하계 세미나

Knowledge Unlearning

정다현

Machine Unlearning of Pre-trained Large Language Models

Jin Yao[1] **Eli Chien**[2] **Minxin Du**[3*] **Xinyao Niu**[4] **Tianhao Wang**[1]
Zezhou Cheng[1] **Xiang Yue**[5†]

[1]University of Virginia [2]Georgia Institute of Technology
[3]The Hong Kong Polytechnic University [4]University of Melbourne
[5]Carnegie Mellon University

rry4fg@virginia.edu ichien6@gatech.edu xyue2@andrew.cmu.edu

ACL 2024

Machine Unlearning

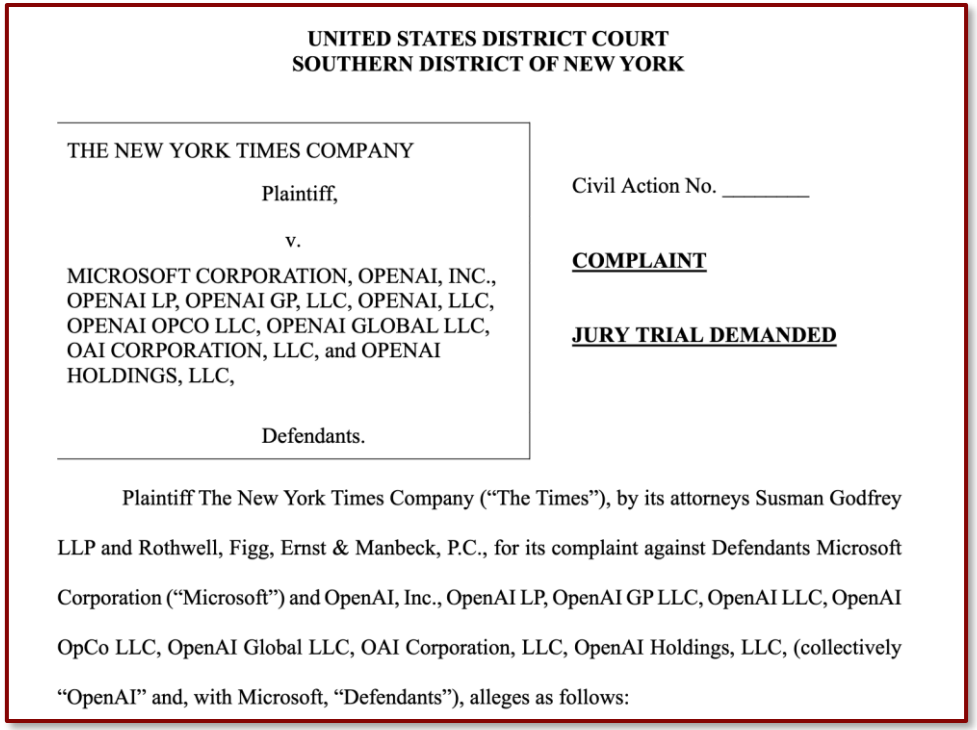
LLM에서 “잊혀질 권리”를 실현시키기 위한 기술

Sensitive, Private, or Copyrighted 데이터를 포함한 대량의 데이터로 학습된 LLM

→ 잠재적으로 윤리적인 이슈를 불러일으킬 수 있음

Introduction

- New York Times에서 OpenAI를 상대로 소송을 진행함
- LLM 훈련 시 Times의 수백만 개의 기사를 사용한 혐의
- 이 소송은 LLM 개발에서 **저작권 침해**라는 중요한 문제를 강조함



UNITED STATES DISTRICT COURT
SOUTHERN DISTRICT OF NEW YORK

THE NEW YORK TIMES COMPANY

Plaintiff,

v.

MICROSOFT CORPORATION, OPENAI, INC., OPENAI LP, OPENAI GP, LLC, OPENAI, LLC, OPENAI OPCO LLC, OPENAI GLOBAL LLC, OAI CORPORATION, LLC, and OPENAI HOLDINGS, LLC,

Defendants.

Civil Action No. _____

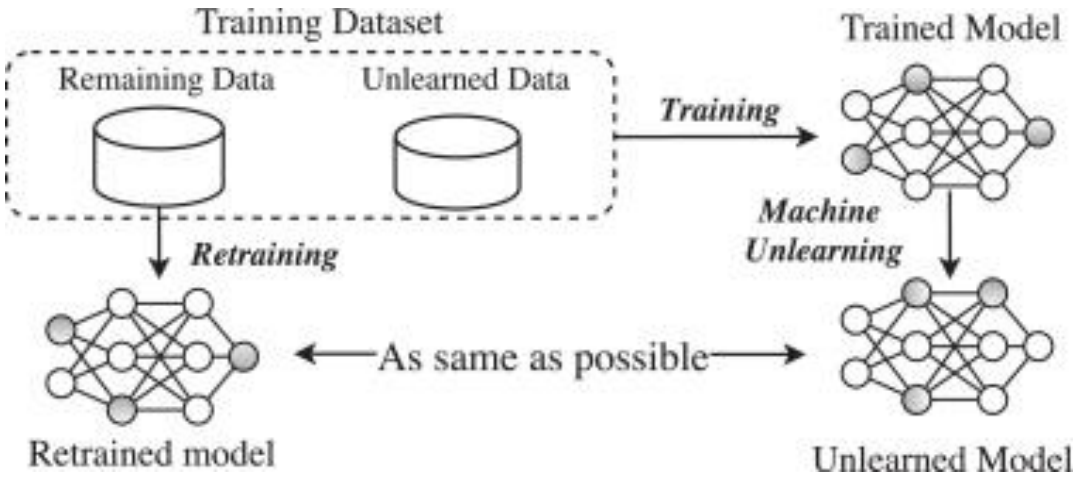
COMPLAINT

JURY TRIAL DEMANDED

Plaintiff The New York Times Company ("The Times"), by its attorneys Susman Godfrey LLP and Rothwell, Figg, Ernst & Manbeck, P.C., for its complaint against Defendants Microsoft Corporation ("Microsoft") and OpenAI, Inc., OpenAI LP, OpenAI GP LLC, OpenAI LLC, OpenAI OpCo LLC, OpenAI Global LLC, OAI Corporation, LLC, OpenAI Holdings, LLC, (collectively "OpenAI" and, with Microsoft, "Defendants"), alleges as follows:

