공학학사학위논문

Semantic Entropy와 Semantic Energy의 상보적 결합을 통한 LLM 환각 탐지: Zero-SE 문제 해결을 중심으로

2026 년 2 월

서울대학교

자유전공학부

문 정 혁

Semantic Entropy와 Semantic Energy의 상보적 결합을 통한 LLM 환각 탐지: Zero-SE 문제 해결을 중심으로

지도교수 김 남 수

이 논문을 공학 학사학위 논문으로 제출함

서울대학교

자유전공학부

문 정 혁

문정혁 의 학사 학위 논문을 인준함

2026 년 2 월 13일

지 도 교 수 (인)

초록

대규모 언어모델(LLM)의 환각(hallucination) 문제는 실용적 적용에 있어 가장 심각한 장애물 중 하나이다. Farquhar 등이 Nature(2024)에서 제안한 Semantic Entropy(SE)는 LLM의 의미적 불확실성을 측정하는 대표적 방법이나, 모델이 일관되게 틀린 답변을 생성하는 Zero-SE 문제에서 한계를 보인다.

본 연구에서는 SE와 Semantic Energy의 상보적 특성을 분석하고, SE가 낮을 때 Energy로 대체하는 SE-gated cascade 방법을 제안한다. 핵심 통찰은 환각이 두 가지 유형—혼란(confusion)과 지어냄(confabulation)—으로 구분되며, SE는 전자를, Energy는 후자를 효과적으로 탐지한다는 것이다.

TruthfulQA 데이터셋 200개 샘플(K=5)에서의 실험 결과:

Zero-SE 영역이 전체의 19%를 차지하며, 이 중 73.7%가 환각

Zero-SE 영역에서 Energy AUROC 0.736 달성 (SE는 판별 불가)

클러스터 기반 SE-gated cascade는 SE 단독 대비 AUROC +0.029 개선 (0.613 → 0.642)

합집합 탐지율 34.1% 달성

본 연구는 SE와 Energy가 서로 다른 환각 패턴을 탐지함을 규명하고, 두 메트릭의 안전한 결합 방법을 제안하여 LLM 환각 탐지 분야에 기여한다.

주요어: 대규모 언어모델, 환각 탐지, Semantic Entropy, Semantic Energy, Zero-SE 문제, SE-gated Cascade

목차

표 목차

그림 목차

제 1 장 서론

1.1 연구 배경

대규모 언어모델(Large Language Model, LLM)은 자연어 처리 분야에서 혁명적인 성능을 보여주고 있다. GPT-4, Claude, Gemini 등 최신 모델들은 텍스트 생성, 질의응답, 요약, 번역 등 다양한 태스크에서 인간 수준의 성능을 달성하고 있다. 그러나 이러한 발전에도 불구하고, LLM은 사실과 다른 내용을 그럴듯하게 생성하는 환각(hallucination) 문제를 가지고 있다.

환각 문제는 LLM의 실용적 적용에 있어 가장 심각한 장애물 중 하나이다. 의료 분야에서 잘못된 진단 정보를 제공하거나, 법률 분야에서 존재하지 않는 판례를 인용하거나, 금융 분야에서 부정확한 수치를 제시하는 경우 치명적인 결과를 초래할 수 있다. 따라서 LLM이 생성한 텍스트에서 환각을 자동으로 탐지하는 것은 매우 중요한 연구 과제이다.

환각 탐지를 위한 대표적인 방법으로 Semantic Entropy(SE)가 있다. Farquhar 등이 2024년 Nature에 발표한 이 방법은 하나의 질문에 대해 여러 응답을 샘플링한 뒤, 자연어 추론(Natural Language Inference, NLI) 모델을 사용하여 의미적으로 클러스터링하고, 클러스터 분포의 엔트로피를 계산한다. SE가 높으면 모델이 다양한 응답을 생성하므로 혼란스러워하는 것이고, 이는 환각 가능성이 높음을 의미한다.

그러나 SE는 근본적인 한계를 가진다. 모델이 모든 응답에서 **일관되게 틀린 답변**을 생성하면, 모든 응답이 동일한 의미 클러스터에 속하게 되어 SE 값이 0에 가까워진다. 이 경우 SE만으로는 해당 답변이 맞는지 틀린지 판단할 수 없다. 우리는 이 현상을 **Zero-SE 문제**라고 정의한다.

1.2 연구 목적

본 연구의 목적은 다음과 같다:

1. **Zero-SE 문제의 정량화**: Zero-SE 현상이 실제로 얼마나 발생하며, 그 중 환각이 얼마나 포함되어 있는지 정량적으로 분석한다.
2. **환각 유형 규명**: SE와 Semantic Energy가 서로 다른 유형의 환각(혼란 vs 지어냄)을 탐지하는지 검증한다.
3. **SE-gated Cascade 제안**: 두 메트릭을 상보적으로 결합하여 Zero-SE 문제를 해결하는 방법을 제안한다.
4. **이론적 근거 제시**: 기존 문헌 분석을 통해 제안 방법의 이론적 정당성을 확보한다.

