

학사졸업논문

모호성 분류와 명확화 질문 생성을 통한  
LLM 쿼리 이해 능력 강화

Enhancing Query Understanding in  
LLM  
via Ambiguity Classification and  
Clarification Question Generation

유지혜, 김다연

한양대학교 정보시스템학과

2025년 12월

# 차 례

국문 요지.....	1
제1장 서 론.....	2
제2장 선행연구.....	3
제3장 방법론.....	4
제2절 라우팅.....	5
제3절 명확화 질문 생성.....	5
제4장 실험 세팅.....	6
제1절 데이터셋.....	6
제2절 모델 선택.....	6
제3절 구현 세부사항.....	7
제5장 실험 결과.....	10
제1절 모호성 분류 모델 평가.....	10
제2절 명확화 질문 생성 모델 평가.....	11
제3절 전체 시스템 평가.....	13
제6장 결 론.....	15
제1절 연구 결과 및 의의.....	15
제2절 한계점.....	15
제3절 향후 확장 가능한 연구.....	16
참고 문헌.....	17
부 록.....	20
A. 모호성 분류 9개 범주.....	20
B. 시스템 프롬프트.....	22

C. 전체 시스템 평가 결과 상세 .....	25
D. 시스템 작동 예시 .....	27

## 국문 요지

거대 언어 모델(LLM)과 사용자 간의 상호작용에서 쿼리의 모호성은 모델의 해석 오류를 유발하고 신뢰성을 저해하는 주요 요인이다. 그러나 기존 LLM은 모호한 입력을 탐지하거나 명시적으로 해소하는 데 한계를 보인다. 본 연구는 이를 해결하기 위해 모호성 유형에 기반하여 명확화 질문을 생성하는 다중 에이전트 시스템을 제안한다. 제안된 시스템은 2단계 파이프라인으로 구성된다. 첫째, 모델이 쿼리의 모호성 유형을 정밀하게 분류한다. 둘째, 식별된 유형 정보에 최적화된 명확화 질문을 생성한다. 실험 결과, 분류 모델은 66.04%의 정확도를 달성하였다. 생성 모델은 직접 선호도 최적화(DPO)를 통해 인간 선호도에 맞게 정렬되었으며, 이를 통해 질문의 품질을 높였다. 이를 결합한 전체 시스템은 99.32%의 높은 구동 안정성을 입증하였다. 본 연구는 경량화된 모델(SLM)만으로 능동적으로 모호성을 해소하는 프로세스를 구현했다는 점에서 중요한 학술적 의의를 갖는다. 본 연구의 코드와 학습 데이터는 <https://github.com/jihyeyu33/LLM-Interactive-Clarification>에서 확인할 수 있다.

## 제1장 서론

거대 언어 모델(LLM)의 발전으로 업무부터 일상생활까지 인간과 AI의 협업이 보편화되었다. 그러나 인간의 의사소통은 본질적으로 불완전하다. 대화 중 맥락을 생략하거나, 중의적인 표현을 사용하거나, 정보를 누락하는 경우가 빈번하다. 인간 간의 대화에서는 되묻기와 같은 상호작용으로 이러한 모호함을 해소한다.

반면, 현재의 LLM은 모호성을 인식하고 적극적으로 명확화를 요청하는 능력이 부족하다. 모호한 질문에 대해 임의로 추론하여 답변하거나, 사용자가 의도하지 않은 해석을 제공한다. 이러한 답변은 LLM의 신뢰성을 저해할 수 있다. 프롬프트 엔지니어링을 통해 입력 과정에서 이를 일부 해소할 수 있지만, 이는 사용자에게 추가적인 학습과 인지적 부담을 요구한다. 결과적으로 비전문가의 접근성을 낮추고 기술의 대중화를 방해하는 요인이 된다. 따라서 LLM이 사용자 입력의 모호성을 스스로 감지하고 숨겨진 의도를 파악하는 시스템이 필수적이다.

본 연구는 모호성 분류와 명확화 질문 생성을 결합한 다중 에이전트 시스템을 제안한다. 시스템은 내재된 모호성 유형을 8개 유형으로 분류하고, 식별된 유형에 따라 맞춤형 명확화 질문을 생성하는 2단계 파이프라인을 구성된다. 특히, 직접 선호도 최적화(DPO) 기법을 도입하여[12] 생성된 질문이 인간의 선호도에 부합하도록 정렬(Alignment)하였다. 아울러, 소형 언어 모델(SLM)과 LoRA(Low-Rank Adaptation) 파인튜닝을 적용하여[4] 자원 효율성을 극대화한 경량화된 방법론을 제시한다.

결과적으로 본 연구는 모델이 8가지 모호성 유형을 스스로 식별하고 질문을 생성한다는 점에서 기존 연구와 차별화된다. 이러한 방식은 사용자가 정교한 프롬프트를 작성해야 하는 부담을 없앤다. 동시에 모델의 자의적 해석으로 인한 오류를 방지하여 답변의 정확도를 높인다. 또한 본 연구는 거대 모델이 아닌 소형 언어 모델(SLM)만으로 이 과정을 구현하였다. 실험을 통해 제한된 컴퓨팅 자원에서도 시스템이 안정적으로 작동함을 입증하였다. 따라서 본 연구는 저비용 고효율의 모호성 해소 시스템을 구축하기 위한 실질적인 대안을 제시한다.

## 제2장 선행연구

**LLM의 모호성** 사용자의 모호한 질의를 효과적으로 처리하기 위한 다양한 벤치마크 데이터셋이 제안되어 왔다. Min 등(2020)의 AmbigQA[10]는 모호한 질문을 명료화된 여러 질문과 답변 쌍으로 매핑하는 구조를 제시했으며, Lee 등(2023)의 CAmbigNQ[7]는 이를 확장하여 명확화 질문과 세분화된 질문을 함께 제공하였다. Li 등(2025)의 CondAmbigQA[9]는 질문의 문맥적 조건에 따라 정답과 근거를 분리하여 모호성을 다루었다. 그러나 이러한 연구들은 모호성의 유형을 체계적으로 분류하지 못했다는 한계가 있다. 이를 보완하기 위해 Zhang 등(2024)은 CLAMBER 벤치마크[20]를 제안하여, 모호성을 3개의 차원과 8개의 세부범주로 구체화한 분류체계를 확립하였다. 본 연구는 이 CLAMBER의 분류체계를 핵심 기반으로 삼는다.

