여름방학 세미나

홍성태
ghdchlwls123@korea.ac.kr
2024.08.29

Not all Layers of LLMs are Necessary during Inference

Siqi Fan[2], Xin Jiang[1], Xiang Li[1], Xuying Meng[3], Peng Han[2], Shuo Shang[2*], Aixin Sun[4], Yequan Wang[1*], Zhongyuan Wang[1]

[1]Beijing Academy of Artificial Intelligence, Beijing, China
[2]University of Electronic Science and Technology of China, Chengdu, China
[3]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[4]School of Computer Science and Engineering, Nanyang Technological University, Singapore

Arxiv 2024.05

Not all Layers of LLMs are Necessary during Inference

Introduction

LLM inference cost problem

- LLM 추론 단계에서 비용이 매우 많이 요구됨
- 추론 시 기존의 성능을 유지하면서 적은 계산 자원을 사용해야 함

LLM Pruning

- LLM에서 효율적 추론을 위한 방법으로 Parameter를 직접적으로 변경하여 일반화 능력의 손상 위험하며 분석하기 어려움

(1) LLM 추론을 가속화하기 위한 방법으로 뉴런의 수를 동적으로 감소시키는 방법론을 고려

(2) 인간의 사고 과정으로부터 영감 ("Easy" tasks activate at shallower layers while "hard" ones at deeper layers.)

(3) Input Instance에 따라 언제 멈출지를 결정하는 것이 자연스러운 접근법임을 주장

👉 Parameter를 변경하지 않는 Early Termination 전략을 사용하여 효율성 최적화하는 **AdaInfer** 제안

Efficiency Analysis of LLM inference

Not all Layers are Necessary

- Can we allocate fewer computational resources per input instance instead of the same substantial budget?
- 이를 위해 태스크별 accuracy와 activation Layers간의 Correlation에 대한 통계적 분석 진행

⭐ Observation ⭐

1) *Not all Layers of LLMs are Necessary during Inference*

2) *Varying Task Difficulties, Different Activation Layers: Stop Simpler Tasks Sooner, Let Complex Ones Go Deeper*

Not all Layers of LLMs are Necessary during Inference

Efficiency Analysis of LLM inference

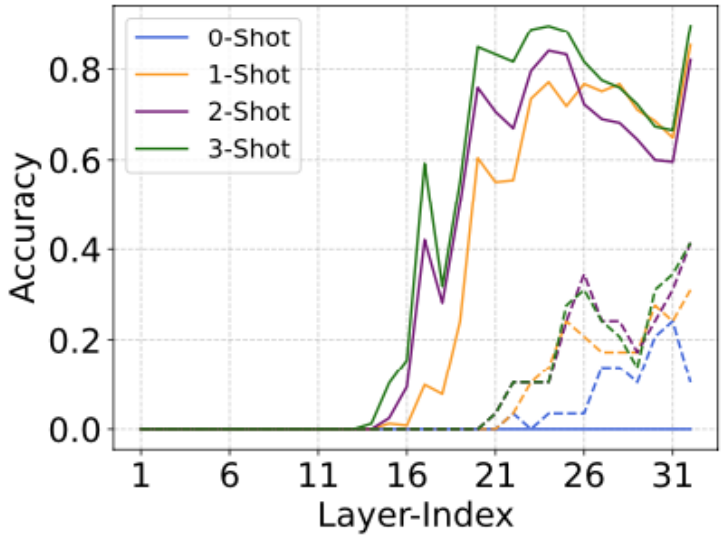


Figure 1: LLama2-7B model zero/few-shot performance across all decoder layers: solid line for sentiment analysis while dashed line for MMLU tasks.

Observation 1 & 2

Ob1-1) 최종 레이어까지 사용한 실험의 acc를 이전 layer에서도 관찰할 수 있음

Ob2-1) 감정분류와 같은 쉬운 태스크는 24번째 layer에서 Final layer output과 유사한 정확도를 보임

Ob2-2) MMLU와 같은 태스크에서 더 깊은 layer에서 정확도가 향상

→ 위 분석을 이용, 추론 효율성에 뛰어난 이점을 가지는 Adaptive Inference 구현 가능성 확인

Not all Layers of LLMs are Necessary during Inference

AdaInfer

어떻게 early stop signal을 찾을 수 있는가?

