생성형 검색을 위한 Prompt 경량화 - 보고서

Outline

🤖 LLM에 정답의 근거가 될 만한 Context를 제공하여 **LLM Hallucination 문제를 개선**하려는 프로젝트입니다.

이때, Context를 추출하는 Retrieval Model의 성능을 높임으로써, **LLM에 전달하는 Context의 크기를 줄이고 더 나은 답변을 얻는 방법(Prompt 경량화)**에 대해 연구했습니다.

Team Role

이름	역할
전민수	Query-side Fine-tuning Loss 연구 및 구현, Baseline Code 수정
조민우	논문 리딩 및 아이디어 탐색, Retrieval 알고리즘 수정 및 실험
조재관	Query Expansion 및 Knowledge Distillation 훈련
진정민	논문 탐색 및 아이디어 구현, Query Expansion 및 Query-side Fine-tuning 실험 진행
홍지호	Query-side Fine-tuning, Retrieval 알고리즘 수정 및 실험

Skill

- Python, Pytorch, HuggingFace, LangChain, Gradio, Wandb

Data

Preprocessed Natural Questions Dataset

- DensePhrases 논문에서 사용된 Dataset
- **Natural Questions Dataset**을 **재가공**하여 만들어짐

💡Natural Questions Dataset: 구글에 입력된 실제 Query를 이용해 만든 Dataset

Data 예제

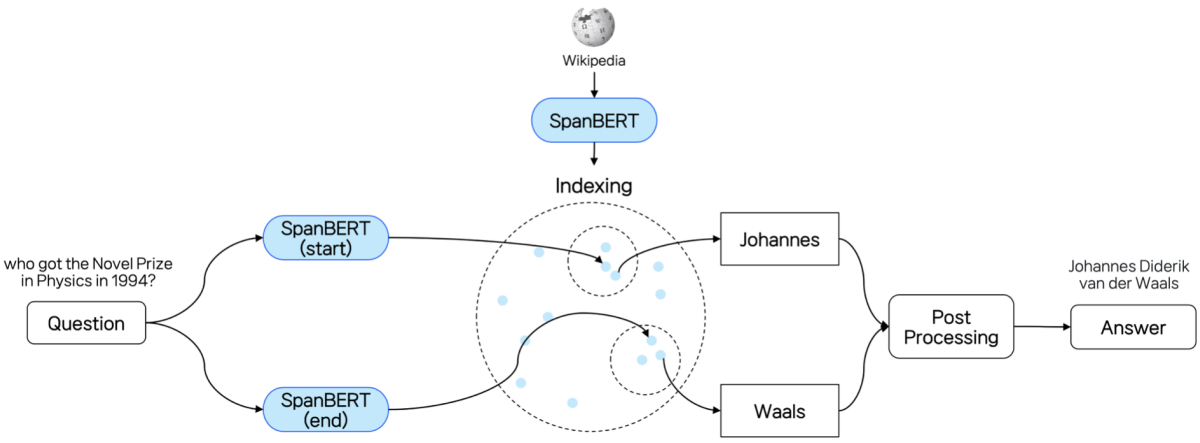

```
{
  "id": "train_13859",
  "question": "when was van halen's first album released",
  "answers": ["February 10 , 1978"]
}
```


Data 개수

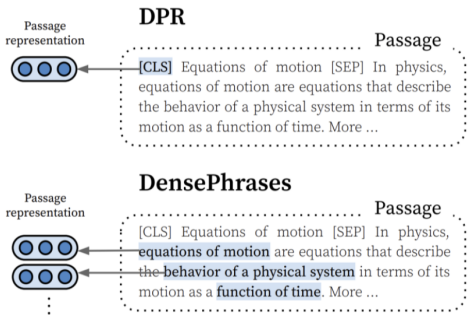
	Train	Validation	Test
#	79,168	8,757	3,610

Model

DensePhrases



- Open Domain Question Answering 모델
- 일반적으로 사용되는 DPR의 경우 [CLS] 토큰만을 반환하여 Fine-grained Retrieval 불가능
- DensePhrases의 경우 Query에 대해 **Phrase, Sentence, Passage, Document로 모두 반환 가능**



