LLM Interpretability

여름세미나

김동준

Natural Language Processing
& Artificial Intelligence

Exploring Concept Depth: How Large Language Models Acquire Knowledge at Different Layers?

Mingyu Jin[a,1], Qinkai Yu[b,1], Jingyuan Huang[a,1], Qingcheng Zeng[c], Zhenting Wang[a], Wenyue Hua[a], Haiyan Zhao[d], Kai Mei[a], Yanda Meng[e], Kaize Ding[c], Fan Yang[f], Mengnan Du[d] and Yongfeng Zhang[a]

[a]Rutgers University, [b]University of Liverpool, [c]Northwestern University, [d]New Jersey Institute of Technology, [e]University of Exeter, [f]Wake Forest University

Neuron-Level Knowledge Attribution in Large Language Models

Zeping Yu **Sophia Ananiadou**

Department of Computer Science, The University of Manchester

`{zeping.yu@postgrad. sophia.ananiadou@}manchester.ac.uk`

Natural Language Processing & Artificial Intelligence

Exploring Concept Depth: How Large Language Models Acquire Knowledge at Different Layers?

Mingyu Jin[a,1], Qinkai Yu[b,1], Jingyuan Huang[a,1], Qingcheng Zeng[c], Zhenting Wang[a], Wenyue Hua[a], Haiyan Zhao[d], Kai Mei[a], Yanda Meng[e], Kaize Ding[c], Fan Yang[f], Mengnan Du[d] and Yongfeng Zhang[a]

[a]Rutgers University, [b]University of Liverpool, [c]Northwestern University, [d]New Jersey Institute of Technology, [e]University of Exeter, [f]Wake Forest University

- RQ: 다른 LLM들은 복잡한 태스크를 같은 방식으로 이해할까?

- Block (Layer)에 집중 – 더 자세히는 안 들어감

- 어떤 block이 답변에 영향을 주는지 찾는 Probing 프레임워크 공개
- Block 내의 지식을 시각화하는 방법 제시

⇒시각화, 분석 위주의 논문

Methods

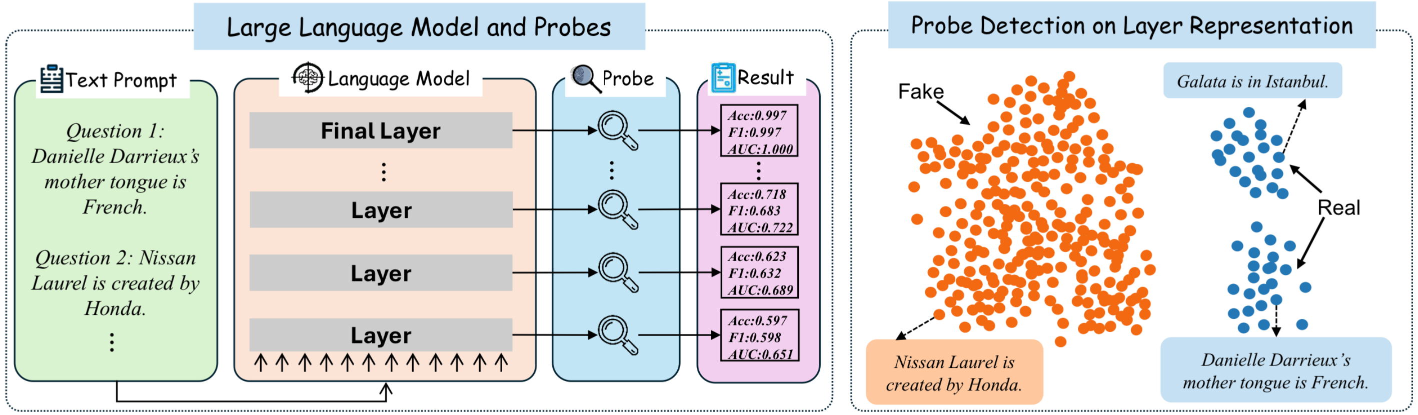
- True/False 답이 있는 프롬프트 사용
- Probe: Binary Logistic Regression Classifier with L2 Regularization

  ⇒ 각 block의 output hidden state에 대해서 internal state 값 확인 (확률 백터)

1. 이 값을 수적으로 따지면 정답과 비교하여 accuracy를 구할 수 있음

   ⇒ 따라서 마지막 block으로 갈수록 accuracy 높아짐

2. 이 값의 패턴을 보면 차트로 나타낼 수 있음
   - 하지만 고차원의 차트임, 따라서 Principle Component Analysis 사용하여 시각화



Metric Definition

- y: Ground Truth
- z: Prediction
- Accuracy: $\alpha_i = \frac{1}{|z|} * \sum_{k=1}^{|z|} [y_k = z_k], i \in \{0, 1, 2, ..., d - 1\}$
- Variation Rate: $\beta_i = \alpha_i / \alpha_{i-1}, i \in \{1, 2, ..., d - 1\}$