- Gap / Top Prob을 이용하여 동적으로 계산 후 사용
- (1) Feature Selection & (2) Classifier 두가지 모듈을 포함

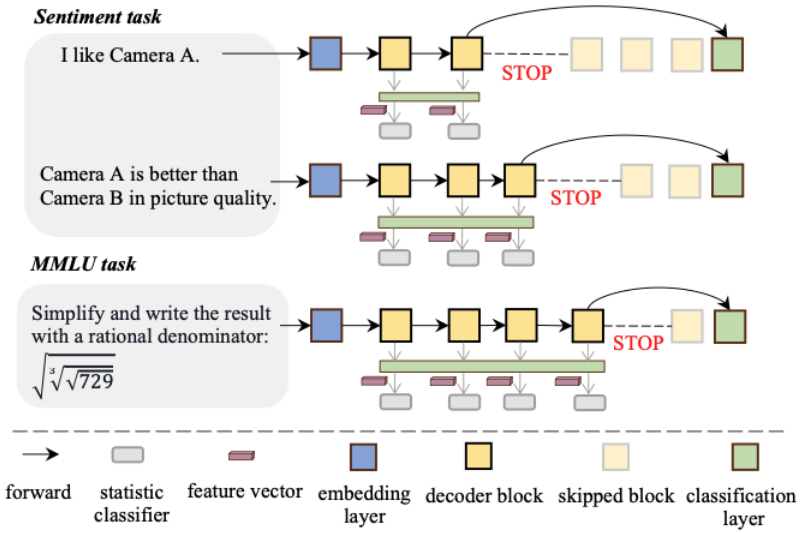
(1) Feature Selection

- Input Instance를 위한 Feature Vector 생성

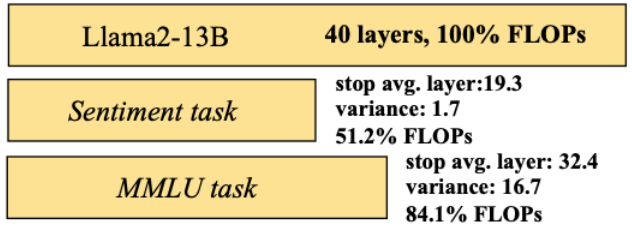
(2) Classifier

- Stopping Signal 평가

→ Signal이 충분히 강하면 early termination되어 이후 디코더 레이어 생략



(a) A workflow of AdaInfer processing three input instances, involving two for sentiment analysis and one for a knowledge-based question answering task. It shows that the early-exit moment varies across the instances.



(b) After implementing AdaInfer, LLMs can reduce computational costs through adaptive early-exit strategies.

Figure 2: An illustration of AdaInfer's processing and computational savings.

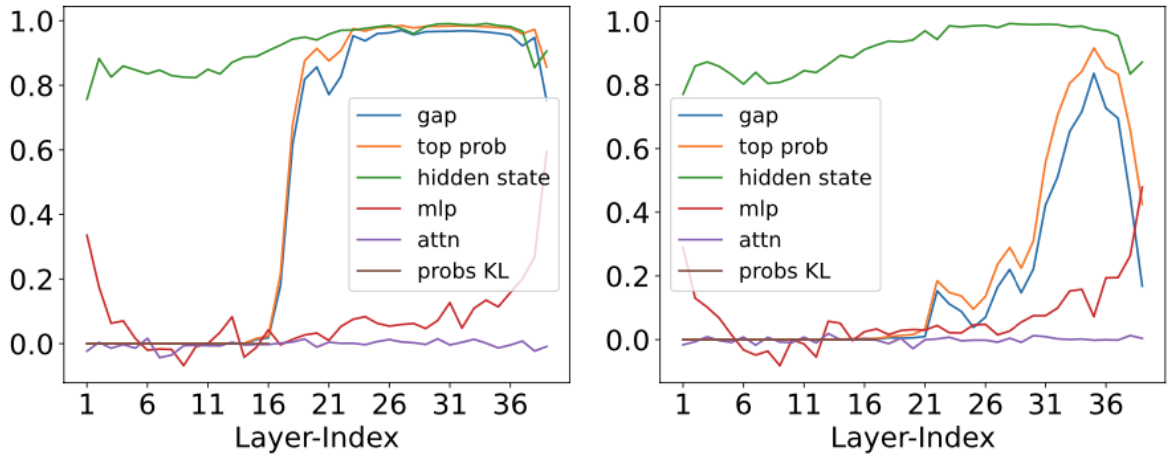
Not all Layers of LLMs are Necessary during Inference

AdaInfer: Feature Selection

: LLM Parameter 수정에 추가적인 훈련과 높은 비용이 요구됨

: Parameter 변경 없이 모델의 능력을 유지하는 효율적인 방법 필요성

: 통계적 분류기를 활용하여 종료 신호를 평가하기 위한 feature 모색



(a) Llama2 on sentiment (b) Llama2 on MMLU

Problem: The lack of feature for decision-making

- LLMs가 초기 계층에서 Coarse-grained feature를 계산하고, 깊은 층에서 자세하고 fine-grained된 feature로 발전
- Input Instance에 대해 Shallow-level의 Representation이 충분하다는 것을 입증하는 Universial Level 특징이 부족

