논문 세미나 발표

2026. 03. 12.

윤태건

Improving Neural Retrieval with Attribution-Guided Query Rewriting

Moncef Garouani
Moncef.Garouani@irit.fr
IRIT, UMR5505 CNRS
Université Toulouse Capitole
Toulouse, France

Josiane Mothe
Josiane.Mothe@irit.fr
IRIT, UMR5505 CNRS
UT2J, Université de Toulouse
Toulouse, France

Attribution-Guided Query Rewriting

1. 기존 Query Rewriting 방법의 한계

- LLM 기반 Query Rewriting은 사용자 쿼리를 더 유창하거나 확장된 형태로 변환 가능
- 그러나 대부분의 방법은 검색기의 피드백을 고려하지 않고
- 언어적 타당성 또는 의미적 유사성에만 초점
- 그 결과
  - 검색 성능 개선이 일관되지 않음
  - 불필요하거나 노이즈가 있는 용어가 추가될 수 있음
  - 오히려 검색 성능이 저하될 가능성 존재

2. Explainable IR 방법의 한계

- Explainable IR은 토큰 수준 분석 기법을 통해 쿼리 내 랭킹 결정에 영향을 주는 부분을 식별
- 대표적인 방법
  - **Gradient-based attribution**
  - **Perturbation-based attribution**
- 그러나 이러한 기법은 주로 사후 분석 용도로 사용됨
- 실제 검색 성능 개선에 직접 활용되지는 않음

Problem Formulation

- **문제 (Problem)**
  - 사용자 질의어 q가 신경망 검색 모델 R에 의해 처리될 때, 질의어의 표면적 표현 때문에 검색 성능이 저하될 수 있음
  - 질의어 내 토큰의 역할 차이
    - 일부 토큰 → **관련 문서 검색에 긍정적 기여**
    - 일부 토큰 → **가중치가 낮거나 관련 없는 문서로 검색을 유도**
- **목표 (Goal)**
  - 검색 모델 R을 수정하거나 재훈련하지 않고 개선된 질의어 q'를 생성하여 검색 성능 향상
  - 실제 시스템에서는 검색 모델을 변경하기 어려운 경우가 많기 때문에

    → **질의어 단계에서의 개선이 현실적인 접근 방법**
- **제약 조건 (Constraint)**
  - 재작성된 질의어는 사용자의 원래 의도를 유지해야 함
  - 동시에 다음 문제를 해결해야 함
    - **약한(weak) 질의어 토큰**
    - **모호한(ambiguous) 질의어 구성 요소**

Attribution-Guided Prompting for QueryRewriting

1. 상위 문서 검색

- 원본 질의어 q를 사용하여
- 기본 검색 모델 R로부터 상위 k개의 문서 검색

$$D_{top_q}$$

- 실험에서는 k = 5 사용

2. 토큰 어트리뷰션 계산

- **Integrated Gradients (IG)** 기법 사용
- 각 질의어 토큰 t_i가

  문서 $d \in D_{top_q}$의 관련성 점수 $s(q, d)$에 얼마나 기여하는지 계산

3. 어트리뷰션 점수 집계

- 상위 k개 문서에 대한 기여도를 평균하여

  각 토큰의 최종 어트리뷰션 점수 계산

$$\alpha_i = \frac{1}{k} \sum_{d \in D_{top_q}} IG(t_i, s(q, d))$$

- $IG(t_i, s(q, d))$:

  토큰 t_i가 문서 d의 관련성 점수에 기여하는 정도

4. 문제 토큰 식별

- α_i가 낮거나 음수인 토큰
  - 검색 점수에 약하게 기여
  - 관련 없는 문서로 검색을 유도할 가능성

-> 명확화 또는 모호성 제거 대상 토큰으로 활용

LLM-Based Query Reformulation

➢ LLM의 역할

- 어트리뷰션 분석은 어떤 토큰이 검색에 도움이 되거나 방해되는지 식별
- 그러나 다음 정보는 제공하지 못함
  - 토큰을 어떻게 수정해야 하는지
  - 예: 약어 확장, 의미 구체화, 모호성 해소