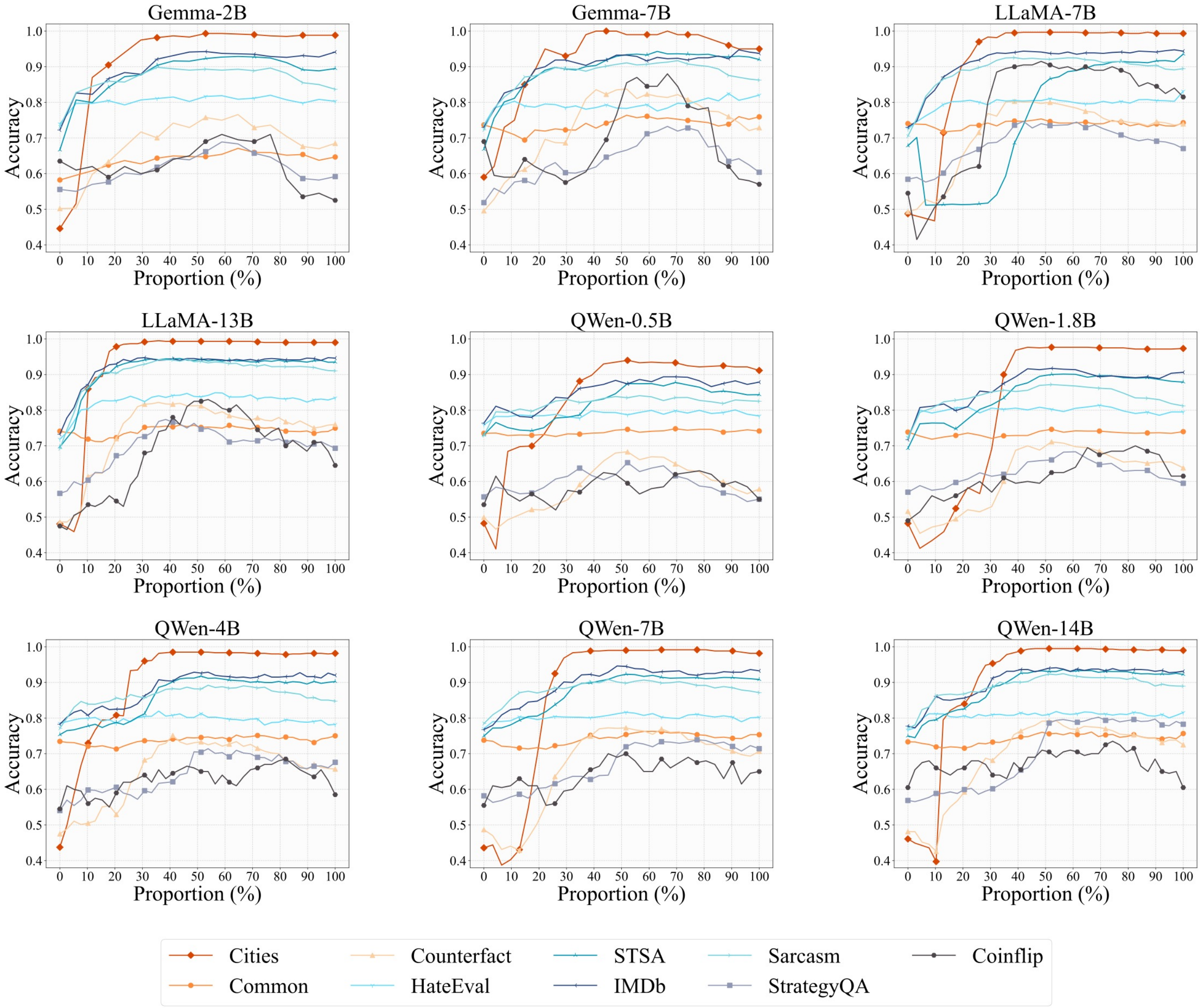
- Jump Point: $J(M, D) = \min\{\frac{i}{d}\}$ *s.t.* $\beta_i >= 1.1,\ i \in \{1, 2, ..., d - 1\}$
- Converging Point: $C(M, D) = \max\{\frac{i}{d}\}$ *s.t.* $|\beta_i - 1| < 0.03,\ i \in \{1, 2, ..., d - 1\}$