Solution: Logits reflect mutation

: LLM 각 블록 내 레이어 전반에 대해 시각적으로 분석

- Top Prob: 출력할 가능성이 가장 높은 토큰에 대한 확률 P (==Confidence)
- Gap: P(top token) – P(second token)
- Cosine Similarity: 현재 블록과 이전 블록의 유사성 평가, attention(value), MLP, hidden states 포함

Not all Layers of LLMs are Necessary during Inference

AdaInfer: Classifier

Classifier Objective

- 이진분류로 단순화하기 위해 통계적 분류기를 사용하여 Early Terminate 결정
- Gap / Top Prob을 Feature로 사용하여 학습

Objective

- Feature Selection module을 통해 feature vector(x_d) 생성
- 각 Layer에서의 출력이 정답을 제공하는지에 따라 분류기를 훈련
- L-layers의 LLM에서 총 L개의 데이터 쌍 생성
- 두가지 유형의 분류기를 고려: SVM, CRF

$$y_c = \begin{cases} 1 & \text{if } \hat{y} = y, \\ 0 & \text{otherwise.} \end{cases}$$

Not all Layers of LLMs are Necessary during Inference

Setup

Tasks

1) Question Answering

- MMLU, CommonsenseQA, SQuAD

2) Text Classification

- SST-2, AG News

Models

1) Llama-2

- 7B, 13B, 70B

2) OPT

- 13B

Metrics

1) Accuracy

- Top-1에 대한 Accuracy 측정

2) FLOPs

$$\frac{2l'(6h+s)+V}{2l(6h+s)+V}$$

- Computational Efficiency 측정
- **l**: total / terminate layer, **h**: dim, **s**: seq_len, **v**: vocab

Train

- 분류기 학습에 평가 데이터 셋 중 하위 태스크 셋(71개)에서 6~9개 샘플링하여 AdaInfer 훈련, 사용한 분류 모델은 SVM

5. Results

Not all Layers of LLMs are Necessary during Inference

Table 2: Performance and efficiency in question answering tasks, with accuracy (%) denoted by 'Acc'. Results include few-shot learning with sample sizes of 5, 10, 15, and 20, showcasing the average values.

Setting	Model	MMLU		CommonsenseQA		SQuAD		Avg	
		Acc↑	FLOPs↓	Acc↑	FLOPs↓	Acc↑	FLOPs↓	Acc↑	FLOPs↓
Zero-shot	OPT-13B	7.95	100	8.20	100	20.00	100	12.05	100
	AdaInfer	8.67	97.55	2.80	97.55	23.00	97.55	11.49	97.55
Few-shot	OPT-13B	23.60	100	21.45	100	26.12	100	23.72	100
	AdaInfer	22.59	83.94	21.62	86.05	25.95	88.31	23.39	86.10
Zero-shot	Llama2-13B	2.54	100	1.00	100	19.20	100	7.58	100
	AdaInfer	2.48	98.14	0.70	98.37	25.90	85.34	9.69	93.95
Few-shot	Llama2-13B	53.31	100	64.92	100	52.9	100	57.04	100
	AdaInfer	52.44	93.55	62.48	89.10	48.35	80.66	54.42	87.77

Table 3: Performance and efficiency in classification and rule understanding, with accuracy (%) denoted by 'Acc'. Results include few-shot learning with sample sizes of 5, 10, 15, and 20, showcasing the average values.

Setting	Model	Sentiment		AG News		Avg		Rule Understanding	
		Acc↑	FLOPs↓	Acc ↑	FLOPs↓	Acc↑	FLOPs↓	Acc↑	FLOPs↓
Zero-shot	OPT-13B	0.00	100	0.10	100	0.05	100	3.38	100
	AdaInfer	0.00	96.87	0.10	100	0.05	98.44	3.86	92.52
Few-shot	OPT-13B	92.58	100	72.83	100	82.71	100	58.48	100
	AdaInfer	92.97	80.28	72.83	100	82.90	90.14	52.83	89.74
Zero-shot	Llama2-13B	0.00	100	0.10	100	0.05	100	2.32	100
	AdaInfer	0.00	97.43	0.10	88.37	0.05	92.90	6.14	85.76
Few-shot	Llama2-13B	95.90	100	77.53	100	86.72	100	69.36	100
	AdaInfer	92.65	59.70	76.43	87.69	84.54	73.70	61.87	80.61

Compared with Baseline Methods

: Top1 Accuracy & FLOPs 측정

ACC

: QA, Classification에서 성능이 1% 이내로 유지

1. Early Termination의 효과에 대한 가능성
2. 특정 태스크에서 AdaInfer가 기존의 성능을 능가
3. Deep Layer가 추론 시 성능을 저해할 가능성 제기

FLOPs

: 감성분석에서 41%절감, MMLU에서 2%절감

1. 태스크에따라 FLOPs비율이 59% ~ 98%로 다양
2. Input Instatnce에 따라 서로 다른 Early Termination 결정

