논문 리뷰_eTracer: Polarity-Aware Evidence Grounding

생성된 문장이 어떤 소스를 참고하는지 검증하는 법

Deep Shark Lab 학부생 **이예정**

2026. 11. 27

eTracer: Polarity-Aware Evidence Grounding

Bohao Chu, Qianli Wang, Hendrik Damm, Shuning Zhang, Hui Wang, Andrea Papenmeier, Georg Lodde, Livingstone, Christoph M. Friedrich, Norbert Fuhr

Abstract

Evidence grounding makes generated content verifiable by linking statements to their sources. However, effective verifiability requires two properties that existing methods fail to jointly provide: *verification granularity*, the ability to localize fine-grained evidence for each statement, and *verification completeness*, the ability to capture both support and contradiction. Existing methods either operate at granularities too coarse for rapid inspection or are support-only. [illegible] sentence to its contextual evidence with a [illegible] human-annotated benchmark with 7,905 supporting and contradicting sentence pairs across seven domains, covering both explicit and implicit contradictions. We further propose **eTracer**, a post-hoc framework whose context-aware signed scorer recovers the full evidence [illegible] per response, with localization and polarity emerging jointly from one mechanism. eTracer achieves the best overall Citation F1 of **82.39**, driven by a Contradict F1 of **89.20**. A pre-registered within-subjects user study (n = 30) shows that polarity-aware grounding nearly doubles contradiction detection over the document-level baseline (**27.9% → 51.9%**, [illegible]) and reveals a [illegible] bias under which support-only grounding fails to improve over this baseline (**23.4% vs. 27.9%**). Dataset and code: https://github.com/chubohao/eTracer.

Anthology ID: 2026.emnlp-main.314
Volume: Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing
Month: October
Year: 2026
Address: Budapest, Hungary
Editors: Hung-Yi Lee, Sunipa Dev, Junyi Jessy Li, Desmond Elliott
Venue: EMNLP
SIG: SIGDAT
Publisher: Association for Computational Linguistics
Note: –
Pages: 7224–7253
Language: –
URL: https://aclanthology.org/2026.emnlp-main.314/
DOI: 10.18653/2026.emnlp-main.314

✳Contents

1. 서론 및 연구 목표
2. 벤치마크 데이터셋 구축
3. eTracer 프레임워크
4. 평가 및 분석
5. 결론 및 한계점

✳ 논문 정보

제목 및 연구진

ETRACER: Polarity-Aware Evidence Grounding

Bohao Chu[1,*] **Qianli Wang**[2] **Hendrik Damm**[3]
Shuning Zhang[4] **Hui Wang**[1] **Andrea Papenmeier**[5] **Georg Lodde**[6]
Elisabeth Livingstone[6] **Christoph M. Friedrich**[3] **Norbert Fuhr**[1]
[1]University of Duisburg-Essen, [2]Technische Universität Berlin,
[3]University of Applied Sciences and Arts Dortmund,
[4]Tsinghua University, [5]University of Twente, [6]University Hospital Essen,
[*]**Correspondence**: bohao.chu@qq.com

