JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 31 No. 2, pp. 65-73, February 2026
https://doi.org/10.9708/jksci.2026.31.02.065

ProbCert: Probabilistic Certification Framework for Black-box LLM Outputs

Jong Wook Kim*

*Professor, Dept. of Computer Science, Sangmyung University, Seoul, Korea

[Abstract]

Large language models (LLMs) are widely used in natural language processing applications such as classification, summarization, and question answering. However, commercial LLMs are typically provided as black-box APIs, making it difficult to interpret the causes of their outputs or to quantitatively assess their reliability. In particular, existing approaches fail to provide a probabilistic characterization of how often output changes occur in response to input variations, or how consistently such changes arise. To address this limitation, this paper proposes ProbCert, a framework for estimating and certifying output change probabilities under input perturbations in black-box LLM settings. ProbCert repeatedly generates semantically valid input variations, observes whether output changes occur, and estimates the corresponding change probability, while continuing queries until a user-specified confidence level and error tolerance are satisfied. The framework integrates multiple confidence interval estimation methods, including the Wilson score interval, Empirical Bernstein bound, and the Clopper–Pearson interval, enabling systematic comparison of estimation accuracy and query efficiency under a unified procedure. Experimental results on both classification and generation tasks demonstrate that all variants of ProbCert reliably satisfy the specified confidence and error requirements. In particular, the Wilson score–based variant achieves certification with the fewest LLM queries, highlighting its practical efficiency in commercial LLM environments.

▸ **Key words**: Output change probability, Explainable artificial intelligence, Large language model

[요 약]

대규모 언어 모델(LLM)은 분류, 요약, 질의응답 등 다양한 자연어 처리 응용에서 널리 활용되고 있으나, 상업적 LLM은 대부분 블랙박스 API 형태로 제공되어 출력 결과의 근거를 해석하거나 신뢰성을 정량적으로 평가하기 어렵다. 특히 입력의 어떤 요소가 출력 변화에 영향을 미치는지, 그리고 이러한 변화가 얼마나 일관되게 발생하는지를 확률적으로 제시하지 못한다는 한계가 있다. 본 논문에서는 이러한 문제를 해결하기 위해, 블랙박스 LLM 환경에서 입력 변형에 따른 출력 변화 확률을 추정하고 이를 유한 표본 기반으로 인증하는 프레임워크인 ProbCert를 제안한다. ProbCert는 의미적으로 타당한 입력 변형을 반복적으로 생성하고, 각 변형에 대한 출력 변화 여부를 관측하여 변화 확률을 추정하며, 사용자 지정 신뢰수준과 허용 오차를 만족할 때까지 질의를 수행한다. 또한 Wilson score interval, Empirical Bernstein bound, Clopper–Pearson interval 등 다양한 신뢰구간 계산 방식을 통합하여, 동일한 절차 하에서 추정 정확도와 질의 효율을 비교할 수 있도록 한다. 분류 및 생성 작업을 대상으로 한 실험 결과, 제안 기법은 모든 설정에서 사용자 지정 신뢰수준과 허용 오차를 안정적으로 충족하였으며, 특히 Wilson score 기반 변형은 가장 적은 LLM 질의로 인증을 달성하여 실제 상업적 환경에서 효율적인 활용 가능성을 보였다.

▸ **주제어**: 출력 변화 확률, 설명 가능 인공지능, 대규모 언어 모델

• First Author: Jong Wook Kim, Corresponding Author: Jong Wook Kim
*Jong Wook Kim (jkim@smu.ac.kr), Dept. of Computer Science, Sangmyung University
• Received: 2025. 12. 29, Revised: 2026. 01. 27, Accepted: 2026. 02. 19.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

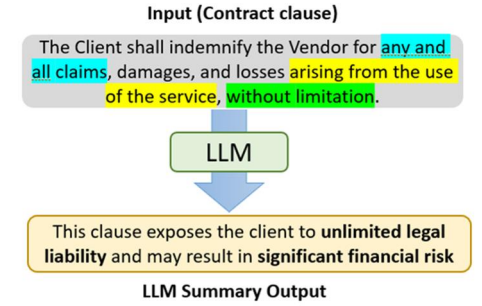
I. Introduction

대규모 언어 모델(LLM)은 분류, 요약, 추천, 질의응답 등 다양한 자연어 처리 응용 분야에서 널리 활용되고 있다 [1,2]. 그러나 상업적 LLM은 대부분 API 형태로 제공되며 (즉, 사용자는 입력을 전달하고 생성된 출력만을 확인할 수 있음), 이로 인해 특정 출력이 생성된 원인, 즉 입력의 어떤 요소가 결과에 실질적인 영향을 미쳤는지를 파악하기 어렵다 [3]. 이러한 불투명성은 의료, 금융, 법률, 보안과 같이 의사결정에 대한 책임성이 요구되는 영역에서 특히 문제가 된다. 출력의 근거를 검증할 수 없다면 오류를 사전에 식별하거나 책임 있는 의사결정을 내리기 어렵고, 이는 모델에 대한 신뢰 확보와 실제 환경에서의 안전한 활용을 제한한다.

예를 들어, 법률 상담 지원 시스템에서 상업적 API 기반 LLM이 계약서의 전문 조항을 일반인이 이해할 수 있는 문장으로 재작성하는 상황을 생각해볼 수 있다(그림 1). 계약서에는 “무제한 손해배상”과 같이 해석에 따라 당사자에게 중대한 불이익이 될 수 있는 표현이 포함되며, LLM은 이러한 내용을 요약하여 책임 위험이 크다는 경고 문장을 생성할 수 있다. 이때 중요한 것은 어떤 입력 표현이 이러한 강한 경고를 유도했는지, 그리고 의미적으로 유사한 표현으로 입력을 바꾸더라도 동일한 경고가 일관되게 유지되는지 여부이다. 따라서 이러한 문서 재작성 환경에서는 입력 표현의 변화가 생성 결과에 미치는 영향을 확률적으로 정량화할 수 있는 접근이 필요하다.