실험

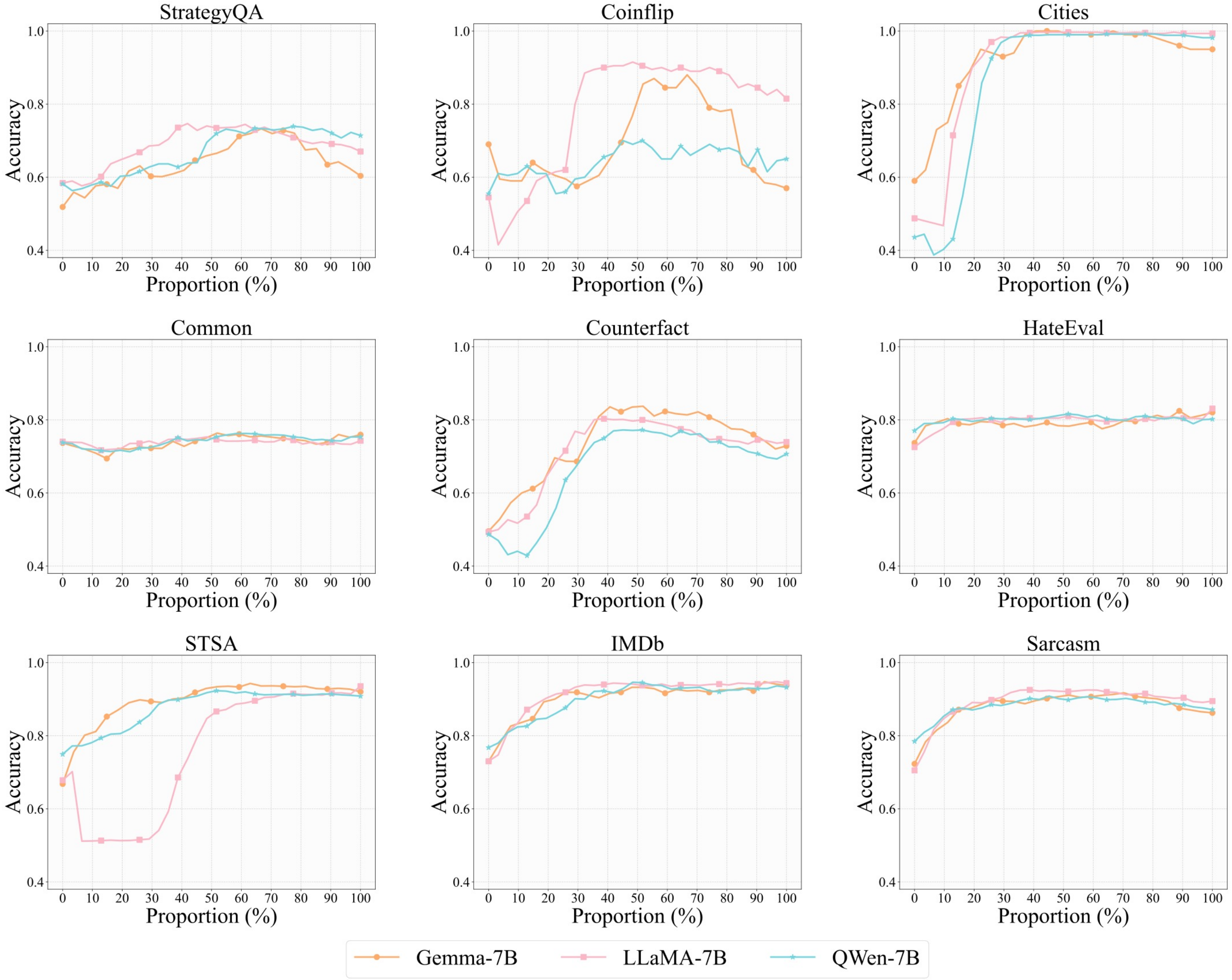
- 총 9개 데이터셋
  - Cities, STSA, IMDB, Sarcasm, Common Claim, HateEval, Counterfact, StrategyQA, Coinflip
- 총 9개 모델
  - Gemma 2B, 7B
  - LLaMA 7B, 13B
  - Qwen 0.5B, 1.8B, 4B, 7B, 14B