-> **역할: LLM을 재작성(Query Rewriting) 모듈로 활용**

- 검색기에서 얻은 토큰 수준 신호를 기반으로 질의어 재작성 수행

➢ Attribution Score 활용(Soft Guidance)

- 어트리뷰션 점수는 LLM에게 **Soft Guidance** 역할 수행
- 높은 점수 토큰 = 보존 또는 강조
- 낮거나 음수 점수 토큰 = 의미 명확화 / 구체화 / 모호성 제거

➢ LLM 입력(Input)

- **LLM에 전달되는 정보**
  - 원본 질의어: q
  - 질의어 토큰: $t_1, \ldots, t_n$
  - 토큰 어트리뷰션 점수: $\alpha_1, \ldots, \alpha_n$
- **중요한 점**
  - 어트리뷰션이 낮은 토큰을 제거하지 않음
- **이유**
- 낮은 점수는 토큰이 불필요하다는 의미가 아니라
- 검색기가 해당 토큰을 제대로 활용하지 못할 가능성을 의미

Prompt for attribution-guided query rewriting.

```
prompt = """You are given:
1) An original user query.
2) A list of query tokens with their attribution scores,
     where higher scores indicate a stronger positive
     contribution to retrieval effectiveness, and lower
     or negative scores indicate weak or misleading
     contributions.

Your task is to rewrite the query to improve retrieval
     effectiveness.

Guidelines:
- Preserve the original user intent.
- Do not remove important concepts.
- Tokens with high attribution scores should be
     preserved or emphasized.
- Tokens with low or negative attribution scores may be
     clarified, specified, or disambiguated.
- Avoid adding new concepts that are not implied by the
     original query.
- Produce a single rewritten query, concise and
     well-formed.

Original query: "{original_query}"
Token attributions: "{zip(tokens, attributions)}" """
```

1. 원본 사용자 질의 (Original Query)
2. 토큰별 기여도 점수 (Token Attribution Scores)

- 당신의 임무

- 원본 사용자 의도 보존
- 중요 개념 제거 금지
- 높은 기여도 토큰 보존 또는 강조
- 낮거나 음의 기여도 토큰 명확화/구체화/모호성 해소
- 원본 질의에 암시되지 않은 새로운 개념 추가 금지
- 간결하고 잘 구성된 단일 재작성 질의 생성

- Original query: "{original_query}“
- Token attributions: "{zip(tokens, attributions)}"

재작성 질의 처리 (Rewritten Query Processing)

재작성 질의 적용 과정

- LLM이 생성한 재작성 질의 q'를 동일한 신경망 검색기 R에 다시 입력

  -> 새로운 문서 순위 결과 생성

처리 방식

- **One-shot Closed-loop 방식**
  - 질의 재작성 → 검색 수행
  - 단일 단계로 수행
- **장점**
  - 연산 효율성 유지
  - 여러 번의 재작성 과정에서 발생할 수 있는 오류 전파(error propagation) 방지

Datasets, Retrievers, and LLM

1. 실험 데이터셋 (Datasets)

- **BEIR Benchmark**
  - 다양한 도메인의 **18개 정보검색(IR) 데이터셋**으로 구성된 **표준 신경 정보검색 벤치마크**
  - 다양한 검색 환경에서 모델 성능 평가 가능

- **실험에 사용된 주요 데이터셋**

1. SciFact (과학 분야)
- 문서: **5,000개**
- 쿼리: **300개**
- 평균 쿼리 길이: **12.37**

2. iQA-2018 (금융 분야)
- 문서: **57,000개**
- 쿼리: **648개**
- 평균 쿼리 길이: **10.77**

3. NFCorpus (생의학 분야)
- 문서: **3,600개**
- 쿼리: **323개**
- 평균 쿼리 길이: **3.30**

Datasets, Retrievers, and LLM(1)

2. Retriever Models

a) SPLADE

- 희소(sparse) + 의미 기반 표현을 결합한 검색 모델

b) TCT-ColBERT

- Contextualized Dense Retrieval 모델
- TCT-ColBERT 모델은 MS MARCO와 같은 대규모 QA 컬렉션에서 훈련되었지만, NFCorpus 및 FiQA-2018과 같은 이질적인 데이터셋에서는 zero-shot 일반화 성능이 제한적이기 때문에, 실험 전에 해당 데이터셋에 대해 추가로 fine-tuning을 수행했습니다.