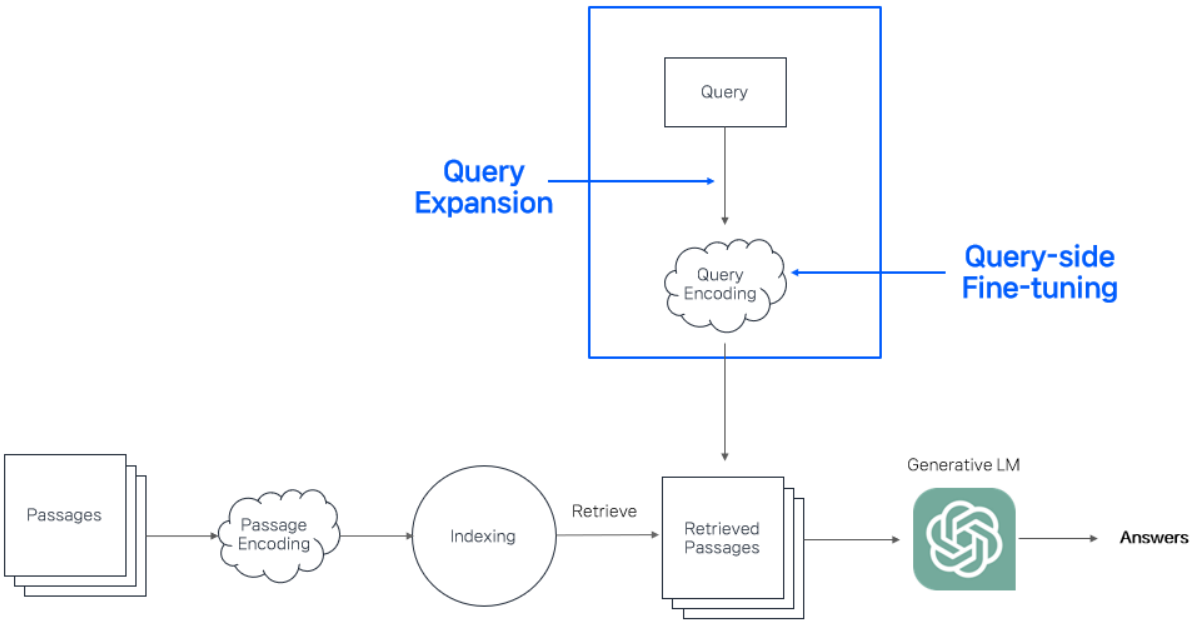
DensePhrases 구조

- Query를 사전학습모델 Spanbert를 사용해 Start / End 벡터로 Embedding
- Embedding 된 Query Vector를 Indexing 된 문서 Vector와 내적하여 최댓값을 추출하는 **MIPS 방식으로 검색 진행**
- 검색을 통해 추출된 Phrase를 후처리하여 원하는 Context를 추출