Introduction

- 이러한 법적이고 윤리적인 문제를 해결하기 위해, 모델에서 특정 데이터를 제거하여 데이터가 전혀 포함되지 않은 것처럼 작동하도록 하는 작업이 주목을 받음



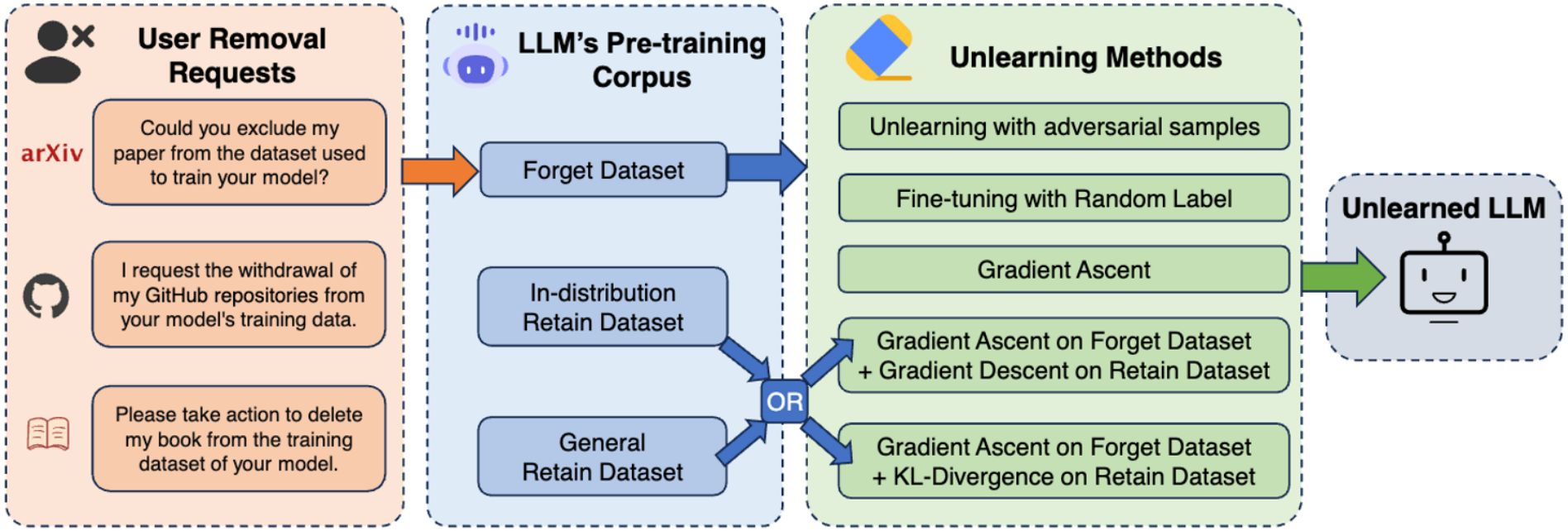
- Fine-tuning 데이터셋에는 일반적으로 10^6 이하의 토큰이 포함되지만, Pre-training은 10^{12} 정도임
- 결과적으로 LLM 재학습은 비현실적 → **Machine Unlearning**

본 논문의 Contribution

- LLM의 unlearning을 위해 **7가지의 방법에 대한 종합적인 프레임워크**를 제시
- 이전 연구들은 짧은 길이의 소규모 데이터셋으로 실험하지만, 본 논문에서는 arXiv, books, GitHub code의 **3가지 도메인에서 수천개의 청크**에 대해 unlearning을 수행
- LLM 재학습의 비실용성을 우회하기 위해 approximate **재학습 평가 베이스라인**을 제안함
- 일반적으로 비공개된 실제 사전 훈련 데이터를 공개함

Introduction

본 논문의 프레임워크



Unlearning Methods

- **Exact Unlearning**: unlearning 타겟 데이터를 제외하고 재학습하는 방법
- 이는 컴퓨팅 자원이 비효율적으로 많이 필요하므로
- 본 논문에서는 fine-tuning을 사용한 **Approximate Unlearning 방법**을 제안함

Unlearning Methods

Unlearning의 목표

(1) 모델에서 제거하고 싶은 데이터를 제거함

(2) 유지하고자 하는 데이터를 잃어버리지 않도록 하여 일반적인 모델의 성능을 유지함

- 이를 위해 approximate unlearning 프레임워크를 제안함
- U: 제거하고자 하는 데이터셋
- R: 유지하고자 하는 데이터셋

$$\sum_{w\in\mathcal{U}}\sum_{t=1}^{T}\mathbb{E}_{q_t\sim Q_{w_t}}\log P_M(q_t|w_1,w_2,...,w_{t-1}) + \sum_{z\in\mathcal{R}}\sum_{t=1}^{T}\log P_M(z_t|z_1,z_2,...,z_{t-1}), \quad (1)$$

Unlearning Methods

1. Gradient Ascent (or Negative Gradient)

- (1)에서 두 번째 항을 무시하고, $Q_{wt} = \delta_{wt}$로 설정하고, 기울기에 -1을 곱함
- **Wt가 나올 확률을 억제하는 방법**
- 너무 많은 에포크를 진행하면 다른 정보들까지 같이 잃어버리기 때문에 적은 에포크만 수행함

$$-\sum_{w\in\mathcal{U}}\sum_{t=1}^{T}\mathbb{E}_{q_t\sim Q_{w_t}}\log P_M(q_t|w_1,w_2,...,w_{t-1}) + \sum_{z\in\mathcal{R}}\sum_{t=1}^{T}\log P_M(z_t|z_1,z_2,...,z_{t-1}), \quad (1)$$