최근 연구들은 데이터셋 구축을 넘어, LLM의 모호성 처리 능력을 다각도로 분석하고 있다. Shore 등(2025)[14]은 상호 참조(Coreference) 해소 작업에서 LLM이 모호성 해소와 탐지를 동시에 효과적으로 수행하지 못하는 'CORRECT-DETECT 상충 관계'를 발견했다. Chen 등(2025)[2]은 LLM이 답변 생성 전 쿼리에 대한 지식 경계를 판단하는 Query-Level Uncertainty 개념을 제안하였다. 이들은 별도의 훈련 없이 내부 레이어의 자체 평가를 활용하는 Internal Confidence 방법을 통해 불확실성 탐지가 가능함을 보였다. 또한 Ruis 등(2023)[13]이 화용론적 함의 해결 능력을 평가한 결과, 인스트럭션 튜닝(Instruction tuning)을 거친 모델만이 우수한 성능을 보였다. 이는 모호성 처리 능력이 모델 크기보다 파인튜닝 전략에 더 크게 의존함을 시사한다.

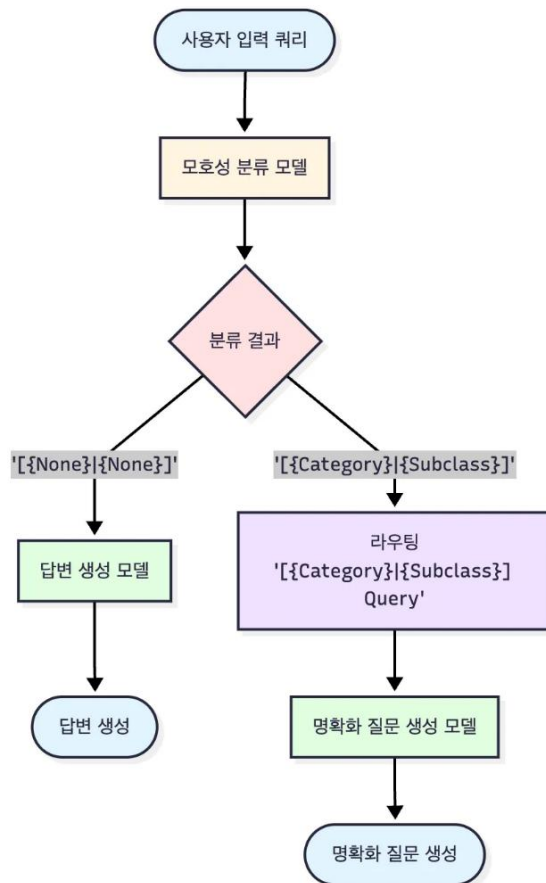
**명확화 요청** LLM이 모호성을 해소하기 위해 명확화 질문을 생성하는 다양한 방법론이 연구되었다. 우선 프롬프팅 기반의 접근이 있다. Kuhn 등(2023)[6]은 Few-shot만으로 질문 분류, 생성, 정보 통합을 수행하는 CLAM 프레임워크를 제안했다. Deng 등(2023)[3]은 Proactive Chain-of-Thought(ProCoT) 프롬프팅을 통해 LLM이 Zero/Few-shot 환경에서도 모호성을 식별하고 명확화 질문을 생성하도록 유도했다.

상호작용 비용을 최적화하거나 모델을 인간의 의도에 맞게 정렬(Alignment)하는 기법들도 제안되었다. Zhang과 Choi(2023)는 INTENT-SIM[19]을 활용해 질문 생성 시점을 최적화하고, 상호작용 빈도를 제한하여 효율성을 높였다. Kim 등(2024)[5]은 모델이 내재적 지식을 통해 모호성을 스스로 인지 및 정량화하고, 이를 기반으로 명확화를 요청하도록 정렬하는 기법을 제시했다. Andukuri 등(2024)[1]은 자기 개선 알고리즘을 도입하여 명확화 질문의 품질 자체를 향상시키는 연구를 수행했다. 나아가 Zhang 등(2025)[18]은 미래 대화 턴에서의 기대 효과를 시뮬레이션 하여 선호도 레이블을 부여하는 RLHF 방법을 제안했다.

### 제3장 방법론

본 연구의 핵심 목표는 모호한 입력에 대해 그 유형에 적합한 명확화 질문을 생성하는 것이다. 이를 달성하기 위해 두 개의 모델이 결합된 다중 에이전트 시스템을 제안한다. 본 장에서는 제안하는 전체 파이프라인을 개괄한 후, 각 단계별 세부 내용을 기술한다.

전체 파이프라인 본 연구는 두 개의 모델을 활용한 2단계 시스템을 구축하였다. 시스템의 처리 과정은 다음과 같다. 첫째, 사용자 입력 쿼리는 모호성 분류 모델을 거치며 모호성 유형이 식별된다. 둘째, 식별된 결과에 따라 경로가 분기된다. 모호성이 존재할 경우(NONE이 아닌 경우), 해당 유형 정보와 원본 쿼리가 명확화 질문 생성 모델로 전달된다. 반면, 모호성이 없는 경우(NONE), 쿼리는 답변 생성 모델로 전달되어 즉시 답변을 반환한다. 셋째, 명확화 질문 생성 모델은 입력 받은 모호성 유형과 쿼리를 바탕으로 명확화 질문을 생성한다.



**그림 1** 시스템 파이프라인. 시스템은 모호성 분류 결과에 따라 두 경로로 분기된다. NONE 으로 분류된 쿼리는 즉시 답변 생성으로, 모호성이 탐지된 쿼리는 유형 정보와 함께 명확화 질문 생성 모델로 전달된다.

## 제1절 모호성 분류

본 모델은 사용자 쿼리를 ‘NONE’ (명확함) 또는 8가지 모호성의 유형 중 하나로 분류한다. 분류 기준은 CLAMBER 데이터셋[20]의 체계를 따르며, 상세 기준은 [부록 A]에서 확인할 수 있다. 모델 학습에는 지도 미세 조정(SFT) 방식을 기반으로 LoRA 기법을 적용하였다[4]. 모델은 모호한 쿼리 문자열을 입력 받아 '[카테고리|서브클래스]' 형식의 결합된 문자열로 유형을 반환한다.

## 제2절 라우팅

분류 모델의 결과에 따라 데이터 흐름을 제어한다. 'NONE'으로 분류된 경우, 사용자 쿼리는 답변 생성 모델로 전달된다. 'NONE' 이외의 유형으로 분류된 경우, 분류 모델이 반환한 유형 정보와 사용자 원본 쿼리를 결합하여 명확화 질문 생성 모델의 입력값으로 전달한다.

## 제3절 명확화 질문 생성

본 단계는 지도 미세 조정(SFT)과 직접 선호도 최적화(DPO)의 두 단계로 구성된다. 먼저, CLAMBER 데이터셋[20]의 '명확화 질문' 필드를 활용하여 입력 쿼리에 대한 질문을 생성하도록 모델을 지도 학습시켰다. 이후, 모델의 출력을 사용자 선호에 맞게 조정하기 위해 직접 선호도 최적화(DPO)를 수행하였다[12]. 이를 위해 SFT 모델이 생성한 두 개의 답변 중 인간 선호도가 높은 답변을 선별하여 데이터셋을 구축하였고, 해당 데이터셋을 기반으로 DPO를 진행하였다.

