

# Nota 기업 연계 프로젝트

NLP-06

팀명 : 어쩔돌멩이

팀원 : 김민호 김성은 김지현 서가은 홍영훈

# INDEX

## 1. Intro

1.1. 문제 정의 & 주제 제안

1.2. 팀 및 팀원 소개

1.3. Timeline

---

## 2. Dataset, Baseline model 소개

2.1. Dataset

2.2. Baseline model

---

## 3. Model & Research

3.1. QSFT

3.2. Multi Unit Retrieval

---

## 4. Result & Conclusion

---

## 5. Reference

---

# 1. Intro

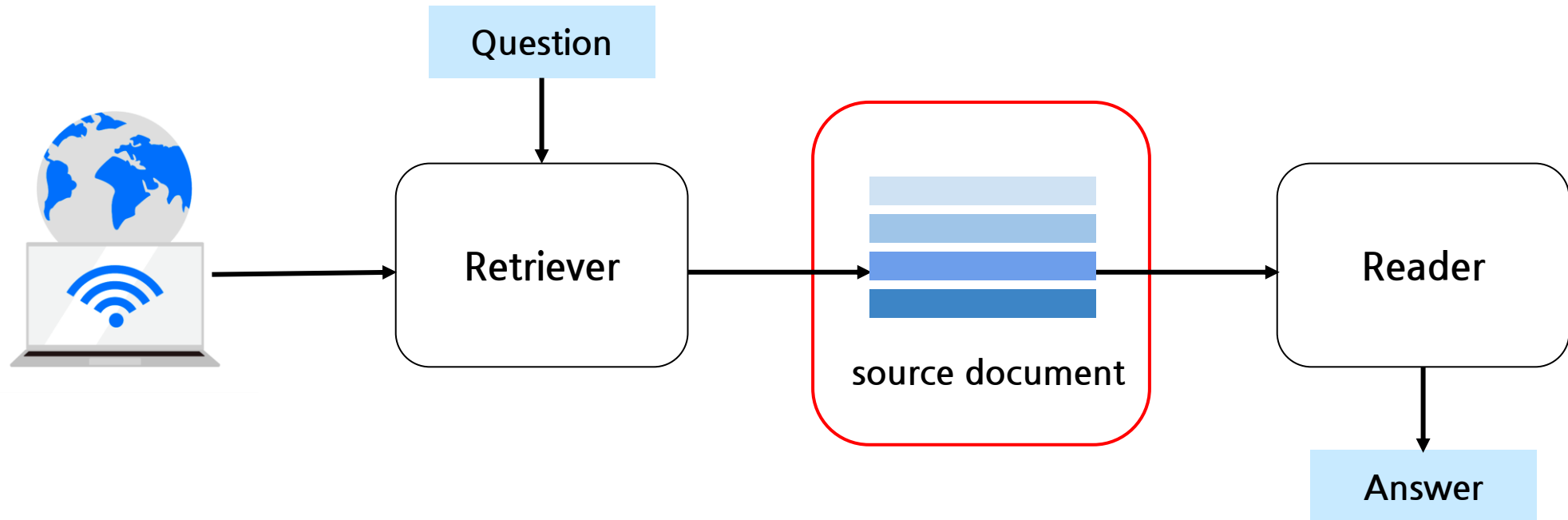
---

문제 정의 & 주제 제안

팀 및 팀원 소개

Timeline

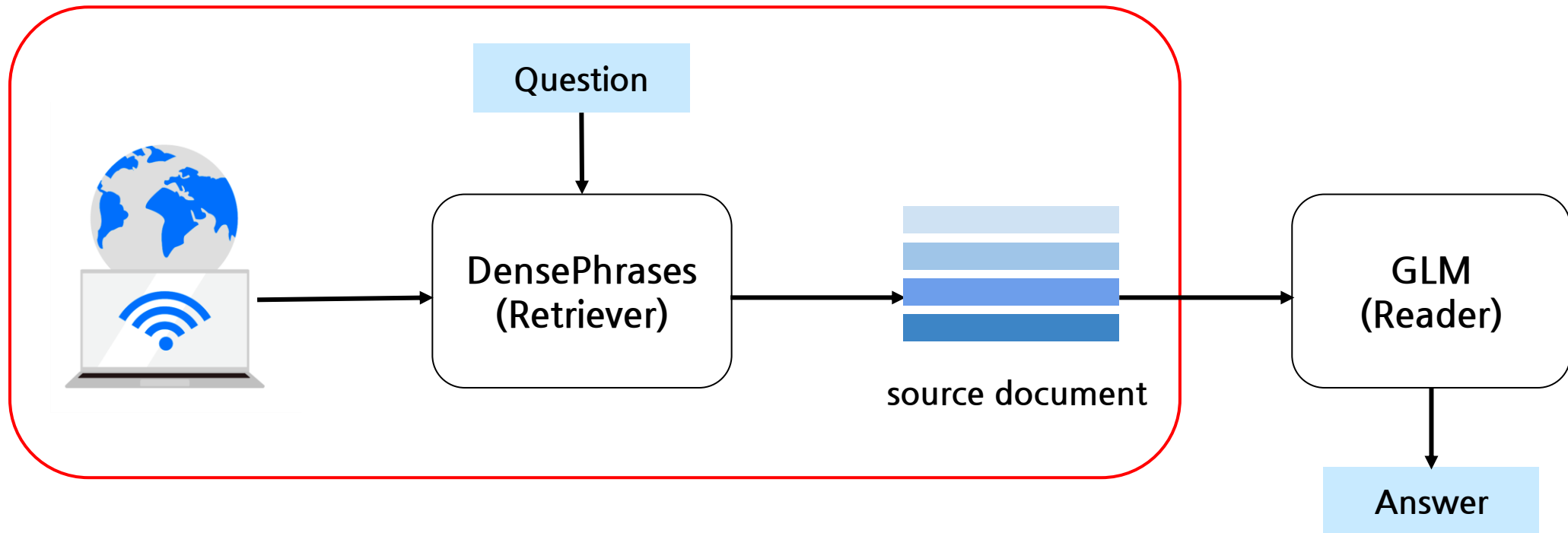
## 1.1. 문제 정의 & 주제 제안



ODQA(Open-domain Question Answering)