기존의 설명 가능 인공지능(Explainable Artificial Intelligence, XAI) 기법들[4,5,6]은 입력이 출력에 미치는 영향을 정성적으로 설명하는 데에는 유용하지만, 그 영향을 확률적으로 근거화하지는 못한다. 대부분의 방법은 특정 입력 표현이 중요하다는 점을 강조하거나, 입력을 일부 변경했을 때 출력이 달라질 수 있음을 사례 수준에서 보여주는 데 그친다. 그러나 이러한 설명은 입력 변형이 출력 변화를 얼마나 자주, 그리고 얼마나 일관되게 유발하는지를 정량적으로 제시하지 않으며, 관측된 변화가 우연적인 현상인지를 판단할 수 있는 근거도 제공하지 않는다. 그 결과, 그림 1과 같이 출력 변화의 가능성과 그에 대한 신뢰도를 함께 제시하는 설명을 제공하기에는 한계가 있다.

상업적 API 기반 LLM을 신뢰성 있게 활용하기 위해서는, 의미적으로 타당한 입력 변형에 대해 출력이 변화할 가능성을 확률로 정량화할 필요가 있다. 또한 이러한 확률 추정은 유한한 질의 결과에 기반하므로, 추정 오차를 함께 제시하여 사용자가 지정한 신뢰 수준에서 결과를 인증



ProbCert Result

Input Phrases	Output Change Probability (95% Confidence)
without limitation	0.92 ± 0.05
any and all claims	0.47 ± 0.05
arising from the use of the service	0.14 ± 0.05

Fig. 1. Motivating example. ProbCert estimates the output change probability with confidence intervals for selected contiguous input phrases, highlighting which parts of the original contract clause most strongly influence the LLM-generated explanation.

(certification)할 수 있어야 한다. 이를 위해 본 논문은 블랙박스 LLM 환경에서 입력 변형에 따른 출력 변화 확률을 추정하고, 이를 확률적 신뢰구간으로 인증하는 프레임워크인 ProbCert를 제안한다. ProbCert는 의미적으로 자연스러운 입력 변형을 반복적으로 생성하고, 각 변형에 대한 출력 변화 여부를 기록하여 변화 확률을 추정하며, 추정 오차가 사전에 설정한 허용 범위 이내로 인증될 때까지 질의를 반복한다. 본 논문의 주요 기여는 다음과 같다.

- 블랙박스 LLM 환경에서 입력 변형에 따른 출력 민감도를 확률적으로 추정하고, 이를 유한 표본 기반의 신뢰구간으로 인증할 수 있는 범용 프레임워크 ProbCert를 제안한다.
- ProbCert 내에 Clopper-Pearson (CP) interval, Empirical Bernstein (EB) bound, Wilson score (WS) interval을 하나의 통합된 인증 모듈로 설계하여, 서로 다른 통계적 인증 기법을 동일한 절차와 종료 조건 하에서 적용할 수 있도록 한다.
- 분류 및 생성 작업을 대상으로 실험을 수행하여, 인증 기법 선택에 따른 추정 정확도와 질의 효율 간의 구조적 차이를 분석한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 정리하고, 3장에서는 본 연구에서 다루는 문제를 정의한다. 4장에서는 제안 프레임워크인를 상세히 설명한다. 5장에서는 실제 데이터셋을 이용한 실험을 통해 제안 기법의 성능을 평가하고, 마지막으로 6장에서 결론을 제시한다.

II. Related Work

LLM 관련 설명 가능성 연구의 다수는 LLM을 도구로 활용하여 기존 기계학습 모델을 설명하는 데 집중해 왔으며, LLM 자체의 판단 과정과 출력 생성 메커니즘을 직접적으로 해석하려는 연구는 상대적으로 부족하다. 이러한 LLM 해석 연구는 주로 입력 변조 기반 방법과 그래디언트 기반 방법으로 구분된다.

입력 변조 기반 방법[4,5,7,8]은 입력의 일부를 마스킹하거나 제거한 뒤, 이러한 변형이 모델 출력에 어떤 변화를 유발하는지를 관찰함으로써 입력 요소의 중요도를 추정한다. 일부 연구에서는 계층적 마스크를 적용하거나 특정 입력 구간을 패딩한 뒤, 변형 전후의 출력 차이를 비교하여 입력 기여도를 분석하기도 한다. 이러한 방법들은 직관적인 해석을 제공할 수 있으나, 입력 변형의 방식이나 범위에 따라 결과가 달라질 수 있으며, 해석의 일관성을 보장하기는 어렵다.

그래디언트 기반 방법[6,9]은 입력 특징의 변화가 출력에 미치는 영향을 미분 정보를 통해 분석하여 중요도를 추정한다. LLM 환경에서는 특정 출력의 생성 확률을 입력 토큰 임베딩에 대해 미분함으로써, 문맥 내에서 어떤 토큰이 출력에 민감하게 작용하는지를 파악한다. 이러한 방법은 세밀한 민감도 분석이 가능하다는 장점이 있으나, 모델 내부 파라미터와 중간 표현에 대한 접근을 필요로 하므로 상업적 API 기반 LLM 환경에서는 적용이 어렵다.

이외에도 LLM 해석 가능성을 다른 관점에서 다룬 연구들이 존재한다. 생성 결과와 함께 근거를 제시하도록 모델을 평가하거나, 입력 토큰이 생성 과정에 미치는 영향을 모델 비의존적인 방식으로 분석하는 시도들이 대표적이다 [10,11]. 또한 외부 문서를 인용하는 방식으로 응답을 생성하도록 학습시켜, 출력의 출처를 명시적으로 제공하는 접근도 제안되었다 [12]. 이러한 방법들은 출력의 해석 가능성을 높이는 데 기여하지만, 입력 변형이 출력에 미치는 영향을 정량적으로 평가하는 데에는 한계가 있다.