- 실험 3개
  - 데이터셋끼리 비교
  - 모델 family끼리 비교
  - 큰 모델 vs 작은 모델

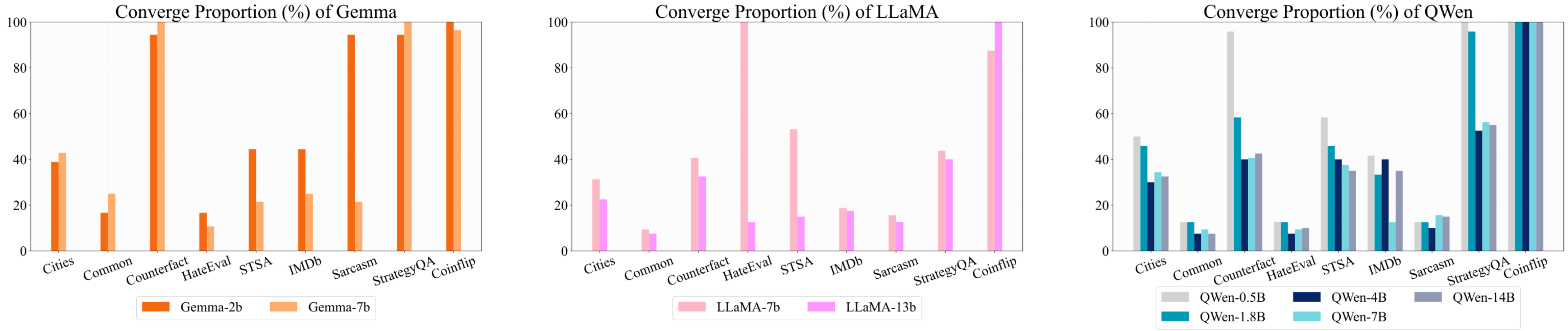
실험 1: 모델 고정, 데이터 셋 비교



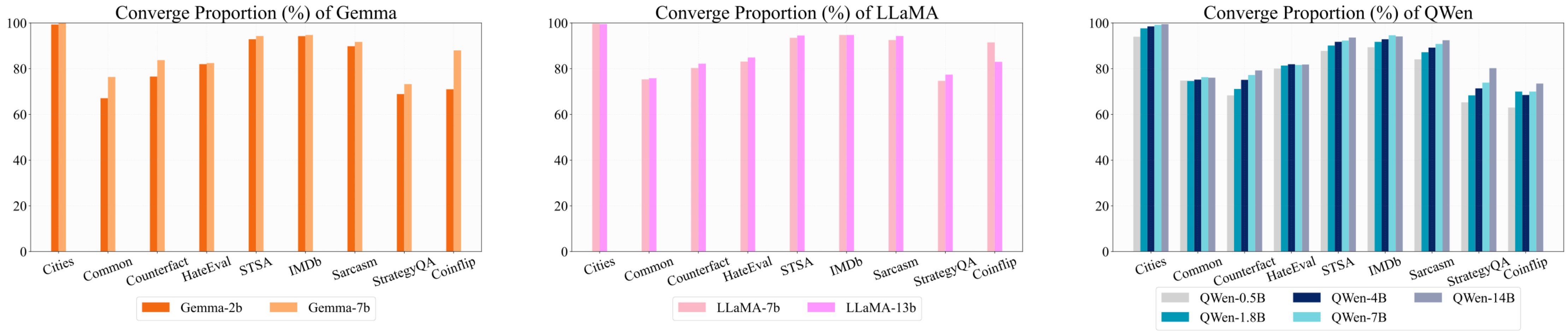
실험 2: 모델 family 비교



실험 3: 모델 사이즈 비교



(a) The converging point of each dataset on Gemma, LLaMA, and QWen represented by the percent depth proportion.



Neuron-Level Knowledge Attribution in Large Language Models

Zeping Yu **Sophia Ananiadou**
Department of Computer Science, The University of Manchester
{zeping.yu@postgrad. sophia.ananiadou@}manchester.ac.uk

- Response에 영향을 가장 많이 미치는 뉴런들을 pinpoint 할 수 있는 방법
  - Response에 직접적인 영향을 주는 Value 뉴런을 찾은 후
  - Value 뉴런을 activate 시키는 Query 뉴런을 찾음

- 실험:
  - 뉴런 내의 어떤 부분이 영향을 줄까?
  - Knowledge는 어디에 저장되어 있을까?

Methodology

1. 뉴런들에 인한 distribution change 분석

2. Distribution에서 영향력이 큰 Value 뉴런 찾기

3. 위의 Value 뉴런을 활성화시키는 Query 뉴련 찾기

Methodology

Background

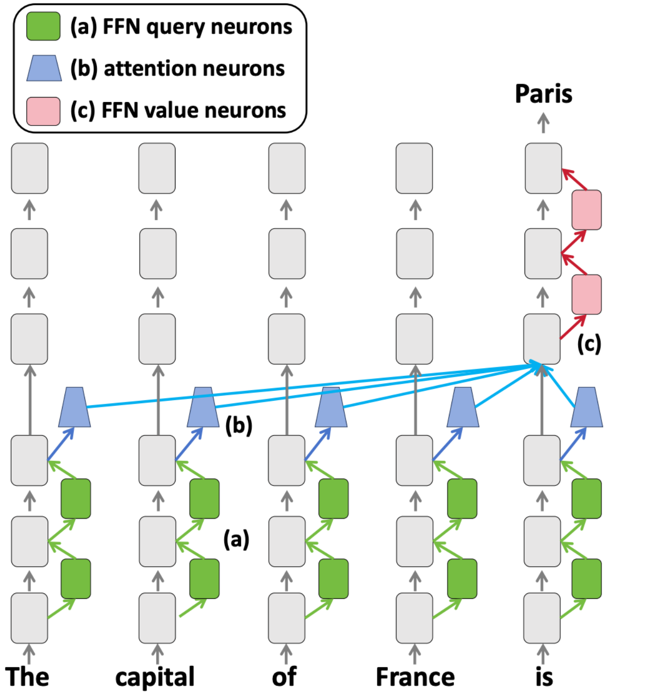


Figure 1: (a) Query neurons in shallow FFN layers. (b) Attention query/value neurons in attention heads. (c) Value neurons in deep FFN layers.