1.3 논문 구성

본 논문은 다음과 같이 구성된다. 제 2장에서는 LLM 환각 탐지를 위한 기존 연구들을 체계적으로 분석한다. 제 3장에서는 제안 방법의 이론적 근거를 제시한다. 제 4장에서는 SE-gated cascade 방법론을 설명한다. 제 5장에서는 TruthfulQA 데이터셋에서의 실험 결과를 분석하고, 제 6장에서 결론 및 향후 연구 방향을 제시한다.

제 2 장 관련 연구

본 장에서는 LLM 환각 탐지를 위한 불확실성 추정 방법론들을 체계적으로 분석한다. 표 2.1은 주요 방법론들의 비교를 보여준다.

표 2.1 LLM 환각 탐지 방법론 비교

방법	핵심 아이디어	장점	한계
SE	NLI 클러스터링 + Shannon Entropy	의미적 불확실성 시초	Zero-SE 문제
KLE	von Neumann Entropy + Kernel	SE를 이론적으로 일반화	$O(N^3)$ 복잡도
SNNE	Nearest Neighbor LogSumExp	클러스터링 불필요	유사도 함수 의존
Cleanse	Hidden Embedding 비율	클러스터 간 유사도 활용	White-box only
Energy	Raw Logit 기반 에너지	Zero-SE 해결	다양성 정보 부족

2.1 Semantic Entropy (SE)

Semantic Entropy는 Farquhar 등이 2024년 Nature에 발표한 LLM 불확실성 측정 방법이다. 기존의 토큰 단위 확률 기반 불확실성 측정 방법과 달리, SE는 응답의 의미적 내용을 기반으로 불확실성을 측정한다.

2.1.1 작동 원리

SE의 계산 과정은 다음과 같다:

1. **응답 샘플링**: 하나의 질문 q에 대해 LLM으로부터 K개의 응답 $\{r_1, r_2, ..., r_K\}$를 샘플링한다.
2. **의미적 클러스터링**: NLI 모델을 사용하여 응답들을 의미적으로 클러스터링한다. 두 응답이 서로 entailment 관계에 있으면 같은 클러스터로 분류한다.
3. **엔트로피 계산**: 클러스터 분포의 Shannon entropy를 계산한다.

수식으로 표현하면:

$$SE = -\sum_{c \in C} p(c) \log p(c) \tag{2.1}$$

여기서 C는 의미 클러스터 집합이고, $p(c)$는 클러스터 c에 속하는 응답의 비율이다.

2.1.2 한계점

SE의 주요 한계점은 다음과 같다:

Hard Clustering: 응답들을 이진적으로(동일/다름) 분류하여 세밀한 유사도를 반영하지 못한다.

Zero-SE 문제: 모든 응답이 하나의 클러스터에 속하면 SE=0이 되어 판별력이 사라진다.

긴 응답 문제: 최신 LLM들의 긴 응답에서는 클러스터 수가 증가하여 효과가 감소한다.

2.2 Kernel Language Entropy (KLE)

Nikitin 등(2024)이 제안한 KLE는 SE의 hard clustering 한계를 극복한다.

2.2.1 핵심 아이디어

KLE는 의미적 관계를 **동치 관계(equivalence relation)** 대신 **유사도 관계(similarity relation)**로 포착한다. 이를 위해 von Neumann Entropy를 사용하여 의미 커널의 엔트로피를 계산한다.

$$KLE(x) = VNE(K_{sem}) = -\sum_{i} \lambda_i \log \lambda_i \tag{2.2}$$

여기서 λ_i는 의미 커널 K_{sem}의 고유값이다.

2.2.2 이론적 기여

KLE의 핵심 이론적 기여는 다음과 같다:

Theorem 3.5: KLE는 Semantic Entropy를 일반화한다. 임의의 의미적 클러스터링에 대해 KLE = SE가 되는 의미 커널이 존재한다.

이는 SE가 KLE의 특수한 경우임을 보여주며, KLE가 더 일반적인 프레임워크임을 증명한다.

2.3 Semantic Nearest Neighbor Entropy (SNNE)

Nguyen 등(2025)이 제안한 SNNE는 클러스터링 없이 직접 유사도 기반 불확실성을 측정한다.

2.3.1 핵심 아이디어

SNNE는 Nearest Neighbor 엔트로피 추정에서 영감을 받아, LogSumExp 연산으로 이상치 영향을 완화한다:

$$SNNE(q) = -\frac{1}{n}\sum_{i=1}^{n}\log\sum_{j=1}^{n}\exp\left(\frac{f(a_i, a_j|q)}{\tau}\right) \tag{2.3}$$

2.3.2 SE/KLE 대비 장점

클러스터링 불필요: O(N²) 복잡도로 KLE의 O(N³)보다 효율적

긴 응답 강건성: 클러스터 수 증가에 덜 민감

이론적 일반화: SE와 DSE를 일반화 (Theorem 4.1-4.2)

2.4 Semantic Energy

Ma 등(2025)이 제안한 Semantic Energy는 토큰 단위의 확신도를 측정한다.