기존 연구들을 종합하면, 현재의 LLM 해석 방법들은 입력 변형이 출력에 미치는 영향을 빈도와 일관성의 관점에서 확률적으로 평가하지 못한다. 이로 인해 실제 응용 환경에서 요구되는 수준의 신뢰도와 검증 가능성을 확보하기에는 한계가 있다.

III. Problem Definition

LLM은 자연어를 입력으로 받아 출력을 생성하므로, 함수 $f: \Sigma^* \rightarrow O$로 정의된다. 여기서 Σ^*는 입력 토큰 시퀀스의 집합이며, O는 모델의 출력 공간을 의미한다. 본 논문에서는 f가 텍스트 입력과 출력만을 제공하는 API 형태로만 접근 가능하다고 가정하며, 이는 일반적인 상업용 LLM의 사용 방식에 해당한다.

사용자 입력을 $P = (s_1, s_2, ..., s_m)$으로 정의하자. 이때 각 s_j는 입력 내에서 서로 겹치지 않는 연속된 입력 부분을 의미한다 (예: 토큰, 구문). 입력 P에 대한 모델의 원래 출력은 $Y = f(P)$로 정의된다.

각 입력 s_j에 대해, 문맥을 유지하면서 의미적으로 타당한 대안 입력 부분들로 구성된 분포를 Q_j로 정의하자. 분포 Q_j로부터 대안 입력 부분 $s' \sim Q_j$을 샘플링하여, 입력 부분 s_j를 s'으로 치환한 입력을 $P_{j \leftarrow s'}$으로 정의하자. 또한 두 출력이 서로 다른지를 판별하는 이진 함수를 다음과 같이 정의할 수 있다.

$$d(Y_1, Y_2) = \begin{cases} 1, \text{ if } Y_1 \neq Y_2 \\ 0, \text{ if } Y_1 = Y_2 \end{cases}$$

이때, 함수 d는 LLM이 수행하는 작업 따라 달라진다. 예를 들어, 분류 작업의 경우 d는 두 출력의 예측 레이블이 서로 다른지를 검사하며, 자연어 생성 작업의 경우에는 두 생성 결과 간에 의미적 차이가 존재하는지를 판단한다.

이때 입력 부분 s_j에 대한 출력 변화 확률은 다음과 같이 정의된다.

$$I_j = \Pr_{s' \sim Q_j}[d(f(P), f(P_{j \leftarrow s'}) = 1]$$

즉, 입력 부분 s_j를 의미적으로 타당한 대안 입력 부분으로 변조했을 때, 모델의 출력이 변화할 확률을 의미한다.

본 논문에서는 블랙박스 모델 f에 대해 유한한 질의만으로 출력 변화 확률 I_j를 추정하고, 해당 추정 오차에 대해 통계적 신뢰도를 보장하는 것을 목표로 한다. 이를 위해 사용자 지정 허용 오차 ϵ_j와 신뢰 수준 $1-\alpha$가 주어졌을 때, 다음 조건을 만족하는 추정치 $\hat{I}_j$를 구하고자 한다.

$$\Pr(|\hat{I}_j - I_j| \leq \epsilon_j) \geq 1 - \alpha$$

이는 입력 부분 s_j를 변조했을 때 출력이 변화할 확률을 추정하고, 그 추정 오차가 지정된 범위 이내임을 주어진 신뢰 수준에서 보장한다는 의미이다. 예를 들어, $1-\alpha = 0.99$, $\epsilon_j = 0.05$인 경우, 실제 출력 변화 확률 I_j가 $[\hat{I}_j - 0.05, \hat{I}_j + 0.05]$에 속한다고 99% 신뢰수준에서 보장할 수 있다.

IV. Proposed Method

본 장에서는 제안 프레임워크인 ProbCert를 설명한다. 그림 2는 ProbCert의 의사코드를 나타내며, 전체 절차는 세 단계로 구성된다. 첫 번째 단계에서는 입력의 특정 부분에 대해 의미적으로 타당한 대안 입력을 생성하고, 이를 기반으로 샘플링 분포를 구성한다(2줄). 두 번째 단계에서는 생성된 대안 입력을 원래 입력에 치환하여 모델 출력을 비교함으로써 출력 변화 여부를 반복적으로 관측한다(5-8줄). 마지막 단계에서는 관측 결과를 누적하여 출력 변화 확률을 추정하고, 설정한 신뢰 수준과 허용 오차를 만족하는지를 기준으로 종료 여부를 판단한다(9-15줄).

Algorithm 1: PROBCERT

1: **Inputs:** LLM f, input $P = (s_1, \ldots, s_m)$, index j, tolerance ε_j, confidence parameter α, CI method $\mathcal{M}$
2: Construct candidate set $\mathcal{C}_j$ and sampling distribution Q_j
3: $t \leftarrow 0$, $S_t \leftarrow 0$, $Y \leftarrow f(P)$
4: **while** true **do**
5: $\quad t \leftarrow t + 1$
6: $\quad$ Sample substitution $s'_{j,t} \sim Q_j$ and build $P_{j \leftarrow s'_{j,t}}$
7: $\quad$ Query the LLM: $Y_t \leftarrow f(P_{j \leftarrow s'_{j,t}})$
8: $\quad$ Compute: $\Delta_{j,t} \leftarrow d(Y, Y_t)$
9: $\quad$ Update: $S_t \leftarrow S_t + \Delta_{j,t}$, $\quad \hat{I}_{j,t} \leftarrow S_t / t$
10: $\quad$ Compute CI:
11: $\quad\quad [\mathrm{LB}_{j,t}, \mathrm{UB}_{j,t}] \leftarrow \mathrm{CI}(\mathcal{M}, t, S_t, \{\Delta_{j,1:t}\}, \alpha)$
12: $\quad \mathrm{Err}_{j,t} \leftarrow \max(\hat{I}_{j,t} - \mathrm{LB}_{j,t},\ \mathrm{UB}_{j,t} - \hat{I}_{j,t})$
13: $\quad$ **if** $\mathrm{Err}_{j,t} \leq \varepsilon_j$ **then**
14: $\quad\quad$ **break**
15: $\quad$ **end if**
16: **end while**
17: **return** $\hat{I}_{j,t}$ and $[\mathrm{LB}_{j,t}, \mathrm{UB}_{j,t}]$