For l^{th} layer, j^{th} head, p^{th} position

$$A_i^l = \sum_{j=1}^{H} ATTN_j^l(h_1^{l-1}, h_2^{l-1}, \cdots, h_T^{l-1})$$

$$= \sum_{j=1}^{H}\sum_{p=1}^{T} \alpha_{i,j,p}^l \cdot W_{j,l}^o (W_{j,l}^v h_p^{l-1})$$

$$\alpha_{i,j,p}^l = Softmax(W_{j,l}^q h_i^{l-1} \cdot W_{j,l}^k h_p^{l-1})$$

$$F_i^l = \sum_{k=1}^{N} m_{i,k}^l \cdot fc2_k^l$$

$$m_{i,k}^l = \sigma(fc1_k^l \cdot \underbrace{(h_i^{l-1} + A_i^l)}_{x})$$

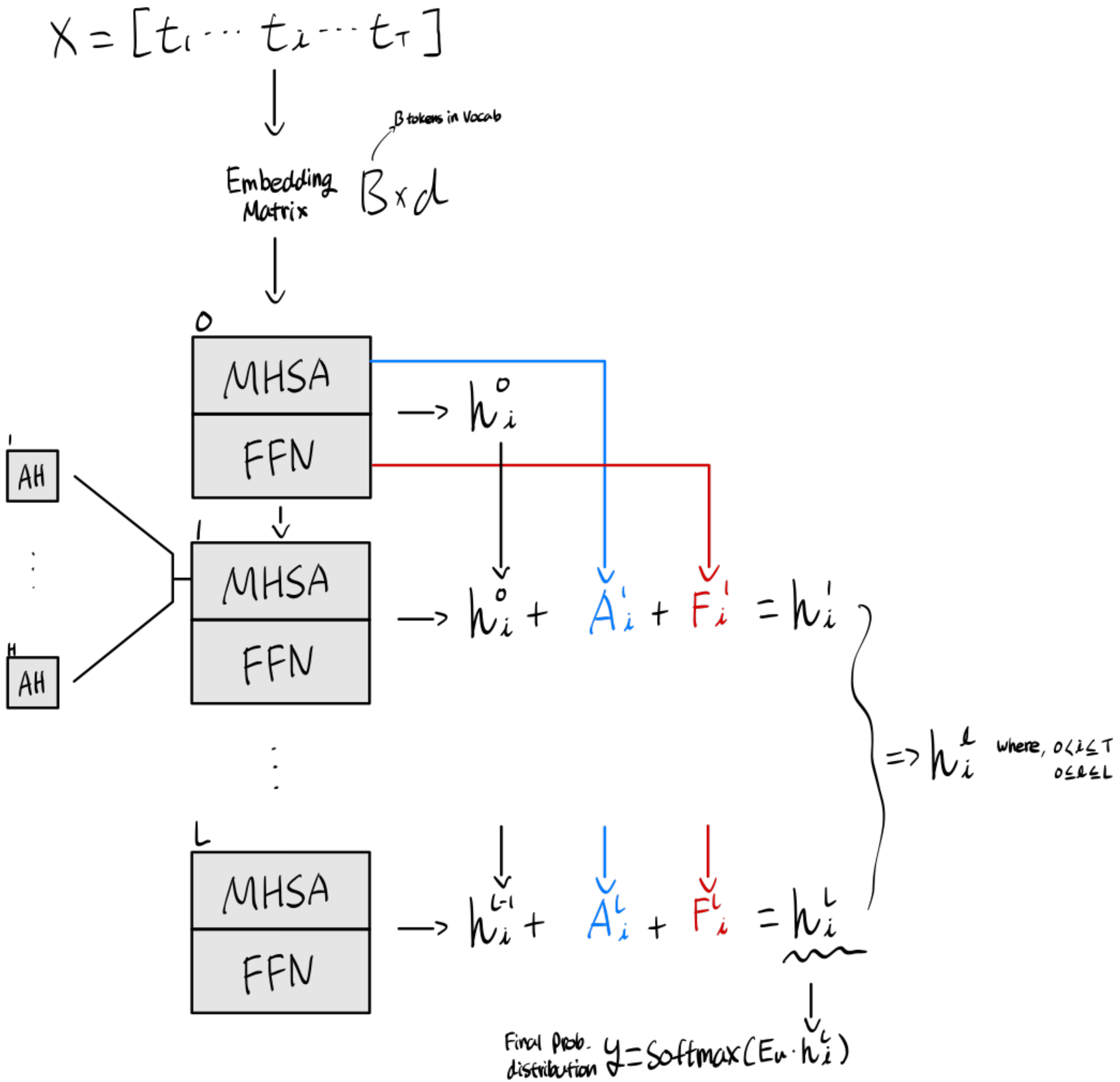
$A_i^1 = A(h_i^0)$

let $x = h_i^0 + A_i^1 = h_i^0 + A(h_i^0)$

$F_i^1 = F(x)$

let $y = x + F(x) = h_i^0 + A_i^1 + F_i^1 = h_i^1$

=> residual output



Definition

$$h_T^l = x\ (residual\ vector) + v\ (crucial\ neuron\ vector)$$

Before Softmax: $bs_w^x = e_w \cdot x$

e_w: w^{th} row of the unembedded matrix E_u

$$bs(x) = [bs_1^x, bs_2^x, ..., bs_w^x, .., bs_B^x]$$

$$bs(x+v) = bs(x) + bs(v)$$

Probability Change: $p(w|x+v) - p(w|x)$

$$p(w|x) = \frac{exp(bs_w^x)}{\Sigma_{j=1}^B exp(bs_j^x)} = \frac{e^{e_w \cdot x}}{\Sigma_{j=1}^B e^{e_j \cdot x}}$$

Methodology

Probability Change 계산 예시

Probability Change: $p(w|x+v) - p(w|x)$

Assume:

- $bs(x)$=[1,2,3,4]
- Corresponding probability distribution $p(x) = [0.03, 0.09, 0.24, 0.64]$

1. **Define** v:

$bs(v) = [1, 1, 1, 3]$

2. **Calculate** $bs(x+v)$
3. $bs(x+v) = bs(x) + bs(v) = [1,2,3,4] + [1,1,1,3] = [2,3,4,7]$
4. **Compute the new probability distribution** $p(x+v)$:

Apply the softmax function to $bs(x+v)$:

$$exp(2) = 7.389 \qquad exp(3) = 20.086 \qquad exp(4) = 54.598 \qquad exp(7) = 1096.633$$

Sum of exponentials:

$7.389 + 20.086 + 54.598 + 1096.633 = 1178.706$

New probabilities:

$$p(1 \mid x+v) = \frac{exp(2)}{1178.706} = \frac{7.389}{1178.706} \approx 0.0063 \approx 0.01 \qquad p(2 \mid x+v) = \frac{exp(3)}{1178.706} = \frac{20.086}{1178.706} \approx 0.0170 \approx 0.02$$

