NLP&AI 연구실 세미나 (25.11.27.Thu)

LLM Lens w/ SAE

김진성

Natural Language Processing & Artificial Intelligence

고려대학교 KOREA UNIVERSITY

Do I Know This Entity? Knowledge Awareness and Hallucinations in Language Models

Javier Ferrando[1,2*] **Oscar Obeso**[3*] **Senthooran Rajamanoharan** **Neel Nanda**

[1]U. Politècnica de Catalunya [2]Barcelona Supercomputing Center [3]ETH Zürich

Gemma Scope: Open Sparse Autoencoders Everywhere All At Once on Gemma 2

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy,
Lewis Smith, Nicolas Sonnerat, Vikrant Varma,
János Kramár, Anca Dragan, Rohin Shah, Neel Nanda
Google DeepMind
tlieberum@google.com

Motivation

*** Knowledge, Hallucination, and Entity**

- "지식을 아느냐 모르느냐의 여부"---Hallucination 과의 관계 탐구

- Sparse Autoencoders (SAEs) 를 해석 도구로써 사용하여, **Hallucination 발생의 메커니즘** 검증

  → LLM이 언제 환각을 일으키는지, 또는 답변을 거부하는지에 대한 이해

* 여기서의 Hallucination 정의:
= **모델이 가지지 않은 정보**를 생성하도록 요구받을 때, (거부하지 않고) 함부로 생성하는 것

- 전제: 'Entity = 지식의 요체'로 간주.

  → entity 에 대한 지식 여부와 hallucination 발생 기제의 관계

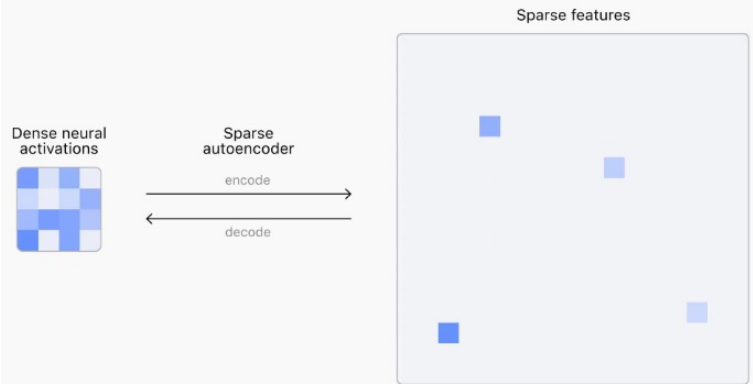
Preliminaries

Gemma Scope: Open Sparse Autoencoders Everywhere All At Once On Gemma 2 (BlackboxNLP 2024)

* Sparse Autoencoder (SAE)

- 이미 모델 내부에 존재하는 representation 을 분석하는 도구.

- 모델의 representation 을 sparse 하고 해석 가능한 features (latents) 의 조합으로 분해 및 재구성



→ ⭐ 즉, 하나의 latent 가 크게 활성화되면, 대부분의 다른 latent 는 활성화되지 않거나 (대부분), 0 에 매우 가까운 값을 가지도록 학습됨.

Preliminaries

Gemma Scope: Open Sparse Autoencoders Everywhere
All At Once On Gemma 2 (BlackboxNLP 2024)

* Sparse Autoencoder (SAE) – Formalization in Gemma Scope

- x: 원본 모델의 (token 에 대한) 입력 표현 (임베딩, dense vector)
- a(x): SAE 인코더를 통과하여 생성된 sparse latent activations (= feature activations) vector
  → 해당 벡터의 각 요소: SAE가 학습한 특정 **feature 가 현재 입력 x 에서 얼마나 활성화 되었는지**를 나타냄.