Unlearning Methods

2. Fine-tuning with Random Labels

- (1)의 두번째 항을 무시하고, 가능한 모든 토큰이 무작위로 나올 수 있도록 설정
- U를 알지 못하는 모델이 무작위 추측으로 행동해야 함
- 언뜻 보면 합리적으로 보일 수도 있지만, 균일한 분포는 보편적으로 적합하지 않음
- 실제로 이는 유용성과 성능 모두에서 현저한 감소로 이어짐

$$\sum_{w \in \mathcal{U}} \sum_{t=1}^{T} \mathbb{E}_{q_t \sim Q_{w_t}} \log P_M(q_t | w_1, w_2, ..., w_{t-1}) + \sum_{z \in \mathcal{R}} \sum_{t=1}^{T} \log P_M(z_t | z_1, z_2, ..., z_{t-1}), \quad (1)$$

3. Unlearning with Adversarial Samples

- Wt가 아닌 가장 가능성이 높은 토큰 a를 선택

$$a_t = \arg\max_{a \neq w_t} P_M(a|w_1, w_2, \ldots, w_{t-1}). \quad (2)$$

4. Gradient Ascent + Descent or KL Divergence on Retained Set

- (1)의 두번째 항을 사용하여 유지할 데이터셋을 fine-tuning하는 방법은 Gradient ascent 기술과 통합되어 두 항을 모두 최적화하는 하이브리드 접근 방식을 형성
- **Unlearning 효과와 기존 데이터셋 유지를 동시에 실현**

- 두 번째 항을 최적화하기 위해 gradient descent 및 KL-divergence constraint method 채택
- **General data**: 일반적인 사전 훈련 데이터
- **In-distribution data**: 잊어버릴 데이터셋과 일치하는 도메인의 데이터

$$-\sum_{w\in\mathcal{U}}\sum_{t=1}^{T}\mathbb{E}_{q_t\sim Q_{w_t}}\log P_M(q_t|w_1,w_2,...,w_{t-1})$$
$$+\sum_{z\in\mathcal{R}}\sum_{t=1}^{T}\log P_M(z_t|z_1,z_2,...,z_{t-1}), \qquad (1)$$

Experiments

Settings

- PLM에서 **저작권이 있는 데이터를 제거**하는 작업을 수행

- Forget Set: 지워져야 하는 데이터
- Retain Set: 유지되어야 하는 데이터
- General Downstream Tasks: 일반적인 LLM의 성능이 유지되었음을 나타내는 지표

- Yi-6B 모델을 사용 (대부분의 LLM은 사전 훈련 데이터를 오픈 소스로 제공하지 않아 Forget 데이터셋 수집이 불가능)

Experiments

Settings

- Approximate Retrain Baseline
- Exact unlearning은 최적의 표준이지만, LLM을 재훈련하는 작업은 비현실적임
- 따라서 이와 유사한 베이스라인을 선정하기 위해서 다음을 가정함:

  바닐라 모델을 unseen 데이터에 대해 평가하면 approximate unlearning된 모델과 일관된 성능을 나타낼 것
- Forget set과 동일한 도메인에서 approximate set을 수집함
- **Approximate set에 대한 바닐라 모델의 성능으로** Forget set에 대해 unlearning된 모델의 성능을 추정할 수 있음

Experiments

Settings

- Membership Inference Attack (MIA)
- LLM의 복잡성으로 인해 모델에서 특정 시퀀스가 완전히 삭제되었음을 직접적으로 검증하는 것은 불가능함
- 비멤버 (unseen) 예제가 멤버 (seen) 예제보다 특히 낮은 확률을 갖는 이상치 단어를 포함하는 경향이 있다는 전제
- 예측 확률이 가장 낮은 토큰의 비율이 높으면 비멤버 예제
- 비멤버와 멤버의 분류 작업은 AUC (Area Under Curve)를 사용하여 정량적으로 평가
- Forget set으로 모델을 평가할 경우 MIA가 높을수록 목표 시퀀스가 훈련 데이터셋 내에서 여전히 식별 가능하다는 것을 의미하는 반면, 무작위 추측 결과를 나타내는 점수인 0.5에 가까울수록 unlearning이 효율적이었음을 보여줌

Experiments

- arXiv paper

Models	Forget Set			Retain Set		Downstream Task Accuracy ↑				
	ACC↓	PPL↑	MIA↓	ACC↑	PPL↓	MMLU	ARC	HumanEval	GSM8K	Avg.
Vanilla Model	69.02	3.65	50.77	52.68	9.24	63.37	68.49	16.46	33.59	45.48
Approximate Retrain	68.98	3.69	-	-	-	-	-	-	-	-
Gradient Ascent	68.79	3.70	50.28	52.66	9.26	63.45	68.77	15.85	34.04	45.53
Fine-tuning with Random Labels	68.92	3.69	50.55	52.67	9.25	63.37	68.38	14.02	32.22	44.50
Unlearning with Adversarial Samples	68.87	3.69	50.52	52.68	9.25	63.32	68.74	15.24	33.13	45.11
Gradient Ascent + Descent on retain set										
- Descent on in-distribution data	68.87	3.69	50.18	52.66	9.26	63.32	68.52	15.24	33.74	45.21
- Descent on general data	68.81	3.69	50.33	52.93	9.04	63.40	67.87	15.24	33.13	44.91
Gradient Ascent + KL divergence										
- KL on in-distribution data	68.82	3.69	50.29	52.65	9.27	63.40	68.57	15.24	33.89	45.28
- KL on general data	68.79	3.70	50.25	52.65	9.27	63.27	68.38	15.85	33.81	45.33