저널 정보

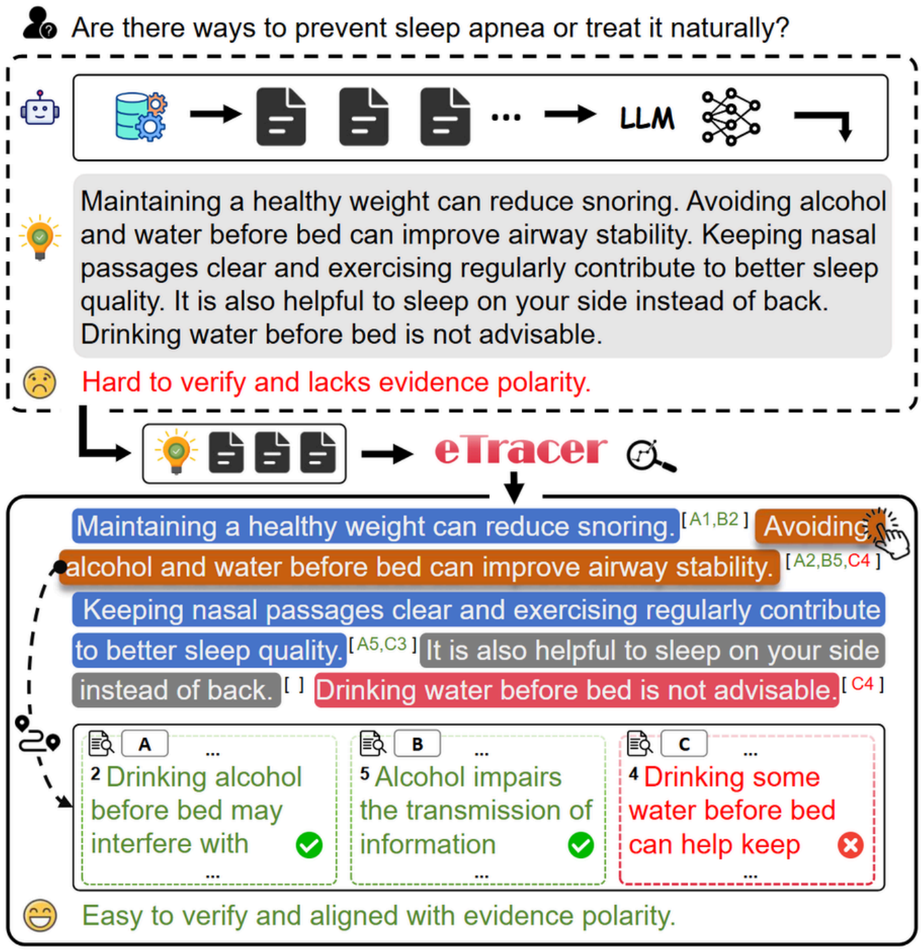
- EMNLP 2026
  - Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing

서론 및 연구 목표

✳ 연구 배경

기존 근거 연결(Evidence Grounding) 방법의 검증성 한계

- RAG 기반의 응답 검증 방식은 검증 세분성 부족, 검증 완전성 부족의 문제가 있다



검증 세분성 부족

- 검증 세분성이란? 각 진술에 대한 세밀한 **근거의 위치를 식별**하는 것이다
  - 주장이나 응답의 근거가 어디 있는가? 라는 질문으로 바꿀 수 있다.

- 그러나, 기존 검증에서는 문서, 문단 단위로만 근거를 파악할 수 있다
- 정확히 어떤 문장을 참조한 답변인지 확인하기 어렵다

검증 완전성 부족

- 검증 완전성이란? 지지 근거와 모순 근거를 **모두 포착하여 검증**하는 것이다
  - 주장이나 응답의 근거를 모두 찾을 수 있는가? 라는 질문으로 바꿀 수 있다

- 그런, 기존 검증에서는 지지 근거에만 초점을 맞추어 모순 근거를 충분히 식별하지 못한다

✳ 연구 목표

연구 목표는?

- **세밀한 근거 위치 식별:** 응답 문장과 출처 문장 사이의 문장-문장 단위 근거 관계를 식별한다
  - 기존 문서-문장 등의 형식에서 벗어난 문장-문장 단위의 근거 관계를 찾는다

- **근거 극성 분류:** 응답과 출처 사이의 관계를 지지(Support), 모순(Contradict), 무관(Irrelevant)으로 분류한다

- **맥락 기반 근거 분석:** 전체 응답과 출처 문서의 맥락을 고려하여 문장 간 복합적인 근거 관계를 분석한다
  - 문장 하나만 보는 것이 아닌, 전체 문장에서 드러나는 맥락을 근거 분석에 활용한다

- **효율적인 근거 연결:** 단일 순전파를 통해 전체 근거 관계 행렬을 산출하는 프레임워크를 개발한다

- **검증 유효성 평가:** 극성을 고려한 근거 제공 방식이 실제 사용자의 정보 검증 능력에 미치는 영향을 분석한다
  - 실제 연구진이 제시한 모델과 다른 모델들을 비교한다
  - 모델에서 특정 조건을 바꾸어 유효성을 검증한다

벤치마크 데이터셋 구축

✳기존 데이터셋 확보

사용 데이터

- LFRQA 데이터셋의 구조를 활용하여 데이터를 증강시켰다
  - LFRQA는 RAG 기반 질의 응답용 데이터셋이다
  - 질문, 응답, 참고 문서와 인용 정보로 구성되어 있다

질문

기업은 왜 주식을 공개하는가?