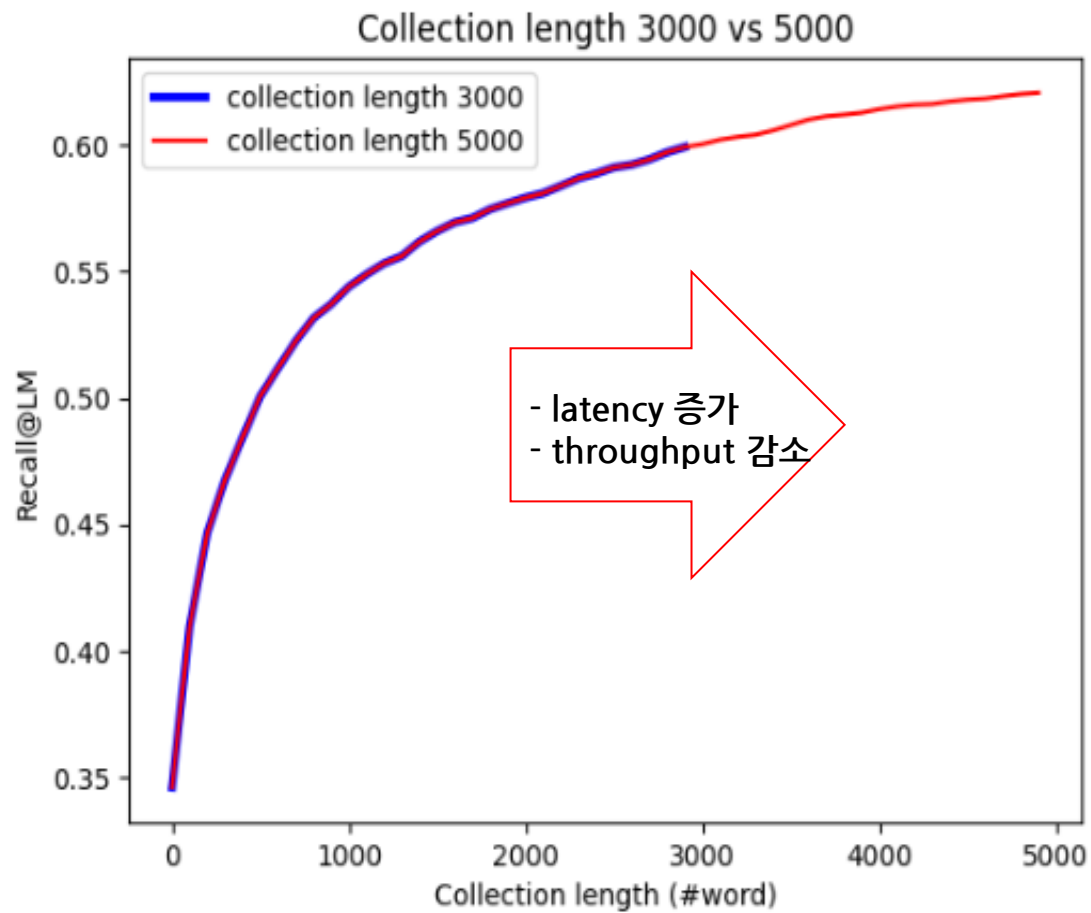
질의가 주어지면 주어진 질의에 답할 수 있는 문장들을 Retriever가 지식 베이스로부터 찾아 근거 문서를 구성하고, 구성된 근거 문서(source document)를 기반으로 Reader가 답변하는 시스템