Experiment

모델 구조

- Phrase Retrieval Model + Reader Model 구조의 검색 모델



실험 방향

1) Loss Function

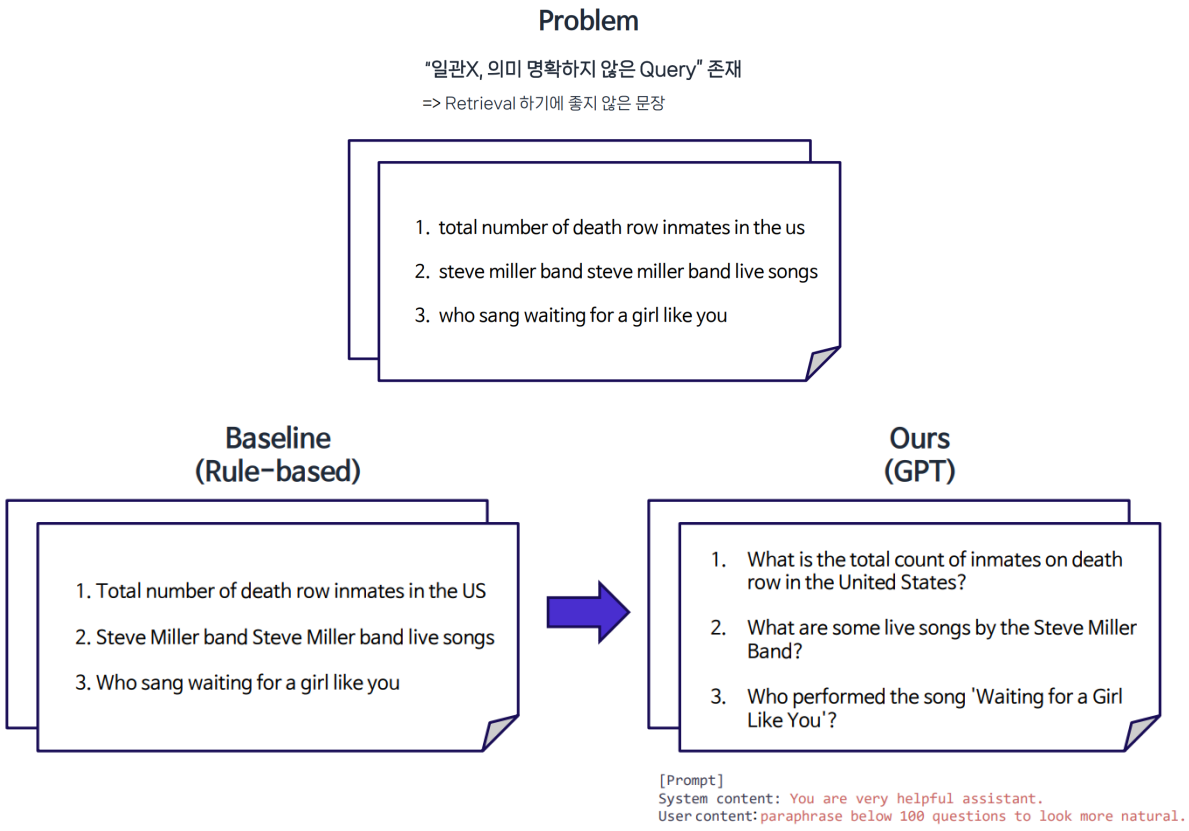
💡 기존

- Loss_phrase: 추출된 Phrase가 **정답 텍스트와 일치할 경우** True Label로 간주
- Loss_document: 추출된 Phrase가 **Golden Document 내에 있는 경우** True Label로 간주

변경

- Loss_sentence: 추출된 Phrase가 **포함된 Sentence 안에 정답이 있는 경우** True Label로 간주
- Loss_passage: 추출된 Phrase가 **포함된 Passage 안에 정답이 있는 경우** True Label로 간주