$$p(3 \mid x+v) = \frac{exp(4)}{1178.706} = \frac{54.598}{1178.706} \approx 0.0463 \approx 0.05 \qquad p(4 \mid x+v) = \frac{exp(7)}{1178.706} = \frac{1096.633}{1178.706} \approx 0.9304 \approx 0.93$$

Therefore, $p(x+v) = [0.01, 0.02, 0.05, 0.93]$

Distribution Change Analysis

- 임의의 뉴런 벡터 (v) 선택해서 probability change 변화 확인

Probability Change: $p(w|x+v) - p(w|x)$

$bs(x)$=[1,2,3,4]

Corresponding probability distribution $p(x) = [0.03, 0.09, 0.24, 0.64]$

$bs(v)$	$bs(x+v)$	$p(x+v)$
$[1,1,1,3]$	$[2,3,4,7]$	$[0.01,0.02,0.05,0.93]$
$[3,1,1,1]$	$[4,3,4,5]$	$[0.20,0.07,0.20,0.53]$
$[6,4,4,4]$	$[7,6,7,8]$	$[0.20,0.07,0.20,0.53]$
$[6,2,2,2]$	$[7,4,5,6]$	$[0.64,0.03,0.09,0.23]$
$-[6,2,2,2]$	$[-5,0,1,2]$	$[0.00,0.09,0.24,0.67]$

Table 1: Probability distribution of $p(x+v)$.

[1,1,1,3] & [3,1,1,1]에서 볼 수 있듯이, V의 역할은 distribution에 가중치를 주는 것으로 생각해볼 수 있음

Experiment - Attribution Method 비교

- GPT2-large, Llama-7B 사용
- T/F를 확인 할 수 있는 TriviaQA 데이터셋 사용

- 데이터셋에 있는 문제들을 필터링 시킴
  - 모델의 FFN에 뉴런의 중요도를 확인할 수 있는 7가지의 다른 방법들을 적용시킴
  - 기준은:
    - 최종 top 10 probability 안에 있고
    - 같은 카테고리의 다른 답변보다 높은 probability

  - GPT2-large: 데이터 1,350개
  - Llama-7B: 데이터 3,141개

Experiments

Attribution Method

```
len(methods) = 7
eval_metrics = ['Mean Reciprocal Rank', 'Probability', 'Log Probability']

for s in sentences:
  for m in methods:
    neuron = m.top_10(s)
    for e in eval_metrics:
      e.eval(neuron)
```

a) (proposed method) log probability increase: $log(p(w|mv^l + A^l + h^{l-1})) - log(p(w|A^l + h^{l-1}))$
b) log probability: $log(p(w|mv^l))$, which attributes the same neurons with $p(w|mv^l)$
c) probability increase: $p(w|mv^l + A^l + h^{l-1}) - p(w|A^l + h^{l-1})$
d) norm: $|v^l|$
e) coefficient score: $|m|$
f) ranking in vocabulary space: $1/rank(w)$
g) $|m| \times |v^l|$
h) $|m| \times 1/rank(w)$