- GitHub code repository files

Models	Forget Set			Retain Set		Downstream Task Accuracy ↑				
	ACC↓	PPL↑	MIA↓	ACC↑	PPL↓	MMLU	ARC	HumanEval	GSM8K	Avg.
Vanilla Model	80.65	2.40	81.93	52.68	9.24	63.37	68.49	16.46	33.59	45.48
Approximate Retrain	72.91	3.42	-	-	-	-	-	-	-	-
Gradient Ascent	78.19	3.53	74.28	52.60	9.31	63.45	68.40	14.63	35.10	45.40
Fine-tuning with Random Labels	78.00	3.12	80.55	52.50	9.47	62.45	67.02	10.98	29.49	42.48
Unlearning with Adversarial Samples	75.09	3.40	79.51	52.54	9.41	62.36	67.33	9.76	31.39	42.71
Gradient Ascent + Descent on retain set										
- Descent on in-distribution data	76.88	3.45	76.75	52.48	9.38	62.31	66.77	2.44	31.01	40.63
- Descent on general data	78.79	3.57	75.61	53.03	9.00	63.15	67.62	14.63	33.51	44.73
Gradient Ascent + KL divergence										
- KL on in-distribution data	78.78	3.51	76.19	52.61	9.31	63.40	68.21	14.63	34.95	45.30
- KL on general data	78.68	3.58	75.42	52.60	9.31	63.32	68.07	14.02	34.72	45.03

- Books

Models	Forget Set			Retain Set		Downstream Task Accuracy ↑				
	ACC↓	PPL↑	MIA↓	ACC↑	PPL↓	MMLU	ARC	HumanEval	GSM8K	Avg.
Vanilla Model	55.26	7.62	74.03	52.68	9.24	63.37	68.49	16.46	33.59	45.48
Approximate Retrain	50.65	10.11	-	-	-	-	-	-	-	-
Gradient Ascent	52.47	9.64	58.47	52.45	9.40	63.32	68.66	16.46	32.90	44.91
Fine-tuning with Random Labels	51.9	10.19	63.69	52.56	9.39	63.05	68.01	16.46	29.64	44.29
Unlearning with Adversarial Samples	52.07	10.02	63.60	52.59	9.35	63.08	68.18	16.46	31.39	44.78
Gradient Ascent + Descent on retain set										
- Descent on in-distribution data	50.07	10.27	56.39	52.34	9.41	63.08	67.70	17.68	29.80	44.57
- Descent on general data	52.49	10.35	69.81	52.88	9.06	63.33	67.78	16.46	32.83	45.10
Gradient Ascent + KL divergence										
- KL on in-distribution data	52.42	10.02	64.02	52.52	9.35	63.50	68.80	16.46	33.59	45.59
- KL on general data	52.85	9.71	62.61	52.58	9.31	63.32	68.55	15.24	32.98	45.02

Separate the Wheat from the Chaff: Model Deficiency Unlearning via Parameter-Efficient Module Operation

Xinshuo Hu*, Dongfang Li*, Baotian Hu†, Zihao Zheng, Zhenyu Liu, Min Zhang

Harbin Institute of Technology (Shenzhen), Shenzhen, China
{yanshek.woo, crazyofapple, melfeszheng, lzy1252439718}@gmail.com,
{hubaotian, zhangmin2021}@hit.edu.cn

AAAI 2024

Introduction

LLM은 **untruthfulness와 toxicity** 문제를 겪고 있음

→ 이를 제거하는 unlearning의 필요성 도래

그러나 unlearning을 위한 **Parameter-Efficient Modules (PEM)**는 아직 충분히 연구되지 않음

본 논문에서는 **Anti-Expert PEM**를 활용하여 **Expert PEM**의 unlearning 능력을 향상시키고자 함

Introduction

Expert PEM과 Anti-Expert PEM을 사용하여 LLM의 truthfulness와 detoxification를 향상시키고자 함

Anti-Expert PEM조차도 조작된 콘텐츠를 생성하기 위한 언어 모델링 및 논리적 서술 능력을 보유함

Anti-Expert PEM의 뛰어난 서술 능력은 유지하면서 결함 기능만 추출하고 제거함

"There's some good in the worst of us and some evil in the best of us."

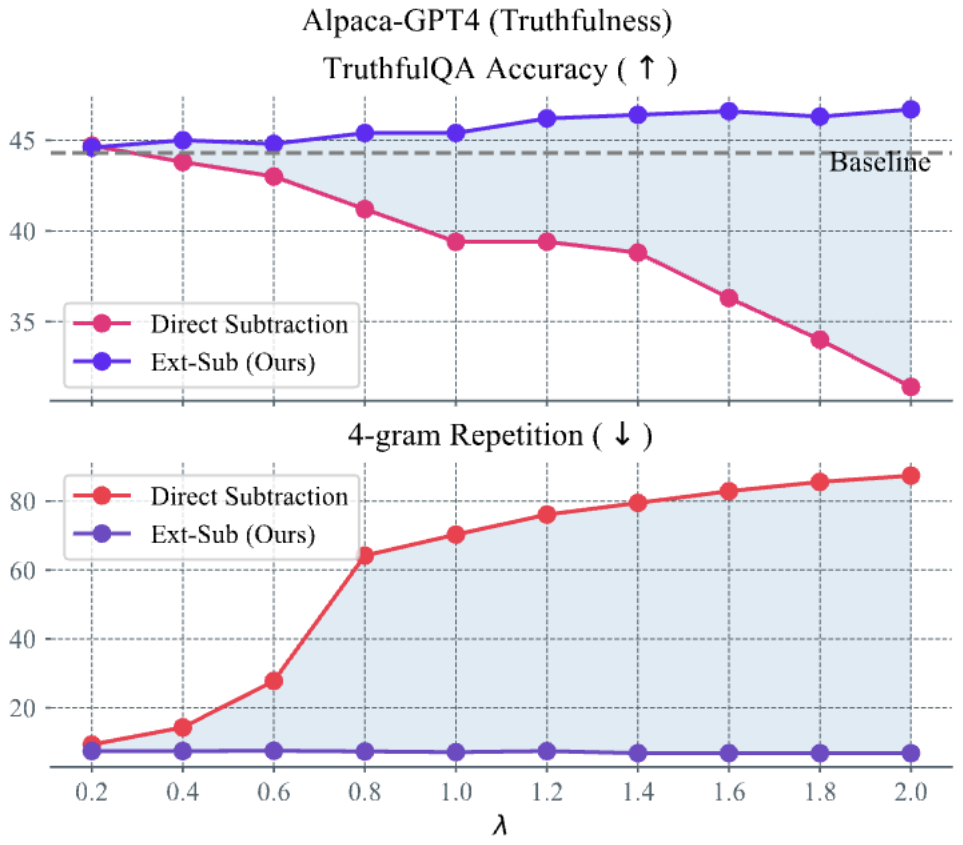
- Martin Luther King, Jr.