## 제4장 실험 세팅

### 제1절 데이터셋

CLAMBER 본 연구에서는 모호성 분류 모델과 명확화 질문 생성 모델의 지도 미세 조정(SFT)을 위해 CLAMBER 벤치마크를 활용한다[20]. 총 3,202개의 샘플로 구성된 이 데이터셋은 모호한 쿼리와 그렇지 않은 쿼리를 50:50 비율로 포함한다. 모호성 유형은 3개의 상위 카테고리 및 8개의 하위 서브클래스로 체계화되어 있으며, 각 쿼리에 대응하는 명확화 질문을 포함하고 있어 학습 데이터로 적합하다.

**인간 선호도 데이터셋** 명확화 질문 생성 모델의 직접 선호도 최적화(DPO)를 위해 인간 선호도 데이터셋을 새롭게 구축하였다. 지도 미세 조정(SFT)된 모델을 통해 모호한 입력 하나당 2개의 명확화 질문을 생성하였다. 이후 LLM-as-a-judge 방식을 적용하여[8] 평가자 모델이 두 답변을 쌍대 비교하였다. 이 과정을 통해 더 우수한 답변을 선정하고 라벨링을 수행하였다.

CLAQUA 본 연구는 시스템의 종합 평가를 위해 CLAQUA 데이터셋을 활용하였다[17]. 이는 약 4만 개의 오픈 도메인 예제를 포함하며, 모호성 식별, 명확화 질문 생성, 답변 예측의 3단계 작업으로 구성된다. 본 연구는 단일 턴 방식의 모호성 식별(Task 1)과 질문 생성(Task 2) 테스트 데이터를 선별하였다. Task 1의 쿼리 및 레이블과 Task 2의 정답 명확화 질문을 매칭하여 최종 평가 데이터셋을 구축하였다.

### 제2절 모델 선택

**베이스 모델** 모호성 분류 및 질문 생성 모델의 기반으로 Microsoft의 Phi-4-mini-reasoning을 선정하였다[16]. Phi 시리즈는 Microsoft에서 개발한 소형 언어 모델(SLM)군이며, 특히 reasoning 버전은 추론 능력에 특화되어 있다. Phi-4-mini-reasoning은 DeepSeek-R1 기반의 고품질 CoT(Chain-of-Thought) 데이터로 학습되어, Math-500 벤치마크에서 더 큰 파라미터를 가진 경쟁 모델들(7B~8B)을 상회하는 성과를 입증한 바 있다. 3.8B의 경량 파라미터 구조임에도 불구하고 우수한 추론 성능을 보유하고 있어 본 연구의 목적에 부합한다.

**평가자 모델** 본 연구는 LLM-as-a-judge 수행을 위해 Qwen-2.5-7B-instruct를 평가자 모델로 선정하였다[11]. 이 모델은 인간 선호도 데이터셋 구축과 SFT 모델과 DPO 모델 간 성능 평가에 활용되었다. 선행 연구 벤치마크 결과[8], 해당 모델은 오픈소스 LLM 중 가장 탁월한 평가 능력을 입증하였다. 위치 편향 등 일부 항목을 제외한 대부분의 지표에서 상용 모델인 GPT-3.5-turbo를 능가하는 성능을 보였다. 특히 인간 평가와의 일치도 측면에서도 GPT-3.5-turbo보다 높은 수치를 기록하여, 평가자 모델로서의 적합성을 확보하였다.

**답변 생성 모델** 답변 생성을 위해 Meta의 Llama-2-7b-chat-hf를 선정하였다[15]. 이 모델은 인간의 선호도에 정렬되어 대화형 지시를 따르는 데 최적화되어 있다. 사용자의 의도를 반영하여 명확한 답변을 생성하는 데 적합하다. 또한 7B 파라미터 규모로 효율적인 연산이 가능하다.

### 제3절 구현 세부사항

**모호성 분류 모델** 본 모델의 학습은 자원 효율성을 극대화하기 위해 bf16 정밀도와 8-bit AdamW 옵티마이저를 사용하여 진행되었다. 또한 트랜스포머의 어텐션 모듈인 qkv\_proj와 o\_proj를 타겟으로 하여 LoRA(Low-Rank Adaptation) 기법을 적용하였다. 최종 모델로 검증 손실이 가장 낮은 체크포인트를 선정하였다. 상세한 학습 하이퍼파라미터 설정은 표 1과 같다. 또한, 데이터셋의 50%를 차지하는 'NONE' 클래스로 인한 편향을 해소하기 위해 손실 함수에 클래스별 가중치를 적용하였다. 가중치는 각 클래스의 역빈도에 제공근을 취한 값을 기준으로 하되, 빈도가 가장 높은 클래스의 가중치가 1.0이 되도록 정규화하여 산출하였다. 적용된 클래스별 가중치 값은 표 2와 같다.

**명확화 질문 생성 모델** 본 모델은 1차적으로 지도 미세 조정(SFT)을 수행하였다. 자원 효율성을 위해 8-bit AdamW 옵티마이저를 사용하였으며, 학습률은  $5e^{-5}$ 로 설정하였다. LoRA 설정은 앞선 모호성 분류 모델과 동일하게 적용되었다. 상세 설정은 표 3과 같다.

SFT 모델을 기반으로 직접 최적화(DPO)를 적용하여 인간 선호도를 반영한 고품질 질문 생성 능력을 강화하였다. 본 학습에는 1,124개의 선호도 쌍으로 구성된 데이터셋을 활용하였다. 학습률은  $5e^{-7}$ 로 설정하여 기존 SFT 모델의 급격한 변화를 방지하고 안정적인 학습을 유도하였다. KL 발산 페널티를 조절하는 beta 값은 0.3으로 설정하였으며, 손실 함수는 시그모이드(sigmoid) 타입을 사용하였다. DPO 학습 시 모델 어댑터와 참조 어댑터를 분리하여 관리함으로써 효율적인 학습을 구현하였다. 상세한 DPO 학습 파라미터는 표 4와 같다.