비교 결과

a~h 방법을 사용하여 top 10 뉴런 뽑고, 강제로 0으로 만들어서
기존 결과 (o)에 비해 얼만큼의 성능 저하가 일어났는지 확인

	GPT2-large			Llama-7B		
	MRR	prob	logp	MRR	prob	logp
o)	0.361	7.1	-3.15	0.551	21.5	-2.24
a)	**0.201**	**3.4**	**-4.06**	**0.312**	**9.2**	**-3.91**
b)	0.214	3.6	-3.91	0.339	10.8	-3.35
c)	0.219	3.7	-3.92	0.345	10.0	-3.57
d)	0.363	7.1	-3.14	0.549	21.3	-2.25
e)	0.439	8.6	-3.10	0.529	22.9	-2.35
f)	0.306	5.8	-3.40	0.493	18.1	-2.49
g)	0.394	8.1	-3.06	0.523	22.6	-2.39
h)	0.232	4.0	-3.80	0.389	13.0	-3.06

Table 2: Results of attribution methods on two models.

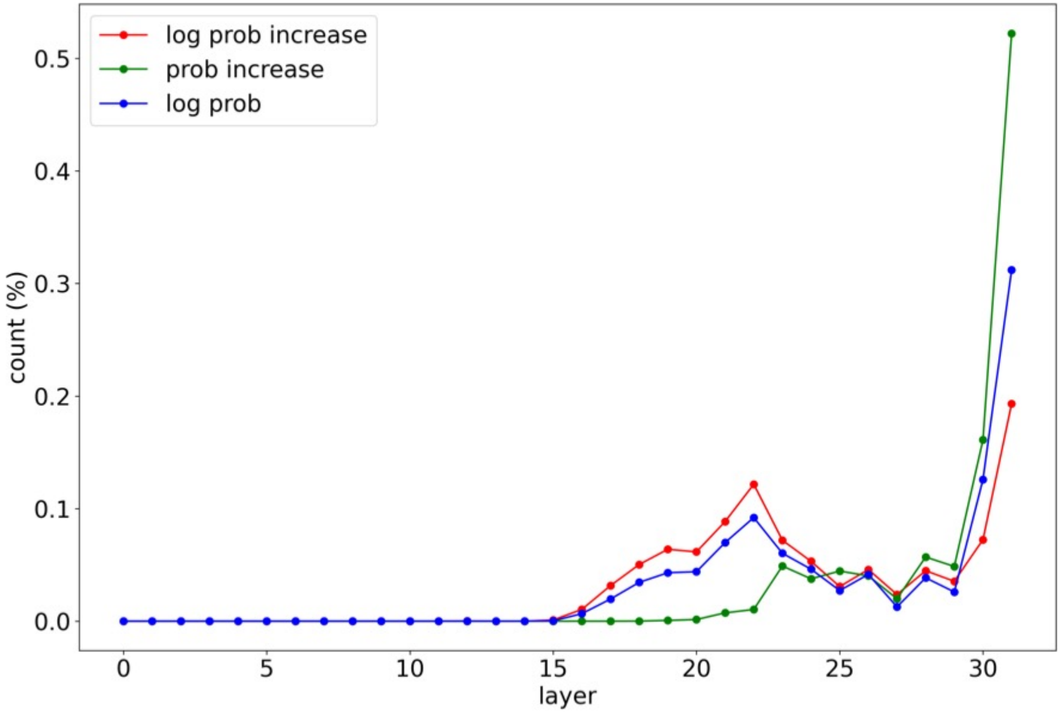


Figure 2: Neuron distribution on all layers in Llama-7B.

Experiment - Knowledge Exploration

- 앞 실험에서 성능이 가장 좋았던 a) log probability increase 방법 사용
- FFN, Attention 레이어 전부에 관하여 실험 진행