# Project Overview



## 1.1. 문제 정의 & 주제 제안

### Recall@LM과 collection length(근거 문서의 길이)



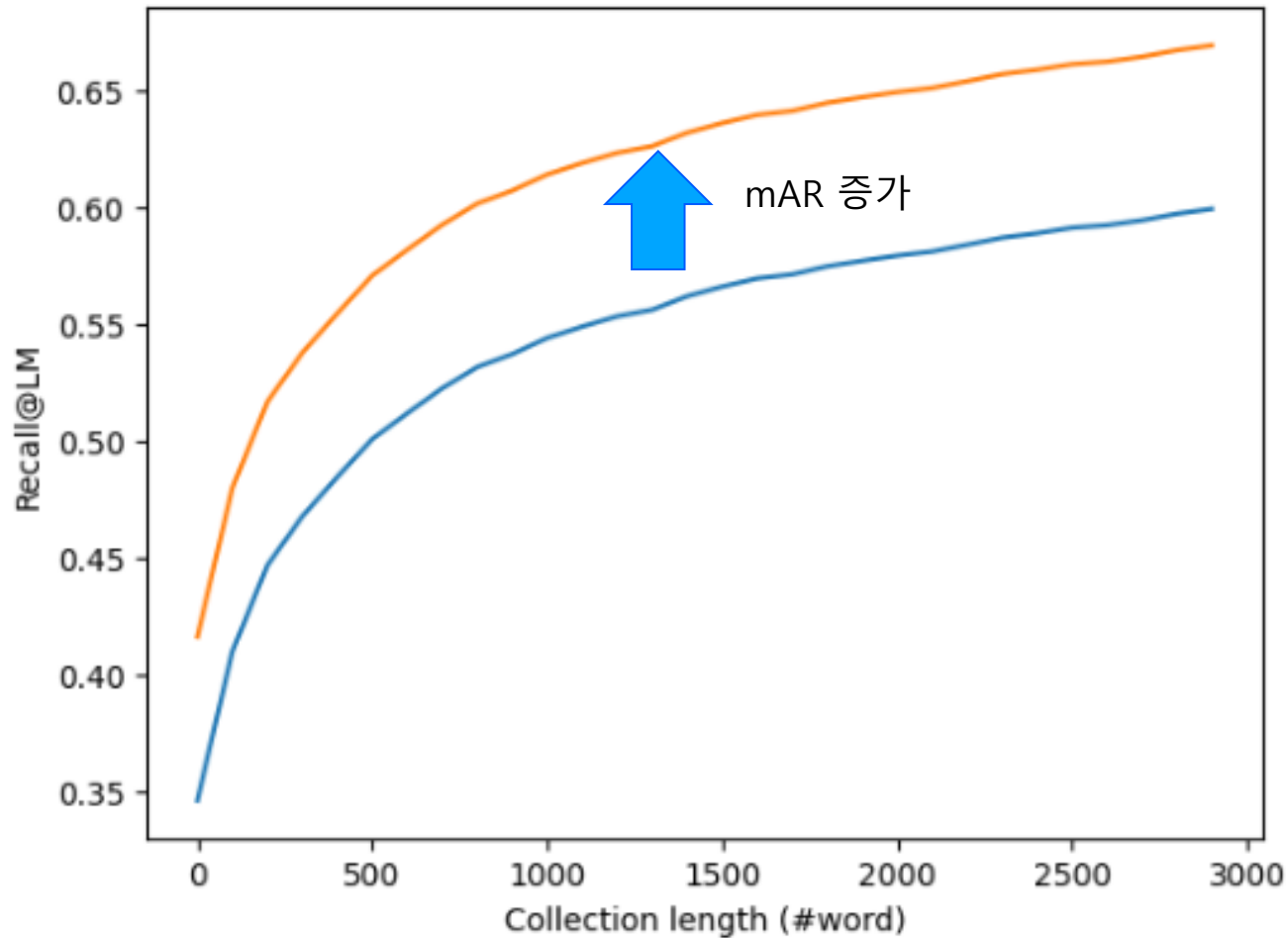
Reader에게 전달되는 근거 문서(prompt) 길이 증가

→ 정답 포함율(Recall@LM) 증가

→ **BUT** 정답 생성에 필요한 비용 역시 증가

## 1.1. 문제 정의 & 주제 제안

- 프로젝트 목표 : mAR 증가를 통한 검색 최적화



- **MAR(Mean Average Recall)**

- 근거 문서의 길이에 따른 평균 정답 포함율
- 정답 생성에 필요한 비용과 정답 포함율을 모두 고려한 지표

$$\text{mAR} = \frac{\sum_{i=1}^L \text{Recall@LM}(i)}{L}$$

$$\text{Recall@LM}(i) = \frac{\sum_{q \in Q} \text{recall}(q, i)}{n(Q)}$$

$$L = \text{max collection length}$$

- mAR을 높이기 위해서는 **같은 collection length**에서 **더 높은 Recall@LM**을 가져야 함
- 본 프로젝트에서는 mAR을 높이기 위해 두 가지 측면에서 접근
  - retrieval 알고리즘 수정
  - Query Side Fine-tuning 수행