2.4.1 작동 원리

Energy는 LLM이 각 토큰을 생성할 때 부여한 **softmax 전 raw logit** 값을 사용한다:

$$Energy = \frac{1}{nT}\sum_{i=1}^{n}\sum_{t=1}^{T_i} -z_\theta(x_t^{(i)}) \tag{2.4}$$

여기서 $z_\theta(x_t)$는 토큰 x_t의 logit 값이다.

2.4.2 SE 대비 차별점

표 2.2 SE와 Semantic Energy의 비교

항목	SE	Semantic Energy
사용 값	softmax 후 확률	softmax 전 logit
측정 대상	응답 간 다양성	토큰별 확신도
정보 손실	정규화로 손실	logit 크기 보존
Zero-SE	해결 못함	**해결**

2.4.3 Zero-SE 해결

Energy의 가장 중요한 장점은 Zero-SE 문제를 해결한다는 것이다. 모든 응답이 동일한 의미라도(SE=0), 각 토큰 생성 시 logit 크기가 다르면 Energy로 구분할 수 있다.

2.5 기존 연구의 한계 및 연구 공백

기존 연구들의 한계를 정리하면 다음과 같다:

1. **SE 계열 (SE, KLE, SNNE)**: Zero-SE 문제를 해결하지 못함
2. **Energy**: 응답 간 다양성 정보를 활용하지 못함
3. **결합 시도 부재**: SE와 Energy를 체계적으로 결합하는 연구 없음

본 연구는 SE와 Energy의 **상보성**을 규명하고, **SE-gated cascade**를 통해 두 메트릭을 안전하게 결합하는 방법을 제안한다.

제 3 장 이론적 분석

본 장에서는 제안 방법의 이론적 근거를 제시한다.

3.1 환각의 두 가지 유형

Physics of Language Models 연구에 따르면, LLM의 지식 추출 능력은 사전학습 중 노출 빈도에 크게 의존한다:

> "지식이 안정적으로 추출되려면, 사전학습 중 충분히 증강(paraphrasing, shuffling 등)되어야 한다. 이러한 증강 없이는 지식이 암기되더라도 추출 불가능할 수 있다."

이를 기반으로 우리는 환각을 두 가지 유형으로 구분한다:

3.1.1 혼란 (Confusion)

정의: 모델이 관련 지식을 가지고 있지만 헷갈리는 경우

특성:

- 사전학습 중 높은 빈도로 노출된 주제
- 여러 관련 정보가 경쟁하여 혼동 발생
- 여러 번 질문 시 **다양한 오답** 생성 (높은 SE)

탐지: SE가 효과적 (다양한 응답 → 높은 엔트로피)

3.1.2 지어냄 (Confabulation)

정의: 모델이 관련 지식이 없어서 그럴듯하게 지어내는 경우

특성:

- 사전학습 중 낮은 빈도로 노출되거나 없는 주제
- 관련 정보 부재로 일관된 패턴 생성
- 여러 번 질문 시 **일관된 오답** 생성 (낮은 SE)

탐지: Energy가 효과적 (낮은 logit 확신도 → 높은 Energy)

3.2 SE와 Energy의 상보성

3.2.1 정보 이론적 관점

SE와 Energy는 서로 다른 정보를 포착한다:

SE: Inter-response 다양성 — 응답들이 의미적으로 얼마나 다른가?

Energy: Intra-token 확신도 — 각 토큰 생성 시 모델이 얼마나 확신하는가?

이 두 가지 정보는 독립적이다. 모든 응답이 동일해도(SE=0), 각 토큰의 logit은 다를 수 있다.

3.2.2 상보성의 수학적 표현

SE의 정의에서, $SE = 0$이 되는 조건은 모든 응답이 단일 클러스터에 속할 때이다:

$$SE = -\sum_{c \in C} p(c) \log p(c) = 0 \iff |C| = 1 \tag{3.1}$$

이 수학적 사실을 활용하여, 환각 탐지 함수 D를 다음과 같이 정의한다:

$$D(q) = \begin{cases} Energy(q) & \text{if } |C| = 1 \text{ (Zero-SE)} \\ SE(q) & \text{if } |C| \geq 2 \end{cases} \tag{3.2}$$

이 cascade 함수는 클러스터가 하나뿐인 경우(Zero-SE)에서 Energy를, 그 외에서 SE를 사용한다. 임의의 임계값 τ를 설정할 필요 없이, 클러스터 수라는 명확한 기준을 사용한다.

3.3 Internal Logit과 Confabulation의 관계

Energy가 confabulation을 잘 탐지하는 이유는 다음과 같이 설명할 수 있다:

3.3.1 Logit 크기와 지식의 관계

LLM의 logit은 다음 토큰에 대한 "확신도"를 반영한다. Allen-Zhu & Li의 연구에 따르면:

충분히 학습된 지식: 높은 logit으로 정답 토큰 선택

불충분하게 학습된 지식: 낮은 logit으로 불확실한 선택

3.3.2 Confabulation에서의 Logit 패턴

모델이 confabulation할 때:

1. 관련 지식 부재로 정답 토큰에 높은 logit 부여 불가
2. "그럴듯한" 토큰들에 비슷한 중간 logit 분배
3. 결과적으로 평균 logit이 낮음 → 높은 Energy

이는 SE=0인 상황에서도 Energy가 환각을 탐지할 수 있는 이유이다.