3. LLM 설정

Query Rewriting 모델

- 사용된 LLM: **Mistral-7B-Instruct-v0.3**

Baselines And Evaluation Metrics(2)

Baselines

- Original Query (Org)
- LLM-only Rewriting (LLM)
- Top-Attribution Tokens Only (Tkn)

vs **Attribution-guided LLM rewriting (GLLM)**

Evaluation Metrics

- nDCG@k (Normalized Discounted Cumulative Gain)
- MAP@k (Mean Average Precision)
- Precision@k (P@k)
- 평가 지표의 k 값으로는 {1, 3, 5, 10, 100}이 사용되어 다양한 깊이에서의 검색 성능을 확인

Table 1: Retrieval effectiveness for original queries (Org), top-attribution tokens only (Tkn), LLM-only rewriting (LLM), andattribution-guided LLM rewriting (GLLM). Best results are in bold -blue.

Collection	k	SPLADE												TCT-ColBERT											
		NDCG				MAP				P				NDCG				MAP				P			
		Org	Tkn	LLM	GLLM	Org	Tkn	LLM	GLLM	Org	Tkn	LLM	GLLM	Org	Tkn	LLM	GLLM	Org	Tkn	LLM	GLLM	Org	Tkn	LLM	GLLM
Nfcorpus	@1	.433	.260	.394	**.461**	.055	.026	.046	**.059**	.442	.272	.402	**.476**	.444	.249	.373	**.476**	.054	.023	.045	**.056**	.464	.263	.393	**.495**
	@3	.392	.250	.357	**.398**	.095	.052	.081	**.099**	.368	.244	.332	**.375**	.416	.231	.354	**.428**	.095	.046	.075	**.098**	.401	.229	.345	**.407**
	@5	.366	.244	.327	**.390**	.108	.063	.092	**.113**	.312	.220	.277	**.339**	.407	.227	.348	**.414**	.117	.055	.093	**.120**	.367	.210	.321	**.369**
	@10	.335	.228	.298	**.364**	**.127**	.073	.107	.123	.244	.177	.218	**.248**	.390	.228	.339	**.394**	.149	.071	.120	**.152**	**.313**	.190	.279	**.313**
	@100	**.296**	.221	.268	.295	**.154**	.097	.131	.151	.074	.070	.066	**.079**	.399	.254	.353	**.401**	.227	.120	.189	**.229**	**.116**	.086	.106	.114
FiQA-2018	@1	.333	.214	.266	**.337**	.173	.113	.144	**.178**	.333	.214	.266	**.338**	.345	.158	.290	**.382**	.175	.080	.147	**.196**	.345	.158	.290	**.382**
	@3	.298	.209	.252	**.308**	.233	.158	.197	**.236**	.190	.136	.160	**.195**	.319	.157	.279	**.328**	.243	.117	.213	**.254**	.211	.105	.184	**.218**
	@5	.314	.221	.261	**.321**	.253	.173	.211	**.255**	.142	.103	.116	**.147**	.329	.169	.294	**.336**	.262	.130	.231	**.272**	**.154**	.084	.140	.150
	@10	.341	.244	.288	**.337**	.269	.186	.227	**.278**	.091	.069	.077	**.099**	.356	.192	.324	**.366**	.282	.142	.250	**.292**	**.099**	.058	.091	**.099**
	@100	**.398**	.306	.351	.389	**.284**	.201	.241	.280	**.015**	.013	.014	.013	.424	.265	.394	**.433**	.299	.157	.267	**.309**	.016	.013	.016	**0.17**
SciFact	@1	.560	.190	.546	**.603**	.530	.172	.524	**.572**	.560	.190	.546	**.603**	.479	.023	.456	**.526**	.479	.020	.442	**.502**	.503	.023	.456	**.526**
	@3	.632	.255	.612	**.654**	.604	.231	.588	**.630**	.247	.117	.235	**.250**	.558	.044	.559	**.602**	.558	.038	.527	**.573**	.234	.021	.224	**.235**
	@5	.653	.286	.635	**.676**	.619	.250	.604	**.646**	.162	.088	.156	**.165**	.567	.051	.571	**.615**	.567	.042	.535	**.582**	.150	.016	.142	**.151**
	@10	.677	.309	.660	**.696**	.631	.260	.616	**.656**	.090	.052	.087	**.090**	.577	.066	.598	**.636**	.577	.048	.548	**.592**	**.083**	.0133	.082	**.083**
	@100	.705	.369	.693	**.724**	.637	.271	.624	**.662**	**.010**	.007	**.010**	**.010**	.585	.122	.631	**.668**	.585	.058	.555	**.600**	**.010**	.004	**.010**	**.010**

