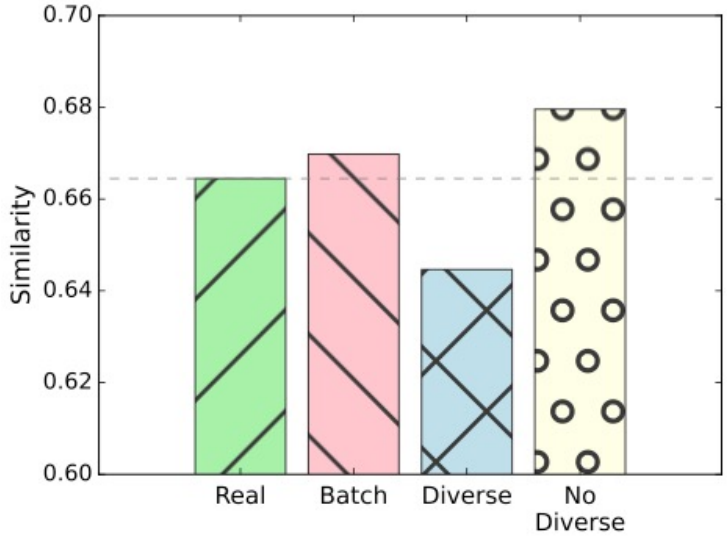
  (2) Prompting with diversity hints
  : "new", "diverse", and "creative" pseudo-input instances 를 제공하도록 모델에 명시적 inst.

  (3) Prompt without diversity hints
  : inst.에서 keywords "new", "diverse", and "creative" 제거하고 나머지 세팅은 동일하게 유지

Analysis

* Different Approaches for Generating Pseudo-Inputs

- 각 방법으로 생성된 pseudo-inputs 과 test input 간의 코사인 유사도 계산한 결과 보여줌
- batch inference가 real inputs과 가장 유사한 pseudo-inputs 생성함



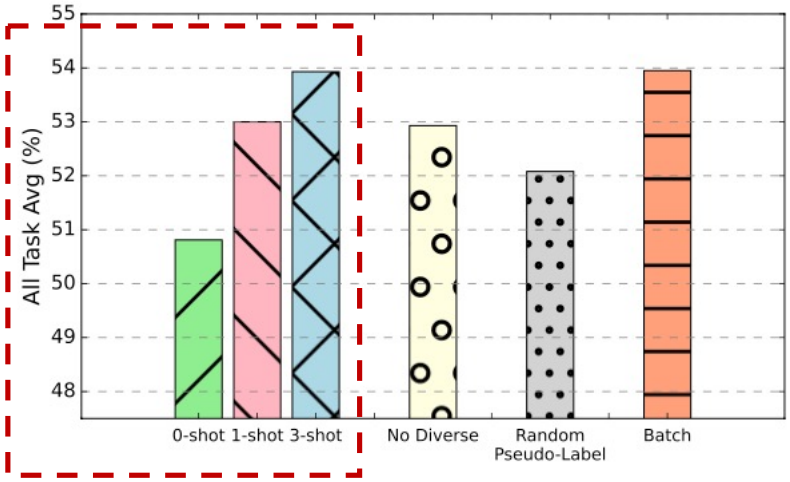
Analysis

*** Effect of Different Number of Shots**

- 다양한 pseudo-demonstration shots 수에서 SELF-ICL의 성능을 조사함

- 3-shot setting 은 SELF-ICL의 메인 실험에서 채택한 방법이며, 1-shot setting 은 3-shot setting 에서 랜덤 샘플링해서 사용
- 0-shot setting ZS-Direct 베이스라인임

→ 1-shot 은 3-shot 보다는 성능 떨어지지만 제로샷보다는 눈에 띄게 좋은 성능 보이며 SELF-ICL의 효과성 입증



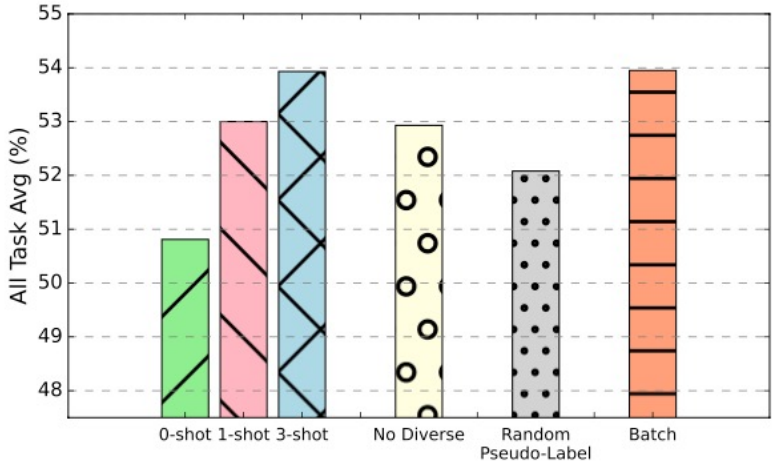
Analysis

* Effect of Random Pseudo-Labels

- Pseudo-labels의 품질을 검증하기 위해, Step 2에서 생성된 pseudo-labels 중 랜덤으로 뽑아 대체하여 random pseudo-labels 로 pseudo- demonstration 를 구성하여 test input's answer를 예측함