## 1.2. 팀 및 팀원 소개



김민호

- Query loss의 단위 변경
- Loss의 구성 요소 추가



김성은

- Query loss의 단위 변경
- Dynamic retrieval



김지현

- Dataset 전처리
- Query loss의 단위 변경
- Knowledge Distillation



서가은

- Query loss의 단위 변경
- Dynamic retrieval

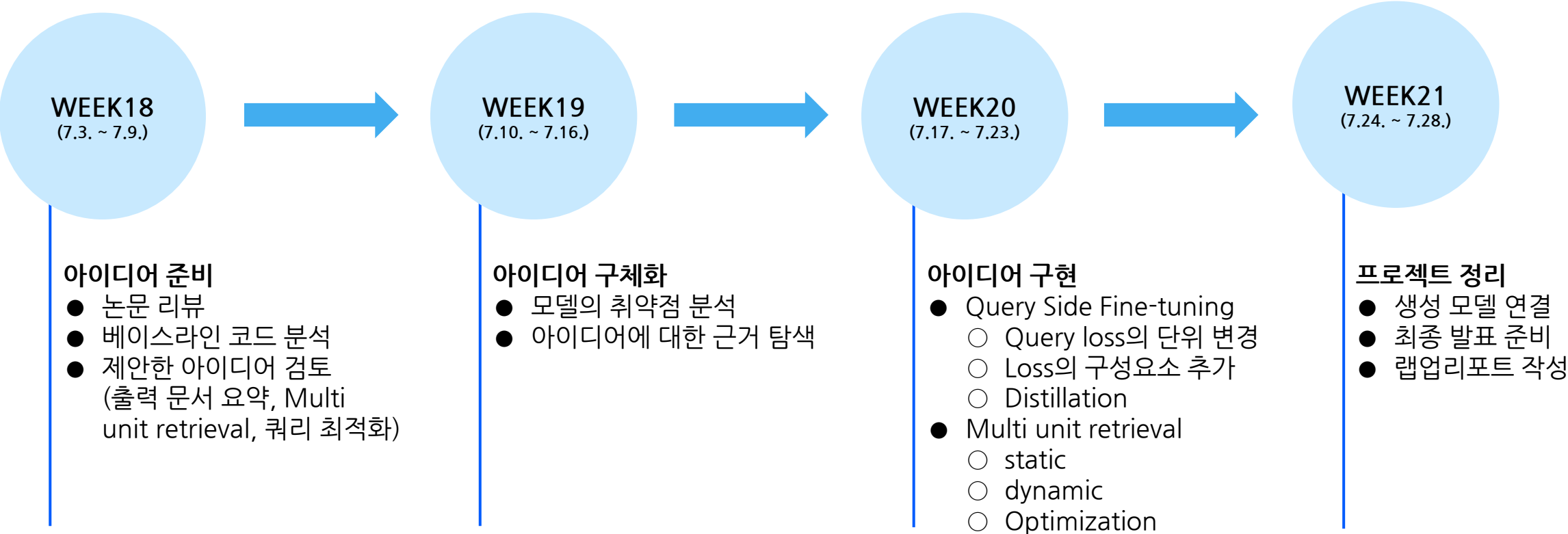


홍영훈

- Query loss의 단위 변경
- Static retrieval
- Optimization



# 1.3. Timeline



---

# 2. Dataset, Baseline model 소개

---

Dataset

Baseline model

## 2.1. Dataset

**NQ (Natural Question) Dataset** : QA (Question Answering) Task의 Benchmark Dataset 중 하나

- id : 질문 id
- question : 질문 텍스트
- titles : 질문의 답이 속한 문서 제목
- sentence : 정답들이 속한 문장 리스트
- context : 정답들이 속한 문단
- answers : 정답들로 구성된 리스트

id	2126085010748850659
question	when does season 5 of bates motel come out
titles	Bates Motel (season 5)
sentence	[' The fifth and final season of "Bates Motel" premiered on <b>February 20, 2017</b> , and concluded on April 24, 2017. ']
context	' The fifth and final season of "Bates Motel" premiered on February 20, 2017, and concluded on April 24, 2017. The season consisted of 10 episodes and aired on Mondays at 10 p.m. ET/PT on A&E. The series itself is described as a "contemporary prequel" to the 1960 film "Psycho", following the life of Norman Bates and his mother Norma prior to the events portrayed in the Hitchcock film. However, the final season of the series loosely adapts the plot of "Psycho". The series takes place in the fictional town of White Pine Bay, Oregon. '
answers	[February 20, 2017]

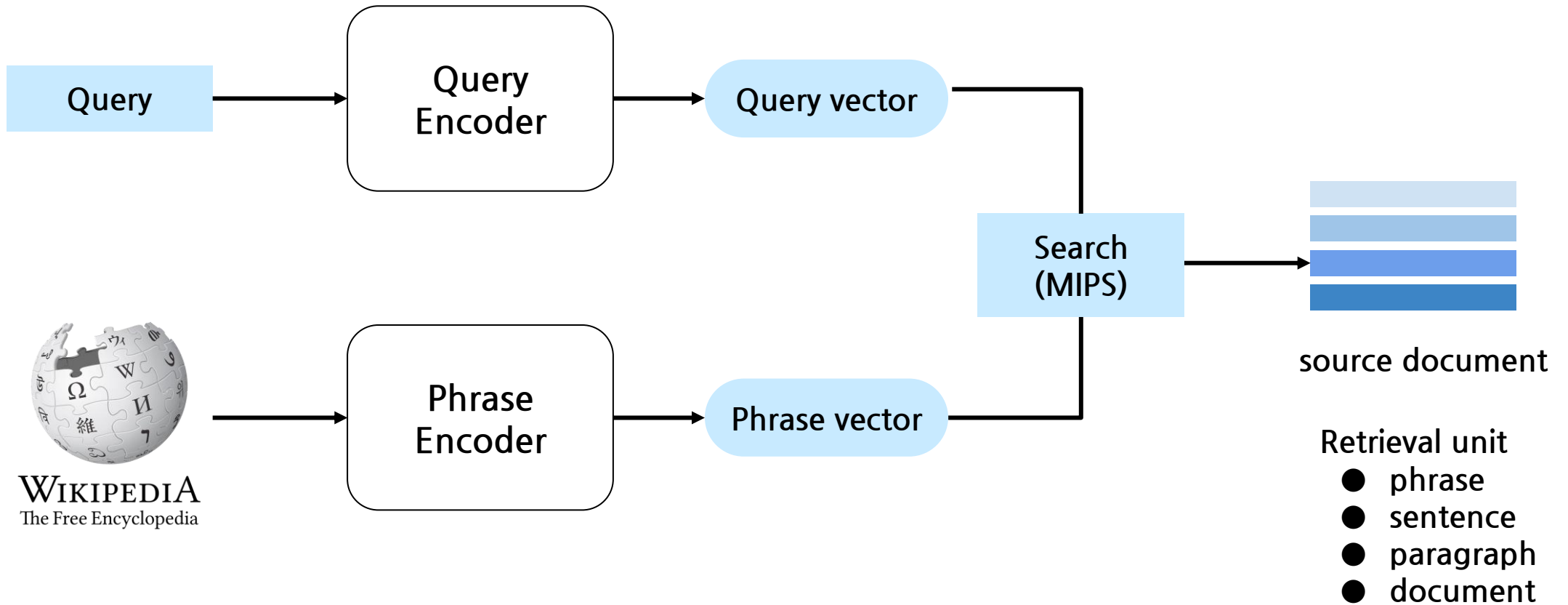
## 2.1. Dataset

**NQ (Natural Question) Dataset** : QA (Question Answering) Task의 Benchmark Dataset 중 하나

- id : 질문 id
- question : 질문 텍스트
- titles : 질문의 답이 속한 문서 제목
- sentence : 정답들이 속한 문장 리스트
- context : 정답들이 속한 문단
- answers : 정답들로 구성된 리스트