2) Query Expansion



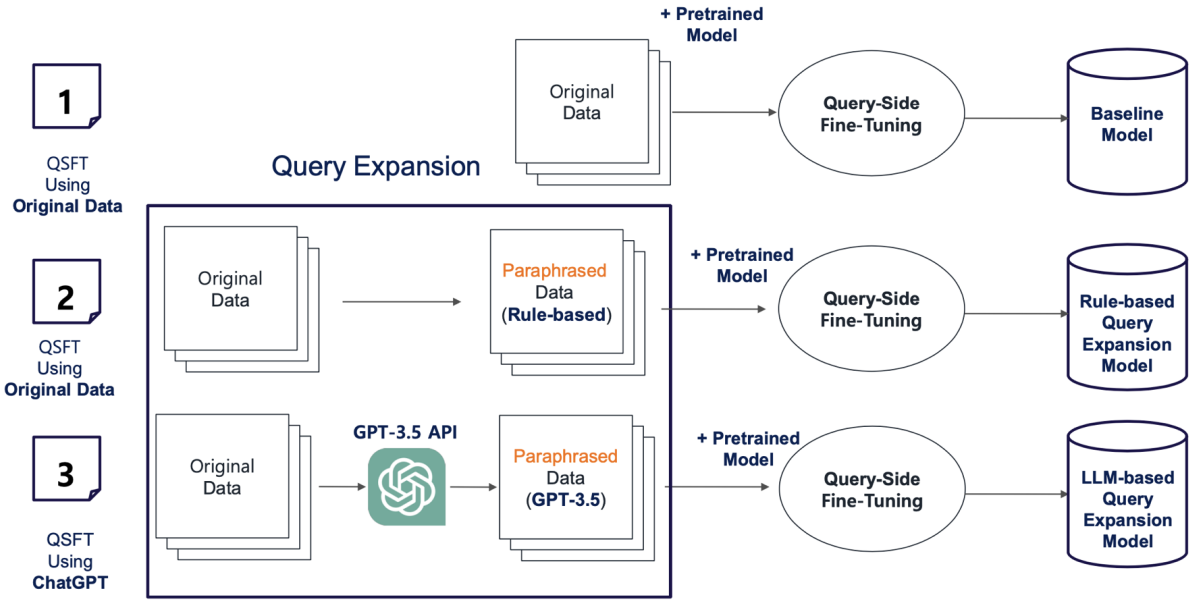
기존

- Origin: Query Expansion을 적용하지 않음
- Rule-based**:** 사전에 저장된 Vocab에 기반한 대소문자 변환

변경

- **GPT-3.5**: GPT-3.5 API를 활용한 Query Expansion
- **T5**: GPT-3.5를 활용하여 만들어진 Data를 이용해 Query Expansion 학습

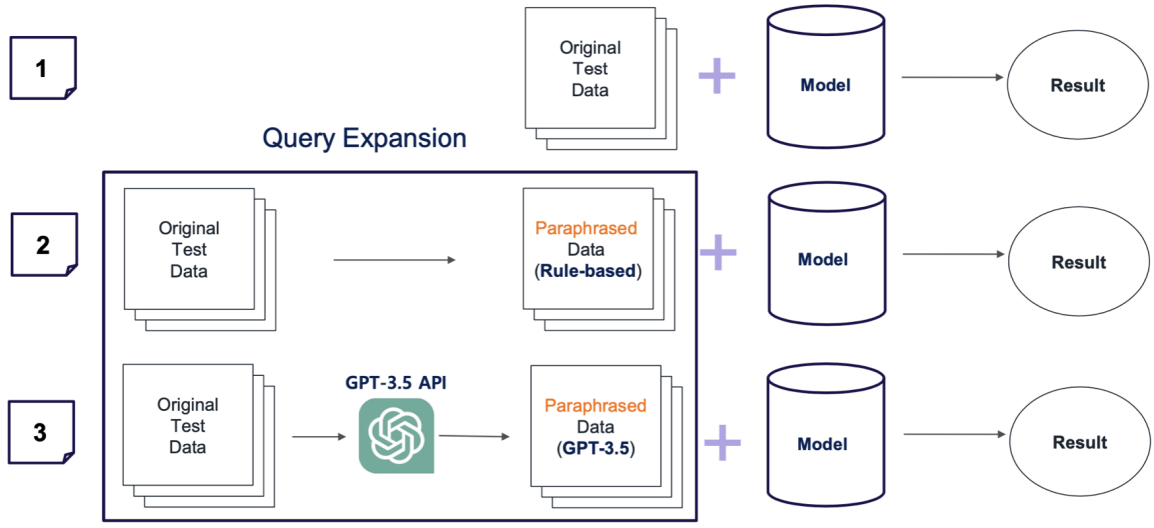
A. Train



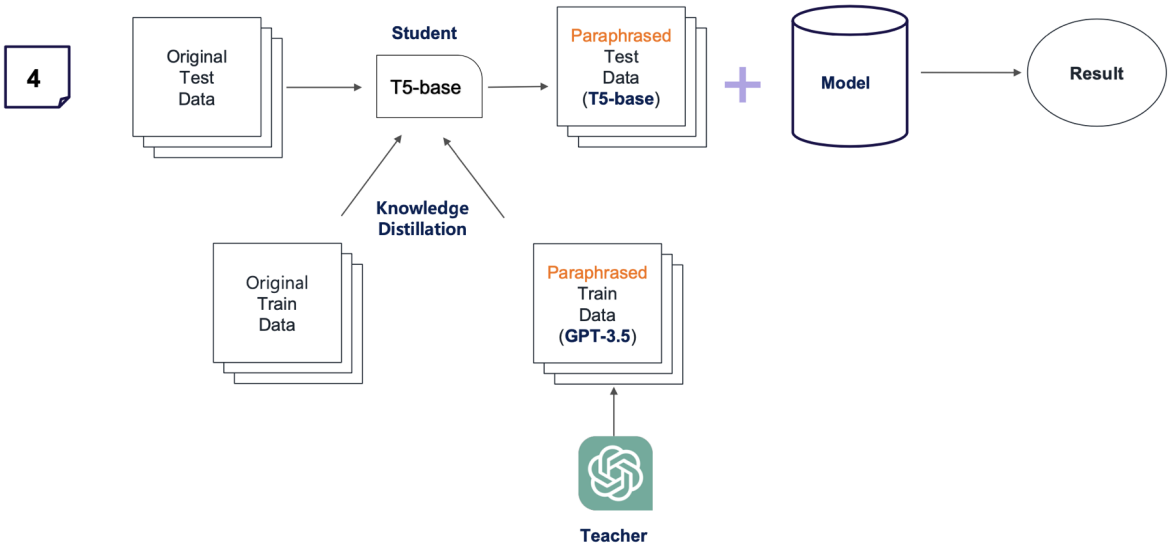
- Query-side Fine-tuning 과정에서 원본 훈련 Data와 이를 Query Expansion 한 Data를 사용해 훈련
- Query Expansion: Rule-based와 GPT-3.5를 활용하여 데이터를 Paraphrasing 하여 학습