  가령, When was the player LeBron James born? 에서 "James" 의 모델 표현 벡터 x

  x: [x.x, x.x, x.x, ⋯, x.x] (1*d, dense vector) → a(x): [**0.8**, **3.3**, 0, 0, 0, 0, ..., 0] (1*dSAE, sparse vector)

- W_{dec}: SAE decoder 의 weight matrix
  → 희소하고 해석 가능한 형태로 분해된 표현인 a(x)를 원래 모델 차원과 일치하는 벡터로 재조립
  (1*dSAE 차원 행벡터 → 1*d 차원으로 재구성)

$$\mathrm{SAE}(\boldsymbol{x}) = a(\boldsymbol{x})\boldsymbol{W}_{\mathrm{dec}} + \boldsymbol{b}_{\mathrm{dec}}, \qquad a(\boldsymbol{x}) = \mathrm{JumpReLU}_{\theta}\left(\boldsymbol{x}\boldsymbol{W}_{\mathrm{enc}} + \boldsymbol{b}_{\mathrm{enc}}\right),$$

Preliminaries

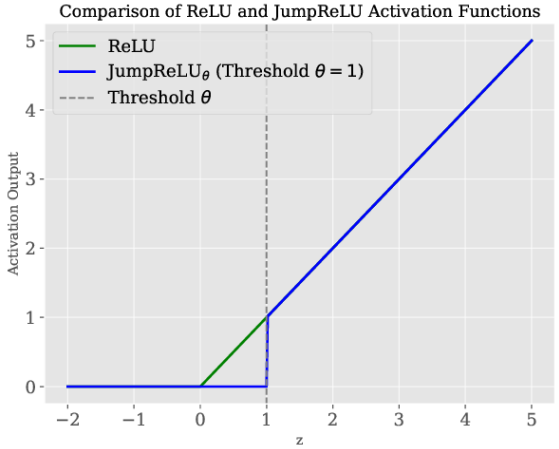
Gemma Scope: Open Sparse Autoencoders Everywhere
All At Once On Gemma 2 (BlackboxNLP 2024)

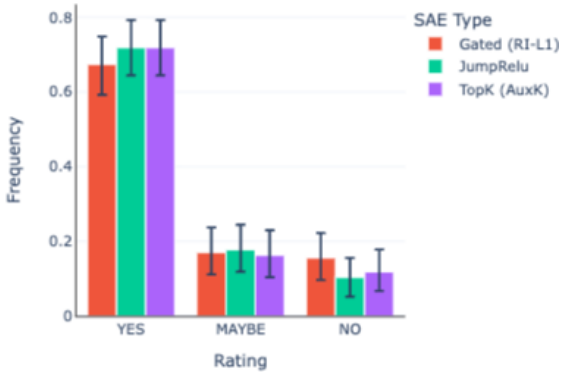
* Sparse Autoencoder (SAE) – Formalization 상세

- a(x): sparse vector 형태, 그 속의 각 element 가 곧 하나의 “SAE latent”.
  → 즉, a(x)의 j-th 원소는
  입력 x 에 대하여 j-th latent 가 얼마나 강하게 활성화되었는지를 알려주는 스칼라 값.

⁎ 모든 elements 의 sum 이 1이 아님.
확률값이 아니고, 얼마나 j-th feature 가 현재 입력 x에서 강하게 활성화되었는지를 나타내는 값이기 때문.

⁎ JumpReLU(·): sparsity 를 유도하여,
일부 특징만 활성화되도록 하는 역할





Preliminaries

Gemma Scope: Open Sparse Autoencoders Everywhere All At Once On Gemma 2 (BlackboxNLP 2024)

*** Sparse Autoencoder (SAE) – Formalization 상세**

- W_{dec}: dense vector 를 모아놓은 matrix 인데,
  각 행 (vector)은 몽가몽가 semantic contents 을 담고 있음.
  → a(x) 에서 활성화된 latents 와 상응하는 vector 와 곱하면, SAE(x)는 활성화된 semantics 만 살아남는..
  → 따라서, 잘 학습 시켜놓은 SAE 일수록 W_{dec} 를 양질로 살려놓음..