id	2126085010748850659
question	when does season 5 of bates motel come out
titles	Bates Motel (season 5)
sentence	[' The fifth and final season of "Bates Motel" premiered on <b>February 20, 2017</b> , and concluded on April 24, 2017. ']
context	' <b>The fifth and final season of "Bates Motel" premiered on February 20, 2017, and concluded on April 24, 2017.</b> The season consisted of 10 episodes and aired on Mondays at 10 p.m. ET/PT on A&E. The series itself is described as a "contemporary prequel" to the 1960 film "Psycho", following the life of Norman Bates and his mother Norma prior to the events portrayed in the Hitchcock film. However, the final season of the series loosely adapts the plot of "Psycho". The series takes place in the fictional town of White Pine Bay, Oregon. '
answers	[ <b>February 20, 2017</b> ]

# DensePhrases



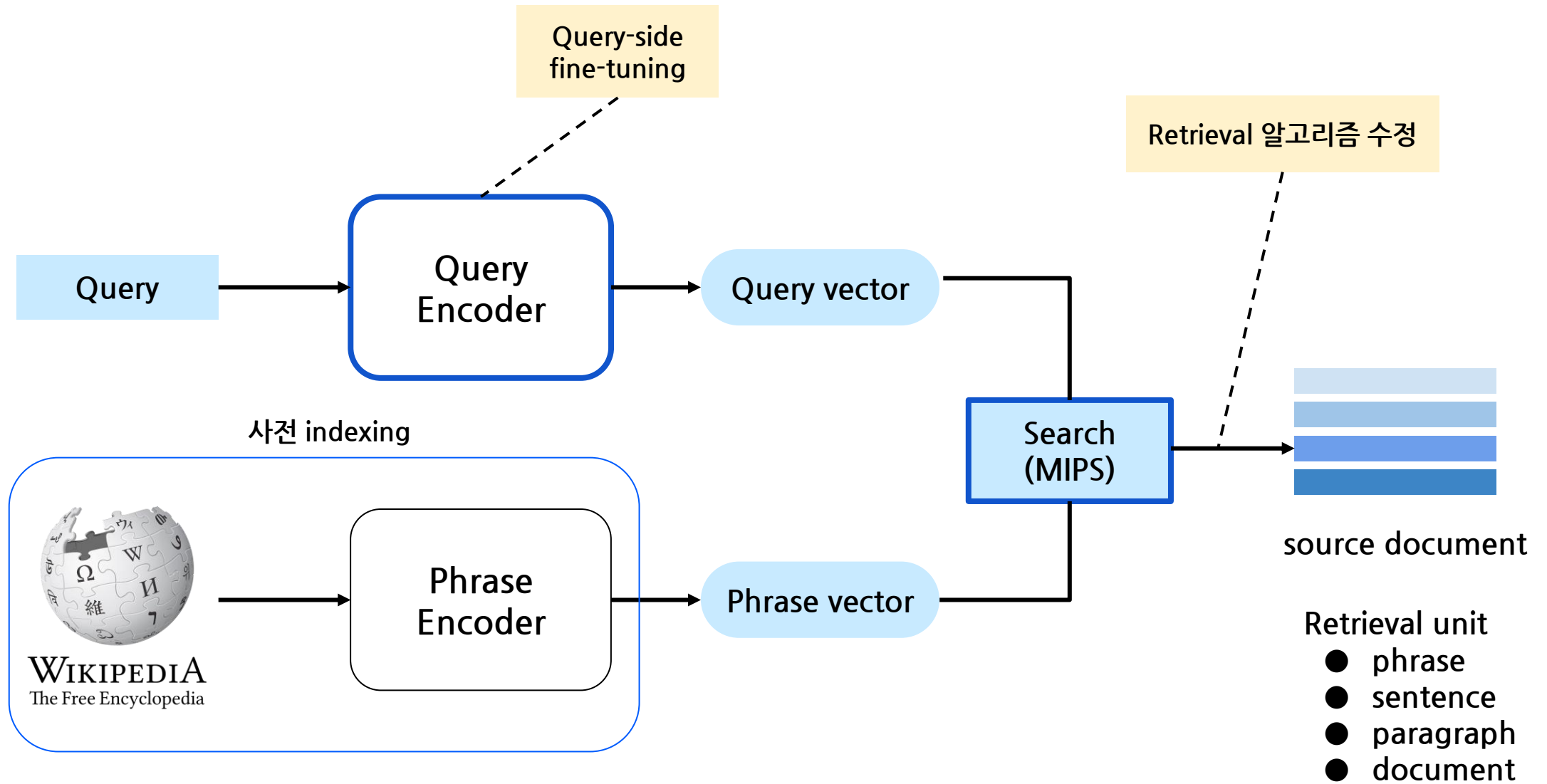
---

# 3. Model & Research

---

QSFT(Query Side Fine-tuning)