Fig. 2. Pseudocode of ProbCert

4.1. Generating the Counterfactual Sampling

$P = (s_1, s_2, \ldots, s_m)$를 사용자 입력, s_j가 현재 평가 대상 입력 부분이라고 가정하자. s_j의 대안 표현을 생성하기 위해 ProbCert는 LLM을 사용한다. 구체적으로, s_j를 마스크 토큰으로 치환한 입력 P_{-j}를 다음과 같이 정의하자.

$$P_{-j} = (s_1, \ldots, s_{j-1}, <mask>, s_{j+1}, \ldots, s_m)$$

ProbCert는 마스크 입력 P_{-j}를 LLM에 전달하여, 주어진 문맥을 바탕으로 $<mask>$ 위치에 들어갈 수 있는 상위 k개의 대안 후보를 생성한다. 이를 $C_j = \{c^{(1)}, c^{(2)}, \ldots, c^{(k)}\}$로 나타내자. 여기서 $c^{(1)}$는 LLM이 가장 높은 순위로 제시한 후보이고, $c^{(k)}$는 가장 낮은 순위의 후보이다.

후보 집합 C_j를 실제 샘플링에 사용할 확률 분포로 변환하기 위해, 순위가 높을수록 선택될 가능성이 높도록 순위 기반 가중치를 부여한다. 즉, h번째 후보 $c^{(h)}$의 선택 확률 $p^{(h)}$는 다음과 같이 정의된다.

$$p^{(h)} = \frac{(h+\beta)^{-r}}{\sum_{x=1}^{k} (x+\beta)^{-r}}, \quad \beta \geq 0,\ \gamma > 0.$$

여기서 β와 γ는 각각 확률 분포의 완화 정도와 감소율을 조절하는 하이퍼 파라미터이다. 위의 식을 통해 선택 확률을 부여함으로써, 상위 후보가 우선적으로 선택되도록 가중치를 조절할 수 있다. 다음으로 분포 Q_j를 다음과 같이 정의한다.

$$Q_j = \{(c^{(1)}, p^{(1)}), (c^{(2)}, p^{(2)}), \ldots, (c^{(k)}, p^{(k)})\}$$

첫 번째 단계에서 생성한 분포 Q_j는 다음 단계에서 입력 부분 s_j에 대한 대안 입력을 샘플링하기 위해 사용된다.

4.2. Output Comparison under Input Substitution

그림 1에 보이듯이 두 번째 단계는 반복적으로 수행된다. t번째 반복에서 ProbCert는 분포 Q_j로부터 대안 입력 $s'_{j,t}$를 샘플링하고, 이를 입력 P의 j번째 위치에 치환하여 반사실적(counterfactual) 입력 $P_{j \leftarrow s'_{j,t}}$를 생성한다.

$$P_{j \leftarrow s'_{j,t}} = (s_1, \ldots, s_{j-1}, s'_{j,t}, s_{j+1}, \ldots, s_m)$$

그런 다음, $P_{j \leftarrow s'_{j,t}}$를 LLM에 질의하여 변형된 출력 $Y_t = f(P_{j \leftarrow s'_{j,t}})$를 구한 후, 이를 원래 출력 $Y = f(P)$와 비교하여 출력 변화 여부를 다음과 같이 계산한다.

$$\Delta_{j,t} = d(Y, Y_t)$$

여기서 함수 d는 LLM이 수행하는 작업에 따라 정의된다. 분류 작업의 경우에는 두 출력(즉, Y_t, Y)의 예측 레이블이 서로 다른지를 비교하여, 서로 다르면 $\Delta_{j,t} = 1$로 설정한다. 반면 생성 작업에서는 의미적 유사도(semantic similarity)를 사용하며, 두 출력의 유사도가 사전에 정한 임계값보다 작으면 출력이 변화한 것으로 판단하여 $\Delta_{j,t} = 1$로 설정한다.

4.3. Computing Confidence Intervals

마지막 단계에서는 반복을 수행하면서 사용자 지정 허용 오차 ϵ_j와 신뢰 수준 $1-\alpha$를 만족하는지를 확인하고, 해당 조건이 충족되면 알고리즘을 종료한다. 이때 ProbCert는 특정 확률 기법에 의존하지 않는 범용적 프레임워크이며, 다양한 확률적 방법을 적용할 수 있다. 본 연구에서는 대표적인 방법에 해당하는 CP interval, EB bound, WS interval을 사용하지만, 이 외에도 다양한 확률적 신뢰구간 추정 기법을 동일한 절차에 적용할 수 있다.

t번째 반복까지 관측된 누적 출력 변화 횟수 S_t는 다음과 같이 계산된다.

$$S_t = \sum_{i=1}^{t} \Delta_{j,i}$$

여기서 $\Delta_{j,i} = 1$은 i번째 반복에서 출력이 변화했음을 의미한다. 이때 출력 변화 확률의 추정치는 다음과 같다.