→ "Simple" Samples에 대해 적은 자원을 할당하는 것이 계산 효율성을 향상시킬 수 있다라는 주장과 일치

5. Results

Not all Layers of LLMs are Necessary during Inference

Task	Setting	AdaInfer *w.* Rule		AdaInfer *w.* CRF	
		Acc↑	FLOPs↓	Acc↑	FLOPs↓
MMLU	Zero-shot	5.35	90.84	4.77	97.40
	Few-shot	47.09	84.10	52.72	97.15
CommonsenseQA	Zero-shot	1.10	92.78	1.40	97.28
	Few-shot	55.33	79.57	65.72	96.40
SQuAD	Zero-shot	24.60	73.17	23.10	93.03
	Few-shot	43.43	71.19	51.75	89.94
Sentiment	Zero-shot	0.00	88.25	0.00	97.27
	Few-shot	91.45	51.25	95.60	73.07
AG News	Zero-shot	0.10	77.82	0.10	94.04
	Few-shot	69.17	70.65	76.77	93.08
Rule Understanding	Zero-shot	9.90	74.80	3.43	90.29
	Few-shot	53.78	70.38	65.82	90.29

Evaluation Different Exit Strategy

: 메인 실험 SVM사용, Rule-based와 CRF를 사용한 결과 제공

: Rule-based는GAP=0.8로 설정

1. CRF와 GAP모두 3~50%까지 비용을 줄이며, 성능도 유지

Task	Setting	Llama2-7B		AdaInfer	
		Acc↑	FLOPs↓	Acc↑	FLOPs↓
MMLU	Zero-shot	4.19	100	4.63	96.13
	Few-shot	43.05	100	43.73	93.76
CommonsenseQA	Zero-shot	5.30	100	4.80	95.26
	Few-shot	53.50	100	53.00	90.46
SQuAD	Zero-shot	20.40	100	23.80	89.98
	Few-shot	48.08	100	45.82	87.06
Sentiment	Zero-shot	0.00	100	0.00	96.37
	Few-shot	95.20	100	95.30	68.05
AG News	Zero-shot	0.10	100	0.10	91.36
	Few-shot	79.65	100	79.72	94.51
Rule Understanding	Zero-shot	5.47	100	5.32	91.55
	Few-shot	66.80	100	66.92	88.41

Evaluation across Scailing Law

Llama2-7B

1. 1%미만의 Acc 손실, 4%~32% 비용 절감

Llama2-70B

1. 제로샷에서 기존 모델과 일치하거나 초과하는 성능
2. 평균 10%~50% 절감

Not all Layers of LLMs are Necessary during Inference

Table 6: Generalization performance of statistic classifier on sentiment task on Llama2-7B (32 layers), Inter-Model refers to Llama2-13B (40 layers).

Classifier	Generalization	Acc	Layers	Variance	FLOPs
SVM	Intra-Task	94.90	18.15	0.45	60.58
CRF		0.00	0.00	0.00	0.00
SVM	Inter-Task	95.50	19.20	4.40	63.80
CRF		94.90	20.20	4.55	66.87
SVM	Inter-Model	90.70	20.60	3.70	54.55
CRF		87.75	19.20	2.75	51.09

Generalization Study

: 분류기 학습 및 feature model 변경에 따른 실험
: 분류기의 일반화 성능을 평가하기 위함

1) Intra-Task

감성분석 태스크를 해당 훈련 데이터셋으로 학습된 분류기로 평가

2) Inter-Task

지식기반 태스크 데이터셋으로 훈련된 분류기로 평가

3) Inter-Model

Llama2-7B로 훈련된 분류기로 Llama2-13B에서 감성분석 평가

1. SVM 분류기의 경우 일반화 성능에 대해 만족
2. CRF의 경우 Feature Selection에 대해 적합하지 않음

Not all Layers of LLMs are Necessary during Inference

Conclusion

1. 추론시 모든 레이어가 필요하지 않다는 증거 제시
2. Input Instance에 따라 적절한 순간에 추론을 종료하여 효율성을 향상시키는 알고리즘 소개
3. 평균 14.8% 계산 자원 절감, 쉬운 태스크에서 최대 50%절감
4. 효율성뿐만 아니라 성능 향상의 케이스도 존재

Limitation

1. Single Forward path에 의존, Sequential generative Tasks로는 확장 X

⚡PLUG: Leveraging Pivot Language in Cross-Lingual Instruction Tuning

Zhihan Zhang[✉1†] , **Dong-Ho Lee**[2†], **Yuwei Fang**[3], **Wenhao Yu**[1],
Mengzhao Jia[1], **Meng Jiang**[1], **Francesco Barbieri**[3]
[1]University of Notre Dame [2]University of Southern California [3]Snap Inc.
zzhang23@nd.edu