Conclusion And Future Direction

정리:

- 이 연구는 **신경망 검색기(neural retriever)**의 **token-level attributions**를 활용하여 LLM기반의 질의어 재작성을 안내하는 방법을 제안

향후 연구 방향:

- **추가적인 신경망 검색기와 벤치마크 컬렉션에 대한 평가.**
- **반복적(iterative) 및 다단계(multi-stage) 재작성 전략 탐색.**
- **다국어 및 특정 도메인 검색 시나리오에서의 적용성 평가.**
- **대규모 언어 모델과의 상호작용을 고려한 대체 attribution 방법 연구.**

Thank you

Enhanced Retrieval Effectiveness through Selective Query Generation

Seyed Mohammad Hosseini
mohammad.hosseini75@gmail.com
Toronto Metropolitan University
Toronto, Ontario, Canada

Negar Arabzadeh
narabzad@uwaterloo.ca
University of Waterloo
Waterloo, Ontario, Canada

Morteza Zihayat
mzihayat@torontomu.ca
Toronto Metropolitan University
Toronto, Ontario, Canada

Ebrahim Bagheri
bagheri@torontomu.ca
Toronto Metropolitan University
Toronto, Ontario, Canada

CIKM

Introduction

1. 쿼리 변형 생성 (Query Variant Generation)

- 사전 훈련된 **Transformer 모델** 활용
- 초기 검색 단계에서 획득한 **의사 관련 문서(pseudo-relevant documents)** 기반 다양한 **잠재적 대체 쿼리(candidate queries)** 생성

2. 최적 쿼리 선택 (Optimal Query Selection)

- 생성된 변형된 커리들 중 최적 쿼리 선택
- **Cross-Encoder 모델**을 활용하여 **검색 효과** 예측
- 가장 성능이 높은 **target query** 선택

➢ 핵심 목표

- 원래 쿼리가 요구하는 정보를 유지하면서 검색 성능을 향상시키는 최적의 쿼리 생성

연구 목표 (Research Objective)

- **목표**
  - 원래 질의 **q**를 **target query variant** $\hat{q}_t$으로 **재구성하는 것**

- **표적 질의 변형이 만족해야 할 조건**

1. **동일한 정보 요구 유지**
   - $\hat{q}_t$는 원래 질의 q와 동일한 정보 요구(information need)를 가져야 함
   - 즉, 두 질의는 의미적으로 동일해야 함
   - $I_q = I_{\hat{q}_t}$
2. **검색 효율성 향상**
   - $\hat{q}_t$는 검색 시스템이 더 잘 처리할 수 있는 형태
   - 결과적으로 검색 성능(retrieval effectiveness) 향상

수학적 목표 (Mathematical Objective)

본 연구는 다음 조건을 만족하는 **최적의 질의 변형** $\hat{q}_t$를 찾는 것을 목표로 함

$$\boldsymbol{\mu}\left(\hat{\boldsymbol{q}}_{\boldsymbol{t}}, \boldsymbol{D}_{\hat{\boldsymbol{q}}_{\boldsymbol{t}}} \mid \boldsymbol{R}_{\boldsymbol{q}}\right) > \boldsymbol{\mu}\left(\boldsymbol{q}, \boldsymbol{D}_{\boldsymbol{q}} \mid \boldsymbol{R}_{\boldsymbol{q}}\right)$$

- 좌변: 재구성된 질의 $\hat{q}_t$사용 시 검색 성능
- 우변: 원래 질의 q사용 시 검색 성능

  ➞ 목표: 재구성된 질의가 **원래 질의보다 더 높은 검색 성능** 달성

Query Variants Generation

1. **초기 검색 (Initial Retrieval)**
   - 원본 쿼리 q를 사용하여 초기 검색 수행
   - 상위 k개의 문서 집합 D_q^k획득
   - 해당 문서들은 쿼리에 대한 잠재적 관련 문서(pseudo-relevant documents) 로 가정