응답

기업은 주식 거래와 신규 주식 발행을 통해 자금을 조달하기 위해 주식을 공개한다. [2,3]
또한, 주식을 보유한 직원들의 경제적 이익을 창출할 수 있다. [1]

예시 문장 출처 논문: RAG-QA Arena: Evaluating Domain Robustness for Long-Form Retrieval-Augmented Question Answering

출처 문서

문서 1
기업 공개를 통해 기업의 가치가 상승하면 주식을 보유한 작원들도 경제적 이익을 얻을 수 있음.

문서 2
기업은 주식 거래와 신규 주식 발행을 통해 사업에 필요한 자금을 조달할 수 있음.

문서 3
기업 공개의 목적은 자금 확보에 있음.

✳ ETRACER-BENCH 데이터셋

LFRQA

장문 응답 (문서 인용)

월요일에 운영한다. [D1]
오후 6시에 문을 닫는다. [D2]

D1: 월요일 정상 운영.
주차장 무료 이용 가능.
D2: 오후 6시 폐관.

대응하는 문장을 찾은 후, 그 문서를 인용

문서 단위 근거 연결

→

ETRACER-BENCH

응답 문장 R

r_1 월요일 운영 / r_2 오후 6시 폐관

근거 문장 S (s_4: 추가 모순 예시)
s_1 월요일 운영 s_2 무료 주차
s_3 오후 6시 폐관 s_4 오후 5시 폐관

	s_1	s_2	s_3	s_4
r_1	+1	0	0	0
r_2	0	0	+1	-1

+1 지지 -1 모순 0 무관

문장 단위 근거 연결 + 극성 주석

✳ 데이터셋 구성

구분	세부 항목	수량 / 비율
데이터 규모	선정 응답 / 도메인	**700 / 7**
	응답 문장	**1,736**
	근거 후보 문장	**16,794**
근거 관계	지지 (Support)	**4,565**
	자연 모순	**94**
	인위적 명시 모순 (증강)	**1,205**
	인위적 암시 모순 (증강)	**2,041**
데이터 분할	학습 / 검증 / 테스트 (응답 단위)	**70% / 10% / 20%**

✳ 데이터셋 카테고리 구성

도메인	주요 내용	원본 데이터 출처
BioASQ(생의학)	의학, 생명과학, 질병	**BioASQ**
FiQA(금융)	금융, 투자, 경제	**FiQA**
Technology (Tech)	컴퓨터, 소프트웨어, 기술	**LoTTE**
Science (Sci)	자연과학 및 과학적 지식	**LoTTE**
Lifestyle (Life)	일상생활 및 생활 정보	**LoTTE**
Writing (Writ)	문법, 작문, 글쓰기 기법	**LoTTE**
Recreation (Rec)	취미, 게임, 여가 활동	**LoTTE**

✳ ETRACER-BENCH 데이터셋의 특징

문서 기준이 아닌 문장 기준의 데이터

- 기존의 데이터는 문서를 기준으로 인용되어 있지만, ETRACER-BENCH 데이터셋은 문장을 인용하게 한다
- 이 방법으로 검증 세분성을 입증한다

각 문장별로 근거 관계를 분류한 데이터

- 각 문장별로 지지, 모순, 관계 없음 세 가지의 근거 관계를 제시한다
  - **지지(Support):** 근거 문장이 응답 문장의 내용을 뒷받침하는 관계
  - **모순(Contradict):** 근거가 응답 문장의 주장과 충돌하는 관계
  - **무관(Unrelated):** 응답을 지지하거나 모순하지 않는 관계
- 이 데이터셋은 위의 **3방향 극성 레이블(Three-Way Polarity Label)**을 활용했다

	s_1	s_2	s_3	s_4
r_1	지지	무관	무관	무관
r_2	무관	무관	지지	모순

✳ 근거 극성 분류법

응답(Response)와 근거(Source)의 관계

$R = \{r_i\}_{i=1}^{n}$ **응답** R : 전체 응답 문장 집합
r_i : 긴 전체 응답의 i번째 문장

$S = \{s_j\}_{j=1}^{m}$ **근거** S : 전체 근거 문장 집합
s_j : 긴 전체 근거 문장의 j번째 문장

ϕ 극성 인식 근거 연결 함수를 뜻한다
r_i, s_j 를 input으로 넣었을 때, 다음과 같은 조건으로 output이 나온다

$$\phi(r_i, s_j) = \begin{cases} +1, & \text{if } s_j \text{ supports } r_i, \\ -1, & \text{if } s_j \text{ contradicts } r_i, \\ 0, & \text{if } s_j \text{ is irrelevant to } r_i. \end{cases}$$