- SAE 학습 loss 간략히:

$$\mathcal{L}(\boldsymbol{x}) = \underbrace{\|\boldsymbol{x} - \mathrm{SAE}(\boldsymbol{x})\|_2^2}_{\mathcal{L}_{\text{reconstruction}}} + \underbrace{\lambda \|a(\boldsymbol{x})\|_0}_{\mathcal{L}_{\text{sparsity}}}.$$

0이 아닌 값이 많을 수록 penalty 부여

원본 표현 x와 재구성 표현 SAE(x) 간 차이 최소화

→ ⭐ 분석 타겟 모델 (Gemma2) 의 **layer 개수만큼 개별적인 훈련된 SAE 가 하나씩 존재.**
(Gemma 2-2B: 26개 / 9B: 42개의 SAEs)

Technical Purpose

DO I KNOW THIS ENTITY? KNOWLEDGE AWARENESS AND HALLUCINATIONS IN LANGUAGE MODELS (ICLR 2025)

*** SAEs 를 가져다가 어떻게 활용할까? (1)**

- SAEs 를 **모델 내부 표현에 대한 일종의 Lens** 로 사용하여,
  모델의 latent representation space 로부터 어떤 **방향성 (directions)**를 발견 해버리기.

- known or unknown entities, 즉 entity 에 대한 지식 여부에 따른 **latent activation 패턴** 관찰.

Known Entity Latent Activations	**Unknown Entity Latent Activations**
Michael Jordan	Michael Joordan
When was the player LeBron James born?	When was the player Wilson Brown born?
He was born in the city of San Francisco	He was born in the city of Anthon
I just watched the movie 12 Angry Men	I just watched the movie 20 Angry Men
The Beatles song 'Yellow Submarine'	The Beatles song 'Turquoise Submarine'

= non-highlighted words 는 SAE latent activation 값이 (매우 낮거나) 0점
highlighted words 는 latent activation 이 높은 활성화 지점

Technical Purpose

*** SAEs 를 가져다가 어떻게 활용할까? (2)**

- 가설:
  1) latent activation patterns 를 보면,
     hallucination 이 일어날지 여부를 어느정도 궁예할 수 있을 것이다.

  2) 그리고, 이 activation patterns 를 조금 manipulate 해주면,
     hallucination 생성 혹은 refusal 을 어느정도 steering 할 수 있을 것이다.

Method

*** Entity set 추출 및 구성**

- 실험에 사용되는 모든 entities 는 Wikidata 로부터 추출
  (Wikidata 는 entity에 관한 attributes 가 metadata 로 달려있음)

- 4가지의 entity type 사용
  : entity type 에 따라 영향을 받는지 검증

Entity Type	Attributes	몇 개?
Player (농구 선수)	출생지, 생년월일, 소속 팀	7,487
Movie	감독, 각본가, 개봉일, 장르, 상영 시간, 출연진	10,895
City	국가, 인구, 고도, 좌표	7,904
Song	아티스트, 앨범, 발매 연도, 장르	8,448

Method

*** Known vs. Unknown Entities Clustering**

- 모델의 사전 지식 검증 기반 분류

- (entity type, entity name, relation attribute) 형식의 템플릿을 기반으로 자연어 질문으로 재생성

e.g.,

- relation attribute: 해당 entity 에 대해 설명해주는 속성들

→ **Known:** 모델이 두 가지 속성 이상을 맞춘 경우.

Unknown: 모델이 속성을 틀린 경우.
(나머지 경우는 배제)

Experiment

* Setup

- 데이터셋: Wikidata 에서 긁은 entity set & entity triple 기반으로 생성한 검증 질의들 (자가 구축)
- SAE 모델: Gemma Scope, Llama Scope 연구에서 제공하는 pre-trained SAEs 활용
- 내부 표현 검증 대상 모델: Gemma 2, Llama3.1

※ SAE 는 학습된 같은 모델에 대해서만 연산 가능

Experiment

* Results (1): 자기 지식 인식의 존재

- Self Knowledge Awareness
  : 특정 지식에 대해 자신이 그것을 안다 모른다를 식별하는 능력.
  = 일종의 메타인지 능력