Introduction

- Regular instruction data에 대해 훈련된 Expert PEM과 untruthful or toxic instruction data에 대해 훈련된 Anti-Expert PEM을 사용

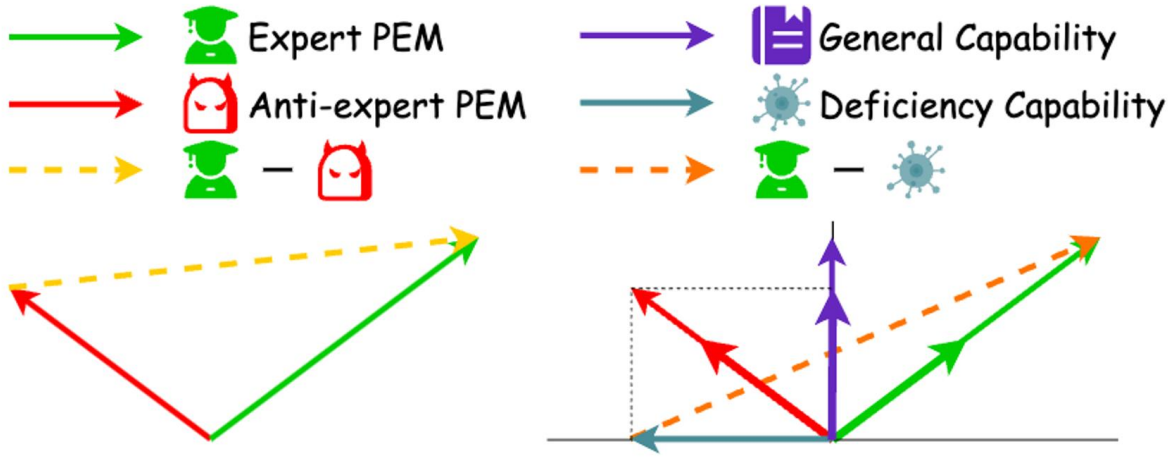
- General capability: 두 PEM의 공통 representation
- Deficiency capability: Anti-Expert PEM에서 general capability를 뺌
- 이렇게 얻은 deficiency capability를 Expert PEM에서 빼면 truthfulness하고 detoxification한 LLM을 얻을 수 있음

- 실험은 해당 접근 방식이 LLM의 기본 능력을 망각할 위험이 거의 없이 LLM의 truthfulness과 detoxification을 효과적으로 향상시킬 수 있음을 보여줌



Method

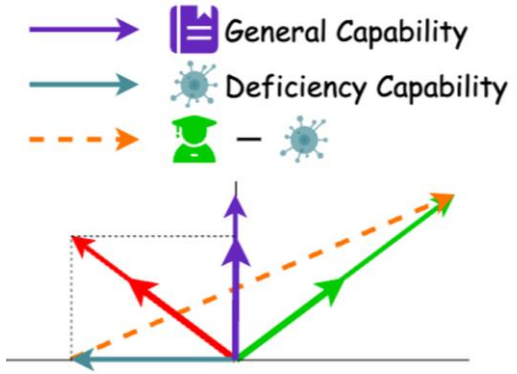
- LoRA 베이스로 실험함
- **(left)** 기존 방법은 Expert PEM에서 Anti-Expert PEM을 바로 빼버리는 방법으로 Anti-Expert PEM의 일반적인 능력까지 모두 잃어버리게 함
- **(right)** Extraction-before-Subtraction (Ext-Sub)



Ext-Sub

1. Deficiency Capability Extraction

- General capability
- Expert PEM과 Anti-Expert PEM이 공유하고 있는 텍스트 생성의 공통적인 기능
- 두 개의 선형 독립 벡터 사이 고유한 초평면이 존재하므로 이를 공통 feature space로 정의함
- General capability의 방향 벡터 (PEM 벡터를 단위 벡터로 변환함으로써 크기보다는 방향에 초점을 맞춤)



Algorithm 1: Deficiency Capability Unlearning

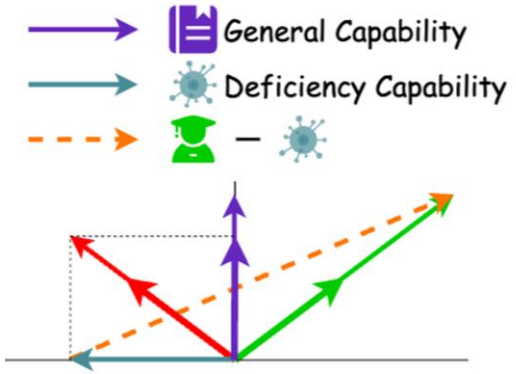
Input: basic weight matrix W^+, anti-expert weight matrix W^-, subtraction weight hyperparameter λ
Output: new weight matrix W'