Multi Unit Retrieval



# 3.1. QSFT (Query Side Fine-tuning)

- Query loss의 단위 변경
- Loss의 구성 요소 추가
- Knowledge Distillation



## 3.1.1. Query loss의 단위 변경

QSFT 단계의 loss 수정

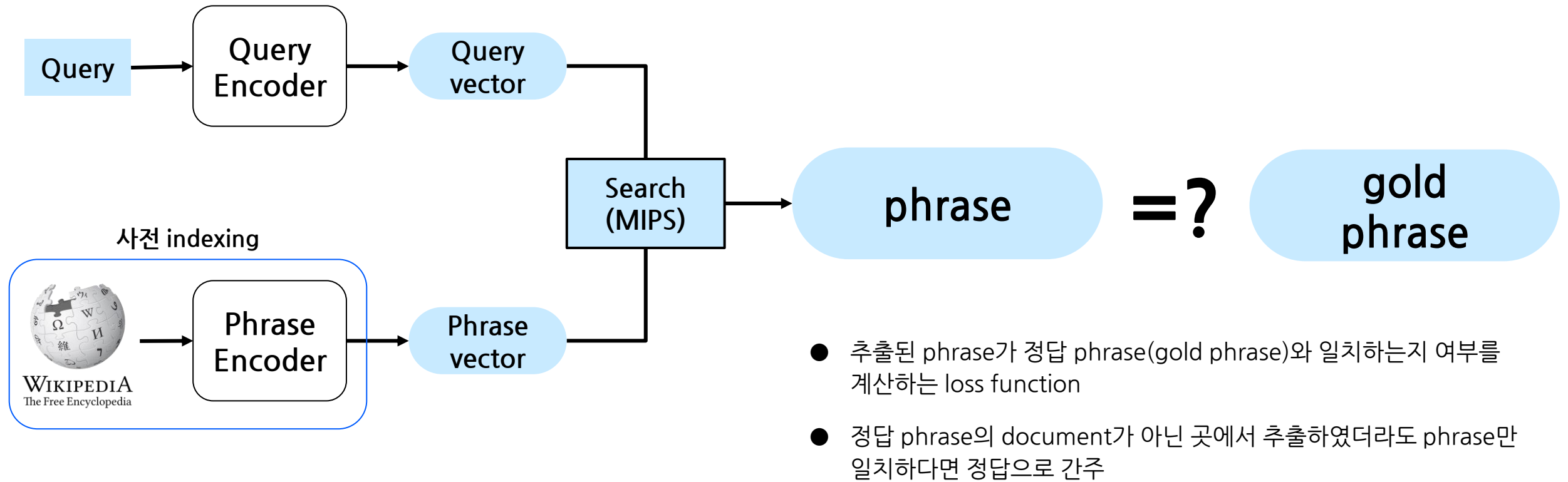
$$1. \mathcal{L}_{\text{query}} = -\log \frac{\sum_{s \in \tilde{S}(q), \text{TEXT}(s)=a^*} \exp(f(s|\mathcal{D}, q))}{\sum_{s \in \tilde{S}(q)} \exp(f(s|\mathcal{D}, q))}$$

$$2. \mathcal{L}_{\text{doc}} = -\log \frac{\sum_{s \in \tilde{S}(q), d(s) \in \mathcal{D}^*} e^{f(s,q)}}{\sum_{s \in \tilde{S}(q)} e^{f(s,q)}}$$

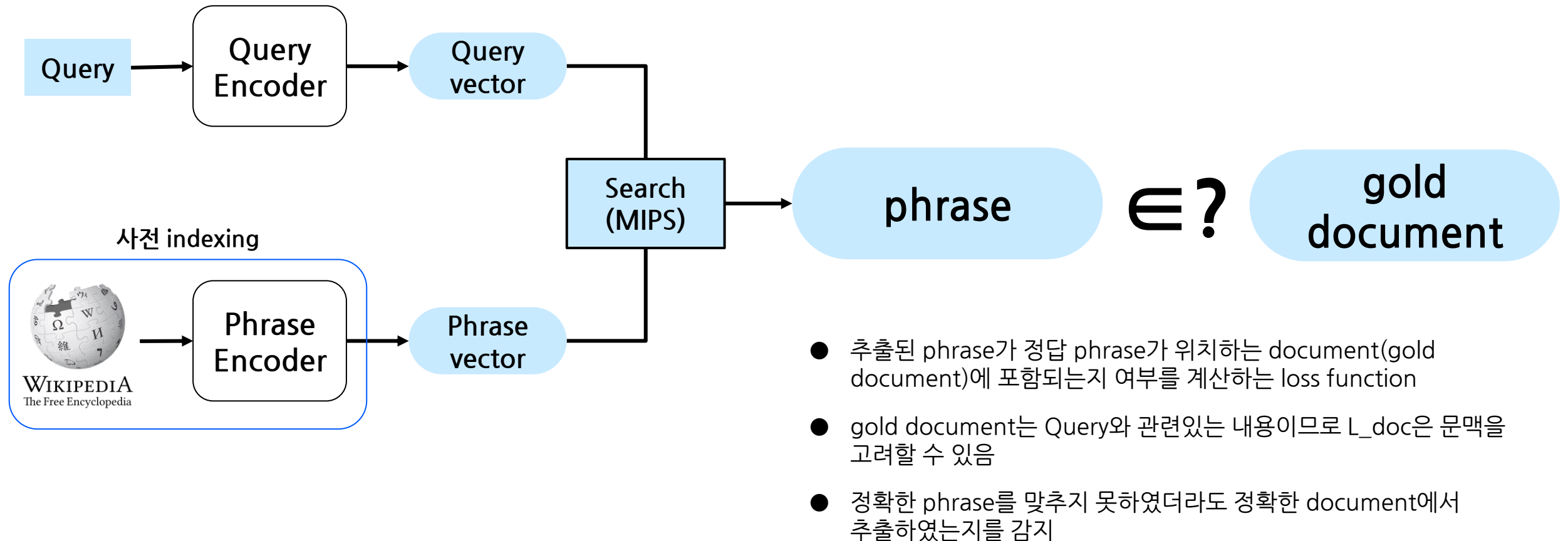
$$3. \mathcal{L}_{\text{context}} = -\log \frac{\sum_{s \in \tilde{S}(q), c(s) \in \mathcal{C}^*} e^{f(s,q)}}{\sum_{s \in \tilde{S}(q)} e^{f(s,q)}}$$

$$4. \mathcal{L}_{\text{sentence}} = -\log \frac{\sum_{s \in \tilde{S}(q), \text{sent}(s) \in \mathcal{S}^*} e^{f(s,q)}}{\sum_{s \in \tilde{S}(q)} e^{f(s,q)}}$$