- 실험 결과 random pseudo-labels는 few shot보다는 성능 낮지만 제로샷보다는 높은 성능 보여줌

- random labels 을 사용했을 때 나타나는 성능 저하는 일부 인스턴스에서 LLM이 demonstration labels을 정답으로 인식하고 그에 따라 예측을 수행했기 때문



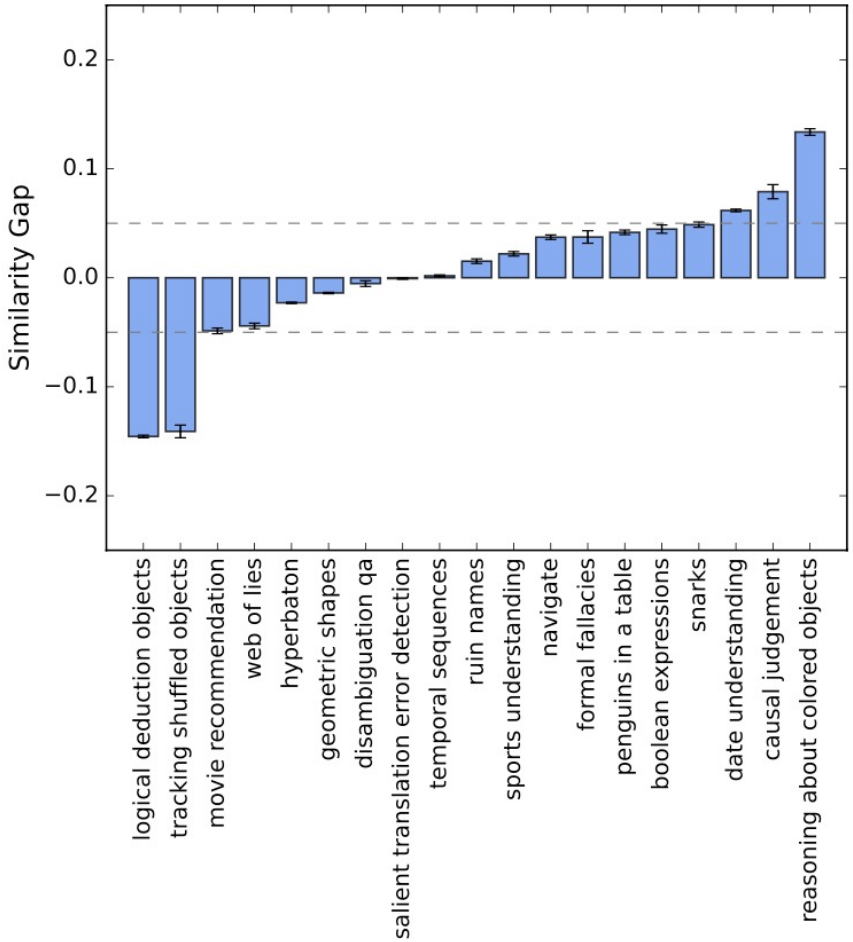
Analysis

* A Deeper Look of SELF-ICL's Pseudo-Inputs

- generated pseudo-inputs의 다양성과 copying effect 위험성을 완화하기 위해 프롬프트에 phase 추가
  - prompting LLMs to be diverse with key words "new", "diverse", and "creative".

- generated pseudo-inputs이 훈련 데이터에서 랜덤으로 샘플링된 real inputs과 비교하여 충분히 유사한지 정량적으로 검증
  - query-input distance와 real-inputs 간의 유사도 차이

- 격차가 클수록 쿼리가 real-inputs sampled from training set보다 pseudo-inputs에 더 가깝고 copying effect의 영향을 받을 가능성이 높다는 것을 나타냄



Insights

- 두 논문 모두 real-world 시나리오를 언급하면서 설득
  → 문제는 거시적으로, 해결은 타겟팅해서..

- CESAR 처럼 기존에 존재하는 데이터셋에 추가 태깅하여 사용하면 조금은 쉽게 contribution 인정해주는듯

- LLMs 성능 향상을 위해 양질의 instruction 중요한데 이거를 자동화 해보자!
  - CoT, Task decomposed 등 프롬프트 방법론은 너무 다양하니 자동화 쪽으로 트는 것 같기도..

감사합니다