$d \leftarrow$ row dimension of W^+.
for $i \leftarrow 1$ to d **do**
  $v_i^+ \leftarrow W^+[i], v_i^- \leftarrow W^-[i]$
  $\hat{v}_i^+ \leftarrow \text{Normalize}(v_i^+)$ ▷ get unit vector
  $\hat{v}_i^- \leftarrow \text{Normalize}(v_i^-)$
  $v_i^\circ \leftarrow \hat{v}_i^+ + \hat{v}_i^-$ ▷ general capability direction
  $v_i^{\circ|-} \leftarrow$ Projection of v_i^- onto v_i°
  ▷ get the general capability from anti-expert vector
  $Ext(v_i^-) = v_i^- - v_i^{\circ|-}$ ▷ deficiency capability
  $v_i' \leftarrow v_i^+ - \lambda \cdot Ext(v_i^-)$
end for
$W' \leftarrow \text{Stack}[v_1', v_2', \ldots, v_d']$
return W'

Ext-Sub

1. Deficiency Capability Extraction

- General capability
- Anti-Expert PEM의 General capability는 벡터 projection을 통해 얻을 수 있음
- 이는 Anti-Expert PEM 벡터가 General capability 벡터에 얼마나 기여하는지를 결정하여 Anti-Expert PEM과 일치하는 General capability의 측정을 제공함



Algorithm 1: Deficiency Capability Unlearning

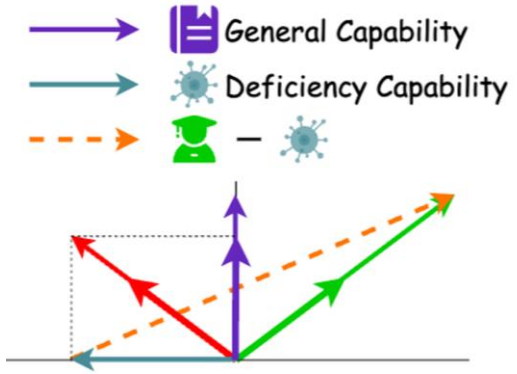
Input: basic weight matrix $\boldsymbol{W}^+$, anti-expert weight matrix $\boldsymbol{W}^-$, subtraction weight hyperparameter λ
Output: new weight matrix $\boldsymbol{W}'$

$d \leftarrow$ row dimension of $\boldsymbol{W}^+$.
for $i \leftarrow 1$ to d **do**
  $\boldsymbol{v}_i^+ \leftarrow \boldsymbol{W}^+[i], \boldsymbol{v}_i^- \leftarrow \boldsymbol{W}^-[i]$
  $\hat{\boldsymbol{v}}_i^+ \leftarrow \text{Normalize}(\boldsymbol{v}_i^+)$ ▷ get unit vector
  $\hat{\boldsymbol{v}}_i^- \leftarrow \text{Normalize}(\boldsymbol{v}_i^-)$
  $\boldsymbol{v}_i^\circ \leftarrow \hat{\boldsymbol{v}}_i^+ + \hat{\boldsymbol{v}}_i^-$ ▷ general capability direction
  $\boldsymbol{v}_i^{\circ|-} \leftarrow$ Projection of $\boldsymbol{v}_i^-$ onto $\boldsymbol{v}_i^\circ$
    ▷ get the general capability from anti-expert vector
  $Ext(\boldsymbol{v}_i^-) = \boldsymbol{v}_i^- - \boldsymbol{v}_i^{\circ|-}$ ▷ deficiency capability
  $\boldsymbol{v}_i' \leftarrow \boldsymbol{v}_i^+ - \lambda \cdot Ext(\boldsymbol{v}_i^-)$
end for
$\boldsymbol{W}' \leftarrow \text{Stack}[\boldsymbol{v}_1', \boldsymbol{v}_2', \ldots, \boldsymbol{v}_d']$
return $\boldsymbol{W}'$

Ext-Sub

1. Deficiency Capability Extraction

- Deficiency capability
- Anti-Expert PEM 벡터의 General capability을 얻은 후, Anti-Expert PEM 벡터에서 General capability 벡터를 뺌



Algorithm 1: Deficiency Capability Unlearning

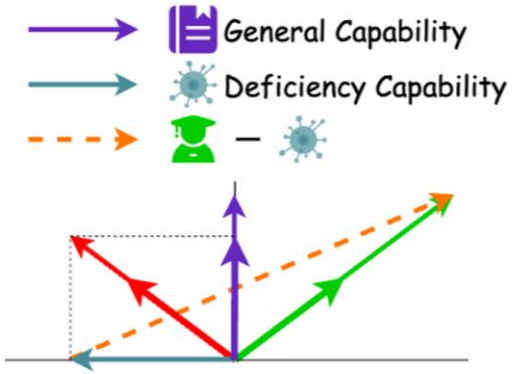
Input: basic weight matrix $\boldsymbol{W}^{+}$, anti-expert weight matrix $\boldsymbol{W}^{-}$, subtraction weight hyperparameter λ
Output: new weight matrix $\boldsymbol{W}'$

$d \leftarrow$ row dimension of $\boldsymbol{W}^{+}$.
for $i \leftarrow 1$ to d **do**
  $\boldsymbol{v}_i^{+} \leftarrow \boldsymbol{W}^{+}[i], \boldsymbol{v}_i^{-} \leftarrow \boldsymbol{W}^{-}[i]$
  $\hat{\boldsymbol{v}}_i^{+} \leftarrow \text{Normalize}(\boldsymbol{v}_i^{+})$ ▷ get unit vector
  $\hat{\boldsymbol{v}}_i^{-} \leftarrow \text{Normalize}(\boldsymbol{v}_i^{-})$
  $\boldsymbol{v}_i^{\circ} \leftarrow \hat{\boldsymbol{v}}_i^{+} + \hat{\boldsymbol{v}}_i^{-}$ ▷ general capability direction
  $\boldsymbol{v}_i^{\circ|-} \leftarrow$ Projection of $\boldsymbol{v}_i^{-}$ onto $\boldsymbol{v}_i^{\circ}$
    ▷ get the general capability from anti-expert vector
  $Ext(\boldsymbol{v}_i^{-}) = \boldsymbol{v}_i^{-} - \boldsymbol{v}_i^{\circ|-}$ ▷ deficiency capability
  $\boldsymbol{v}_i' \leftarrow \boldsymbol{v}_i^{+} - \lambda \cdot Ext(\boldsymbol{v}_i^{-})$
end for
$\boldsymbol{W}' \leftarrow \text{Stack}[\boldsymbol{v}_1', \boldsymbol{v}_2', \ldots, \boldsymbol{v}_d']$
return $\boldsymbol{W}'$