ACL 2024

⚡PLUG: Leveraging Pivot Language in Cross-Lingual Instruction Tuning

Introduction

Instruction Tuning

- 주어진 Instruction에 대해 적절한 Response를 생성하도록 Tuning
- IT의 성공적은 Case들은 대부분 High Resource Lang에서 이루어짐
- Target Lang을 통한 IT에도 단일 언어 생성시 응답 품질 향상 폭이 크지 않음

Low Resource lang Problem

- IT에서 고자원 언어(영어)에서는 성공적이지만, 저자원 언어에서는 쉽지 않음
- 사전 학습 데이터에서 언어 간 자원의 불균형에 기인
- LLM이 익숙하지 않은 언어에서 직접 생성하도록 훈련하는 것은 어렵다.

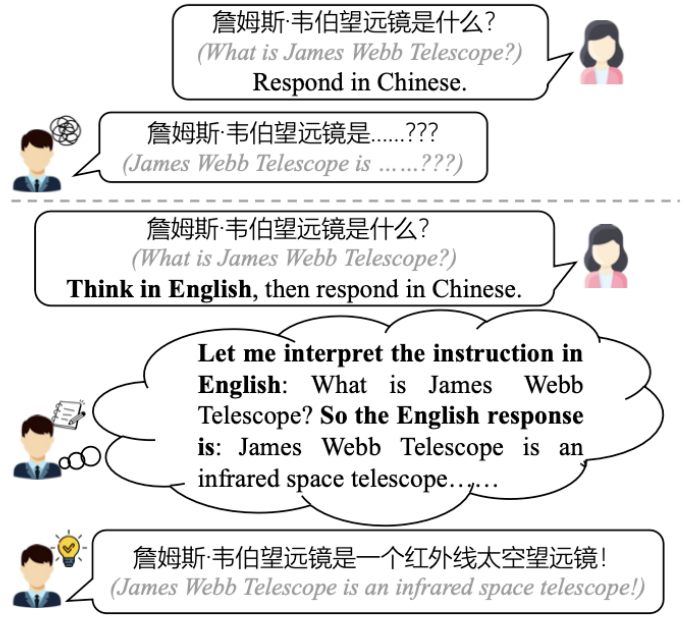


Figure 1: When humans struggle to learn a second language, they tend to comprehend the instruction and draft a response in their native language, before finally responding in the target language. With a similar philosophy, we train LLMs to utilize a high-resource language as the *pivot language* when responding to instructions in the target language.

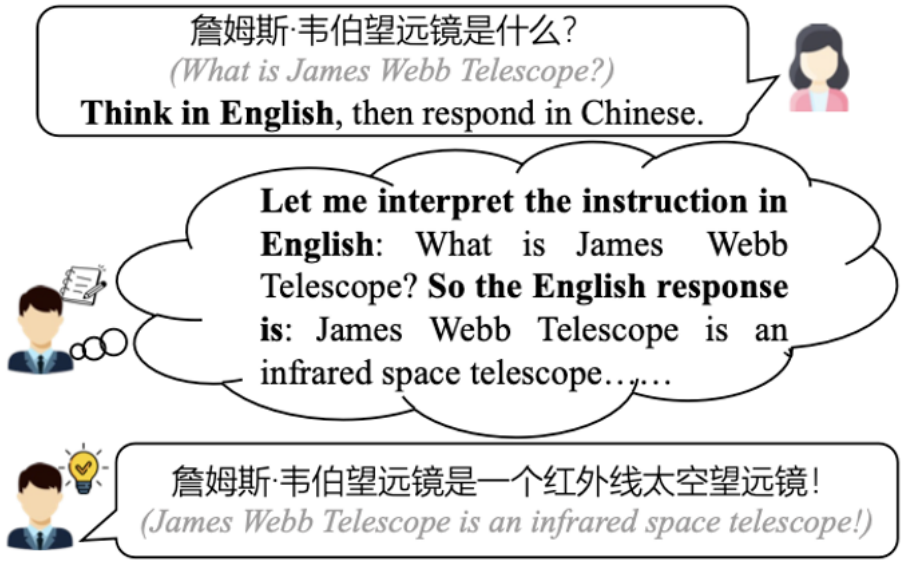
→ 고자원 언어에서의 LLM의 우수한 능력을 고려하여, 제2의 언어를 배울 때 사용하는 인지 전략을 반영한 훈련 방법을 제안

⚡PLUG: Leveraging Pivot Language in Cross-Lingual Instruction Tuning

Introduction

PLUG (pivot language guided generation)