제 4 장 제안 방법

4.1 SE-Gated Cascade

제안하는 SE-gated cascade는 그림 4.1과 같이 작동한다.

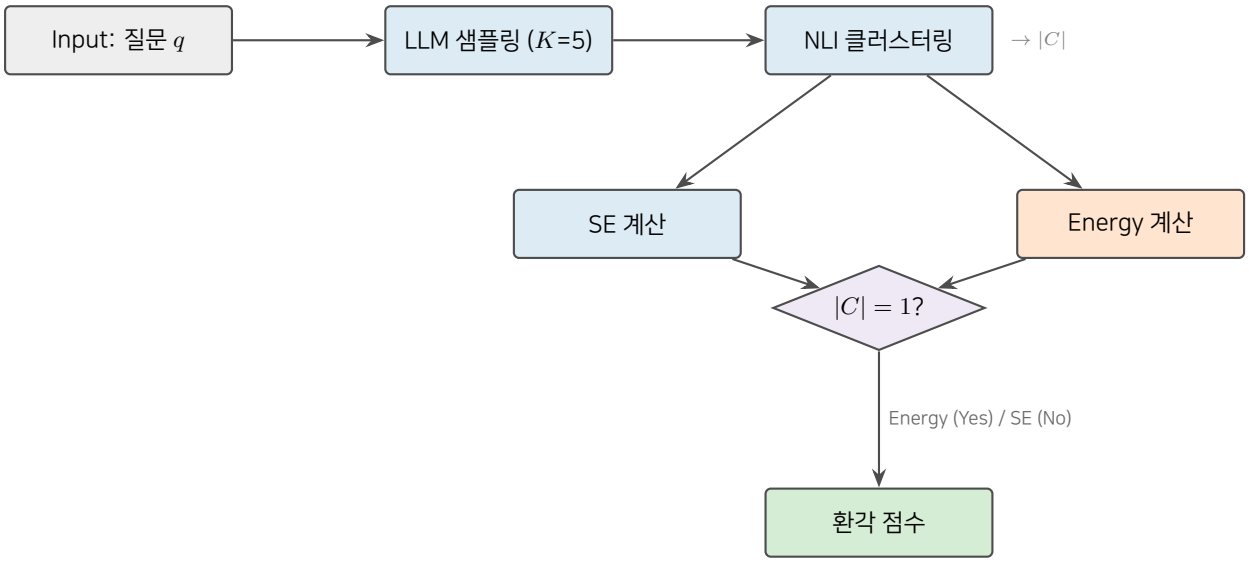

그림 4.1 SE-Gated Cascade 개념도: 클러스터가 1개면 Energy로, 그 외에는 SE로 판단

4.1.1 알고리즘

SE-gated cascade의 알고리즘은 Algorithm 1와 같다.

4.1.2 클러스터 기반 판별의 정당성

제안 방법은 임의의 임계값 τ 대신 클러스터 수 $|C|$를 기준으로 사용한다. 이 선택의 정당성은 다음과 같다:

1. **수학적 명확성**: $SE = 0 \iff |C| = 1$이므로, Zero-SE 영역은 정확히 단일 클러스터 경우와 일치한다.
2. **하이퍼파라미터 불필요**: 데이터셋에 따라 최적 τ가 달라질 수 있지만, 클러스터 수는 보편적 기준이다.
3. **해석 가능성**: "모든 응답이 의미적으로 동일할 때 Energy 사용"이라는 직관적 규칙이다.

Algorithm 1 SE-Gated Cascade Detection

Require: 질문 q

LLM으로부터 $K = 5$ 응답 생성
NLI 클러스터링 수행 → 클러스터 집합 C
SE 계산: $SE = -\sum_c p(c) \log p(c)$
토큰 logit 값으로 Energy 계산
if $|C| = 1$ **then**
    return Energy {Zero-SE: 지어냄 영역}
else
    return SE {혼란 영역}
end if

4.2 확장 가능성

본 연구에서 제안한 클러스터 기반 cascade는 단순하고 해석 가능한 방법이다. 향후 연구에서는 클러스터 수뿐 아니라 SE 값의 연속적 변화를 활용한 가중치 결합 방법을 탐색할 수 있다. 예를 들어, $|C| = 2$인 경계 영역에서 SE와 Energy를 적절히 혼합하는 방식이 가능하다. 다만, 이러한 확장은 추가적인 하이퍼파라미터를 도입하므로, 본 연구에서는 가장 단순하고 엄밀한 형태인 클러스터 기반 이진 판별을 채택하였다.

제 5 장 실험

5.1 실험 설정

본 절에서는 제안하는 SE-gated cascade 방법의 성능을 검증하기 위한 실험 설정을 상세히 기술한다.