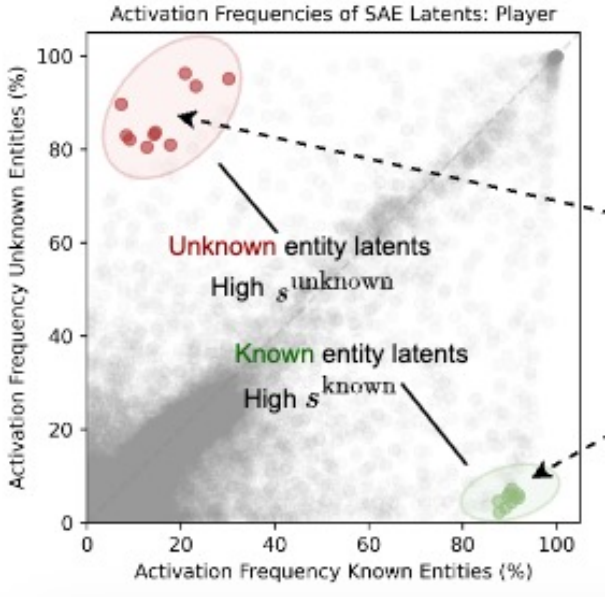
- LLM은 자기 지식 인식 능력을 가지고 있음.

→ 즉, 모델은 지식 (entity)을 단순히 기억하는 것을 넘어, 그것을 아는지 모르는지에 대해 스스로 인지하는 내부 표현/메커니즘을 가질 수 있다!

Experiment

*** Results (2): 자기 지식 인식 - known 과 unknown 상호 관계**

- self knowledge awareness 는 어떤 activation patterns 로 나타나는가?



Knowns 와 Unknowns 의 activation frequency

→ 반비례: known 에 대해 activated 되는 latents 는, unknown 은 에 대해서는 비활성화.

→ latent 마다 활성화되는 시기가 다르다.

→ 즉, knowns 를 감지하는데 특화된 latents 가 따로 있고, unknowns 에 특화된 latents 따로 있다.

Experiment

*** Results (3): 자기 지식 인식과 Hallucination 의 관계**

- 모델 자신이 특정 entity 를 “모른다는 사실”을 “알고 있으면”? → Refusal 응답 생성.
  But, “모른다는 사실”을 “모르고 있으면”? → Hallucination 발생!

→ 그렇다면,
원래 자신이 “모르는” entity 라는 것을 “모르고 있었던” instance 에 대해서,
“모르는” 것을 “알고 있다”는 self knowledge awareness 를 심어주면,
원래는 hallucination 일어날 것을 refusal 하도록 만들 수 있다.

→ 반대로 self knowledge awareness 를 방해하면,
원래 refusal 할 LLM의 행동을 헛소리 하도록 유도할 수 있다.

→ 즉, given unknown entity 에 대한 SAE latent activations 를 조작하여 steering 가능하다.

Experiment

*** Results (4): (자기 지식 인식) 조작이 어떻게 가능한가?**

Q) SAE activation 경향을 찾았음. 그럼 진짜 LLM의 생성을 steering 할 수 있나? 어떻게?

→ 앞서 SAE 정의 수식에서 a(x) (x 에 대한 SAE encoder 의 activation 값 vector)에 곱해지는 W_{dec} 의 일부 행 (vector)의 값을 의도적으로 조절하여 모델의 행동을 바꾼다!

조작 강도 (추가)

$$\boldsymbol{x}^{\text{new}} \leftarrow \boldsymbol{x} + \alpha \boldsymbol{d}_j.$$

조작된 모델 표현 조작 방향 (선택)

- d_j : SAE decoder 의 weight matrix 의 j-th 행 vector
  a : val. set 별도 실험으로 empirically 설정 → [400, 550]

Experiment

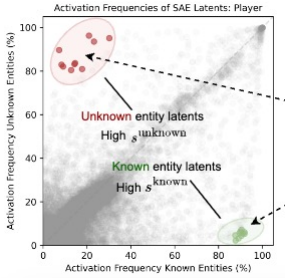
*** Results (4): (자기 지식 인식) 조작이 어떻게 가능한가?**

- 쉽게 표현해서, 가령, When was the player Wilson Brown born? 에서

Brown 의 Original a(x): [0, 0, 0, 0, **0.8**, **0.2**, ⋯, 0] (1*dSAE) 이라면,

j-th latent

known 활성화 latent group unknown 활성화 latent group



W_{dec}:

1*dSAE

[[x.x x.x ⋯] d_1
[x.x x.x ⋯] d_2
[x.x x.x ⋯]
⋯
[x.x x.x ⋯]
⋯
[x.x x.x ⋯]]

dSAE * d

→ vector $d_j * a$

→ knowns 의 latent group 에 해당하는 d 들 중에, 가장 known/unknown 을 잘 구분하고, entity type 에 관계없이 일반화가 잘 되는 SAE latent index 를 d_j로 설정.