2. **트랜스포머 기반 쿼리 생성**
   - 각 문서 $d \in D_q^k$에 대해 문서→쿼리 변환 함수 T적용
   - 동일 문서에 대해 m번 반복 적용 $T^m(d)$
   - 비결정적(non-deterministic) 생성 방식 → 다양한 쿼리 변형 생성 가능

3. **쿼리 변형 집합 구성**
   - 생성된 모든 쿼리를 모아 쿼리 변형 집합 형성

$$\hat{Q}_q^{k,m} = \{T^m(d) \mid d \in D_q^k\}$$

- $\hat{Q}_q^{k,m}$: 원본 쿼리 q에서 생성된 쿼리 변형 집합
- D_q^k: 초기 검색으로 얻은 상위 k개 문서
- $T^m(d)$: 문서 d에 Transformer를 m번 적용하여 생성된 쿼리

Query Variant Classification (1)

Query Variant Classification의 필요성 (Noise Problem)

- **문제:** 생성된 모든 쿼리 변형이 유효한 쿼리가 되는 것은 아님 → **노이즈(noise) 쿼리 존재 가능**

R1. 관련성 낮은 문서 기반 생성 문제

- 초기 검색에서 얻은 상위 문서 집합 D_q가 항상 완전히 관련된 문서라고 보장할 수 없음
- 이 문서들로 생성된 쿼리 변형 $\hat{q}_t$은 원본 쿼리의 정보 요구(I_q) 를 정확히 반영하지 못할 수 있음

$$I_q \neq I_{\hat{q}_t}$$

➡ 원본 쿼리와 의미가 다른 쿼리 변형 생성 가능

R2. 변환 함수 T의 불완전성

- 문서를 쿼리로 변환하는 Transformer 함수 T는 완벽하지 않음
- 관련 문서에서 생성되더라도
  - 의미적 일관성 유지 실패 가능
  - 검색 성능 향상 보장 불가

Query Variant Classification (2)

- **분류기 레이블 정의**

$$\phi(q, \hat{q}) = \begin{cases} 1 & if\ \mu(\hat{q}, D_{\hat{q}} \mid R_q) > \mu(q, D_q \mid R_q) \wedge I_q = I_{\hat{q}} \\ 0 & otherwise \end{cases}$$

- **긍정 샘플:** $\phi(q, \hat{q}) = 1$

  다음 두 조건을 모두 만족하는 경우

  1. **검색 성능 향상**

$$\mu(\hat{q}, D_{\hat{q}} \mid R_q) > \mu(q, D_q \mid R_q)$$

  2. **동일한 정보 요구 유지**

$$I_q = I_{\hat{q}}$$

- **부정 샘플:** $\phi(q, \hat{q}) = 0$
  - 위 조건을 만족하지 못하는 모든 경우

Query Variant Classification (3)

$$\widehat{\boldsymbol{\phi}}(\boldsymbol{q}, \hat{\boldsymbol{q}}) = \boldsymbol{\mathcal{F}}(\boldsymbol{q} \oplus \boldsymbol{d}_{\boldsymbol{q}}^{\boldsymbol{1}} \oplus \boldsymbol{S}(\boldsymbol{q}, \boldsymbol{d}_{\boldsymbol{q}}^{\boldsymbol{1}}), \hat{\boldsymbol{q}} \oplus \boldsymbol{d}_{\hat{\boldsymbol{q}}}^{\boldsymbol{1}} \oplus \boldsymbol{S}(\hat{\boldsymbol{q}}, \boldsymbol{d}_{\hat{\boldsymbol{q}}}^{\boldsymbol{1}}))$$