- 간단하지만 효과적인 훈련 방법
- 제2 언어 학습 전략에서 착안
- Pivot Lang을 활용한 Instruction Response 생성 방법
- 하나의 단일 패스로 처리



Contributions

👉 Pivot language를 사용하여 Low Resource Lang의 Instruction Response 성능 향상을 패러다임 PLUG 소개

👉 Multilingual Instruction Benchmark인 X-AlpacaEval 구축

⚡PLUG: Leveraging Pivot Language in Cross-Lingual Instruction Tuning

Pivot Language Guided Generation

Traditional Instruction Tuning

: Target Lang으로 제공된 지시를 고려하여 응답하도록 훈련

: 모델의 기본 능력 제한 및 성능 저하

⚡PLUG: Leveraging Pivot Language in Cross-Lingual Instruction Tuning

Pivot Language Guided Generation

PLUG trains the model to leverage the pivot language as the intermediary in the instruction-following process.

- 학습된 코퍼스 내 많이 포함된 언어(Pivot Lang)를 이용하여 Instruction Tuning 진행
- Target Lang Instruction(xt) → Pivot Lang Instruction (xp) & Response (yp) → Target Lang Response (yt)

PLUG 장점

- Pivot lang으로 지시를 처리할 때 더 나은 이해와 실행을 보임, 이를 통해 응답 생성(yt) 난이도 완화 역할
- 중개 언어의 대응물에 의해 더 우수한 응답 품질 & 지시 수행의 상대적 용이성 증대

⚡PLUG: Leveraging Pivot Language in Cross-Lingual Instruction Tuning

(1) Benchmark & Models

1) X-AlpacaEval

: AlpacaEval을 번역한 X-AlpacaEval 구축

: Chinese, Korean, Italian, Spanish

: LLM Judge를 사용하여 평가

2) Truthfulness & Reasoning Benchmarks

: 사실성과 추론 능력 평가를 위해, GPT-4를 통한 번역 이후 zero-shot Evaluation

1. TruthfulQA
2. SVAMP

Foundation Model

- Eng-centric: LLaMA-2-13B
- Multilingual: PolyLM-(Instruct)-13B

Train Dataset

- GPT-4-Alpaca (52k): ChatGPT를 사용하여 번역

⚡PLUG: Leveraging Pivot Language in Cross-Lingual Instruction Tuning

(2) Methods to Compare

5가지 방법론에 대한 실험 진행 (x: Instruction, y: Response, p: pivot lang, t: target lang)

1. **Pivot-only training** [D(xp, yp)]

   : 영어로만 튜닝

2. **Monolingual Response training** [D(xp, yp) ∪ D(xt, yt)]

   : Target lang 데이터를 포함하여 튜닝

3. **Code Switching** [D(xp, yp) ∪ D(xt, yt) ∪ D(xp, yt) ∪ D(xt, yp)]

   : x, y에 대해 switch 한 데이터를 포함 (P <-> T)

4. **Auxlilary translation tasks** [D(xp, yp) ∪ D(xt, yt) ∪ D([Ptrans; xp], xt) ∪ D([Ptrans; yp], yt)]

   : 번역 학습

5. **PLUG** [D(xp, yp) ∪ D(xt, [xp; yp; yt])]

   : Pivot lang으로 풀이 후 target land에 대한 Response학습

⚡PLUG: Leveraging Pivot Language in Cross-Lingual Instruction Tuning

(1) Open-Ended Instruction

Training Method Comparison	Chinese			Korean			Italian			Spanish		
	Win%	Loss%	Δ%	Win%	Loss%	Δ%	Win%	Loss%	Δ%	Win%	Loss%	Δ%
English-Centric Foundation LLM: LLaMA-2-13B												
PLUG *vs.* Pivot-Only	70.9	19.1	+51.8	76.5	12.7	+63.9	67.6	17.8	+49.8	64.0	20.9	+43.1
PLUG *vs.* Mono. Response	58.0	25.2	+32.8	64.1	19.9	+44.2	50.3	25.8	+24.5	53.0	27.6	+25.5
PLUG *vs.* Mono.+Translation	53.0	28.0	+25.1	62.7	20.1	+42.6	50.1	26.6	+23.5	51.3	25.6	+25.7
PLUG *vs.* Mono.+Code-Switch	50.2	31.6	+18.6	55.2	25.6	+29.6	46.2	30.9	+15.3	48.4	29.9	+18.5
Multilingual Foundation LLM: PolyLM-13B												
PLUG *vs.* Pivot-Only	53.2	32.3	+20.9	79.9	11.1	+68.8	65.7	18.5	+47.2	57.4	24.1	+33.3
PLUG *vs.* Mono. Response	45.5	34.5	+10.9	67.3	18.4	+48.9	59.3	22.1	+37.1	44.5	30.7	+13.8
PLUG *vs.* Mono.+Translation	47.0	34.3	+12.7	67.3	20.9	+46.5	51.9	27.5	+24.5	50.2	31.2	+19.0
PLUG *vs.* Mono.+Code-Switch	47.0	37.8	+11.2	57.5	25.1	+32.4	48.8	29.4	+19.4	45.8	34.0	+11.8
Multilingual Instruction-Tuned LLM: PolyLM-Instruct-13B												
PLUG *vs.* Pivot-Only	52.8	31.9	+20.9	77.1	12.9	+64.2	62.0	20.1	+41.9	56.7	26.3	+30.4
PLUG *vs.* Mono. Response	48.5	32.1	+16.4	64.5	19.0	+45.5	54.2	22.9	+31.3	44.8	32.1	+12.7
PLUG *vs.* Mono.+Translation	46.8	33.5	+13.3	65.0	21.8	+43.3	51.1	29.0	+22.1	48.3	32.6	+15.7
PLUG *vs.* Mono.+Code-Switch	46.1	32.8	+13.3	57.8	23.9	+33.9	49.6	29.8	+19.8	45.5	32.9	+12.5

