Llama, OPT 모델을 활용한 Supervised Fine Tuning, Reinforcement Learning, Chain-of-Hindsight 성능 비교

이현민°[1], 나승훈[1], 임준호[2], 김태형[3], 류휘정[3], 장두성[3]
전북대학교[1], 한국전자통신연구원[2], KT[3]
{leehm, nash}@jbnu.ac.kr[1]
joonho.lim@etri.re.kr[2]
{taehyeong_2019.kim, hwijung.ryu, dschang}@kt.com[3]

Comparing the performance of Supervised Fine-tuning, Reinforcement Learning, and Chain-of-Hindsight with Llama and OPT models

Hyeon Min Lee°, Seung Hoon Na[1], Joon Ho Lim[2], Tae Hyeong Kim[3], Hwi Jung Ryu[3], Du Seong Chang[3]
Jeonbuk National University[1], ETRI[2], KT[3]

요 약

최근 몇 년 동안, Large Language Model(LLM)의 발전은 인공 지능 연구 분야에서 주요 도약을 이끌어 왔다. 이러한 모델들은 복잡한 자연어처리 작업에서 뛰어난 성능을 보이고 있다. 특히 Human Alignment를 위해 Supervised Fine Tuning, Reinforcement Learning, Chain-of-Hindsight 등을 적용한 언어모델이 관심 받고 있다. 본 논문에서는 위에 언급한 3가지 지시학습 방법인 Supervised Fine Tuning, Reinforcement Learning, Chain-of-Hindsight 를 Llama, OPT 모델에 적용하여 성능을 측정 및 비교한다.

주제어: Instruction Tuning, Supervised Fine Tuning, Reinforcement Learning, Chain-of-Hindsight

1. 서론

지난 몇년간 인공지능, 특히 자연어처리 분야는 놀라운 발전을 이루었다. Large Language Model(LLM)의 등장은 자연어처리의 성능을 기하급수적으로 향상시키며 발전을 주도하고 있다. GPT3[1]를 필두로 LLM의 중요성은 더욱 대두 되었고 Google의 PaLM[2], Meta AI의 OPT[3], Llama[4] 등의 모델들이 등장하였다. 이러한 LLM의 특징 중 하나는 광범위한 정보와 지식을 포함하고 있어, 다양한 작업에 대한 복잡한 문제 해결 능력을 보유하고 있다는 것이다. 그러나 이 모델의 효과적인 활용을 위해서는 특정 목적 혹은 작업에 맞게 세밀하게 조정할 필요가 있다. 이를 위한 방법론으론 Instruction Tuning, Supervised Fine Tuning(SFT), Reinforcement Learning(RL), Chain-of-Hindsight(CoH) 등이 대표적이다. FLAN[5]에서 처음 제안된 Instruction Tuning 은 다양한 태스크를 지시문 형태의 프롬프트를 부착하여 언어모델을 학습하면, 학습하지 않은 Unseen 태스크에 대한 성능도 증가한다는 것을 증명하였다. InstructGPT[6]는 사람의 선호도가 레이블링된 데이터와 SFT, RL을 언어모델에 적용하여, 지시문장의 응답에 대한 사람의 선호도를 증가 시켰다. Chain-of-Hindsight[7]는 사람의 피드백이 있는 데이터셋을 사용하여 언어모델을 미세조정하여 요약과 대화의 성능을 증가 시켰다. 본 논문에서는 위에 언급한 방법론 SFT, RL, CoH를 Llama(7B, 13B), OPT 6.7B 모델에 적용하여 성능을 측정 및 비교하는 것을 목표로한다.
본 논문의 기여점은 다음과 같다.