- **모델**
  - Cross-Encoder 네트워크 F사용
  - 대조 학습(contrastive learning) 방식으로 훈련
- **입력 구성**
  - 원본 쿼리 정보와 쿼리 변형 정보를 각각 구성한 뒤 결합하여 입력

a) 원본 쿼리 정보

$$q \oplus d_1^q \oplus S(q, d_1^q)$$

b) 쿼리 변형 정보

$$\hat{q} \oplus d_1^{\hat{q}} \oplus S(\hat{q}, d_1^{\hat{q}})$$

- d_1^q: 원본 쿼리 q로 검색된 첫 번째 문서
- $d_1^{\hat{q}}$: 쿼리 변형 $\hat{q}$로 검색된 첫 번째 문서
- $S(\cdot)$: 쿼리–문서 간 관련성 점수
- $\oplus$: 정보 결합(concatenation)

Query Variant Classification (4)

▪ 출력 및 손실 함수

▪ 출력

- Cross-Encoder F는 선형 레이어를 거쳐 스칼라 값 $\hat{\phi}(q, \hat{q})$출력

▪ 학습 방식

- Binary Cross Entropy (BCE) Loss <손실함수>를 사용하여 훈련

$$L = -\frac{1}{N}\sum_{i=1}^{N}[\phi(q_i, \hat{q}_i)\log\left(\hat{\phi}(q_i, \hat{q}_i)\right) + \left(1 - \phi(q_i, \hat{q}_i)\right)\log\left(1 - \hat{\phi}(q_i, \hat{q}_i)\right)]$$

Datasets and Retrievers(1)

- **Datasets**
  - **MS MARCO Passage Collection V1**
    - 대규모 패시지 검색(passage retrieval) 데이터셋
    - 쿼리당 관련 문서 수가 적은 sparse label 구조
    - 특히 어려운 쿼리(difficult queries) 환경에서 검색 성능 평가에 적합
  - **TREC Deep Learning Track (DL-19, DL-20)**
    - MS MARCO 기반 평가 데이터셋
    - 쿼리당 더 많은 관련 문서가 주석된 deep label 구조
    - 검색 시스템의 정밀한 성능 평가에 사용

Datasets and Retrievers(2)

Retrievers

- 제안한 방법의 일반화 성능과 견고성을 검증하기 위해 다양한 유형의 검색기(Retrievers) 에 대해 실험 수행

1. Sparse Retriever

- **BM25:** 전통적인 Bag-of-Words 기반 검색 모델

2. Dense Retrievers

- **Dual-Encoder (DE)**
  - 쿼리와 문서를 임베딩 공간에서 비교하는 기본 구조
- **Dual-Encoder with Nearest Neighbor Negatives (DE-NN)**
  - ANCE 기법 사용
  - 학습 과정에서 hard negative 샘플 활용
  - 검색 성능 향상
- **Dual-Encoder with Late Interaction (DE-LI)**
  - ColBERT 모델 기반
  - 쿼리 토큰과 문서 토큰 간 **세밀한 상호작용(late interaction)** 고려

Transformer Model & Training Set

Query Generation Model

- Transformer Model: **docT5query** 모델 사용

Training Set Construction

Query Variant 생성 방법

- 각 원본 쿼리 q에 대해 100개의 대안 쿼리 생성

생성 과정

1) 초기 검색 수행

- 상위 k = 10개 문서 검색

2) Transformer 적용

- 각 문서에 대해 m = 10번 query 생성

- 총 생성 쿼리 수: $10\ documents \times 10\ queries = 100$

Classification Model & Hyperparameters

Classification Model

- BERT-based-uncased가 cross-encoder를 통해 파인튜닝
- Sentence Transformers 라이브러리 사용

Hyperparameters

Parameter	Value
Learning rate	2e-5
Warm-up	전체 학습 단계의 10%
Batch size	16
Epoch	1

Improvements on hard queries of the MS MARCO Dev set

Architecture	Number of Queries	MRR@10
Sparse	111 out of 4,224	0.165
DE-NN	321 out of 2,908	0.261
DE-LI	293 out of 2,834	0.217
DE	319 out of 3,011	0.239

Table: Improvements on hard queries of the MS MARCO Dev set.

Conclusion

- **정리:**
  - 본 연구는 쿼리 재구성 프레임워크가 다양한 검색 시스템에서 검색 효율성을 크게 향상시킬 수 있음을 입증했으며, 특히 기존 모델들이 어려움을 겪는 쿼리에 대한 해결 능력을 보여줌

Thank you