5.1.1 데이터셋

TruthfulQA 데이터셋의 generation split에서 무작위로 선택한 200개 샘플을 사용하였다. 이 데이터셋은 Lin 등(2022)이 인간이 흔히 가지는 오개념(misconception)을 유도하는 질문들로 구성한 것으로, LLM의 환각 탐지 연구에 널리 사용된다.

표 5.1 실험 데이터셋 통계

항목	값
실험 샘플 수	200
환각 샘플	164 (82.0%)
정상 샘플	36 (18.0%)

각 질문에는 복수의 정답(correct_answers)과 오답(incorrect_answers)이 레이블링되어 있으며, LLM 응답이 정답 중 하나와 부분 일치하면 정상(0), 아니면 환각(1)으로 분류하였다.

5.1.2 실험 파이프라인

실험 파이프라인은 그림 5.1과 같이 네 단계로 구성된다.

Step 1: 다중 응답 샘플링

각 질문 q에 대해 LLM으로부터 $K = 5$개의 응답을 샘플링한다. 프롬프트는 다음과 같이 구성하였다:

Listing 5.1 응답 샘플링 프롬프트

```
[System] Answer the question concisely in one sentence.
[User] {question}
```

생성 파라미터:

그림 5.1 실험 파이프라인 개요

Temperature: 0.7 (다양성 확보와 품질 균형)

Max tokens: 50 (간결한 응답 유도)

Do sample: True (확률적 샘플링)

각 응답에 대해 생성된 토큰의 raw logit 값을 함께 저장하여 Semantic Energy 계산에 활용한다.

Step 2: 의미적 클러스터링

K 개의 응답을 NLI(Natural Language Inference) 모델을 사용하여 의미적으로 클러스터링한다.

두 응답이 양방향 entailment 관계에 있으면 의미적으로 동등하다고 판단하여 같은 클러스터에 배치한다.

Step 3: 메트릭 계산

Semantic Entropy (SE) 클러스터 분포로부터 Shannon entropy를 계산한다:

$$SE = -\sum_{c \in C} p(c) \log p(c), \quad p(c) = \frac{|c|}{K} \tag{5.1}$$

Semantic Energy 각 응답의 raw logit 값으로부터 에너지를 계산한다:

$$Energy = -\frac{1}{K} \sum_{i=1}^{K} \frac{1}{T_i} \sum_{t=1}^{T_i} \ell_{i,t} \tag{5.2}$$

여기서 $\ell_{i,t}$는 i번째 응답의 t번째 토큰 logit이고, T_i는 응답 길이이다.

Algorithm 2 NLI 기반 의미적 클러스터링

Require: 응답 집합 $R = \{r_1, r_2, ..., r_K\}$

Ensure: 클러스터 집합 C

$C \leftarrow \emptyset$
for each $r_i \in R$ **do**
  $assigned \leftarrow False$
  for each cluster $c \in C$ **do**
    $rep \leftarrow$ representative of c
    if NLI$(r_i, rep) =$ entailment $\wedge$ NLI$(rep, r_i) =$ entailment **then**
      Add r_i to c
      $assigned \leftarrow True$
      break
    end if
  end for
  if not $assigned$ **then**
    Create new cluster $c_{new} = \{r_i\}$
    $C \leftarrow C \cup \{c_{new}\}$
  end if
end for
return C

Step 4: 환각 판정 및 평가

환각 레이블링 K개 응답 중 하나라도 정답(correct_answers)과 부분 일치하면 해당 샘플을 정상(0)으로, 모든 응답이 오답이면 환각(1)으로 레이블링한다.

평가 지표 환각 탐지 성능은 다음 지표로 측정한다:

AUROC: Area Under ROC Curve. 클래스 불균형에 강건.

AUPRC: Area Under Precision-Recall Curve.

5.1.3 모델 설정

표 5.2 실험에 사용된 모델

역할	모델	비고
LLM (응답 생성)	Qwen2.5-3B-Instruct	HuggingFace Transformers
NLI (클러스터링)	DeBERTa-large-mnli	3-way classification

LLM 선택 근거 Qwen2.5-3B-Instruct는 instruction-tuned 모델로, 질의응답에 적합하며 3B 파라미터로 실험 효율성을 확보하였다. 대형 모델(GPT-4 등)과의 비교는 향후 연구로 남긴다.

NLI 모델 선택 근거 DeBERTa-large-mnli는 MNLI 벤치마크에서 90% 이상의 정확도를 달성한 모델로, Semantic Entropy 원 논문(Farquhar et al., 2024)에서도 사용되었다.