  → 총 3개의 인코더 모델을 생성

B. Inference



- 각 Encoder의 성능을 확인하기 위해 총 4가지의 Test Dataset 을 이용해 Inference 진행
- 원본 테스트 Data와 함께, 테스트 Data를 각각 Rule-based, GPT-3.5, T5 모델을 이용하여 Query Expansion 한 후 성능을 측정



- 이때, GPT의 비용 문제를 해결하기 위해 Knowledge Distillation을 통해 T5를 훈련하여 사용
  - Teacher Model: GPT-3.5
  - Student Model: T5-base

Query Expansion 방법론 비교

	장점	단점
Rule-based	빠름	Vocab Dictionary 필요 확장성 낮음 대소문자만 변환
GPT-3.5	확장성 높음 문장 형식 수정 가능	매우 느림 비용
T5-base	**확장성 높음** **Fine-tuning을 통한 새로운 Query 학습 가능** **문장 형식 수정 가능**	느림

3) Retrieval Algorithm Modification

- Passage의 길이가 길수록 해당 Passage 안에서 Query와 유사도가 높은(MIPS 값이 큰) Phrase가 발견될 확률이 높음
- BM25의 경우 아래 수식에 따라 Passage 길이에 대한 Penalty를 부여함

$$\sum_{i}^{n} IDF(q_i) \frac{f(q_i, D) \times (k1 + 1)}{f(q_i, D) + k1 \times (1 - b + b \times \frac{fieldLen}{avgFieldLen})}$$

- 동일한 수식을 사용하여 MIPS 값에 Passage 길이에 대한 Penalty를 부여하여 Retrieval 진행
- 20개의 Sentence를 Retrieval 하여 토이 실험을 진행해 본 결과 mAR가 낮아져서 이후 실험 중단
  - mAR: 0.4660 → 0.2688

Result

Query Expansion

Train: Rule-based Data

Inference: Origin / Rule-based / GPT-3.5/ T5 Data

Inference 단계에서 Query Expansion 여부에 따른 성능 평가

	mAR (3000 words)	AR (at 3000 words)
Origin	0.5428	0.5987
Rule-based	0.6348	0.6753
GPT-3.5	0.6380	0.6766
T5-BASE	0.6311	0.6705

훈련 Data

- Rule-based의 Query Expansion이 적용된 Data

추론 Data

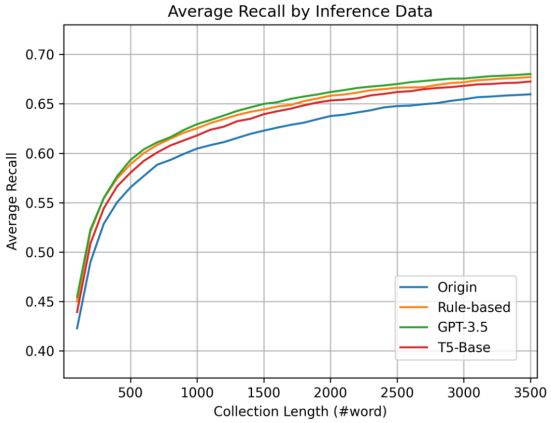
- Origin: Query Expansion이 적용되지 않은 Data
- Rule-based: 통계 기반으로 대소문자 변환
- GPT-3.5: GPT-3.5 모델을 활용한 Query Expansion
- T5-Base: T5-Base 모델을 활용한 Query Expansion