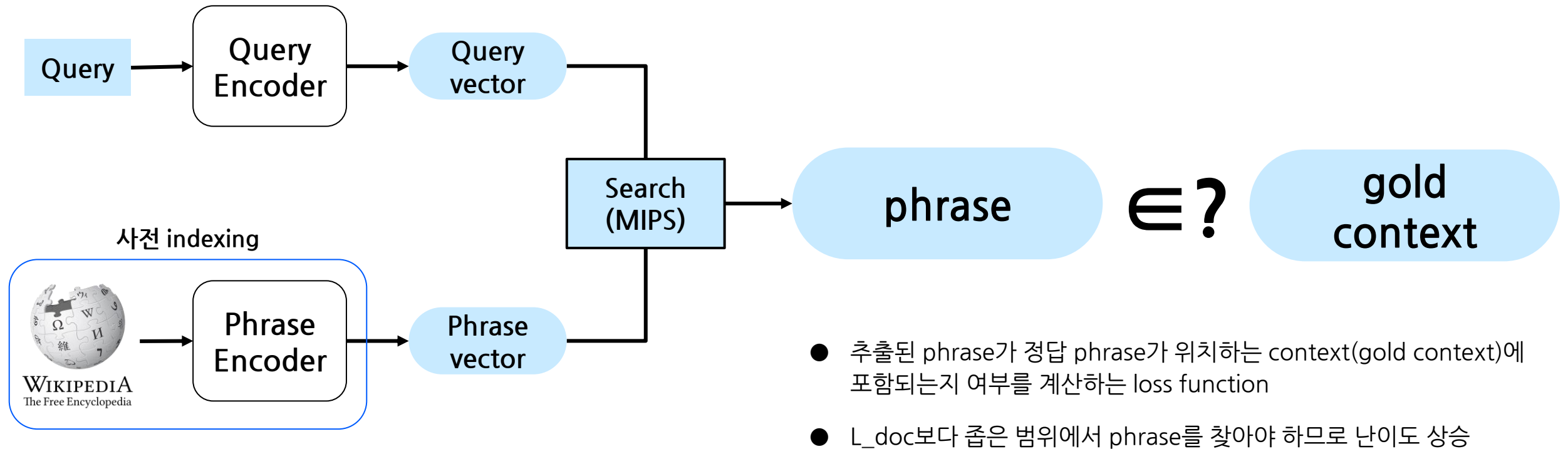
## L\_query



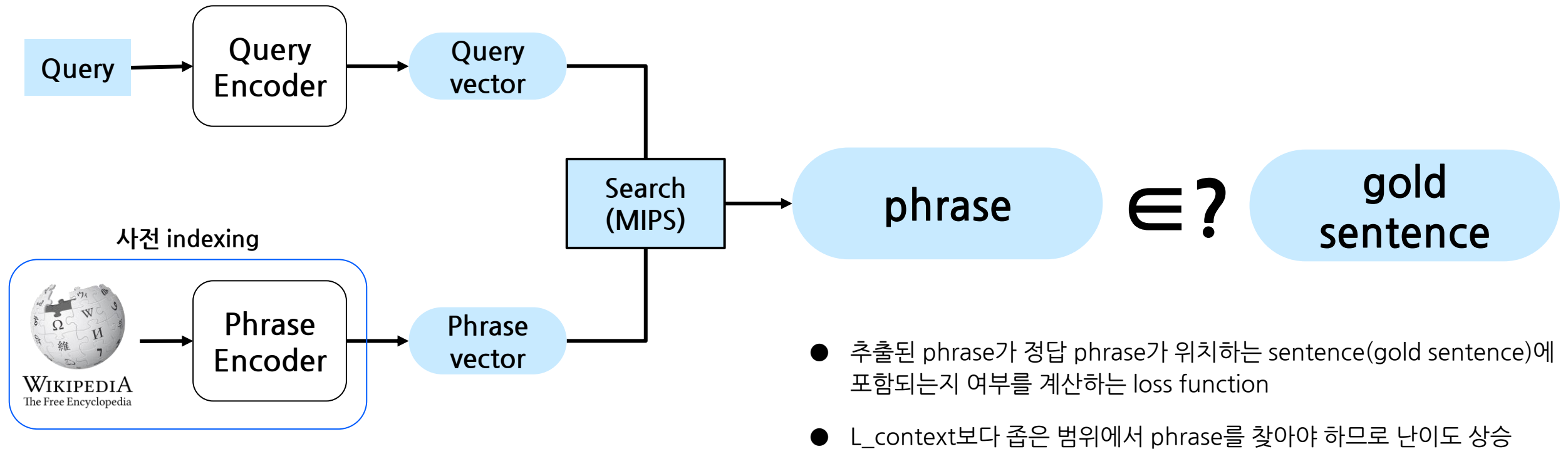
## L\_doc



## L\_context

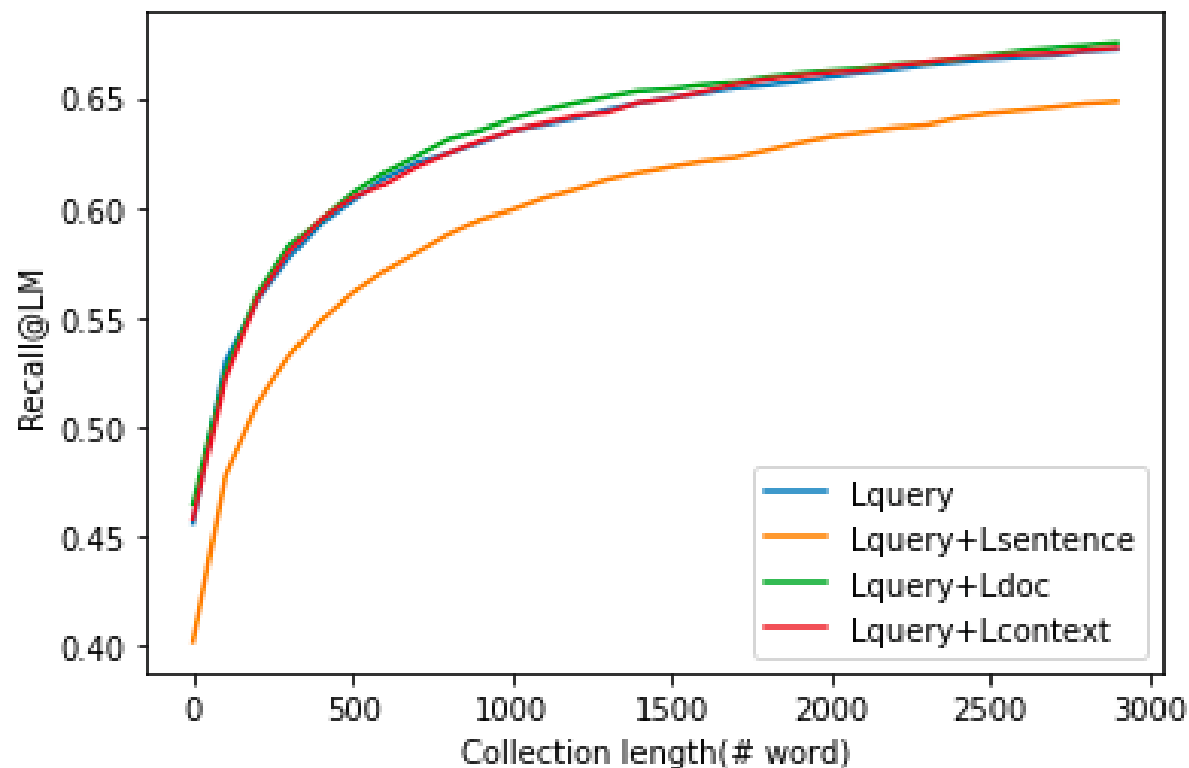