- 오픈소스 대규모 언어모델인 Llama와 OPT 모델을 학습하여 성능비교
- Supervised Fine Tuning과 Chain-of-Hindsight 성능비교 및 결과확인
- Supervised Fine Tuning과 Reinforcement Learning 성능비교 및 결과확인

2. 관련 연구

2.1 Large Language Models

Large Language Model(LLM)의 개념은 Open-AI의 GPT3에서 처음 등장하였다. GPT3는 방대한 양의 파라미터와 학습 데이터를 기반으로 언어모델에 추가적인 학습 없이 사용자가 원하는 문제를 해결하는 Prompting, In-context few shot learning등을 통하여 다양한 문제를 범용적으로 잘 푸는 모습을 보여줬다. 하지만, 성능은 여전히 사람에 미치지 못하였고 개선할 여지를 남겼다. 이후 Google의 PaLM[2]과 Meta AI의 OPT[3], Llama[4] 등이 등장하며 LLM의 성능 경쟁이 시작되었다.

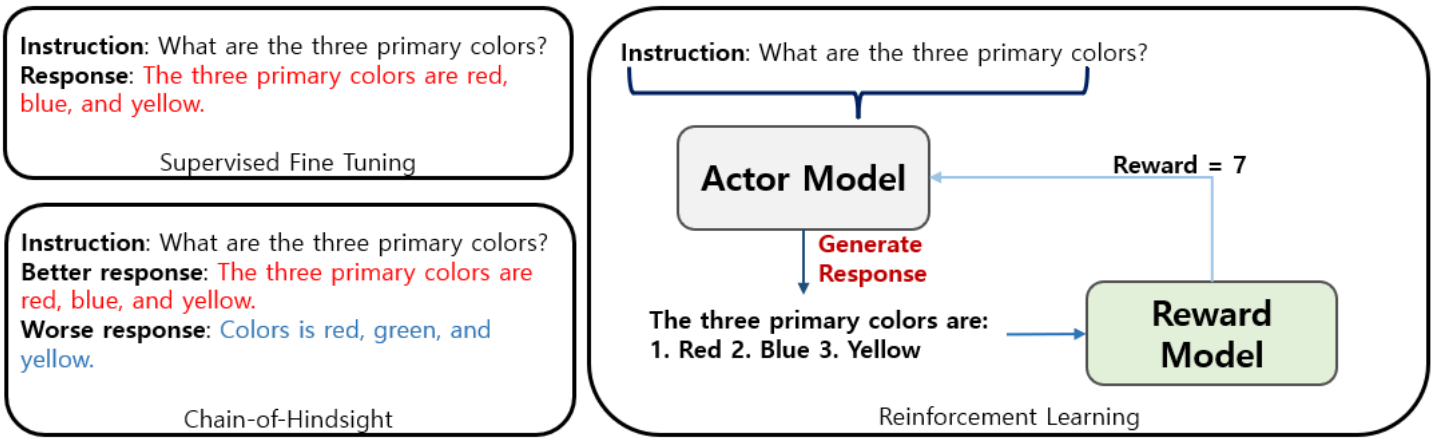

그림 1. Supervised Fine Tuning, Chain-of-Hindsight, Reinforcement Learning 학습 예시

2.2 지시학습

Open-AI의 GPT3는 In-context few shot learning을 통해 파리미터를 수정하지 않고 언어모델의 성능을 증가시키는 방법론을 제시하였지만, 여전히 사람의 성능에는 미치지 못하는 모습을 보여주었다. 부족한 성능을 끌어 올리기 위해선 사전학습 언어모델의 파라미터 업데이트가 필요하지만, 특정 태스크에 대해 미세조정하는 방법은 학습시간면에서 큰 부담이 된다. 이러한 문제를 해결하기 위해 Open-AI는 Supervised Fine Tuning(SFT)과 Reward Modeling(RM), Proximal Policy Optimization(PPO)[8] 알고리즘 적용하여 특정 태스크가 아닌 다양한 태스크에 대한 사람의 선호도를 증가시키는 논문 InstructGPT[6]를 공개하였다.