$$\hat{I}_{j,t} = \frac{S_t}{t}$$

각 반복에서 ProbCert는 선택한 기법에 따라 신뢰수준 $1-\alpha$의 신뢰구간 $[LB_{j,t}, UB_{j,t}]$를 계산한다. 여기서 $LB_{j,t}$와 $UB_{j,t}$는 각각 신뢰구간의 하한과 상한을 나타낸다. 신뢰구간이 주어지면, 현재 반복에서의 최대 추정 오차는 다음과 같이 정의된다.

$$Err_{j,t} = \max(\hat{I}_{j,t} - LB_{j,t}, UB_{j,t} - \hat{I}_{j,t})$$

ProbCert는 다음 조건을 만족하면 반복을 종료한다.

$$Err_{j,t} \le \epsilon_j$$

이 조건은 신뢰구간 $[LB_{j,t}, UB_{j,t}]$의 양 끝이 추정치 $\hat{I}_j$로부터 허용 오차 ϵ_j 이내에 들어왔음을 의미한다. 즉 신뢰구간의 폭이 충분히 좁아졌음을 뜻한다. 다만 본 알고리즘은 반복 중 신뢰구간을 갱신하고 종료 시점을 데이터에 따라 결정하는 순차적 종료 구조를 사용하므로, 고정 표본 기반 신뢰구간을 그대로 적용할 때 종료 시점에서의 명목 신뢰수준 $1-\alpha$가 엄밀히 유지된다고 단정하기는 어렵다.

따라서 본 논문에서는 위 종료 조건을 사용자 지정 허용 오차 ϵ_j를 달성하기 위한 실용적 기준으로 사용하며, 사용자 지정 신뢰수준 $1-\alpha$의 충족 여부는 5장의 포함률(coverage) 실험 결과로 평가한다.

다음 하위 절에서는 각 확률적 방법에 대해 신뢰구간을 계산하는 구체적인 방법을 설명한다.

4.3.1. Clopper-Pearson interval

CP interval은 이항 관측에 대해 유한 표본에서도 신뢰수준을 만족하는 구간을 제공한다. S_t가 관측된 변화 횟수일 때, CP의 신뢰구간의 하한과 상한은 각각 다음과 같이 계산된다 [13].

$$LB_{j,t} = B^{-1}(\frac{\alpha}{2}; S_t, t - S_s + 1),$$

$$UB_{j,t} = B^{-1}(1 - \frac{\alpha}{2}; S_t + 1, t - S_s)$$

여기서 $B^{-1}(p; a, b)$는 Beta(a,b) 분포의 p-분위수이다.

4.3.2 Empirical Bernstein bound

EB bound는 관측값의 분산을 이용해 구간 폭을 조절한다. $t \ge 2$에서의 표본분산과 반경은 각각 다음과 같이 정의된다 [14].

$$\hat{V}_{j,t} = \frac{1}{t-1}\sum_{i=1}^{t}(\Delta_{j,i} - \hat{I}_{j,t})^2$$

$$r_{j,t} = \sqrt{\frac{2\hat{V}_{j,t}\ln(4/\alpha)}{t}} + \frac{7\ln(4/\alpha)}{3(t-1)}$$

이를 이용하여 EB의 신뢰구간의 하한과 상한은 각각 다음과 같이 계산된다.

$$LB_{j,t} = \max(0, \hat{I}_{j,t} - r_{j,t}),$$
$$UB_{j,t} = \min(1, \hat{I}_{j,t} + r_{j,t}).$$

4.3.3 Wilson score interval

WS interval은 정규 근사를 사용하되 작은 표본에서도 비교적 안정적인 구간을 제공한다. WS 구간은 추정치 $\hat{I}_j$를 보정한 중심값 c와, 그 주변의 오차 폭을 나타내는 반경 w로 표현할 수 있으며, 각각 다음과 같이 정의된다 [15].

$$c = \frac{\hat{I}_{j,t} + z^2/2t}{1 + z^2/t},$$

$$w = \frac{z}{1 + z^2/t}\sqrt{\frac{\hat{I}_{j,t}(1-\hat{I}_{j,t})}{t} + \frac{z^2}{4t^2}}$$

여기서 $z = \Phi^{-1}(1-\alpha/2)$는 표준정규분포에서 신뢰수준 $1-\alpha$에 해당하는 임계값이다. 중심값 c와 반경 w를 이용하여 WS의 신뢰구간의 하한과 상한은 각각 다음과 같이 계산된다.

$$LB_{j,t} = c - w, \quad UB_{j,t} = c + w.$$

V. Experiments and Results

5.1. Experiment Setup

제안 기법의 성능 평가는 분류와 생성 두 가지 작업에서 수행하였다. 분류 작업에서는 SST-2 데이터셋[16]을 사용하여 LLM이 문장의 감성을 분류하도록 하였다. 생성 작업에서는 ASSET 데이터셋[17]을 사용하여, LLM이 주어진 입력 문장의 의미를 유지하면서 문장을 요약화하는 작업을 수행하도록 하였다. 생성 작업에서는 출력 변화 판별 함수를 문장 의미 유사도 기반으로 정의하였다. 두 출력 문장은 고정된 문장 임베딩 모델을 사용해 벡터화한 뒤 코사인 유사도(cosine similarity)를 계산하며, 유사도가 0.8 미만인 경우 의미적 차이가 발생한 것으로 판단하여 출력

변화로 판정한다. 해당 기준값은 문장 임베딩 기반 의미 유사도 측정에서 의미 보존 여부를 구분하는 경험적 기준으로 널리 사용되는 범위에 해당한다.