구분	파라미터 (Parameter)	값 (Value)
SFT	Epochs	3
	Batch Size (per device)	4
	Gradient Accumulation	4
	Learning Rate	$2e^{-4}$
	Optimizer	AdamW (8-bit)
LoRA	Rank	8
	Alpha	16
	Dropout	0.05
	Target Modules	qkv_proj, o_proj

**표 1** 모호성 분류 모델의 SFT 파라미터 및 LoRA 설정

모호성 유형	가중치
NONE	1.00
WHAT	2.82
WHEN	2.83
WHERE	2.83
WHOM	2.83
CONT	2.83
SEM	2.83
LEX	2.83
UNF	2.83

**표 2** 모호성 유형 별 클래스 가중치

구분	파라미터 (Parameter)	값 (Value)
SFT	Epochs	3
	Batch Size (per device)	4
	Gradient Accumulation	4
	Learning Rate	$5e^{-5}$
	Optimizer	AdamW (8-bit)
LoRA	Rank	8
	Alpha	16
	Dropout	0.05
	Target Modules	qkv_proj, o_proj

**표 3** 명확화 질문 생성 모델의 SFT 하이퍼파라미터 및 LoRA 설정

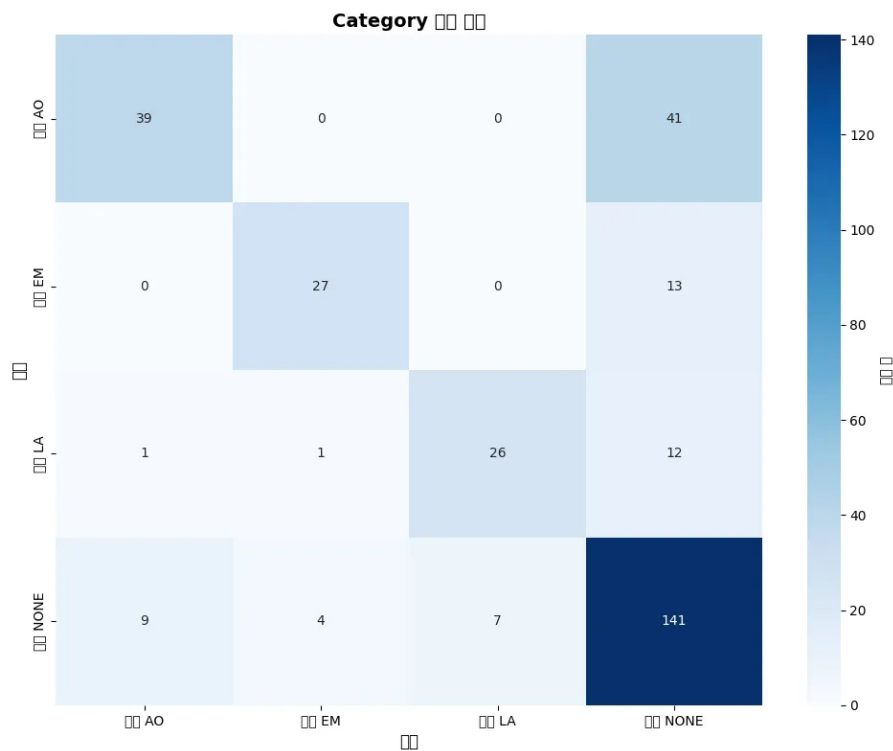
파라미터 (Parameter)	값 (Value)
Epochs	3
Batch Size (per device)	4
Gradient Accumulation	4
Learning Rate	$5e^{-7}$
Beta ( $\beta$ )	0.3
Loss Type	Sigmoid
Optimizer	AdamW

**표 4** 명확화 질문 생성 모델의 DPO 하이퍼파라미터

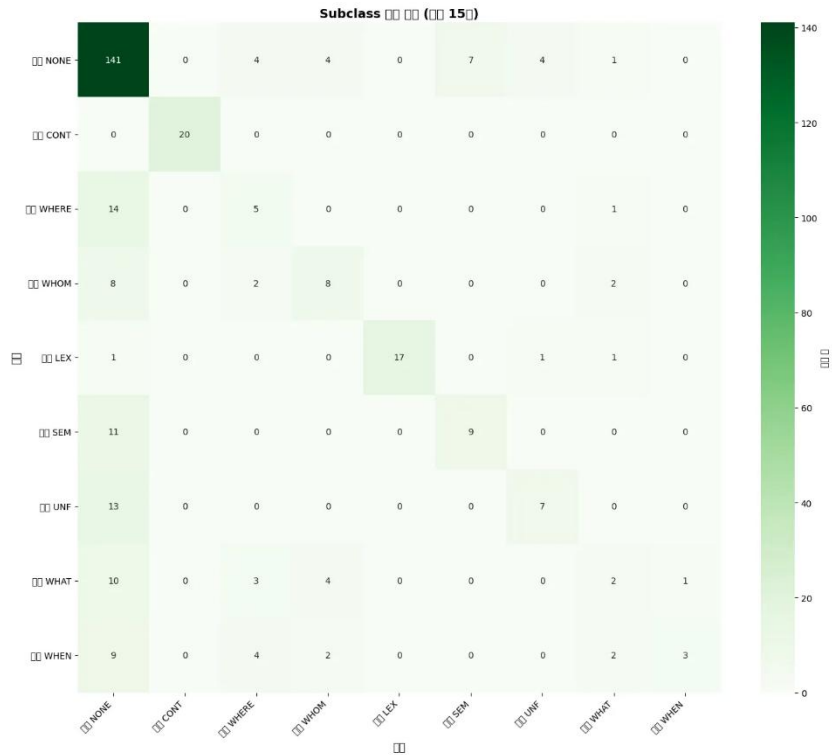
## 제5장 실험 결과

### 제1절 모호성 분류 모델 평가

모호성 분류 모델은 총 321개의 테스트 샘플에 대해 66.04%의 전체 정확도를 달성하였다. 학습 과정에서 최종 훈련 손실은 0.3884, 검증 손실은 0.1156을 기록하며 안정적으로 수렴하였다. 세부 범주(Sub-class) 기준 혼동 행렬을 분석한 결과, 실제 모호한 쿼리임에도 'NONE'으로 예측한 미검출(False Negative) 사례는 66건으로 나타났다. 세부 범주별 성능을 살펴보면 'CONT' 클래스는 100%의 정확도를, 'LEX' 클래스는 85%의 정확도를 기록하여 높은 분류 성능을 보였다. 반면, 'WHAT'(10.00%)이나 'WHEN'(15.00%)과 같은 유형에서는 상대적으로 낮은 정확도를 보였다. 상위 차원(Category) 기준으로는 'NONE' 클래스가 87.58%로 가장 높은 정확도를 기록하였다. 모호성 차원 중에서는 'EM'과 'LA'가 각각 67.50%, 65.00%의 준수한 성능을 보인 반면, 'AO'는 48.75%로 다소 저조한 결과를 나타냈다.



**그림 1** 상위 차원(category) 기준 혼동행렬. NONE 클래스는 87.58%의 높은 정확도를 보였으나, AO 차원은 48.75%로 저조한 성능을 나타냈다. 이는 모델이 명확한 쿼리 식별에는 효과적이지만, 작업별 정보 결락 유형 분류에는 어려움을 겪음을 보여준다.



**그림 2** 세부 범주(Subclass) 기준 혼동 행렬. CONT(100%)와 LEX(85%)는 높은 정확도를 보인 반면, WHAT(10%), WHEN(15%) 등은 저조한 성능을 기록했다. 모델은 구조적 결함 탐지에는 강하지만, 특정 정보 결락 유형의 세밀한 구분에는 한계를 보인다.