해당 논문은 크게 3가지 과정을 거쳐 학습하게 된다. 첫번째 단계에서는 지시문 형식의 프롬프트와 인간 레이블러가 작성한 정답 레이블을 Fine Tuning 한다. 두번째 단계에서는 하나의 프롬프트와 그에 따른 결과물 4 ~ 9개, 결과에 대한 선호도 순위로 구성된 데이터를 활용하여 RM을 학습한다. 세번째 단계에서는 1단계에서 학습한 SFT 모델이 생성한 결과를 바탕으로 RM 모델로 보상값을 측정하고 보상을 최대화 하는 방식으로 학습하게 된다. 위와 같은 방법을 통해 언어모델은 사람의 직접적인 지시를 받아들일 수 있고 출력 결과 또한 사람의 선호도가 증가하게 된다. Chain-of-Hindsight는 인간의 선호도가 반영된 데이터를 사용하여 미세조정을 한다. 프롬프트를 통해 좋은 예제와 나쁜 예제를 학습하게 된다. 좋은 예제는 “A helpful answer”, 나쁜 예제는 “unhelpful answer”와 같은 프롬프트를 가지고 학습하게 된다. 언어모델은 좋은 예제와 나쁜 예제를 모두 학습하게 됨으로써 인간의 선호도를 언어모델의 파라미터에 내재할 수 있게된다. 이를 통해 언어모델은 선호도가 부여된 예제들을 학습하게 되고 추론시에 “A helpful answer, better response”와 같은 프롬프트를 통해 더 좋은 예제를 생성할 수 있게 된다.

3. 학습 방법

3.1 데이터셋

RL과 CoH를 적용하기 위해선 하나의 프롬프트에 사람의 선호도가 부여된 데이터셋이 필요하다. 하지만, 사람의 선호도 데이터셋은 구축하는데 시간과 비용이 많이 들고 비 전문가가 작성한 선호도 데이터 셋은 전문성이 떨어진다는 문제점이 존재한다. 본 논문에서는 이를 해결하기 위해 Open-AI에서 공개한 Instruction Tuning with GPT-4[9] 데이터셋을 사용한다. 해당 데이터셋은 Alpaca[1] 데이터셋의 52,000개 프롬프트를 사용하여 GPT-4[10], GPT-3.5, OPT-IML 모델로 서로 다른 응답을 생성하고, 생성한 응답을 Open-AI의 RM을 사용해 점수를 부여한다. 부여된 점수를 사용하여 높은 점수의 응답을 선호되는 응답으로, 낮은 점수의 응답을 선호되지 않는 응답으로 전처리 할 수 있게된다. 본 논문은 학습 방법에 따라 데이터셋 학습량에 차이가 있다. SFT만 학습한 모델은 프롬프트에 대해 GPT4가 응답한 52,000개의 데이터셋을 사용하여 학습하였다. CoH를 학습한 모델은 프롬프트에 대해 가장 점수가 높은 응답을 선호하는 응답으로, 두번째로 점수가 높은 응답을 선호되지 않는 응답으로 전처리하여 52,000개의 데이터셋을 사용해 학습하였다. RL을 학습한 모델은 52,000 개의 데이터셋을 학습 단계별로 나누어 학습하였는데, 1단계에 10,000개의 데이터셋, 2단계에 10,000개의 데이터셋을 전처리하여 총 20,255개의 데이터셋으로 학습하였다. 3단계에 30,000개의 데이터셋 중 프롬프트만 사용하여 학습하였다. 나머지 2,000개의 데이터는 평가를 위해 사용하였다.

[1]https://crfm.stanford.edu/2023/03/13/alpaca.html

3.2 Supervised Fine Tuning

SFT는 입력된 프롬프트에 대한 출력값만 학습한다.
SFT를 수식으로 나타내면 다음과 같고, 값이 최대가 되도록 학습한다.

$$L = \sum_{(x,y)} logP(y|x^1, ..., x^m) \quad (1)$$

이때 y는 정답 레이블, x는 입력 프롬프트이다.