결과

- Query Expansion을 사용했을 경우 사용하지 않았을 경우보다 **성능이 약 17% 향상**
- Query Expansion 방법론 간 성능 차이는 거의 없었음
- 확장성과 비용을 고려하여 T5를 통한 Query Expansion을 사용하기로 결정

Train: Origin Data

Inference: Origin / Rule-based / GPT-3.5/ T5 Data



Inference 단계에서 Query Expansion 여부에 따른 성능 평가

	mAR (3000 words)	AR (at 3000 words)
Origin	0.6076	0.6546
Rule-based	0.6290	0.6719
GPT-3.5	0.6329	0.6756
T5-BASE	0.6229	0.6682

훈련 Data

- Origin: Query Expansion이 적용되지 않은 Data

추론 Data

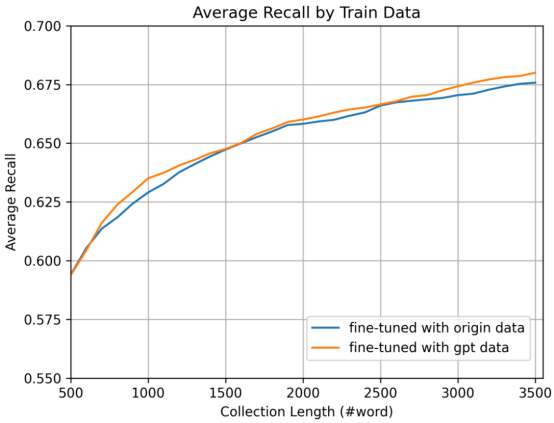
- Origin: Query Expansion이 적용되지 않은 Data
- Rule-based: 통계 기반으로 대소문자 변환
- GPT-3.5: GPT-3.5 모델을 활용한 Query Expansion
- T5-Base: T5 모델을 활용한 Query Expansion

결과

- 전체적인 성능은 Rule-based Query Expansion Data로 학습한 것보다 조금 떨어지지만,
  좀 더 Robust 한 결과를 보임

Train: Rule-based/GPT-3.5 Data

Inference: T5 Data



Training 단계에서 Query Expansion 여부에 따른 성능 평가

	mAR (3000 words)	AR (at 3000 words)
Rule-based	0.6311	0.6705
GPT-3.5	0.6335	0.6742

훈련 Data

- Origin: Query Expansion이 적용되지 않은 Data
- GPT-3.5: GPT-3.5 모델을 활용한 Query Expansion

추론 Data

- T5-Base: T5 모델을 활용한 Query Expansion

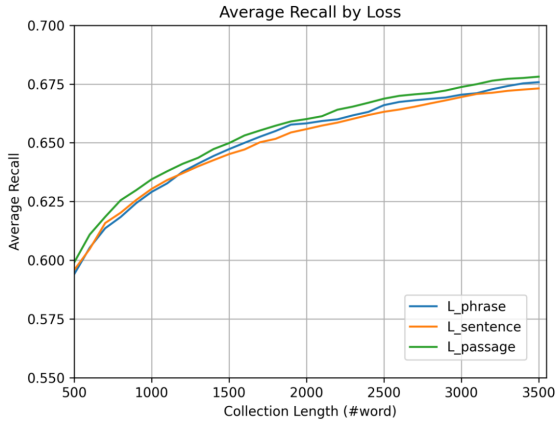
결과

- Query-side Fine-tuning Data에 따른 큰 성능 차이는 없었음

Query-side Fine-tuning Loss

Train: Rule-based Data

Inference: Rule-based Data



Loss 종류에 따른 성능 평가

	mAR (3000 words)	AR (at 3000 words)
L_phrase	0.6348	0.6753
L_sentence	0.6307	0.6700
L_passage	0.6349	0.6737