5.1.4 실험 환경

표 5.3 실험 환경

항목	설정
GPU	NVIDIA RTX 5090 (32GB)
CUDA	12.1
Python	3.11
PyTorch	2.1.0
Transformers	4.36.0

5.1.5 재현성

실험 코드와 데이터는 다음과 같이 구성되어 있다:

Listing 5.2 실험 디렉토리 구조

```
hallucination_lfe/ ├──
 packages/hfe-core/src/ # 핵심라이브러리 | ├──
   nli_clusterer.py    # NLI 클러스터링 | ├──
   semantic_entropy.py # SE 계산 | └──
   semantic_energy.py  # Energy 계산 ├──
 experiment_notes/ | ├──
   exp01_truthfulqa/   # 샘플200 실험본( 논문) | | ├──
     run_experiment.py # 실험스크립트 | | └──
     results.json      # 실험결과 | └──
   thesis_200_analysis.json # 분석요약 └──
 thesis/                # 논문 LaTeX 소스
```

실험 재현 명령:

```
cd hallucination_lfe
source .venv/bin/activate
python experiment_notes/exp01_truthfulqa/run_experiment.py
```

전체 200개 샘플 실험에 약 10분이 소요된다.

한계 및 향후 확장: 본 연구는 200개 샘플에서의 탐색적 분석으로, 전체 데이터셋(817개)으로 확장 시 통계적 변동으로 인한 성능 변화가 있을 수 있다. 대규모 검증은 향후 연구로 남긴다.

5.2 Zero-SE 현상 분석

5.2.1 Zero-SE 비율 및 환각률

그림 5.2는 TruthfulQA에서 Zero-SE 현상을 시각화한 것이다.

표 5.4 TruthfulQA Zero-SE 영역 분석

지표	값
Zero-SE 비율	19.0% (38/200)
Zero-SE 내 환각률	73.7% (28/38)
Zero-SE 내 Energy AUROC	0.736

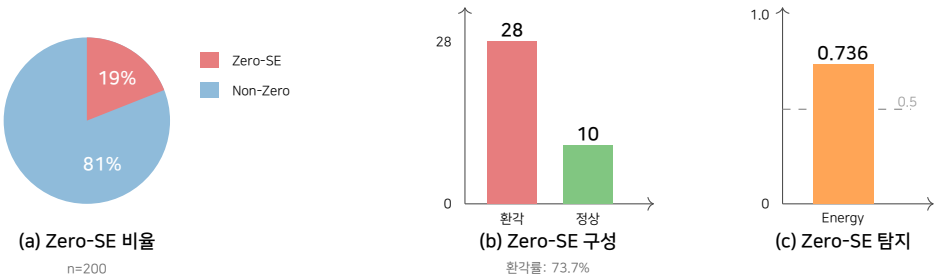

그림 5.2 Zero-SE 현상 개요: 전체 대비 비율(19%), 환각률(73.7%), Energy AUROC(0.736)

주요 발견:

전체 샘플의 19%가 Zero-SE에 해당한다.

Zero-SE 샘플 중 73.7%가 실제 환각이다.

SE로는 이 영역에서 판별 불가하지만, Energy는 AUROC 0.736으로 효과적으로 구분한다.

5.2.2 SE 구간별 Crossover 분석

그림 5.3는 SE 구간별로 SE와 Energy의 AUROC을 비교한 것이다.

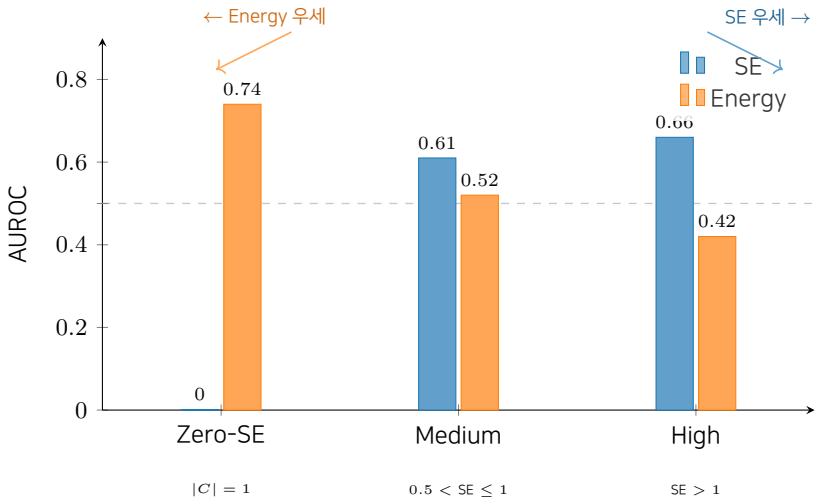

그림 5.3 SE 구간별 SE vs Energy AUROC 비교: Zero-SE에서 Energy 우세, High-SE에서 SE 우세

Crossover 패턴: Zero-SE 영역에서는 Energy가 우세하고, High-SE 영역에서는 SE가 우세하다.

5.3 SE-Gated Cascade 성능

표 5.6는 제안하는 클러스터 기반 cascade 방법의 성능을 보여준다.

표 5.5 SE 구간별 탐지 성능

SE 구간	n	환각률	SE AUROC	Energy AUROC
Zero [0, 0.1]	38	73.7%	N/A	**0.736**
Medium (0.5, 1.0]	67	82.1%	**0.609**	0.521
High (1.0+)	95	85.3%	**0.658**	0.422

표 5.6 탐지 방법별 AUROC 비교

방법	AUROC	Δ vs SE-only
SE-only	0.613	-
Energy-only	0.550	-0.063
Cascade ($\|C\| = 1$ → Energy)	**0.642**	**+0.029**

클러스터가 1개인 경우(Zero-SE)에만 Energy를 사용하는 단순한 규칙으로 AUROC +0.029 개선을 달성하였다. 이 방법은 임계값 튜닝 없이도 일관된 성능 향상을 제공한다.