Table 1: Pair-wise comparison between PLUG and each baseline on `X-AlpacaEval`. Here, Δ indicates the win-loss differential, and thus a higher value indicates a larger gap between PLUG and the baseline.

Comparison	zh	ko	it	es
LLaMA-2-13B				
PLUG *vs.* Pivot-Only	+10.9	+7.6	+10.7	+12.0
PLUG *vs.* Mono. Response	+7.7	+1.2	+8.6	+10.1
PolyLM-13B				
PLUG *vs.* Pivot-Only	+1.2	+3.4	-8.0	+1.2
PLUG *vs.* Mono. Response	+1.6	+4.3	+5.0	+2.2
PolyLM-Instruct-13B				
PLUG *vs.* Pivot-Only	-0.2	+0.7	-0.6	+1.1
PLUG *vs.* Mono. Response	-3.0	-0.4	-3.6	0.0

Table 2: Comparisons in the pivot language (English): Generally, PLUG matches monolingual response and pivot-only training in models' instructability in the pivot language. Comparisons with other baselines exhibit similar trends and are moved to Appendix C.1 for brevity.

Table 1

- 4개의 Target Lang 전반에 걸쳐 응답 품질을 크게 향상 시킴 (Monolingual Tuning 보다 우수)
- Low Resource Lang에서 뛰어난 성능 향상을 보임 (KO, IT)

Table 2

- Pivot lang(eng)에 대한 성능을 비교

⚡PLUG: Leveraging Pivot Language in Cross-Lingual Instruction Tuning

(2) Study of Pivot Languages

Pivot \ Target	Chinese	Korean	Italian	Spanish
English	+21.6	+54.4	+35.9	+30.3
Chinese	–	+36.6	+3.1	-8.7
Korean	-42.2	–	-39.4	-42.1
Italian	-5.7	+36.5	–	+2.9
Spanish	+4.1	+41.9	+17.5	–

Table 4: PLUG *vs.* monolingual response training: The Win-Loss differential (Δ%) using different languages as the pivot, tested on PolyLM.

Pivot Language에 따른 성능 비교

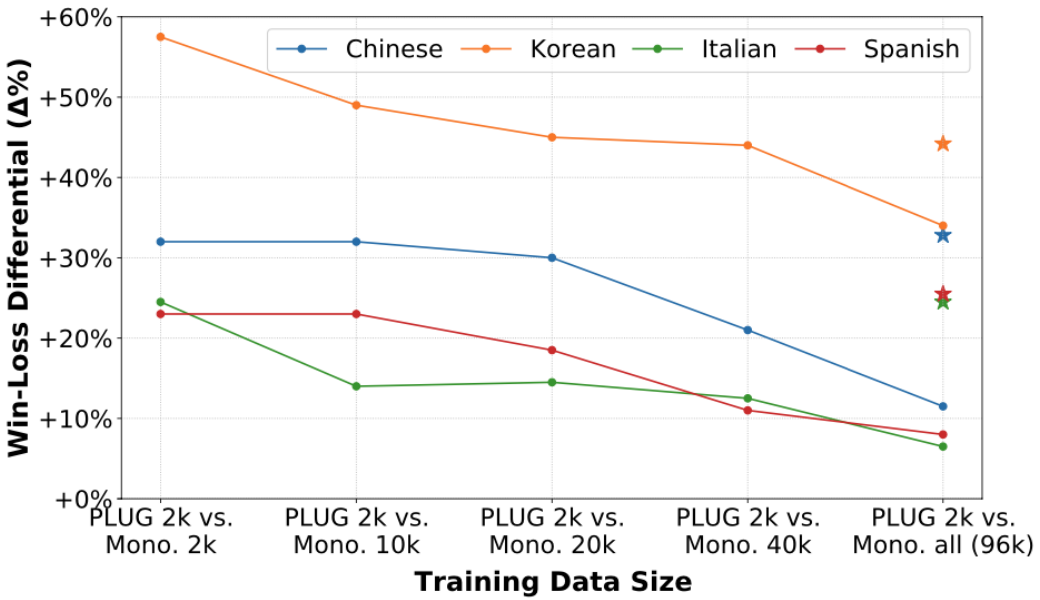
: En은 PolyLM PT Corpus 중 압도적인 양을 가지므로 효과적인 중개언어