'NONE'과 'CONT'에서 보인 높은 정확도는 모델이 쿼리의 문맥적 완결성과 명시적인 구조적 결함을 식별하는 데에는 효과적임을 보여준다. 반면 'WHAT', 'WHEN' 등 특정 정보가 결락된 유형에서의 저조한 성과는 심층적인 의미론적 추론이 필요한 영역에서 여전히 모델의 한계가 존재함을 시사한다. 특히 실제 모호한 쿼리를 'NONE'으로 오분류한 사례들은 명확화 프로세스가 누락되어 잘못된 답변 생성으로 이어질 수 있으므로, 향후 재현율(Recall) 중심의 최적화가 필요함을 암시한다.

## 제2절 명확화 질문 생성 모델 평가

지도 미세 조정 결과 명확화 질문 생성 모델의 1차 지도 미세 조정(SFT) 성능은 실제 생성된 샘플의 정성적 분석을 토대로 적합한 평가 지표를 선정하고, 이를 정량적으로 측정하는 방식으로 검증되었다.

모델의 출력 특성을 파악하기 위해 실제 생성 결과를 분석하였다. [표 5]의 Case 1에서 볼 수 있듯이, 모델은 정답(Ground Truth)과 어휘적 구성은 다르지만 맥락적으로 동일한 의도의 질문을 생성하는 경향을 보였다. 이는 단순히 단어의 등장을 매칭하는 n-gram 기반 지표는 본 연구의 모델 성능을 온전히 반영하기 어려움을 시사한다. 이에 따라 본 연구에서는 두 가지 의미론적 평가 지표를 도입하였다. 첫째, 문장 전체를 벡터 공간에 임베딩하여 코사인 유사도를 측정하는 Semantic Similarity를 통해 전체적인 의미의 유사성을 평가하였다. 둘째, 문맥을 반영한 토큰 간 유사도를 계산하는 BERTScore를 활용하여 토큰 수준에서의 의미 정렬을 측정하였다.

선정된 지표를 바탕으로 한 정량적 평가 결과는 다음과 같다. 첫째, <NO\_CLARIFYING\_QUESTION> 태그 예측 정확도는 99.70%를 기록하였다. <NO\_CLARIFYING\_QUESTION> 태그는 모호성이 없는 쿼리에 대해 질문을 생성하지 않아야 함을 나타낸다. 이는 모델이 불필요한 질문 생성을 효과적으로 억제하고 있음을 보여준다. 둘째, 생성된 질문의 품질을 평가한 결과 BERTScore의 평균 F1 점수는 0.8811, 평균 Semantic Similarity는 0.4722를 기록하였다. 두 지표 간의 상관계수는 0.7504로 나타나 평가의 신뢰성을 확인하였다. 반면, Case 2와 같이 구체적인 정보가 결락된 사례도 관찰되었다. 이는 단순 지도 학습만으로는 사용자의 세밀한 의도를 완벽히 반영하기 어렵다는 한계를 보여준다.

**DPO 학습 결과** DPO 학습의 실효성을 검증하기 위해 SFT 모델과 DPO 모델 간의 쌍대비교를 수행하였다. 무작위 추출된 150개 샘플을 대상으로 LLM-as-a-judge를 활용하여 [8] 더 우수한 명확화 질문을 판정하였다. 이후 인간 검증자의 후처리를 거쳐, 최종적으로 115개의 유효 샘플이 분석에 활용되었다.

두 모델 간의 승률 비교 결과는 [표 6]과 같다. DPO 모델은 48.7%(56개)의 승률을 기록하였다. 이는 SFT 모델 대비 4.4%p의 근소한 우위를 점한 결과이다. 두 답변이 동등하다고 평가된 비율(SAME)은 7.0%(8개)에 불과하였다. 이는 대부분의 사례에서 모델 간 성능 차이가 식별 가능함을 시사한다. 다만, t-검정 결과 p-value는 0.3496(p > 0.05)으로 산출되어 통계적으로 유의미한 성능 차이는 입증되지 않았다.

서브클래스별 승자 분포를 분석한 결과, 모델의 특성에 따른 성능 차이가 관찰되었다. 'WHAT', 'WHEN', 'WHERE' 등 사실적 정보에 기반한 질문 유형에서는 DPO 모델이 우세하였다. 반면, 'LEX'(어휘적 모호성), 'SEM'(의미적 모호성) 등 언어학적 추론이 요구되는 유형에서는 SFT 모델의 선호도가 더 높았다. 한편, 후처리 과정에서 제외된 35개 샘플(전체의 23.3%)을 분석한 결과, 특정 유형에 대한 생성 난이도가 확인되었다. 특히 'WHOM' 유형은 50.0%, 'LEX' 유형은 35.0%가 삭제되었다. 이는 두 모델 모두 해당 유형의 질문 생성에 어려움을 겪고 있음을 보여준다. 이러한 결과는 5.1절에서 언급한 모호성 분류 모델의 성능 저하와 맥락을 같이한다. 따라서 향후 해당 유형에 대한 데이터 증강 및 모델 개선이 필요함을 시사한다.

구분	승리 횟수	비율	비고
DPO	56	48.7%	SFT 대비 + 4.4%p
SFT	51	44.3%	
SAME	8	7.0%	
Total	115	100.0%	p-value = 0.3496

**표 5** DPO 와 SFT 명확화 질문 쌍대 비교 결과. DPO 모델은 SFT 모델 대비 4.4%p 높은 승률(48.7%)을 기록했으나, p-value 0.3496 으로 통계적 유의성은 확보하지 못했다. 이는 DPO 가 질문 품질을 소폭 개선하지만 결정적 차이를 만들지는 못함을 시사한다.

### 제3절 전체 시스템 평가

본 연구는 제안된 파이프라인의 성능을 종합적으로 검증하기 위해 4 가지 평가 지표를 설정하였다:

- **라우팅 정확도 및 F1 점수** 쿼리 분류 능력을 나타낸다.
- **명확화 질문 품질** 모호한 입력에 대해 생성된 명확화 질문의 적절성을 나타낸다.
- **시스템 안정성** 답변 완료율을 나타낸다.
- **응답 속도** 처리 소요 시간을 나타낸다.

**라우팅 정확도 및 F1 점수** 라우팅 정확도는 47.15%, F1 점수는 23.43%를 기록하였다. 특히 재현율이 18.89%로 매우 낮게 나타났는데, 이는 모호성 탐지 능력이 낮음을 의미한다. 정밀도 또한 30.84%로 명확한 쿼리를 모호하다고 판단하는 경우가 많다는 것을 알 수 있다. 반면, 명확한 쿼리에 대한 정탐률은 68.3%로 비교적 양호하였다. 이러한 성능 저하는 학습 데이터와 평가에 사용된 CLAUQA 데이터셋 간의 도메인 불일치에 기인한 것으로 분석된다.