근거 문장이 응답 문장을 지지할 경우 +1

근거 문장이 응답 문장에 모순될 경우 -1

관련 없을 경우 0

✳ 모순에도 종류가 있다?

명시적 모순

- 응답의 주장과 근거의 사실이 직접 충돌하는 경우

암시적 모순

- 근거에 함축된 의미를 해석하거나 추론해야 충돌을 판단할 수 있는 경우

비교 항목	명시적 모순 (Explicit)	암시적 모순 (Implicit)
모순 표현	직접적인 사실 충돌	간접적으로 드러나는 충돌
판단 방식	표면적·의미적 사실 비교	문맥 해석 및 의미적 추론
응답 예시	도서관은 일요일에 운영한다.	도서관은 일요일에 운영한다.
근거 예시	일요일에는 운영하지 않는다.	운영 요일은 월~금으로 한정된다.

- 추후 실험에서 이 두 가지 모순을 전부 잘 찾을 수 있는지 비교 실험을 진행하게 된다

eTracer 프레임워크

✳ eTracer 프레임워크 제안

전체 처리 과정 (Forward Pass & Inference)

Step 1. 입력 시퀀스 구성

R, S + 문장 경계 토큰

→

Step 2. 전체 문장 인코딩

ModernBERT + Self-Attention

→

Step 3. 문장 표현 추출

경계 토큰 은닉 상태 h_{ri}, h_{sj}

→

Step 4. 문장 쌍별 벡터 연결

$[h_{ri} ; h_{sj}]$

↓

Step 5. 관계별 확률 계산

MLP + Softmax

←

Step 6. 관계 결정

Argmax (최대 확률 클래스)

←

Step 7. 최종 근거 관계 행렬

Signed Evidence Map

$\Phi \in \{-1, 0, +1\}^{n \times m}$

✳ eTracer 프레임워크 제안

입력 시퀀스 구성

- 모델에 활용할 input을 구성하기 위해, **BERT encoder 기반 문장 임베딩**을 수행한다
- 응답 데이터 R과 근거 데이터 S를 문장 단위로 분리한다
  - 응답은 [R_SEP]토큰으로 문장을 분리하고, 근거는 [S_SEP]로 문장을 분리한다