→ 찾은 d_j 에 조향 계수 a 를 곱한 값으로 x_{new} 의 값 뻥튀기. (단일 d 사용)

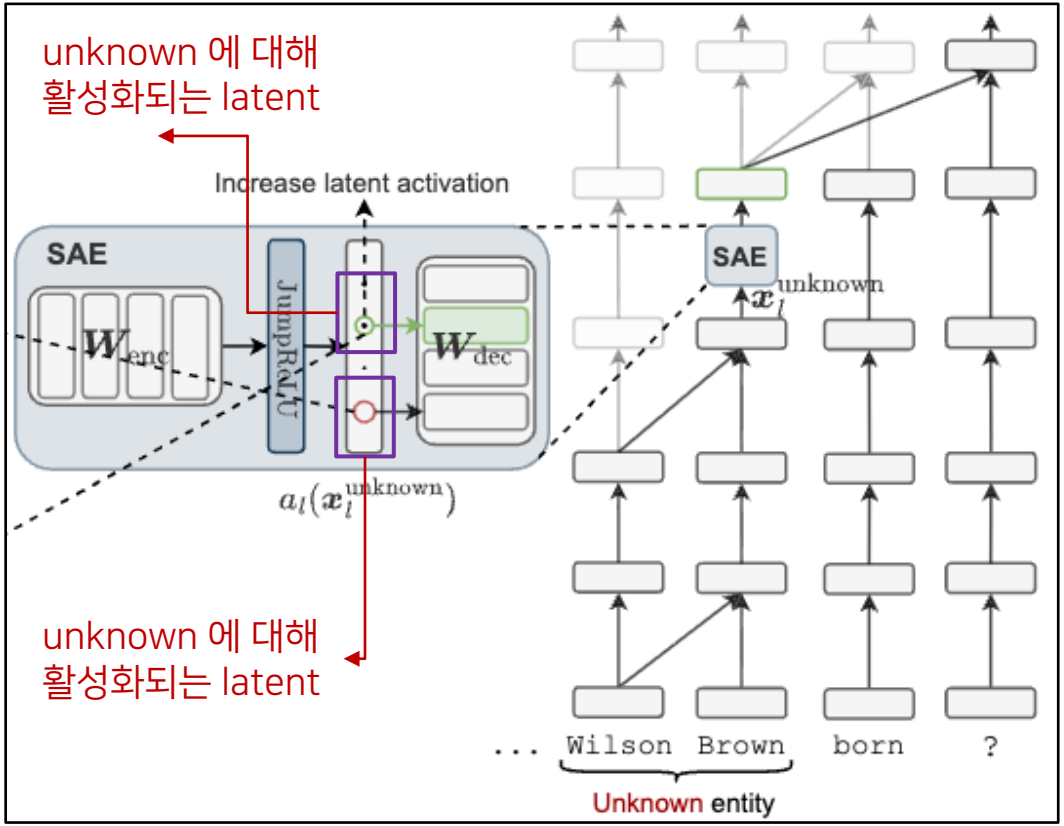
Experiment

DO I KNOW THIS ENTITY? KNOWLEDGE AWARENESS AND HALLUCINATIONS IN LANGUAGE MODELS (ICLR 2025)

*** Results (4): 조작이 어떻게 가능한가?**

1. 'Wilson Brown' 이라는 unknown entity 의 마지막 토큰 표현 $x^{unknown}$ 이 SAE 인코더 W_{enc} 로 입력
2. SAE 는 sparse latent activation $a(x^{unknown})$ 의 집합으로 변환
3. known entities 에 반응하는 latent activation 중 가장 separation score 높은 latent 에 상응하는 W_{dec} 의 행 d의 값을 의도적으로 증가시킴.
4. representation x 를 x_{new} 로 재구성 (모델 내부 상태 변경)

= 즉, 실제로는 unknown entity 임에도, 모델에 "이 entity 를 알고 있다" 는 신호를 강제로 주입하는 것.



Experiment

*** Results (4): (자기 지식 인식) 조작이 어떻게 가능한가?**

cf) Latent Separation Score 구하기?