훈련 Data

- Rule-based: 통계 기반으로 대소문자 변환

추론 Data

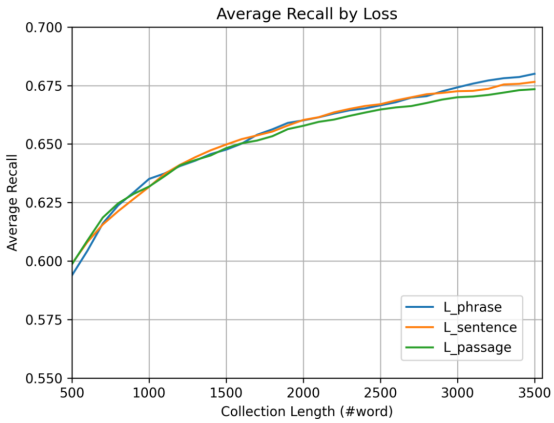
- Rule-based: 통계 기반으로 대소문자 변환

결과

- Loss_passage가 성능이 미세하게 좋았지만, 의미 있는 차이는 없었음

Train: GPT-3.5 Data

Inference: T5 Data



Query Expansion을 적용했을 때 Loss 종류에 따른 성능 평가

	mAR (3000 words)	AR (at 3000 words)
L_phrase	0.6335	0.6742
L_sentence	0.6337	0.6726
L_passage	0.6328	0.6700

훈련 Data

- GPT-3.5: GPT-3.5 모델을 활용한 Query Expansion

추론 Data

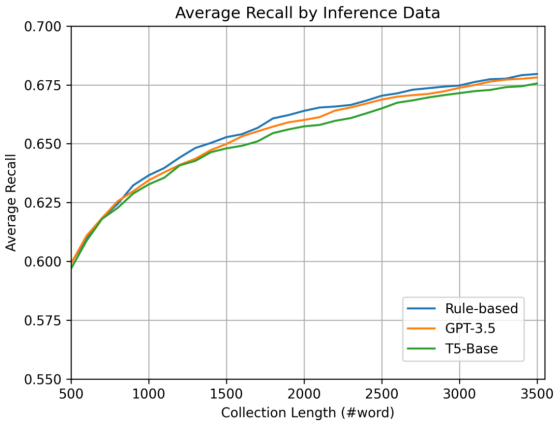
- T5-Base: T5 모델을 활용한 Query Expansion

결과

- Loss_sentence가 성능이 미세하게 좋았지만, 의미 있는 차이는 없었음

Train: Rule-based Data

Inference: Rule-based / GPT-3.5/ T5 Data



Query Expansion을 적용했을 때 Loss 종류에 따른 성능 평가

	mAR (3000 words)	AR (at 3000 words)
Rule-based	0.6369	0.6747
GPT-3.5	0.6349	0.6737
T5-BASE	0.6325	0.6715

> 💡 Auxiliary Loss
>
> 단순히 Loss를 Loss_phrase에서 Loss_sentence나 Loss_passage로 대체하지 않고, Loss_phrase와 Loss_passage의 합을 Loss로 설정하여 전체 Loss에 대해 학습

훈련 Data

- Rule-based: 통계 기반으로 대소문자 변환

추론 Data

- Rule-based: 통계 기반으로 대소문자 변환
- GPT-3.5: GPT-3.5 모델을 활용한 Query Expansion
- T5-Base: T5 모델을 활용한 Query Expansion

결과

- 기존 모델에 비해 미미한 성능 향상이 있었으나, 큰 차이를 발견하지 못함

Conclusion

의의

- LLM Hallucination 문제 해결 방법론 중 Context를 제공하는 방법의 **비용 효율성 향상**
- **Query Expansion만으로도 Retrieval 성능이 크게 향상**되는 것을 확인
- 큰 비용이 드는 LLM 대신 Query Expansion으로 T5를 활용하여 **비용 절감**
- 여러 산업 분야에 적용 가능
  - 특정 도메인 검색 서비스
  - 문서 관리 서비스
  - Chatbot 서비스

한계

- Query Expansion을 위해 T5를 추가하여 Inference 시간 증가
- 학습과 평가에 많은 시간이 소요되어 많은 실험을 진행할 수 없었음
- Retrieval-Read 단계에서 많은 시간이 소요되어 실시간 서비스에 사용되려면 경량화 작업 필요