기존 LLM 설명 가능성 연구들은 출력 변화 확률의 정량화를 시도했으나, 유한 표본 조건에서 신뢰수준과 허용 오차를 사용자가 지정할 수 있도록 보장하지는 못했다. 이로 인해 기존 방법과의 직접 비교는 어렵다. 이에 본 실험에서는 ProbCert 프레임워크 내에서 신뢰구간 계산 방식만을 달리한 세 가지 변형을 비교한다. 구체적으로, ProbCert-CP는 CP interval을 사용하고, ProbCert-EB는 EB bound를, ProbCert-WS는 WS interval을 적용한다. 세 변형은 대안 입력 생성, 샘플링 절차, 출력 비교 방식, 종료 조건을 공유하며, 신뢰구간 또는 오차 경계를 계산하는 방식에서만 차이가 있다.

실험에서는 다음의 지표를 사용하여 각 기법 간 성능을 비교하였다.

- 포함률(Coverage)은 추정확률 $\hat{I}_j$가 실제 확률 I_j로부터 허용 오차 ϵ_j 이내에 포함되는 비율을 의미하며, 사용자 지정 신뢰수준 $1-\alpha$를 만족하는지를 평가한다.
- 평균 절대 오차(Mean Absolute Error, MAE)는 추정치와 실제 확률 간의 평균 절대 오차를 의미하며, 이는 허용 오차 ϵ_j를 만족하는지를 평가한다.
- 효율성은 출력 변화 확률 추정을 완료하는 데 필요한 총 LLM 질의 횟수로 측정한다. 이는 그림 2의 알고리즘에서 반복 횟수에 해당하며, 구체적으로는 9번째 줄에서 수행되는 LLM 질의의 총 횟수를 의미한다.

블랙박스 LLM 환경에서는 출력 변화 확률의 참값을 직접 관측할 수 없으므로, 본 연구에서는 Monte Carlo 샘플링을 통해 이를 근사하였다. 구체적으로, 각 평가 대상 입력 부분에 대해 10,000회의 독립적인 대안 입력 변형을 생성하고, 관측된 출력 변화 비율을 기준 확률로 사용하였다. 충분히 큰 샘플 수를 사용함으로써 근사 오차를 최소화하였으며, 해당 기준 확률은 모든 기법에 대해 동일하게 적용되어 상대적 성능 비교에는 영향을 미치지 않는다.

모든 실험은 상업용 LLM인 Gemini-2.0-flash [18]를 사용하여 수행하였다. 각 실험에서 평가 대상이 되는 입력 부분 s_j는 다음 기준에 따라 선택한다. 전체 입력에서 형용사를 포함하는 첫 번째 명사구를 선택하며, 해당 조건을 만족하는 명사구가 없는 경우에는 가장 앞에 위치한 명사구를 사용한다. 또한 4.1절에서 사용한 후보 집합의 크기는 $k=100$으로 고정하였고, 하이퍼파라미터 β와 γ는 각각 0.5와 1.0으로 설정하였다.

5.2. Experimental Results

표 1과 표 2는 각각 SST-2 데이터셋과 ASSET 데이터셋에서의 포함률과 평균 절대 오차를 보여준다. 본 실험에서는 신뢰수준 $1-\alpha$를 0.9와 0.95로 설정하였고, 허용 오차 ϵ_j는 0.1과 0.15로 설정하였다. 5.1절에서 설명한 바와 같이, 포함률은 사용자 지정 신뢰수준 $1-\alpha$의 충족 여부를 평가하며, 평균 절대 오차는 사용자 지정 허용 오차 ϵ_j의 충족 여부를 평가한다.

표에서 확인할 수 있듯이, 세 가지 제안 기법 모두 모든 실험에서 사용자 지정 신뢰수준과 허용 오차를 충족하였다. 기법별로 살펴보면, ProbCert-EB는 다른 방법들에 비해 포함률이 일관되게 높게 나타나며, 보다 보수적으로 동작하는 경향을 보인다. 이는 출력 변화 확률을 과대 추정하는 방향으로 신뢰구간을 구성함을 의미한다.

Table 1. Coverage and MAE for SST-2 Dataset

1-α	εj	Method	Coverage	MAE
0.9	0.1	ProbCert-EB	1.000	0.008
		ProbCert-WS	0.978	0.016
		ProbCert-CP	0.982	0.015
	0.15	ProbCert-EB	1.000	0.010
		ProbCert-WS	0.980	0.023
		ProbCert-CP	0.984	0.020
0.95	0.1	ProbCert-EB	1.000	0.007
		ProbCert-WS	0.978	0.015
		ProbCert-CP	0.986	0.014
	0.15	ProbCert-EB	1.000	0.008
		ProbCert-WS	0.996	0.020
		ProbCert-CP	0.998	0.018

Table 2. Coverage and MAE for ASSET Dataset

1-α	εj	Method	Coverage	MAE
0.9	0.1	ProbCert-EB	1.000	0.011
		ProbCert-WS	0.968	0.023
		ProbCert-CP	0.970	0.023
	0.15	ProbCert-EB	1.000	0.015
		ProbCert-WS	0.968	0.034
		ProbCert-CP	0.987	0.032
0.95	0.1	ProbCert-EB	1.000	0.010
		ProbCert-WS	0.989	0.022
		ProbCert-CP	0.993	0.021
	0.15	ProbCert-EB	1.000	0.013
		ProbCert-WS	0.987	0.029
		ProbCert-CP	0.991	0.028

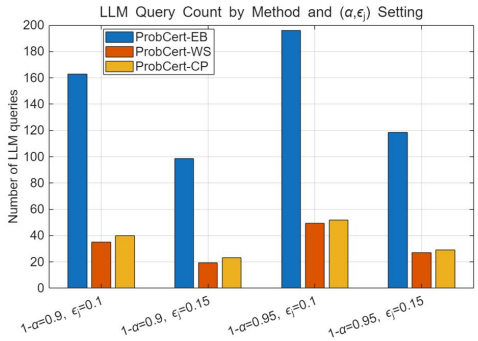
다만 포함률이 목표 신뢰수준을 크게 상회하는 것이 반드시 바람직한 것은 아니다. 본 연구의 목적은 포함률을 최대화하는 것이 아니라, 사용자 지정 신뢰수준과 허용 오차를 만족하는 데 있다. 포함률이 과도하게 높을 경우, 해당 신뢰수준을 달성하기 위해 필요한 샘플 수가 증가하게

되며, 이는 곧 LLM 질의 수 증가로 이어져 효율성 저하를 초래한다. 실제로 이러한 경향은 이후 효율성 실험에서 확인된다. 이러한 결과는 ProbCert-EB가 보수적으로 동작하는 반면, ProbCert-WS와 ProbCert-CP는 동일한 목표 조건을 보다 효율적으로 만족함을 보여준다.