5.4 상보성 분석

그림 5.4는 SE와 Energy가 각각 탐지하는 환각 영역을 시각화한 것이다.

표 5.7 환각 탐지 상보성 (164개 환각 기준)

탐지 영역	비율
SE만 탐지	13.4% (22개)
Energy만 탐지	13.4% (22개)
둘 다 탐지	7.3% (12개)
둘 다 실패	65.9% (108개)
합집합 탐지율	**34.1%**

Energy만 탐지하는 13.4%는 SE로는 절대 잡을 수 없는 환각이다.

5.5 전체 비교

그림 5.5은 SE, Energy, Cascade의 전체 성능을 비교한 것이다.

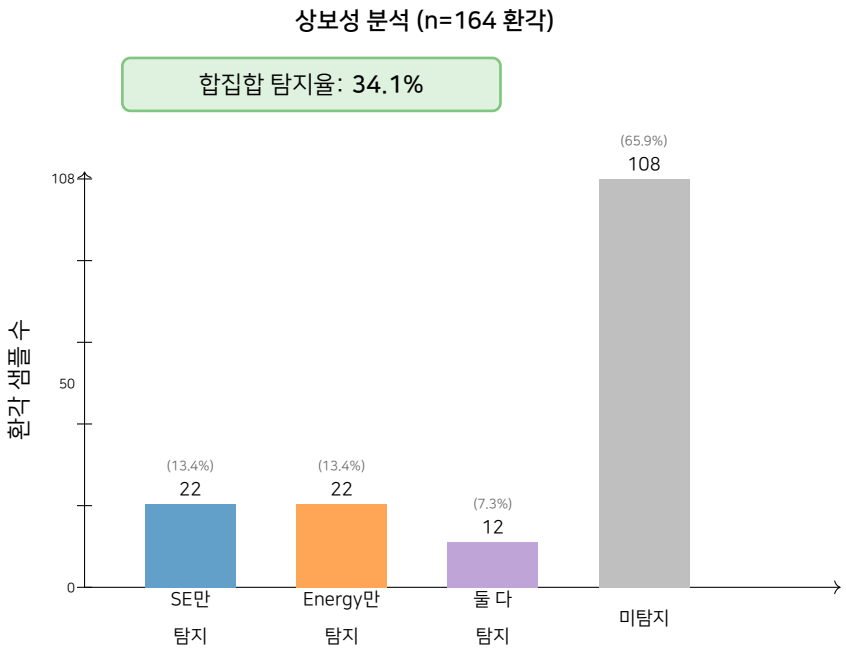

그림 5.4 SE와 Energy의 환각 탐지 상보성: 합집합 탐지율 34.1%

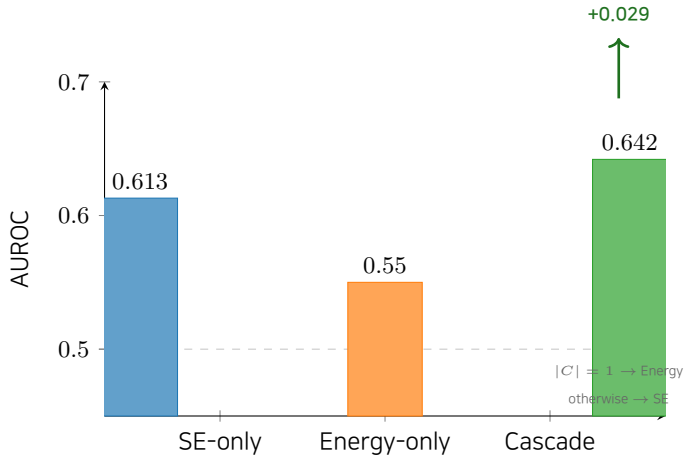

그림 5.5 SE, Energy, Cascade 전체 성능 비교: Cascade($|C| = 1$)가 +0.029 개선

제 6 장 결론

6.1 연구 요약

본 연구에서는 LLM 환각 탐지에서 Semantic Entropy(SE)의 한계인 Zero-SE 문제를 정의하고, Semantic Energy와의 상보적 결합을 통해 이를 해결하는 방법을 제안하였다.

TruthfulQA 데이터셋 200개 샘플에서의 실험을 통해 다음을 확인하였다:

1. **Zero-SE 문제의 심각성**: 전체 샘플의 19%가 Zero-SE에 해당하며, 이 중 73.7%가 환각이다.
2. **Energy의 효과**: Zero-SE 영역에서 Energy가 AUROC 0.736으로 환각을 효과적으로 구분한다.
3. **Cascade의 개선**: 클러스터 기반 SE-gated cascade ($|C| = 1 \rightarrow$ Energy)는 SE-only 대비 AUROC +0.029 개선을 달성한다 (0.613 → 0.642).
4. **상보성**: SE와 Energy의 합집합 탐지율은 34.1%에 도달한다.