3.3 Reward Modeling

RM은 기존의 디코더 기반 언어모델에서 마지막 LM head를 제거하고, 스칼라 값을 출력할 수 있는 순방향 신경망으로 대체하여 학습한다. 하나의 프롬프트에 대해 선호되는 답변, 선호되지 않는 답변이 있는 데이터셋을 활용하여 선호되는 답변은 높은 점수가 부여되고, 선호되지 않는 답변은 낮은 점수가 부여되도록 학습된다. 본 논문에서는 강화학습 적용시에만 RM을 학습한다. RM의 손실값의 수식은 다음과 같다.

$$loss(\theta) = -E_{(x,y_w,y_l)\sim D}[\log(\sigma(r_\theta(x, y_w) - r_\theta(x, y_l)))] \quad (2)$$

이때 D는 데이터셋, x는 프롬프트, y_w, y_l은 각각 선호되는 답변과 선호되지 않는 답변, θ는 파라미터, σ는 시그모이드 함수이다.

3.4 Reinforcement Learning

RL은 SFT 모델에 PPO 알고리즘을 적용하여 학습한다. 학습할 SFT 모델을 사용하여 입력 프롬프트에 대한 응답을 생성한다. 프롬프트와 응답은 RM을 통해 점수화 되고 에피소드를 종료하게 된다. 보상 모델의 과적합을 완화하기 위해 각 토큰에 SFT 모델에서 토큰당 KL 페널티를 추가한다. PPO 알고리즘의 학습 함수는 다음과 같다.

$$objective(\phi) = E_{(x,y)} \sim D_{\pi_\phi^{RL}}[r_\theta(x, y) - \beta \log(\pi_\phi^{RL}(y|x)/\pi^{SFT}(y|x))]$$

이때 π_ϕ^{RL}은 학습되는 모델이고, π^{SFT}는 1단계 학습한 SFT 모델, β 값은 KL 보상계수이다.

3.5 Chain-of-Hindsight

CoH는 사전학습된 모델에 프롬프트를 사용하여 선호되는 응답과 선호되지 않는 응답을 학습한다. “A helpful answer, better response”와 같은 긍정적인 프롬프트 이후에 선호되는 응답을 주고, “unhelpful answer, worse response”와 같은 부정적인 프롬프트 이후에 선호되지 않는 응답을 주고 학습하게 된다. 이를 통해 언어모델은 선호되는 응답과 선호되지 않는 응답을 한번에 학습하게 되고, 추론시에 선호되는 응답을 잘 생성할 수 있게된다. CoH 의 손실함수는 다음과 같다.

$$\log p(x) = \log \prod_{i=1}^{n} p(x_i|x_1, x_2, ..., x_{i-1}) \quad (3)$$

4. 실험 세팅

학습에는 NVIDIA A100-PCIE-40GB 8장을 사용하였다. SFT만 적용한 모델과 CoH를 적용한 모델은 Llama 7B, 13B 모델의 미세조정 하였다. 강화학습을 적용한 모델은 Actor Model: OPT-6.7B, Reward Model: OPT-350m 을 사용 하였으며, 모두 미세조정 하였다. Optimizer는 AdamW를 사용하였으며, deepspeed stage 3를 적용하여 학습하였다

이 외의 SFT, CoH 하이퍼 파라미터 구성은 다음 표 1와 같다.

하이퍼 파라미터	값
Epochs	5
Learning Rate	1e-5
Total Batch Size	128
Cosine Warmup Ratio	0.03
Gradient Clipping	1

표 1. SFT, CoH 하이퍼 파라미터

강화학습을 적용한 모델의 하이퍼 파라미터는 다음 표 2와 같다.

하이퍼 파라미터	1단계(SFT)	2단계(RM)	3단계(RL)
Epochs	5	1	1
Learning Rate	2e-5	5e-5	9e-6, 1e-5
Total Batch Size	32	16	8
Gradient Clipping	1	1	1
KL β	-	-	0.02
Clip Reward	-	-	5
Clip Range	-	-	0.2
γ	-	-	1
λ	-	-	0.95