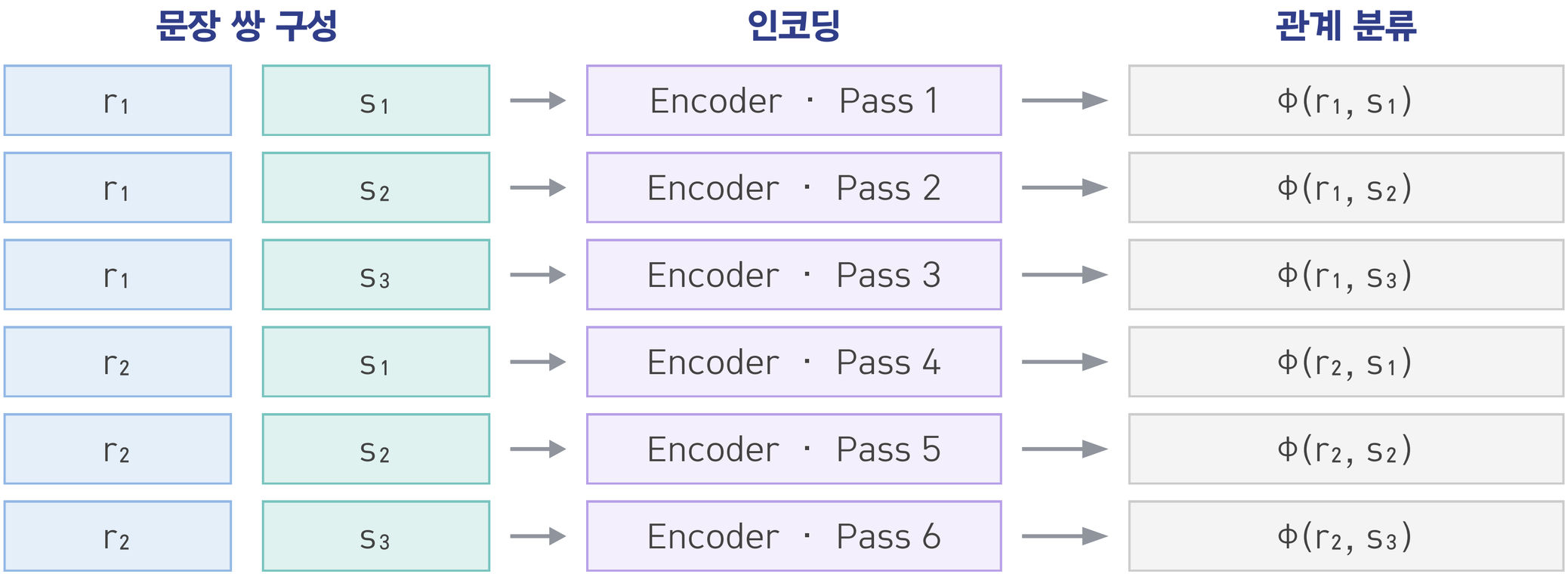
$$[\mathrm{CLS}]\ \underbrace{r_1[\mathrm{R_SEP}]\cdots r_n[\mathrm{R_SEP}]}_{\text{response}}\ [\mathrm{SEP}]\ \underbrace{s_1[\mathrm{S_SEP}]\cdots s_m[\mathrm{S_SEP}]}_{\text{context}}$$

[CLS]	시퀀스 시작	
r_1	도서관은 월요일에 운영한다.	[R_SEP]
r_2	도서관은 오후 6시에 닫는다.	[R_SEP]
[SEP]	응답과 근거 영역 구분	
s_1	월요일은 정상 운영일이다.	[S_SEP]
s_2	도서관은 오후 5시에 폐관한다.	[S_SEP]
s_3	주차장은 무료로 이용할 수 있다.	[S_SEP]

✳ 토큰을 활용해 문장을 구분하는 이유는?

기존 방식과의 차이점

- 기존 방식은 **Pair-wise Scoring** 으로 이루어진다
- 즉, 응답이 2개, 근거가 3개라면 총 6개의 Pair를 준비해서 학습시켜야 한다