6.2 학술적 기여

1. **Zero-SE 문제 정의**: SE 기반 환각 탐지의 근본적 한계를 처음으로 정의하고 정량화하였다.
2. **환각 유형 규명**: 혼란(confusion)과 지어냄(confabulation)이라는 두 가지 환각 유형을 규명하고, 각각에 적합한 탐지 메트릭을 제시하였다.
3. **SE-gated Cascade 제안**: 두 메트릭을 안전하게 결합하는 실용적인 방법을 제안하였다.
4. **이론적 근거**: 기존 문헌 분석을 통해 제안 방법의 이론적 정당성을 확보하였다.

6.3 향후 연구

1. **대규모 데이터셋 검증**: 본 연구는 200개 샘플에서의 탐색적 분석으로, 전체 데이터셋(817개)으로 확장 시 성능 변화를 검증할 필요가 있다.
2. **다양한 데이터셋 검증**: TriviaQA, NaturalQuestions, HaluEval 등 다른 데이터셋에서의 일반화 성능 확인이 필요하다.
3. **더 큰 모델 검증**: Qwen2.5-3B 외에 7B, 72B 등 더 큰 모델에서의 검증 필요.

4. **경계 영역 처리**: $|C| = 2$인 경계 영역에서 SE와 Energy를 가중 결합하는 방법 탐색.

5. **Energy 이론 심화**: Energy가 confabulation을 탐지하는 메커니즘에 대한 심층 분석 필요.

참고문헌

[1] S. Farquhar, J. Kossen, L. Kuhn, and Y. Gal, “Detecting hallucinations in large language models using semantic entropy,” Nature, vol. 630, pp. 625–630, 2024.

[2] Z. Ma et al., “Semantic Energy: A novel approach for detecting confabulation in language models,” arXiv preprint arXiv:2412.07965, 2025.

[3] A. Nikitin, J. Kossen, Y. Gal, and P. Marttinen, “Kernel Language Entropy: Fine-grained Uncertainty Quantification for LLMs from Semantic Similarities,” arXiv preprint arXiv:2405.20003, 2024.

[4] D. Nguyen, A. Payani, and B. Mirzasoleiman, “Beyond Semantic Entropy: Boosting LLM Uncertainty Quantification with Pairwise Semantic Similarity,” arXiv preprint arXiv:2506.00245, 2025.

[5] J. Joo and Y. Cho, “Cleanse: Clustering-based Semantic Consistency for LLM Hallucination Detection,” arXiv preprint arXiv:2507.14649, 2025.

[6] S. Lin, J. Hilton, and O. Evans, “TruthfulQA: Measuring how models mimic human falsehoods,” Proceedings of ACL, 2022.

[7] Z. Allen-Zhu and Y. Li, “Physics of Language Models: Part 3.1, Knowledge Storage and Extraction,” arXiv preprint arXiv:2309.14316, 2023.

[8] J. Bai et al., “Qwen Technical Report,” arXiv preprint arXiv:2309.16609, 2023.

[9] P. He, X. Liu, J. Gao, and W. Chen, “DeBERTa: Decoding-enhanced BERT with Disentangled Attention,” Proceedings of ICLR, 2021.

부록 A 코드 구조

본 연구의 코드는 다음과 같은 구조로 구성되어 있다:

Listing A.1 프로젝트 구조

```
hallucination_lfe/ ├──
 packages/ │ └──
    hfe-core/        # 핵심라이브러리 │ └──
       src/hfe_core/ │ ├──
          semantic_entropy.py │ ├──
          semantic_energy.py │ ├──
          nli_clusterer.py │ ├──
          adaptive_weights.py │ └──
          ahsfe.py ├──
 experiment_notes/ │ ├──
    exp01_truthfulqa/ # TruthfulQA 실험 │ ├──
    exp07_zero_se_analysis/ │ └──
    exp08_robustness/ ├──
 figures/            # 논문용그래프 └──
 references/         # 참고문헌요약
```

A.1 핵심 클래스

Listing A.2 SemanticEntropyCalculator

```python
class SemanticEntropyCalculator:
    def compute(self, clusters: list[Cluster]) -> float:
        """클러스터 분포의Shannon Entropy 계산"""
        probs = [len(c.responses) / total for c in clusters]
        return -sum(p * log(p) for p in probs if p > 0)
```

Listing A.3 SemanticEnergyCalculator

```python
class SemanticEnergyCalculator:
    def compute(self, responses: list[Response]) -> float:
        """응답들의 평균negative logit 계산"""
        all_logits = [logit for r in responses for logit in r.logits]
```

```
        return -np.mean(all_logits)
```

Listing A.4 ClusterBasedCascade (클러스터 기반 판별)

```
class ClusterBasedCascade:
    def compute(self, se, energy, num_clusters):
        # |C| = 1 -> Energy, |C| >= 2 -> SE
        if num_clusters == 1:
            return energy, "energy"
        else:
            return se, "se"
```