표 2. 강화학습 하이퍼 파라미터

Model		SST2			CSQA			MNLI			CoPA		
		k=0	k=1	k=3	k=0	k=1	k=3	k=0	k=1	k=3	k=0	k=1	k=3
Llama-7B	Pretrained	0.11	83.27	77.22	17.28	44.17	50.91	34.43	46.71	60.3	76.2	**93.93**	**95.93**
	SFT	27.29	83.22	91.51	**50.78**	**48.08**	**54.93**	34.8	41.61	55.66	77.4	93.47	93.47
	CoH	**74.43**	**92.74**	**93.12**	48.08	48.02	49.14	**54.15**	**58.72**	**62.45**	**89.0**	91.8	91.53
Llama-13B	Pretrained	51.38	71.87	85.55	17.28	44.17	50.91	33.77	48.33	57.67	89.2	91.73	91.8
	SFT	37.84	85.74	92.05	**60.44**	**61.29**	64.16	41.95	43.63	**66.91**	**91.0**	93.6	93.07
	CoH	**68.35**	**88.15**	**92.7**	56.67	59.57	**65.22**	**44.63**	**50.76**	63.3	90.6	**96.3**	**96.87**

표 3. SFT, CoH 정확도 성능 비교, 단위 %

Prompt	1단계 SFT	3단계 RL
You have a list of words, and you need to come up with a new word or phrase by combining them. Input: Wind, Rise Response:	Winds of change	Here is a possible new word or phrase that combines wind and rise: "Winding Rise"
Given a list of items classify each item as either vegetables or fruits. Input: Carrot, Apple, Broccoli, Banana Response:	- Carrot: Vegetable - Apple: Fruite - Broccoli: Vegetable - Banan	Here is a classification of the items you provided: 1. Carrot: Vegetable 2. Apple: Fruit 3. Broccoli: Vegetable 4. Banana: Fruit

표 4. Supervised Fine Tuning, Reinforcement Learning 결과 예제

5. 결과

5.1 SFT, CoH

평가를 위해 Llama 모델과 SFT만 적용한 모델, CoH를 적용한 모델을 4개의 태스크에 대한 성능 평가를 진행하였다. SST2[11], CommenSense QA[12], MNLI[13], CoPA[14] 데이터셋을 사용하여 성능 평가를 하였다. 모든 태스크는 프롬프트 부여하고 제로샷, 원샷, 퓨샷을 통해 언어모델이 생성된 결과를 평가하였다. 원샷과 퓨샷은 각 태스크의 학습 데이터에서 랜덤하게 예제를 뽑고, 시드값을 변경하여 3번 평가한 값의 평균을 기록하였다. SST2 는 입력된 문장이 긍정인지 부정인지 평가하는 이진 분류 태스크이다. CommenSense QA는 질문과 보기가 주어지고 정답을 선택하는 객관식 문제이다. MNLI는 전제에 대한 가설이 중립, 수반, 모순 중에서 하나를 선택하는 태스크이다. CoPA는 두 문장 중 전제에 대한 원인과 결과가 되는 문장을 선택하는 태스크이다. 각 태스크에 대한 프롬프트는 다음과 같다.

SST2 - Is this movie review positive? [sentence] Response:

CSQA - Select the correct answer to the question. [question, answer] Response:

MNLI: - [premise] Based on the paragraph above can we conclude that [hypothesis]? Response:

CoPA: - [premise] What is the cause(or effect)? Response:

CoH는 동일한 프롬프트에서 "Response"만 "better response"로 변경하여 생성된 결과를 평가하였다. 결과는 다음 표3와 같다. 표의 K값은 In-context few shot example 개수이다.

5.2 1단계(SFT), 3단계(RL)

OPT 모델에 강화학습을 적용한 후, 1단계와 3단계의 생성 결과를 Human Evaluation으로 비교했다. 어떤 모델에서 나온 결과인지 알 수 없는 상태에서 한 사람에게 10개 예제를 평가하게 했다. 그 중 1단계 모델이 더 좋다고 평가받은 예제는 1개, 3단계 모델이 더 좋다고 평가받은 예제는 3개였다. 나머지 6개의 예제는 비슷한 수준의 결과를 보였다. 1단계와 3단계의 생성 결과 비교는 표4와 같다.