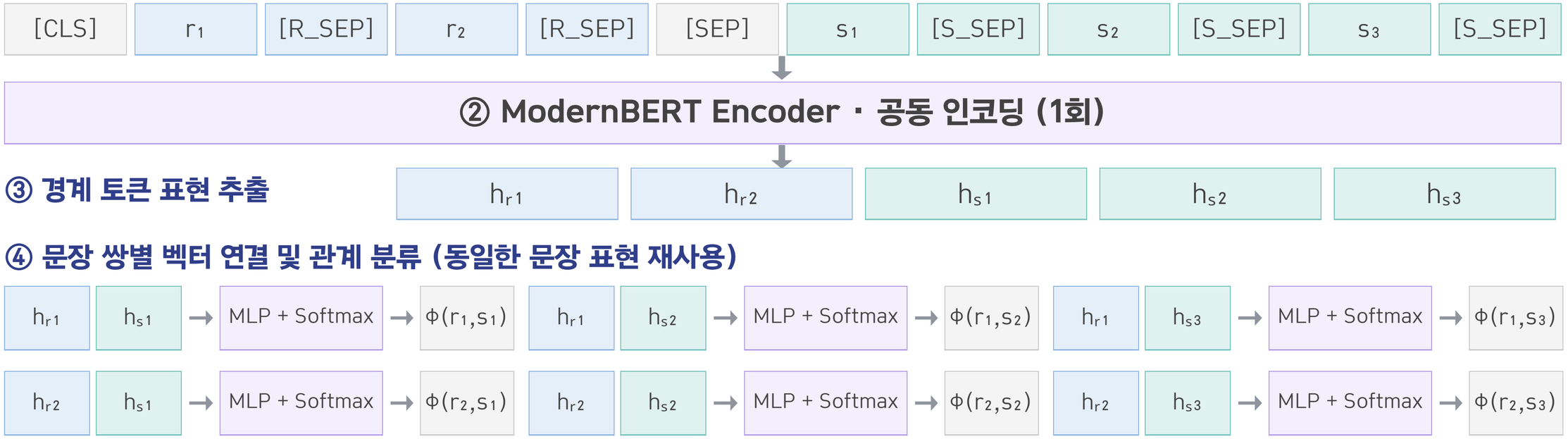


✳ 토큰을 활용해 문장을 구분하는 이유는?

전체 문장의 구조를 유지하는 방법을 사용

- 토큰을 활용하는 방식은 한 번에 전체 문장을 입력할 수 있게 된다
- 문장의 맥락 정보를 해치지 않는다는 장점과, 연산량이 적어진다는 장점을 지닌다

① 전체 입력 시퀀스 구성



✳ 문장 쌍별 관계 확률 계산

확률 계산 수식

$$p(\cdot \mid r_i, s_j) = \text{softmax}(\text{MLP}([h_{r_i}; h_{s_j}]))$$

$p(\cdot \mid r_i, s_j)$ 두 문장이 주어졌을 때의 클래스별 확률 분포

h_{r_i} 응답 문장 r_i의 맥락 정보가 반영된 은닉 상태 벡터

h_{s_j} 근거 문장 s_j의 맥락 정보가 반영된 은닉 상태 벡터

- MLP를 활용하여 **관계별 점수**를 측정한다
- softmax를 통과해 MLP에서 나온 점수를 **총합 1인 확률 분포**로 바꾼다

$$\text{MLP}([h_{r_i}; h_{s_j}]) = \begin{bmatrix} 3.0 \\ 1.0 \\ -2.0 \end{bmatrix}$$



$$p(\cdot \mid r_i, s_j) \approx \begin{bmatrix} 0.8756 \\ 0.1185 \\ 0.0059 \end{bmatrix}$$

✳ 모델의 최종 출력

극성 관계 행렬

- 모든 응답-근거 문장 쌍에 대한 관계별 확률을 산출한다
- 그 후, 가장 높은 극성 레이블을 선택하여, 최종적으로는 $n \times m$ 크기의 행렬이 나온다

문장 쌍	지지 (+1)	무관 (0)	모순 (−1)	합
(r_1, s_1)	0.9200	0.0500	0.0300	1
(r_1, s_2)	0.0400	0.9100	0.0500	1
(r_1, s_3)	0.0200	0.9600	0.0200	1
(r_2, s_1)	0.0800	0.8900	0.0300	1
(r_2, s_2)	0.1142	0.0420	0.8438	1
(r_2, s_3)	0.0300	0.9500	0.0200	1

→
Argmax

	s_1	s_2	s_3
r_1	+1	0	0
r_2	0	−1	0

✳ 로스 함수 설계

클래스 가중 엔트로피 교차 손실(Class-Weighted Cross-Entropy Loss)

- 전체 문장에 대한 손실을 판단하기 위해, 각 오차와 가중치를 계산해 **전부 더한 값을 사용**한다

$$\mathcal{L} = \sum_{i=1}^{n} \sum_{j=1}^{m} w_{y_{ij}} \, \mathrm{CE}(\hat{y}_{ij}, y_{ij})$$

$\hat{y}_{ij}$ 모델이 예측한 문장 쌍의 분류 출력

y_{ij} 문장 쌍의 실제 정답 레이블

$w_{y_{ij}}$ 연구자가 설정한 각 정답 클래스의 가중치

평가 및 분석

✳ 실험 설계

비교 모델

제안 모델	ETRACER (ModernBERT-large 기반 학습 모델)

구분	비교 대상
문장 쌍 분류	Mpair
LLM 기반	GPT-4o, Qwen3-235B (0-shot / 7-shot)
NLI 기반	BART, DeBERTa