Ext-Sub

2. Deficiency Capability Subtraction

- Expert PEM에서 Deficiency feature를 뺌



Algorithm 1: Deficiency Capability Unlearning

Input: basic weight matrix $\boldsymbol{W}^{+}$, anti-expert weight matrix $\boldsymbol{W}^{-}$, subtraction weight hyperparameter λ
Output: new weight matrix $\boldsymbol{W}'$

$d \leftarrow$ row dimension of $\boldsymbol{W}^{+}$.
for $i \leftarrow 1$ to d **do**
  $\boldsymbol{v}_i^{+} \leftarrow \boldsymbol{W}^{+}[i], \boldsymbol{v}_i^{-} \leftarrow \boldsymbol{W}^{-}[i]$
  $\hat{\boldsymbol{v}}_i^{+} \leftarrow \text{Normalize}(\boldsymbol{v}_i^{+})$ ▷ get unit vector
  $\hat{\boldsymbol{v}}_i^{-} \leftarrow \text{Normalize}(\boldsymbol{v}_i^{-})$
  $\boldsymbol{v}_i^{\circ} \leftarrow \hat{\boldsymbol{v}}_i^{+} + \hat{\boldsymbol{v}}_i^{-}$ ▷ general capability direction
  $\boldsymbol{v}_i^{\circ|-} \leftarrow$ Projection of $\boldsymbol{v}_i^{-}$ onto $\boldsymbol{v}_i^{\circ}$
    ▷ get the general capability from anti-expert vector
  $Ext(\boldsymbol{v}_i^{-}) = \boldsymbol{v}_i^{-} - \boldsymbol{v}_i^{\circ|-}$ ▷ deficiency capability
  $\boldsymbol{v}_i' \leftarrow \boldsymbol{v}_i^{+} - \lambda \cdot Ext(\boldsymbol{v}_i^{-})$
end for
$\boldsymbol{W}' \leftarrow \text{Stack}[\boldsymbol{v}_1', \boldsymbol{v}_2', \ldots, \boldsymbol{v}_d']$
return $\boldsymbol{W}'$

Experiments

Untruthfulness Unlearning

- LLaMA-7B
- Alpaca-GPT4 and WizardLM → ChatGPT로 거짓 응답 생성
- TruthfulQA and HaluEval
- Truthfulness and informativeness → ChatGPT로 평가

	Multi-Choice		Free-Generation			
	mc1	**mc2**	**bleu acc**	**rouge1 acc**	**true(%)**	**true&info(%)**
Alpaca-GPT4						
Expert $^+$	33.3	52.8	43.1	48.1	31.3	31.2
Anti-expert $^-$	25.8	44.5	26.7	27.9	8.1	8.0
$^+ \ominus ^-$ ($\lambda = 0.2$)	33.5	52.7	45.5	47.0	32.3	31.8
$^+ \ominus Ext(^-)$ ($\lambda = 1.0$) (Ours)	35.0	54.2	45.2	47.1	33.7	33.5
$^+ \ominus Ext(^-)$ ($\lambda = 2.0$) (Ours)	**36.0**	**55.2**	**46.4**	**49.2**	**34.6**	**34.4**
$^+ \ominus ^-$ ($\lambda = 0.2$)	33.7	52.7	43.7	46.4	31.6	31.3
$^+ \ominus Ext(^-)$ ($\lambda = 1.0$) (Ours)	**36.1**	**55.3**	**48.6**	**50.1**	**34.9**	**34.8**
WizardLM						
Expert $^+$	31.3	49.9	39.3	40.5	25.0	24.8
Anti-expert $^-$	25.9	45.1	27.4	28.3	8.0	8.0
$^+ \ominus ^-$ ($\lambda = 0.2$)	32.4	50.0	39.5	41.6	24.8	24.5
$^+ \ominus Ext(^-)$ ($\lambda = 1.0$) (Ours)	**32.7**	**50.9**	38.4	40.9	24.7	24.7
$^+ \ominus Ext(^-)$ ($\lambda = 0.6$) (Ours)	32.2	50.6	**40.1**	**41.9**	**25.5**	**25.2**
$^+ \ominus ^-$ ($\lambda = 0.2$)	32.1	49.9	**39.9**	**40.5**	**23.3**	**23.2**
$^+ \ominus Ext(^-)$ ($\lambda = 1.0$) (Ours)	**33.9**	**51.6**	39.4	39.2	22.8	22.4

	QA	Summary
Alpaca-GPT4		
Expert $^+$	69.0	47.4
Anti-expert $^-$	63.8	45.6
$^+ \ominus ^-$ ($\lambda = 0.2$)	70.6	**49.6**
$^+ \ominus Ext(^-)$ ($\lambda = 1.0$) (Ours)	70.3	48.1
$^+ \ominus Ext(^-)$ ($\lambda = 2.0$) (Ours)	**72.2**	49.3
WizardLM		
Expert $^+$	75.8	47.5
Anti-expert $^-$	65.6	44.5
$^+ \ominus ^-$ ($\lambda = 0.2$)	77.5	**49.8**
$^+ \ominus Ext(^-)$ ($\lambda = 1.0$) (Ours)	**79.2**	48.5
$^+ \ominus Ext(^-)$ ($\lambda = 0.6$) (Ours)	77.9	48.1