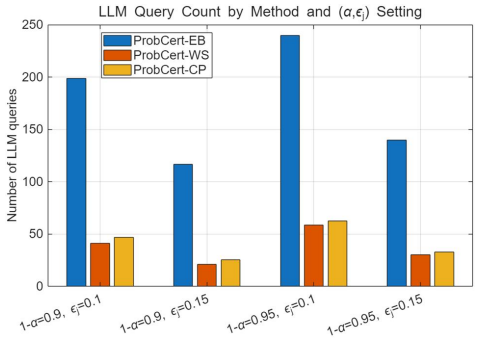
또한 ProbCert-WS와 ProbCert-CP를 비교하면, 두 데이터셋 전반에서 ProbCert-CP가 ProbCert-WS보다 포함률이 소폭 높고 오차도 더 작게 나타난다. 이 차이는 두 방법이 사용하는 신뢰구간 계산 방식이 다르기 때문에 발생한다. Clopper Pearson 기반의 ProbCert-CP는 이산 분포에서의 불확실성을 보다 보수적으로 반영하는 경향이 있어, 동일한 종료 조건이 적용될 때 더 많은 샘플을 요구하는 방향으로 작동하기 쉽다. 그 결과 추정에 사용되는 표본 수가 늘어나고, 추정치의 분산이 줄어 오차가 감소하며, 목표 신뢰수준 대비 포함률도 안정적으로 유지된다. 반대로 Wilson 기반의 ProbCert-WS는 근사적 성격이 강해 일부 구간(특히 표본 수가 크지 않거나 실제 확률이 0 또는 1 근처인 경우)에서 구간이 상대적으로 낙관적으로 좁아질 수 있고, 이때 조기 종료가 발생하면 포함률이 ProbCert-CP보다 낮아지거나 오차가 커지는 결과로 이어질 수 있다.

그림 3은 출력 변화 확률 추정을 완료하는 데 필요한 총 LLM 질의 횟수(평균)를 신뢰수준과 허용 오차 설정에 따라 비교한 결과이다. 여기서 질의 횟수는 반복적 출력 비교 단계에서 수행되는 질의만을 의미하며, 상위 k 후보 생성은 입력당 1회의 질의로 수행되므로 비용 산정에서 제외하였다. 전반적으로 ProbCert-WS와 ProbCert-CP는 ProbCert-EB보다 훨씬 적은 질의로 종료하며, 이러한 경향은 SST-2와 ASSET 모두에서 일관되게 관찰된다.

ProbCert-WS와 ProbCert-CP를 비교하면, 두 데이터셋 모두에서 ProbCert-WS가 가장 적은 질의로 종료하며, ProbCert-CP는 ProbCert-WS보다 소폭 더 많은 질의를 요구한다. 이는 ProbCert-CP가 이항 분포 기반 신뢰구간을 사용함으로써 동일한 종료 조건을 만족하기까지 더 많은 표본을 요구하는, 상대적으로 보수적인 특성을 가지기 때문으로 해석할 수 있다. 반면 ProbCert-WS는 근사적 신뢰구간을 사용하므로 더 이른 시점에 종료되는 경향이 있어 질의 수 측면에서 유리하다.



(a) SST-2 Dataset



(b) ASSET Dataset

Fig. 3. Number of LLM queries required by each method under different confidence levels and error tolerances

설정 변화에 따른 영향도 일관적이다. 허용 오차 ϵ_j를 0.1에서 0.15로 완화하면 모든 기법에서 질의 수가 크게 감소하며, 신뢰수준을 0.9에서 0.95로 높일 경우 질의 수는 증가한다. 이때 증가 폭은 ProbCert-EB가 약 20% 수준인 반면, ProbCert-WS와 ProbCert-CP는 약 25-43%로 더 크게 나타난다. 또한 ASSET은 SST-2에 비해 전반적으로 약 10-22% 더 많은 질의를 요구하는데, 이는 생성 과제가 분류 과제보다 출력 변동성이 크기 때문에 동일한 조건을 만족하기 위해 더 많은 샘플이 필요함을 의미한다.

종합하면, 본 장의 실험 결과는 세 가지 기법 모두 유한 표본 조건에서 사용자 지정 신뢰수준과 허용 오차를 실제로 충족함을 보여주며, 출력 변화 확률을 사전에 지정한 사양에 따라 인증된 형태로 추정할 수 있음을 보여준다. 또한 동일한 조건을 만족하는 과정에서 필요한 LLM 질의 수는 기법에 따라 큰 차이를 보이는데, ProbCert-WS는 모든 설정에서 가장 적은 질의로 종료하여 시간과 계산 비용, 나아가 상업용 LLM 사용 비용을 효과적으로 절감할 수 있음을 보여준다.

VI. Conclusions

본 논문에서는 블랙박스 환경의 상업적 LLM을 대상으로, 입력 변형에 따른 출력 변화 가능성을 확률적으로 정량화하고 이를 유한 표본 기반으로 인증하는 프레임워크 ProbCert를 제안하였다. 제안 기법은 기존 설명 가능성 연구들이 제공하지 못했던 출력 변화의 빈도와 신뢰도를 함께 제시함으로써, 입력의 어떤 요소가 출력에 실질적인 영향을 미치는지를 평가할 수 있도록 한다.