- 대상 모델과 각기 다른 모델들을 비교한다
  - 비교 LLM의 경우엔 프롬프트를 설정하는 것으로 비교한다

- 문장 쌍 분류의 경우, Pair-wise Scoring과 토큰 구분 방식을 비교하기 위해 진행되었다

✳ 실험 설계

평가 방식

구분	정의
Support F1	지지 관계의 F1-score
Contradict F1	모순 관계의 F1-score
Citation F1	Support F1과 Contradict F1의 산술평균

로스 함수 설계 시의 가중치 설정

$$w_{y_{ij}} = [w_{\text{sup}}, w_{\text{irr}}, w_{\text{con}}] = [1, 10, 1]$$

w_{sup} **지지 근거**의 가중치 w_{irr} **무관 근거**의 가중치 w_{con} **모순 근거**의 가중치

- 모델이 '무관'으로 잘못 응답한 경우, 제일 loss를 크게 만든다
- 해당 가중치는 Grid Search를 통해 Citation F1이 가장 높은 조합을 고른 것이다

✳ 실험 설계

실험 1. 전체적인 모델 성능은 어떤가?

- 비교군 모델을 활용하여 연구자의 모델이 얼마나 뛰어난 성능을 보이는지 실험한다
- 지지 관계, 모순 관계의 정밀도, 재현율, F1-score와 Citation F1를 측정한다

실험 2. 명시적 모순과 암시적 모순을 찾아낼 수 있는가?

- 명시적 모순보다도 암시적 모순이 포함되어있을 경우의 분류를 잘 하는지 실험한다
- 전체 모순 관계와 명시적 모순, 암시적 모순의 F1-score를 측정한다

실험 3. 문장 간 정보 Attention을 제한했을 때의 성능 변화가 있는가?

- 전체 문장을 줬을 때, 다른 근거 문장을 주었을 때, 다른 응답 문장을 주었을 때, 둘 다 제공하지 않았을 때를 구분하여, 실제로 문장 사이의 문맥 정보가 분류에 영향을 미치는지 실험한다

실험 4. 문장의 순서가 바뀌었을 때 성능 변화가 있는가?

- 근거 문장의 순서를 무작위로 바꾼 후에 성능에 영향이 있는지 실험한다

✳실험 결과_모델별 성능

지지 관계 관련 | 모순 관계 관련

Category	Method	Arch.	Params	Infer.	S.P	S.R	S.F1	C.P	C.R	C.F1	Cit.F1
Prompt	GPT-4o (0-shot)†	Dec-only	–	1*	**88.60**	68.96	77.56	91.02	48.64	63.40	70.48
	GPT-4o (7-shot)†	Dec-only	–	1*	85.39	71.95	78.10	87.31	64.96	74.50	76.30
	Qwen3-235B (0-shot)†	Dec-only	235B	1*	79.18	72.95	75.94	**92.42**	70.24	79.82	77.88
	Qwen3-235B (7-shot)†	Dec-only	235B	1*	78.80	**77.49**	**78.14**	89.51	81.92	85.55	81.85
Fine-tuned	DeBERTa-v3 (MNLI)‡	Enc-only	435M	73	66.03	34.48	45.30	52.15	69.92	59.74	52.52
	BART-large (MNLI)‡	Enc-Dec	406M	73	69.84	34.15	45.87	40.47	68.32	50.83	48.35
	DeBERTa-v3 (ETB)	Enc-only	435M	73	71.51	73.73	72.60	74.03	76.64	75.31	73.96
	BART-large (ETB)	Enc-Dec	406M	73	74.27	70.40	72.28	71.75	77.60	74.56	73.42
	$\mathcal{M}_{pair}$ (ETB)	Enc-only	395M	73	74.70	70.07	72.31	74.11	66.40	70.04	71.18
	$\mathcal{M}_{con}$ (ETRACER)	Enc-only	395M	**1**	84.10	68.63	75.58	90.59	**87.84**	**89.20**	**82.39**