**명확화 질문의 품질** 생성된 명확화 질문과 정답 간의 평균 의미적 유사도 (Semantic Similarity)는 0.4961로 측정되었다. 이는 모델이 생성한 질문이 정답의 의도를 일부 반영하나, 완전히 일치하지는 않음을 시사한다. 유효한 질문 생성률은 전체 모호한 쿼리의 17.9%에 불과했는데, 이는 앞서 언급한 라우팅 단계의 낮은 재현율과 직결된다. 또한 점수의 표준편차가 0.23으로 높게 나타나, 쿼리 유형에 따라 생성 품질의 편차가 크고 성능이 불안정함을 확인하였다.

**시스템 안정성** 시스템의 구동 안정성을 평가한 결과, 99.32%의 높은 완료율을 달성하였다. 총 1,175건 중 실패 사례는 8건에 불과했으며, 이는 모두 빈 응답으로 인한 것이었다. 정확도와는 별개로, 시스템 아키텍처 자체의 안정성은 확보되었음을 입증하였다.

**응답 속도** 전체 시스템의 평균 응답 속도는 4.5110초로 측정되었다. 단계별 시간을 분석한 결과, 분류 단계는 0.1623초(3.6%)로 매우 신속하였다. 반면, 생성 단계는 4.3487초(96.4%)를 차지하였다. 이러한 지연은 쿼리의 모호성 여부와 무관하게 발생하였다. 4.5초의 대기 시간은 실시간 상호작용 관점에서 다소 느린 수치이다. 따라서 사용자 경험 개선을 위해 생성 모델의 경량화 및 최적화가 필수적이다.

## 제6장 결 론

### 제1절 연구 결과 및 의의

본 연구는 LLM의 쿼리 이해 능력을 강화를 위한 다중 에이전트 시스템을 제안한다. 본 연구는 모호한 입력을 8가지 세부 범주로 정밀하게 분류하였다. 식별된 유형 정보는 생성 모델에 제공되어, 각 모호성에 최적화된 질문을 생성하는 데 기여하였다. 이를 통해 모델은 사용자의 쿼리를 스스로 해석하고 능동적으로 질문할 수 있게 되었다. 실험 결과, DPO를 적용하여 모델을 인간 선호도에 맞게 정렬하였다. 또한 시스템은 99.32%의 높은 안정성을 달성하였다. 비록 분류 재현율과 실시간성 측면에서 과제는 남았다. 그러나 경량화된 모델(SLM)만으로도 능동적인 모호성 해소 프로세스를 구현할 수 있음을 입증했다는 점에서 중요한 학술적 의의를 갖는다.

### 제2절 한계점

본 연구는 모호성 분류와 명확화 질문 생성을 통해 LLM의 쿼리 이해 능력을 강화하는 데 기여하였으나, 다음과 같은 한계점을 갖는다.

**베이스 모델의 한계** 본 연구는 추론 능력 강화를 목적으로 수학적 추론에 특화된 Phi-4-mini-reasoning 모델을 채택하였다. 그러나 훈련 과정에서 추론 능력뿐만 아니라 프롬프트 제어 및 지시 이행(Instruction Following) 능력이 필수적임을 확인하였다. 대화와 지시 이행에 최적화된 모델을 활용했다면 생성된 질문의 자연스러움과 품질이 더욱 향상되었을 가능성이 있다.

**모호성 분류 모델의 성능** 파이프라인의 첫 단계인 분류 모델은 약 66%의 정확도를 기록하였으며, 특히 AO(Ambiguous Option) 카테고리에서 낮은 성능을 보였다. 클래스 가중치를 적용하여 데이터 불균형 문제를 완화하고자 하였으나, 근본적인 성능 개선을 위해서는 해당 카테고리에 대한 추가적인 데이터 확보가 필수적이다.

**실제 사용자 대상의 평가 부재** 본 연구는 정량적 지표를 통한 모델 성능 검증에 집중하였다. 반면, 연구의 궁극적 목표인 실제 사용자 경험 개선을 직접적으로 검증하지 못했다는 한계가 있다. 향후 연구에서는 일반 모델과의 A/B 테스트 및 시스템 로그 분석을 수행하여, 시스템의 실질적 효용성과 만족도를 검증할 예정이다.

**컴퓨팅 자원의 제약** 본 연구는 Colab 환경의 제한된 GPU 자원 내에서 수행되었기에 경량화된 오픈소스 모델(SLM)만을 활용하였다. 더 큰 파라미터를 가진 모델을 적용하거나 대규모 학습을 진행한다면 성능의 추가적인 향상을 기대할 수 있을 것이다.

### 제3절 향후 확장 가능한 연구

**내재적 피드백 루프 도입** 본 연구의 파이프라인은 모호성이 감지되면 즉시 명확화 질문을 생성하는 단일 턴 방식이다. 그러나 모든 모호성이 사용자에게 질문을 해야만 해결되는 것은 아니다. 따라서 향후 연구에서는 '내재적 피드백 루프'를 도입하고자 한다. 이는 모델이 사용자에게 질문을 건네기 전, 내부적으로 모호성 해소를 먼저 시도하는 과정이다. 모델은 생성된 명확화 질문에 대해 스스로 답변을 추론하거나 내부 지식을 검색하여 해결 가능성을 탐색한다. 만약 내부 정보만으로 모호성이 해소된다면 즉시 답변을 제공한다. 반대로 내부 추론의 신뢰도가 낮을 때만 최종적으로 사용자에게 명확화를 요청한다. 이는 잦은 질문으로 인한 사용자의 피로감을 근본적으로 해소할 것이다. 궁극적으로는 사용자의 추가적인 개입이나 노력 없이도, 모호한 입력에 대해 완성도 높은 답변을 제공하는 사용자 친화적 시스템으로 발전시킬 것이다.

## 참고 문헌

- [1] Andukuri, C., Fränken, J. P., Gerstenberg, T., & Goodman, N. D. (2024). Star-gate: Teaching language models to ask clarifying questions. *arXiv preprint arXiv:2403.19154*.
- [2] Chen, L., de Melo, G., Suchanek, F. M., & Varoquaux, G. (2025). Query-level uncertainty in large language models. *arXiv preprint arXiv:2506.09669*. <https://arxiv.org/abs/2506.09669>
- [3] Deng, Y., Liao, L., Chen, L., Wang, H., Lei, W., & Chua, T.-S. (2023). Prompting and evaluating large language models for proactive dialogues: Clarification, target-guided, and non-collaboration. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10602–10621. <https://arxiv.org/abs/2305.13626>
- [4] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., ... & Chen, W. (2022). Lora: Low-rank adaptation of large language models. *ICLR*, 1(2), 3.
- [5] Kim, H. J., Kim, Y., Park, C., Kim, J., Park, C., Yoo, K. M., ... & Kim, T. (2024). Aligning language models to explicitly handle ambiguity. *arXiv preprint arXiv:2404.11972*.
- [6] Kuhn, L., Gal, Y., & Farquhar, S. (2022). Clam: Selective clarification for ambiguous questions with generative language models. *arXiv preprint arXiv:2212.07769*.
- [7] Lee, D., Kim, S., Lee, M., Lee, H., Park, J., Lee, S.-W., & Jung, K. (2023). Asking clarification questions to handle ambiguity in open-domain QA. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. <https://arxiv.org/abs/2305.13808>
- [8] Li, D., Jiang, B., Huang, L., Beigi, A., Zhao, C., Tan, Z., ... & Liu, H. (2025, November). From generation to judgment: Opportunities and challenges of llm-as-a-judge. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing* (pp. 2757–2791).
- [9] Li, Z., Li, Y., Xie, H., & Qin, S. J. (2025). CondAmbigQA: A benchmark and dataset for conditional ambiguous question answering. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP 2025)*. <https://arxiv.org/abs/2502.01523>