- Language, color, number, capital, country, month 지식을 나눠 실험

Knowledge Exploration 결과

- Attention, FFN 섞여있음
  ⇒ 모든 레이어에 지식 저장됨

- Attention에서는 비슷한 semantic의 지식은 비슷한 Attention Head에 들어있음

	top10 important layers
lang	$a_{26}, a_{30}, a_{32}, a_{22}, a_{31}, a_{28}, a_{23}, a_{27}, a_{19}, a_{23}$
col	$a_{32}, f_{32}, a_{33}, f_{29}, f_{31}, a_{31}, a_{26}, f_{33}, f_{28}, a_{22}$
num	$f_{29}, f_{23}, f_{27}, f_{30}, f_{31}, f_{26}, f_{32}, a_{23}, a_{22}, f_{28}$
capi	$a_{26}, a_{28}, a_{30}, a_{25}, a_{22}, f_{26}, f_{28}, a_{19}, f_{27}, f_{30}$
cnty	$a_{26}, a_{30}, a_{28}, a_{22}, f_{29}, a_{31}, f_{26}, a_{32}, a_{25}, a_{19}$
mon	$a_{27}, a_{26}, f_{26}, a_{25}, f_{30}, a_{28}, a_{24}, a_{22}, a_{30}, f_{27}$
lang	$a_{23}, a_{21}, f_{21}, a_{19}, a_{18}, a_{31}, a_{25}, a_{16}, f_{20}, f_{19}$
col	$f_{29}, a_{20}, a_{22}, a_{20}, a_{19}, a_{28}, a_{16}, a_{29}, a_{18}, f_{28}$
num	$f_{31}, f_{26}, f_{29}, f_{27}, a_{26}, f_{23}, f_{24}, a_{28}, f_{17}, f_{30}$
capi	$a_{23}, f_{21}, f_{22}, a_{18}, a_{25}, a_{21}, f_{19}, f_{20}, a_{16}, f_{24}$
cnty	$a_{23}, a_{21}, a_{25}, f_{22}, a_{18}, a_{19}, a_{16}, f_{21}, f_{31}, a_{31}$
mon	$a_{21}, a_{19}, f_{19}, a_{16}, f_{31}, a_{23}, a_{28}, f_{30}, f_{17}, f_{18}$

Table 4: Top10 important layers in GPT2 (first block) and Llama (second block).

type	top10 heads
lang	$a^{6}_{30}, a^{17}_{26}, a^{7}_{26}, a^{11}_{32}, a^{0}_{19}, a^{9}_{31}, a^{13}_{25}, a^{17}_{22}, a^{13}_{28}, a^{2}_{29}$
col	$a^{5}_{33}, a^{1}_{34}, a^{7}_{26}, a^{19}_{24}, a^{18}_{23}, a^{13}_{32}, a^{1}_{30}, a^{8}_{22}, a^{14}_{32}, a^{2}_{28}$
num	$a^{18}_{22}, a^{3}_{17}, a^{8}_{23}, a^{2}_{19}, a^{3}_{30}, a^{19}_{25}, a^{3}_{20}, a^{0}_{30}, a^{2}_{12}, a^{3}_{25}$
capi	$a^{7}_{26}, a^{6}_{30}, a^{17}_{26}, a^{17}_{22}, a^{13}_{25}, a^{13}_{28}, a^{0}_{19}, a^{10}_{19}, a^{2}_{29}, a^{11}_{32}$
cnty	$a^{7}_{26}, a^{6}_{30}, a^{17}_{22}, a^{13}_{28}, a^{17}_{26}, a^{11}_{32}, a^{0}_{19}, a^{13}_{25}, a^{9}_{31}, a^{10}_{19}$
mon	$a^{2}_{27}, a^{7}_{26}, a^{11}_{25}, a^{10}_{19}, a^{2}_{30}, a^{4}_{28}, a^{18}_{23}, a^{17}_{17}, a^{1}_{33}, a^{3}_{17}$
lang	$a^{12}_{23}, a^{31}_{19}, a^{25}_{31}, a^{25}_{25}, a^{5}_{16}, a^{1}_{18}, a^{9}_{21}, a^{22}_{29}, a^{17}_{21}, a^{23}_{18}$
col	$a^{22}_{29}, a^{19}_{28}, a^{27}_{20}, a^{15}_{16}, a^{27}_{17}, a^{21}_{28}, a^{14}_{25}, a^{28}_{18}, a^{1}_{24}, a^{3}_{14}$
num	$a^{19}_{28}, a^{24}_{26}, a^{10}_{23}, a^{13}_{30}, a^{29}_{21}, a^{24}_{13}, a^{24}_{18}, a^{22}_{29}, a^{23}_{17}, a^{1}_{19}$
capi	$a^{12}_{23}, a^{22}_{29}, a^{25}_{25}, a^{25}_{31}, a^{31}_{19}, a^{1}_{18}, a^{15}_{16}, a^{5}_{16}, a^{9}_{21}, a^{23}_{18}$
cnty	$a^{12}_{23}, a^{31}_{19}, a^{25}_{25}, a^{9}_{21}, a^{25}_{31}, a^{15}_{16}, a^{1}_{18}, a^{5}_{16}, a^{22}_{29}, a^{19}_{28}$
mon	$a^{10}_{21}, a^{0}_{16}, a^{22}_{21}, a^{18}_{23}, a^{16}_{28}, a^{20}_{19}, a^{6}_{31}, a^{1}_{19}, a^{3}_{14}, a^{13}_{20}$

Table 10: Top10 important heads in GPT2 (first block) and Llama (second block).

결론

- Attention, FFN 둘다 지식을 포함하고 있음
- Attention에 한에서는 비슷한 지식은 비슷한 head에 들어있음
- 뉴런 값을 조금 바꿔주면 결과에 차이가 확실히 나타남

- 자세하게 뉴런 단위로 지식의 저장을 확인하거나, 답변에 영향을 직접적으로 주는 논문은 많지 않았음

- Knowledge editing이나 XAI 연구들 활용 방안 다양

Thank you