✳ 실험 결과_명시적 모순 vs 암시적 모순

명시적 모순 $Ce.F1$
- 응답의 주장과 근거의 사실이 직접 충돌하는 경우

암시적 모순 $Ci.F1$
- 근거에 함축된 의미를 해석하거나 추론해야 충돌을 판단할 수 있는 경우

Method	C.F1	C_e.F1	C_i.F1	Δ
GPT-4o (7-shot)	74.50	89.50	61.54	−27.96
Qwen3-235B (7-shot)	85.55	91.31	81.48	−9.83
DeBERTa-v3 (MNLI)	59.74	67.26	53.30	−13.96
BART-large (MNLI)	50.83	57.36	45.16	−12.20
DeBERTa-v3 (ETB)	75.31	83.71	69.35	−14.36
BART-large (ETB)	74.56	83.83	68.14	−15.69
$\mathcal{M}_{pair}$ (ETB)	70.04	82.82	59.91	−22.91
$\mathcal{M}_{con}$ (ETRACER)	**89.20**	**95.18**	**85.41**	**−9.77**

✳ 실험 결과_문장의 Attention 제한

System	S.F1	C.F1	Cit.F1	Δ
ETRACER (full)	**75.58**	**89.20**	**82.39**	—
Mask context cross-attn.	57.47	75.58	66.53	−15.86
Mask response cross-attn.	73.78	85.22	79.50	−2.89
Mask both	60.04	67.56	63.80	−18.59

- 당연하게도, 문장 전체의 맥락을 읽는 방법을 택하기 때문에 mask가 들어가면 성능이 감소한다
- 특히나 응답과 근거 문장 간 정보 전달을 차단했을 때 Citation F1이 가장 크게 감소한다

✳ 실험 결과_문장의 순서 변화

Condition	S.F1	C.F1	Cit.F1
$\mathcal{M}_{con}$ (original)	75.58	89.20	82.39
$\mathcal{M}_{con}$ (shuffled)	$70.42_{\pm 0.12}$	$70.35_{\pm 1.60}$	$70.39_{\pm 0.85}$
Δ	-5.16	-18.85	-12.00
$\mathcal{M}_{pair}$ (invariant)	72.31	70.04	71.18

- 지지 관계는 문장의 순서에 영향을 크게 받지 않는다.
- 모순 관계는 문장의 순서에 영향을 크게 받는다.
  - 함께 등장하는 문장의 순서와 위치에 의존을 많이 한다.

결론 및 한계점

✳ 결론

1. 근거를 연결하는 데에 있어 해당 모델의 우수성

- 응답과 근거 문장을 공동 인코딩하여 한 번의 순전파로 근거의 위치와 지지·모순 여부를 함께 식별한다
  - 한 번의 순전파로 검증 세분성, 검증 완전성을 확인할 수 있다
- 타 모델보다 연산 횟수를 줄이면서 문맥 정보를 반영한 결과를 처리할 수 있다

2. 암시적 모순 탐지에 있어 해당 모델의 우수성

- 동일한 학습 데이터를 사용했던 Mpair 방식에 비해 모순 근거를 찾아낸 F1-score 가 높게 측정되었다
- 특히 암시적 모순 F1은 59.91에서 85.41로 향상되어 개별 문장 쌍만으로 판단하기 어려운 모순에서 차이를 보인다

3. 문맥의 유무가 모델 성능에 영향을 준다

- 근거 문장 간 Attention을 차단할 경우 Citation F1이 82.39에서 66.53으로 감소하여 전체적으로 정답률이 떨어지게 된다
- 근거 순서를 섞을 경우에도 Contradict F1이 89.20에서 70.35로 감소하여 정답률이 떨어진다
- 해당 모델은 문장 사이의 문맥이 중요하게 작용하는 모델이다

✳ 한계

1. 자연 모순 데이터의 부족

- 기존 데이터셋에는 모순 데이터가 부족해서 연구진이 모순 데이터를 임의적으로 증강 했다
- 인위적으로 구성한 모순에서의 성능이 실제 여러 문장·문서에 걸친 모순에 대응 가능한지는 추가 검증이 필요하다

2. 긴 문맥 처리의 제약

- 인코더 모델인 ModernBERT의 입력 토큰 한도 때문에 문장이 길어지면 임의로 나눠야 한다
- 여러 윈도우를 사용할 경우 문장 전체를 학습할 수는 있지만, 서로 다른 윈도우 문장에는 접근할 수 없게 된다
- 해당 모델은 문장 간 정보가 차단될 경우 성능이 떨어지게 된다

Q&A

감사합니다.

논문 정보

eTracer: Polarity-Aware Evidence Grounding
https://aclanthology.org/2026.emnlp-main.314
Chu et al., EMNLP 2026