실험 결과, ProbCert의 세 가지 인증 변형은 분류 및 생성 작업 전반에서 사용자 지정 신뢰수준과 허용 오차를 안정적으로 충족하였다. 특히 신뢰구간 계산 방식에 따라 질의 효율에서 뚜렷한 차이가 나타났으며, Wilson score interval을 사용하는 ProbCert-WS는 동일한 인증 조건을 만족하면서 가장 적은 LLM 질의로 종료하여 실제 상업용 LLM 환경에서 비용과 시간 측면에서 가장 효율적인 선택임을 보였다. 이러한 결과는 ProbCert가 단순한 사례 기반 설명을 넘어, 출력 변화 가능성을 정량적이고 인증된 형태로 제공함으로써 책임성과 신뢰성이 요구되는 응용 환경에서 실질적인 설명 도구로 활용될 수 있음을 의미한다.

REFERENCES

[1] L. Wu, X. Yang, X. Shi, C. Ma, Optimizing prompt efficacy in large language models for fake news detection via evolutionary algorithm-based feature selection, Information Sciences, 2025. DOI: 10.1016/j.ins.2025.122539

[2] S. Feng, Z. Lang, J. He, H. Zhang, W. Chen, J. Cao, A group recommendation method based on automatically integrating members' preferences via taking advantages of LLM, Information Sciences, 2025. DOI: 10.1016/j.ins.2025.122067

[3] H. Zhao, H. Chen, F. Yang, N. Liu, H. Deng, H. Cai, S. Wang, D. Yin, M. Du, Explainability for large language models: A survey, ACM Transactions on Intelligent Systems and Technology, 2024. DOI: 10.1145/3639372

[4] R. K. Mothilal, D. Mahajan, C. Tan, A. Sharma, Towards unifying feature attribution and counterfactual explanations: Different means to the same end, Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, 2021. DOI: 10.1145/3461702.3462597

[5] Z. Wu, Y. Chen, B. Kao, Q. Liu, Perturbed masking: Parameter-free probing for analyzing and interpreting bert, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020. DOI: 10.18653/v1/2020.acl-main.383

[6] V. Miglani, A. Yang, A. Markosyan, D. Garcia-Olano, N. Kokhlikyan, Using captum to explain generative language models, Proceedings of the 3rd Workshop for Natural Language Processing Open Source Software, 2023. DOI: 10.18653/v1/2023.nlposs-1.19

[7] J. Enouen, H. Nakhost, S. Ebrahimi, S. O. Arik, Y. Liu, T. Pfister, Textgenshap: Scalable post-hoc explanations in text generation with long documents, arXiv:2312.01279, 2023. DOI: 10.48550/arXiv.2312.01279

[8] B. Cohen-Wang, H. Shah, K. Georgiev, A. Madry, Contextcite: Attributing model generation to context, arXiv:2409.00729, 2024. DOI: 10.48550/arXiv.2409.00729

[9] Y. Chang, B. Cao, Y. Wang, J. Chen, L. Lin, JoPA:Explaining Large Language Model's Generation via Joint Prompt Attribution, arXiv:2405.20404, 2024. DOI: 10.48550/arXiv.2405.20404

[10] T. Gao, H. Yen, J. Yu, D. Chen, Enabling large language models to generate text with citations, arXiv:2305.14627, 2023. DOI: 10.48550/arXiv.2305.14627

[11] Z. Zhao, B. Shan, Reagent: A model-agnostic feature attribution method for generative language models, arXiv:2402.00794, 2024. DOI: 10.48550/arXiv.2402.00794

[12] R. Nakano, J. Hilton, S. Balaji, J. Wu, L. Ouyang, C. Kim, C. Hesse, S. Jain, V. Kosaraju, W. Saunders, X. Jiang, K. Cobbe, T. Eloundou, G. Krueger, K. Button, M. Knight, B. Chess, J. Schulman, Webgpt: Browser-assisted question- answering with human feedback, arXiv:2112.09332, 2022. DOI: 10.48550/arXiv.2112.09332

[13] Sangdon Park, Shuo Li, Insup Lee, Osbert Bastani, PAC Confidence Predictions for Deep Neural Network Classifiers, arXiv:2011.00716, 2020. DOI: 10.48550/arXiv.2011.00716

[14] A. Maurer, M. Pontil, Empirical bernstein bounds and sample variance penalization, arXiv:0907.3740, 2009. DOI: 10.48550/arXiv.0907.3740

[15] L. D. Brown, T. T. Cai, A. Dasgupta, Interval estimation for a binomial proportion, Statistical Science, 2021. DOI: 10.1214/ss/1009213286

[16] R. Socher, A. Perelygin, J.Wu, J. Chuang, C. D. Manning, A. Ng, C. Potts, Recursive deep models for semantic compositionality over a sentiment treebank, Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2013.

[17] F. Alva-Manchego, L. Martin, A. Bordes, C. Scarton, B. Sagot, L. Specia, ASSET: A dataset for tuning and evaluation of sentence simplification models with multiple rewriting transformations, Proceedings of the Annual Meeting of the Association for Computational Linguistics, 2020. DOI: 10.18653/v1/2020.acl-main.424

[18] Gemini-2.0-flash, https://docs.cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/2-0-flash?hl=ko, 2025.

Authors

Jong Wook Kim received the Ph.D. degree in Computer Science Department, Arizona State University, in 2009. He was a Software Engineer with the Query Optimization Group, Teradata, from 2010 to 2013.

Dr. Kim is currently a Professor with the Department of Computer Science at Sangmyung University. His primary research interests include data privacy and query optimization.