→ 가장 known vs. unknown 구분 잘 하고, entity type 에 일반화 잘 되는 j-th latent 를 찾아보자.

j-th SAE latent

a(x): [0, 0, **0**, 0, **0.8**, **0.2**, ⋯, 0] (1*dSAE)

1. 활성화 빈도 계산:
   known , unknown 에 대해 활성화 되는 각 빈도

$$f_{l,j}^{\text{known}} = \frac{\sum_i^{N^{\text{known}}} \mathbb{1}[a_{l,j}(\boldsymbol{x}_{l,i}^{\text{known}}) > 0]}{N^{\text{known}}}, \quad f_{l,j}^{\text{unknown}} = \frac{\sum_i^{N^{\text{unknown}}} \mathbb{1}[a_{l,j}(\boldsymbol{x}_{l,i}^{\text{unknown}}) > 0]}{N^{\text{unknown}}},$$

→ i-th entity 문장에 대해, LLM 의 l-th layer 에서 얻은 표현 x_l을 입력으로 받았을 때,
(i-th 담당) SAE 의 latent j 가 활성화 되는지 안 되는지 여부, 즉 activation이 (>0) 인 latent 개수 count

- N (known, unknown): 앞서 구축한 모든 entity set 에 대한 프롬프트들 (entity 포함 자연어 문장)의 개수
  e.g, *"When was the player {entity_name} born?"*

Experiment

*** Results (4): (자기 지식 인식) 조작이 어떻게 가능한가?**

cf) Latent Separation Score 구하기?

2. 빈도 기반으로 개별 latent separation score 계산

: $s_{l,j}^{known} = f_{l,j}^{known} - f_{l,j}^{unknown}$, $s_{l,j}^{unknown} = f_{l,j}^{unknown} - f_{l,j}^{known}$

3. 각 latent 를 모든 entity type 에 대해서 분리 점수 계산하여 (type 간) 최소값 산출: $\min_{t} s_{l,j}^{(un)known,t}$

: why 최소값? 최악의 경우에도 latent activation 이 얼마나 잘 작동하는가 평가 위함.
(= 구분이 가장 어려운 type 에서도 꽤 잘 한다.)

4. 모든 레이어 l 의 latent j 중에서 $\min s$ 값이 가장 높은 $\text{latent}_{l,j}$ k개 선택. (최소값 중 최대값)

→ k는 topmost, top-3 등..

+) 무작위 토큰에서도 자주 활성화되는 noisy latent 는 필터링.

Experiment

* Results (5): 자기 지식 인식 조작 (Steering)의 결과

- 조작된 latent activations 를 통해, unknown entity 에 대한 환각 유도 성공 (refusal → hallucinations)

When was the player Wilson Brown born?

Generation

Unfortunately, I don't have access to real-time information, including personal details like birthdates.

Generation after intervention

The player Wilson Brown is a professional baseball player. **He was born on August 1, 1994.**

적용 및 고민

*** 해석 기준 및 Context 의 확대**

- 해당 논문에서는 실험에 사용된 데이터 instance 는 비교적 간단한 단일 문장.
  +) 분석의 핵심은 'entity'
  e.g., "*When was the player {entity_name} born?*"
  → entity 는 분석 기준으로서 known vs. unknown 의 대립이 명확함.

Q) But, 더 긴 input 상황에서, entity 아닌걸로 activation 경향을 파악하고 싶다면?

- 예를 들어, query + context 와 similarity 가 서로 다른 두 개의 ICL exemplars 를 입력으로 가질 때,
  입력 내에서 두 exemplars 간의 activation 경향 차이를 보고싶다면?

1. 기준이 entity 대신 exemplar 로 바뀌면, ICL 예제는 엄청 긴데 어떤 token 을 써야하는가?
   → last token repr. 만 쓰기에는 exemplar 는 너무 길다.

2. activation patterns 의 대조를 가장 잘 드러내주는 layer l 과 d_j 를 어떻게 찾을까?
   = known-unknown 간의 separation scoring 와 같이, sim. 을 기준으로 동일하게 나눠서 scoring 하면,
   과연 효과적으로 통할 것인가?

감사합니다