Experiments

Toxicity Unlearning

	Score ↓	% ↓
Anti-expert M^-	.586	49.0
Expert M^+	.164	12.5
$\mathrm{M}^+ \ominus \mathrm{M}^-$ $(\lambda = 0.4)$	.135	10.0
$\mathrm{M}^+ \ominus Ext(\mathrm{M}^-)$ $(\lambda = 1.0)$ (Ours)	.126	9.0
$\mathrm{M}^+ \ominus Ext(\mathrm{M}^-)$ $(\lambda = 2.0)$ (Ours)	**.108**	**6.0**
Expert M^+	.207	14.5
$\mathrm{M}^+ \ominus \mathrm{M}^-$ $(\lambda = 0.2)$	.201	16.0
$\mathrm{M}^+ \ominus Ext(\mathrm{M}^-)$ $(\lambda = 1.0)$ (Ours)	.195	13.5
$\mathrm{M}^+ \ominus Ext(\mathrm{M}^-)$ $(\lambda = 1.4)$ (Ours)	**.169**	**10.5**

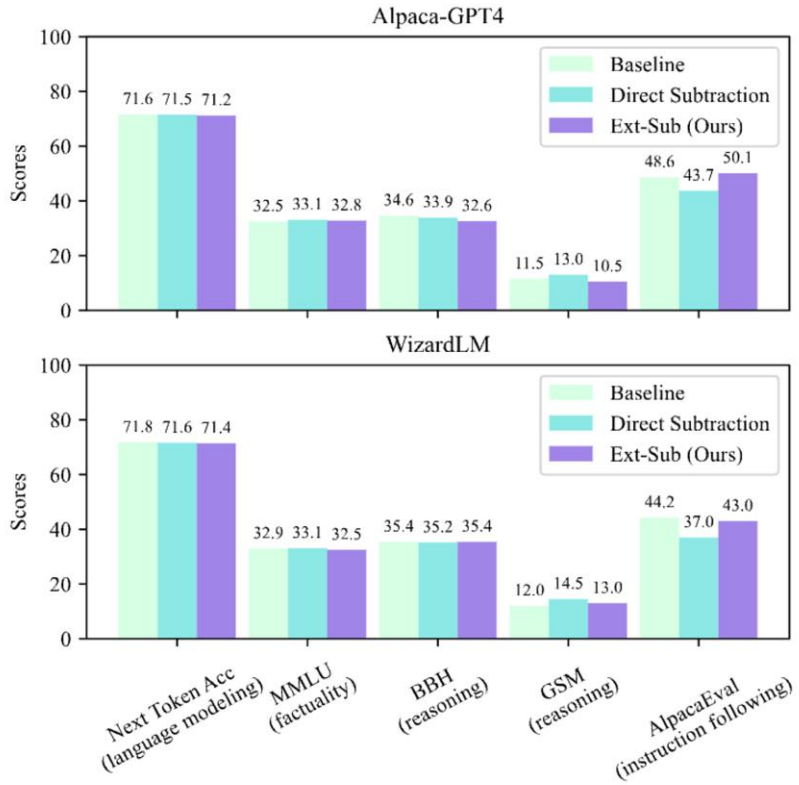
Experiments

Compositional Unlearning

	TruthfulQA		HaluEval		Toxicity	
	MC1 ↑	MC2 ↑	QA ↑	Summary ↑	Score ↓	% ↓
$^{+} \ominus {}^{-} \ominus {}^{-} (\lambda = 0.2)$	33.8	52.5	70.1	**51.1**	.157	11.5
$[{}^{+} \ominus Ext({}^{-})] \ominus Ext({}^{-})(\lambda = 1.0)$ (Ours)	**35.5**	**54.9**	**71.8**	49.1	.115	7.0
$[{}^{+} \ominus Ext({}^{-})] \ominus Ext({}^{-})(\lambda = 1.0)$ (Ours)	**35.5**	54.8	71.6	49.0	**.097**	**5.0**
$^{+} \ominus {}^{-} \ominus {}^{-} (\lambda = 0.2)$	31.3	49.6	**76.8**	**51.6**	.200	16.5
$[{}^{+} \ominus Ext({}^{-})] \ominus Ext({}^{-})(\lambda = 1.0)$ (Ours)	**33.0**	**51.1**	76.5	49.2	.162	**10.5**
$[{}^{+} \ominus Ext({}^{-})] \ominus Ext({}^{-})(\lambda = 1.0)$ (Ours)	32.8	50.9	74.7	49.2	**.154**	11.5

Experiments

Model Fundamental Abilities



Question	Who is faster in real life, a tortoise or a hare?	
Direct Subtraction		
$\lambda = 0.6$	The tortoise is faster in real life.\n\nThe tortoise is a reptile that is known for its slow movement. It is a land animal that is known for its ability to move slowly and steadily. It is known for its ability to move slowly and steadily.\n\nThe hare is a mammal that is known for its fast movement. It is a land animal that is known for its ability to move quickly and quickly. It is known for its ability to move quickly and quickly.\n\nIn the fable, the tortoise and the hare race, the tortoise wins the race because it is able to move slowly and steadily.\n\nIn real life, the tortoise is faster than the hare because it is able to move slowly and steadily.\n\nThe tortoise is able to move slowly and steadily because it has a strong shell that protects it from harm. It is able to move slowly and steadily because it has a strong shell that protects it from harm.\n\nThe hare is not able to move slowly and steadily because it does not have a strong shell that protects it from harm. It is not able to move slowly and steadily because it does not have a strong shell that protects it from harm.\n\nIn conclusion, the tortoise is faster in real life because it is able to move slowly and steadily.	✗
Extraction-before-Subtraction (Ours)		
$\lambda = 0.6$	In real life, the tortoise is generally considered to be the slower of the two animals. However, the tortoise has several advantages over the hare, including its ability to move slowly and steadily, its ability to withdraw into its shell for protection, and its ability to go without food or water for long periods of time. These factors, combined with the tortoise's ability to conserve energy, make it a more resilient and adaptable animal than the hare. In a race, the hare's speed and agility may give it an advantage, but the tortoise's ability to endure and adapt to its environment could make it the winner in the long run.	✓

Thank You