5.3 결과 분석

사전학습된 Llama 모델과 SFT 적용한 모델, CoH를 적용한 모델을 4개의 태스크에 대한 성능 평가 결과, 사전학습된 언어모델보다 SFT, CoH를 적용한 모델들의 성능이 더 높은것을 확인해볼 수 있다. SFT와 CoH의 태스크별 제로샷 성능을 비교해보면, CSQA와 Llama 13B CoPA 에서의 성능을 제외하고 CoH의 방법론의 성능이 더 우수함을 확인할 수 있었다. 특히나 SST2에서의 성능 변화가 눈에 띄는데 이는 선호되지 않는 데이터셋 학습을 통해 문장의 감정을 이해하는 능력이 향상되었음을 시사한다.

OPT 모델에 SFT 적용한 결과와 RL을 적용한 결과를 비교해보면, 사용자의 지시사항을 더 잘 이행하는 것을 확인할 수 있었다. 첫번째 예제의 경우 입력된 SFT 모델은 입력된 지시문을 지키지 못한 반면, RL 모델은 지시사항을 이행하며 답변하는 것을 확인할 수 있다. 두번째 예제의 경우 SFT 모델은 결과를 끝맺음 하지 못하고 생성이 중단된 반면에, RL 모델은 끝까지 생성해낸것을 확인할 수 있다. 또한 지시사항에 단답형으로 대답하는 SFT 모델과 다르게 RL 모델은 유저 친화적인 답변을 생성하는 것을 확인할 수 있었다.

6. 결론

본 논문에서는 SFT, CoH, RL을 각각 Llama, OPT 모델에 적용하여 4개의 태스크와 Human Evaluate을 진행하였다. 실험결과 SFT 모델보다 CoH를 적용한 모델이 CSQA와 Llama 13B CoPA를 제외한 모든 경우에서 제로샷 성능이 더 우수함을 확인할 수 있었다. 또한, SFT만 적용한 모델보다 RL을 적용한 모델이 사용자의 지시사항을 더 잘 이행하고, 유저 친화적인 답변을 생성할 수 있음을 보였다.

참고문헌

[1] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, Vol. 33, pp. 1877–1901, 2020.

[2] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann *et al.*, "Palm: Scaling language modeling with pathways," *arXiv preprint arXiv:2204.02311*, 2022.

[3] S. Zhang, S. Roller, N. Goyal, M. Artetxe, M. Chen, S. Chen, C. Dewan, M. Diab, X. Li, X. V. Lin *et al.*, "Opt: Open pre-trained transformer language models," *arXiv preprint arXiv:2205.01068*, 2022.

[4] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[5] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, "Finetuned language models are zero-shot learners," *arXiv preprint arXiv:2109.01652*, 2021.

[6] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *Advances in Neural Information Processing Systems*, Vol. 35, pp. 27 730–27 744, 2022.

[7] H. Liu, C. Sferrazza, and P. Abbeel, "Chain of hindsight aligns language models with feedback," *arXiv preprint arXiv:2302.02676*, Vol. 3, 2023.

[8] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[9] B. Peng, C. Li, P. He, M. Galley, and J. Gao, "Instruction tuning with gpt-4," *arXiv preprint arXiv:2304.03277*, 2023.

[10] R. OpenAI, "Gpt-4 technical report," *arXiv*, pp. 2303–08 774, 2023.

[11] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, Oct. 2013. [Online]. Available: https://www.aclweb.org/anthology/D13-1170

[12] A. Talmor, J. Herzig, N. Lourie, and J. Berant, "CommonsenseQA: A question answering challenge targeting commonsense knowledge," *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158, Jun. 2019. [Online]. Available: https://aclanthology.org/N19-1421

[13] A. Williams, N. Nangia, and S. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, 2018. [Online]. Available: http://aclweb.org/anthology/N18-1101

[14] P. Kavumba, N. Inoue, B. Heinzerling, K. Singh, P. Reisert, and K. Inui, "When choosing plausible alternatives, clever hans can be clever," *Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing*, pp. 33–42, Nov. 2019. [Online]. Available: https://aclanthology.org/D19-6004