- [10] Min, S., Michael, J., Hajishirzi, H., & Zettlemoyer, L. (2020). AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*. <https://arxiv.org/abs/2004.10645>
- [11] Qwen, :, Yang, A., Yang, B., Zhang, B., Hui, B., ... Qiu, Z. (2025). Qwen2.5 Technical Report. *arXiv [Cs.CL]*. Retrieved from <http://arxiv.org/abs/2412.15115>
- [12] Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., & Finn, C. (2023). Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36, 53728–53741.
- [13] Ruis, L., Khan, A., Biderman, S., Hooker, S., Rocktäschel, T., & Grefenstette, E. (2023). The Goldilocks of pragmatic understanding: Fine-tuning strategy matters for implicature resolution by LLMs. In *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS 2023)*. <https://arxiv.org/abs/2210.14986>
- [14] Shore, A., Scheinberg, R., Agrawal, A., & Lee, S. Y. (2025). Correct-Detect: Balancing performance and ambiguity through the lens of coreference resolution in LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP 2025)*, pages 30032–30046. <https://arxiv.org/abs/2509.14456>
- [15] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., ... & Scialom, T. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- [16] Xu, H., Peng, B., Awadalla, H., Chen, D., Chen, Y. C., Gao, M., ... & Chen, W. (2025). Phi-4-mini-reasoning: Exploring the limits of small reasoning language models in math. *arXiv preprint arXiv:2504.21233*.
- [17] Xu, J., Wang, Y., Tang, D., Duan, N., Yang, P., Zeng, Q., ... Sun, X. (2019, November). Asking Clarification Questions in Knowledge-Based Question Answering. In K. Inui, J. Jiang, V. Ng, & X. Wan (Eds), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (pp. 1618–1629). doi:10.18653/v1/D19-1172

- [18] Zhang, M. J. Q., Knox, W. B., & Choi, E. (2025). Modeling future conversation turns to teach LLMs to ask clarifying questions. In Proceedings of the International Conference on Learning Representations (ICLR 2025). <https://arxiv.org/abs/2410.13788>
- [19] Zhang, M. J. Q., & Choi, E. (2023). Clarify When Necessary: Resolving Ambiguity Through Interaction with LMs. *arXiv [Cs.CL]*. Retrieved from <http://arxiv.org/abs/2311.09469>
- [20] Zhang, T., Qin, P., Deng, Y., Huang, C., Lei, W., Liu, J., Jin, D., Liang, H., & Chua, T.-S. (2024). CLAMBER: A benchmark of identifying and clarifying ambiguous information needs in large language models. In *Proceedings of the Association for Computational Linguistics (ACL 2024)* <https://arxiv.org/abs/2405.12063>

## 부 록

### A. 모호성 분류 9개 범주

본 분류 체계는 Zhang et al. (2024)의 CLAMBER 벤치마크를 기반으로 함.

차원	범주	설명	예시
인식론적 불일치 (Epistemic Misalignment , EM)	생소함 (UNFAMILIAR)	질의가 생소한 개체(entities)나 사실(facts)을 포함함	Find the price of Samsung Chromecast.
	모순 (CONTRADICTION )	질의가 자기 모순(self- contradictions) 을 포함함	Output 'X' if the sentence contains [category withhold] and 'Y' otherwise. The critic is in the restaurant.>X . The butterfly is in the river.>Y. The boar is in the theatre.>?
언어적 모호성 (Linguistic Ambiguity, LA)	어휘 (LEXICAL)	질의가 다중 의미를 가진 용어(terms)를 포함함	Tell me about the source of Nile.
	의미 (SEMANTIC)	질의가 다중 해석을 유발하는 맥락의 부족을 포함함	When did he land on the moon?
우연적 결과 (Aleatoric Output, AO)	누구 (WHO)	질의 응답이 누락된 개인적 요소(personal elements)로 인해 혼란을 포함함	Suggest me some gifts for my mother.

	언제 (WHEN)	질의 누락된 요소 (temporal elements)로 인해 혼란을 포함함	응답이 시간적 How many goals did Argentina score in the World Cup?
	어디 (WHERE)	질의 누락된 요소 (spatial elements)로 인해 혼란을 포함함	응답이 공간적 Tell me how to reach New York
	무엇 (WHAT)	질의 누락된 요소 (task- specific elements)로 인해 혼란을 포함함	응답이 작업별 Real name of gwen stacy in spiderman?
명확함 (Clear)	NONE	질의가 모호성을 포함하지 않으며, 추가적인 명확화 없이 답변 가능함	What is the capital of France?

표 6 모호성 분류 체계.

## B. 시스템 프롬프트

그림 3과 그림 4는 각각 모호성 분류 모델 SFT의 시스템 프롬프트와 명확화 질문 생성 모델 DPO의 시스템 프롬프트이다. 작업에 대한 설명과 모호성 분류 체계를 제공한다.

You are an AI system that determines if the question requires clarification and classifies the ambiguity.

Task:

1. Determine if the question requires clarification: clear(no clarification needed) or ambiguous(clarification needed)

2. Classify the ambiguity:

- If question is clear, set category=NONE and subclass=NONE
- If question is ambiguous, classify category and subclass

Output format: category|subclass

Categories:

- EM (Epistemic Misalignment): Questions with unfamiliar entities or self-contradictions
- LA (Linguistic Ambiguity): Questions with lexical or semantic ambiguity
- AO (Aleatoric Output): Questions with missing contextual information causing confusion
- NONE: Clear questions that don't require clarification

Subclasses:

For EM:

- UNF (UNFAMILIAR): Query contains unfamiliar entities or facts
- CONT (CONTRADICTION): Query contains self-contradictions

For LA:

- LEX (LEXICAL): Query contains terms with multiple meanings
- SEM (SEMANTIC): Query lacks context leading to multiple interpretations

For AO:

- WHOM: Query output contains confusion due to missing personal elements
- WHEN: Query output contains confusion due to missing temporal elements
- WHERE: Query output contains confusion due to missing spatial elements
- WHAT: Query output contains confusion due to missing task-specific

그림 3 모호성 분류 모델 SFT 시스템 프롬프트

You are an AI that generates a single, concise clarifying question when a user's query is ambiguous.

Task:

Generate exactly one clarifying question based on the ambiguity type.