: 다른 언어를 Pivot으로 사용하였을 때도 개선을 보임 → 특정언어에 국한 X

1. 가장 적은 데이터 언어인 한국어에 대해 기타 다른 언어를 중개언어로 사용하였을 때 +42% 향상을 보임
2. 그러나 Low resource lang을 pivot으로 쓰는 경우 망함

⚡PLUG: Leveraging Pivot Language in Cross-Lingual Instruction Tuning

(3) Data Efficiency of PLUG



Compared to Monolingual tune

: PLUG (2000) vs mono (96,000)

1. PLUG는 적은 데이터에서도 높은 효율성을 보이며, 데이터 양을 늘릴 경우 더 큰 성능 향상
2. PLUG는 데이터 크기의 증가에이득

⚡PLUG: Leveraging Pivot Language in Cross-Lingual Instruction Tuning

(4) Truthfulness & reasoning

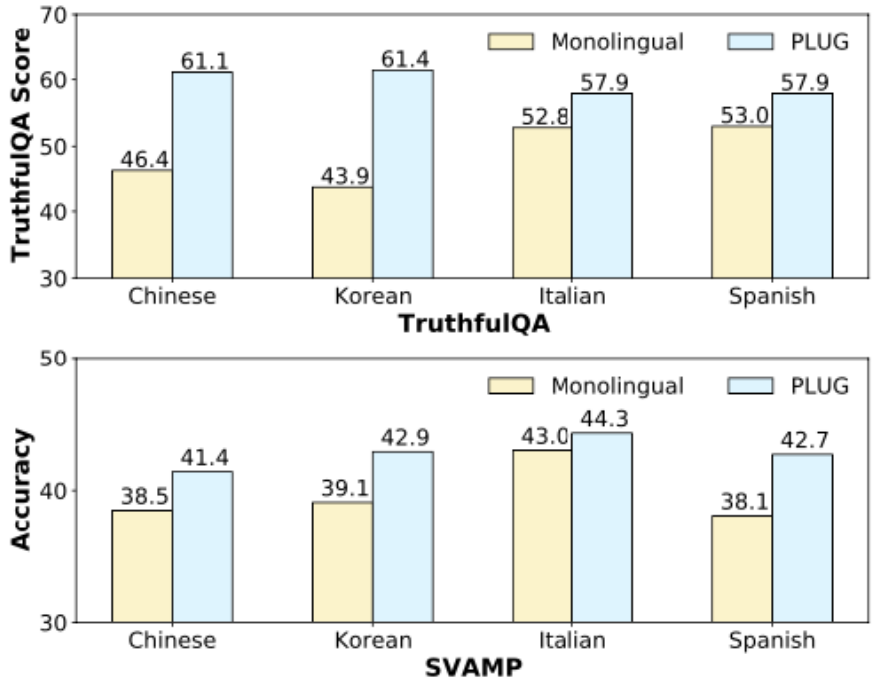


Figure 4: TruthfulQA and SVAMP experiments on LLaMA-2. TruthfulQA scores are the percentage of generations that are both truthful and informative.

Truthful & Reasoning performance

: General domain Instruction 성능뿐만 아니라 사실적 질문에 대한 진실성 및 수학 문제 추론 능력에서도 성능 향상을 보임

1. TruthfulQA에서 한국어는 39.9%, 중국어는 31.7%의 상대적 향상
2. SVAMP에서 스페인어는 12.1%의 향상

⚡PLUG: Leveraging Pivot Language in Cross-Lingual Instruction Tuning

Conclusion

1. High Resource Lang을 이용하여 쉽게 low lang의 성능을 높이는 Instruction Tuning에 용이
2. Monolingual Tuning에 비해 Response Quality 향상
3. Minor한 언어가 아니라면 Pivot lang으로 사용하여 충분한 성능 향상을 이루어 낼 수 있음

Limitation

1. Instruction 길이가 긴 경우, 매우 비효율적이며, 불필요한 토큰 생성으로 인한 overload발생
2. 데이터 번역을 위힌 경제적 비용 증가

Thank you

Q&A