Output format: One clarifying question

Categories:

- EM (Epistemic Misalignment): Questions with unfamiliar entities or self-contradictions
- LA (Linguistic Ambiguity): Questions with lexical or semantic ambiguity
- AO (Aleatoric Output): Questions with missing contextual information causing confusion

Subclasses:

For EM:

- UNF (UNFAMILIAR): Query contains unfamiliar entities or facts
- CONT (CONTRADICTION): Query contains self-contradictions

For LA:

- LEX (LEXICAL): Query contains terms with multiple meanings
- SEM (SEMANTIC): Query lacks context leading to multiple interpretations

For AO:

- WHOM: Query output contains confusion due to missing personal elements
- WHEN: Query output contains confusion due to missing temporal elements
- WHERE: Query output contains confusion due to missing spatial elements
- WHAT: Query output contains confusion due to missing task-specific elements

그림 4 명확화 질문 생성 모델 DPO 시스템 프롬프트

## C. 전체 시스템 평가 결과 상세

### 라우팅 정확도 평가

	실제: 명확함 (0)	실제: 모호함 (1)
예측: 답변 생성 (0)	459 (TN)	408 (FN)
예측: 명확화 질문 (1)	213 (FP)	95 (TP)

표 7 라우팅 평가 혼동 행렬

### 명확화 질문 품질 평가

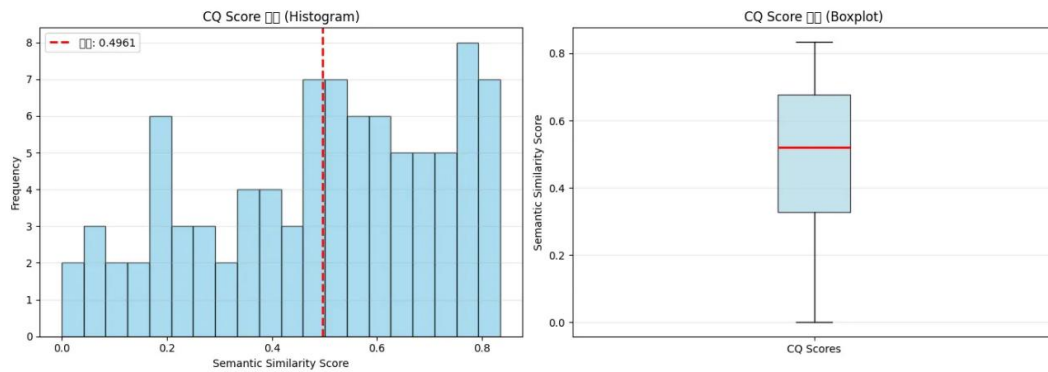


그림 6 생성된 명확화 질문의 Semantic Similarity 분포

Score	Query	Generated	Gold
0.8346	What is sport of Buffalo Bulls	Is the sport of Buffalo Bulls a team name or a type of game?	Are you talking about the Buffalo Bulls baseball team or the Buffalo Bulls men's basketball team...
0.8282	What is occupation of Robert Johansson?	What do you mean by "occupation" in the context of Robert Johansson?	Do you mean politician Robert Johansson or actor Robert Johansson...
0.8269	A list of movies with Diane Peterson in them	What type of movies does Diane Peterson appear in?	Are you referring to actor Diane Peterson or actress Diane Peterson...
0.8032	What is stated on the software license of Write!	Is it the software license of Write! or the software license of Write!	Are you talking about word processor Write! or text editor Write!...
0.8011	What is cause of David Clark's death	Please provide more context or details about the cause of David Clark's death.	Are you talking about non player David Clark or player David Clark...

표 8 가장 우수한 명확화 질문 상위 5 개 샘플

Score	Query	Generated	Gold
0.0000	Why is James Burke known?	When is the best time to use the word "what"?	Do you mean non producer James Burke or producer James Burke...
0.0275	Where did Bulletproof originate from?	When did the first human encounter with the entity occur?	When you say the originating country of bulletproof...
0.0776	How is Ruby classified?	When is the best time to go to the beach?	Are you talking about Asian elephant Ruby or parrot Ruby...
0.0803	What is the postcode of rhodes?	Which one?	When you say the zip code of the place, are you referring to...
0.0815	What is Mobile Revelers's sport?	Is it a person or a thing?	When you say the sport mobile revelers, are you talking about...

표 5 가장 낮은 명확화 질문 상위 5 개

## D. 시스템 작동 예시

제안된 2단계 시스템의 실제 작동 예시를 보여준다. 각 사례는 사용자 쿼리, 모호성 분류 결과, 생성된 명확화 질문(또는 최종 답변)으로 구성된다.

### 모호한 쿼리 처리 사례

#### 사례 1: WHERE 유형

질문 입력	Give me a list of good coffee shops?
분류 결과	AO WHERE
최종 답변	Please provide the city or specific location to narrow down the search.

시스템은 쿼리에서 공간적 요소(위치 정보)가 누락되었음을 정확히 식별하였다. 사용자에게 구체적인 도시나 지역을 명시하도록 요청하는 적절한 명확화 질문을 생성하였다.

## 명확한 쿼리 처리 사례

### 사례 2: NONE 유형

질문 입력	In which geological period did Dysdera first appear?
분류 결과	NONE NONE
최종 답변	Dysdera is an extinct genus of trilobites that lived during the Paleozoic Era, specifically during the Devonian period, which lasted from about 416 to 359 million years ago. The first appearance of Dysdera is believed to have been during the Early Devonian, around 405 million years ago. The genus was present throughout the Devonian period and went extinct at the end of the period, around 359 million years ago. So, to summarize, Dysdera first appeared during the Early Devonian period, around 405 million years ago.

시스템은 쿼리가 충분히 명확하다고 판단하여 추가적인 명확화 질문 없이 즉시 답변을 생성하였다. 질문에 포함된 모든 필수 정보(대상: Dysdera, 질문 내용: 출현 시기)가 명시되어 있어 모호성이 없는 것으로 정확히 분류되었다.

## 영문 요지

In interactions between Large Language Models (LLMs) and users, query ambiguity is a primary factor that causes interpretation errors and compromises system reliability. However, existing LLMs exhibit limitations in detecting ambiguous inputs or explicitly resolving them. To address this issue, this study proposes a multi-agent system that generates clarification questions based on specific ambiguity types. The proposed system consists of a two-stage pipeline. First, the model precisely classifies the ambiguity type of the user query. Second, it generates clarification questions optimized for the identified type information. Experimental results demonstrate that the classification model achieved an accuracy of 66.04%. Furthermore, the generation model was aligned with human preferences via Direct Preference Optimization (DPO), thereby enhancing the quality of the generated questions. The integrated system demonstrated high operational stability of 99.32%. This study holds significant academic value by demonstrating that an active ambiguity resolution process can be implemented using only Small Language Models (SLMs). The code and training data for this study are available at <https://github.com/jihyeyu33/LLM-Interactive-Clarification>.