## L\_sentence



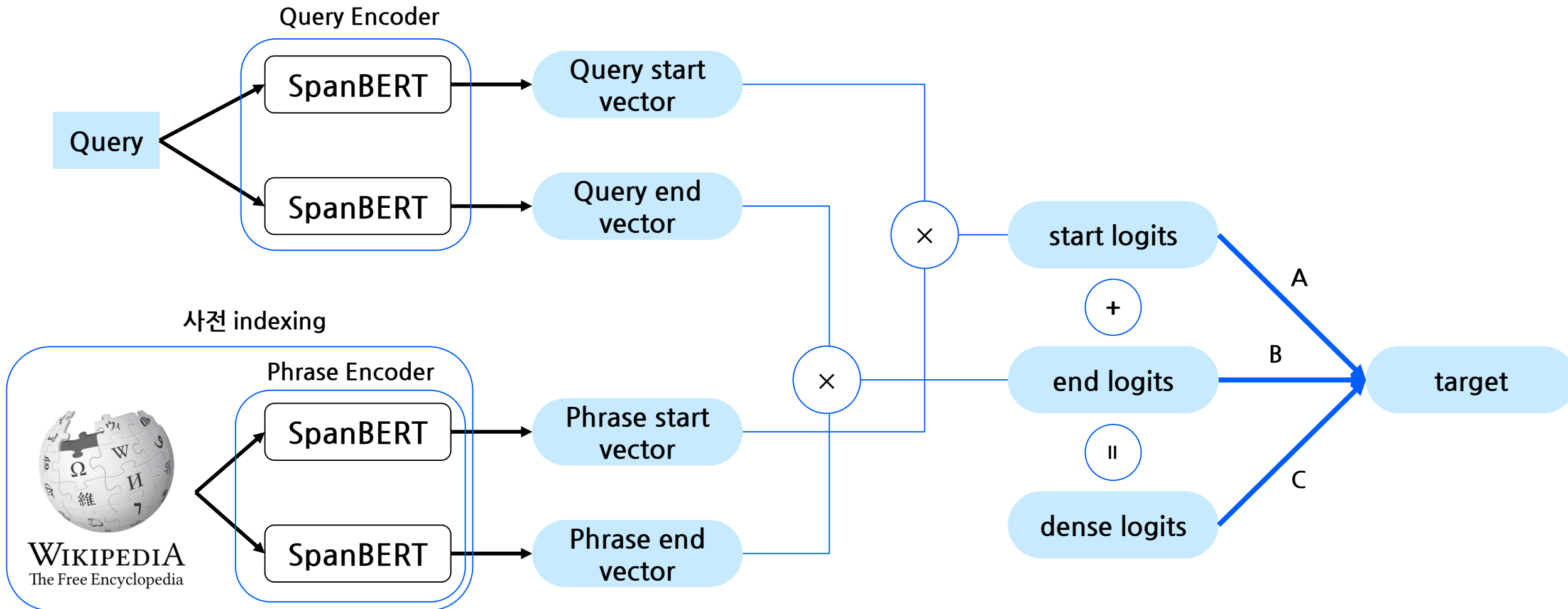
## Query loss의 단위 변경: Result

- R\_unit : sentence
- Top\_k : 200
- Collection length(# word) = 3000

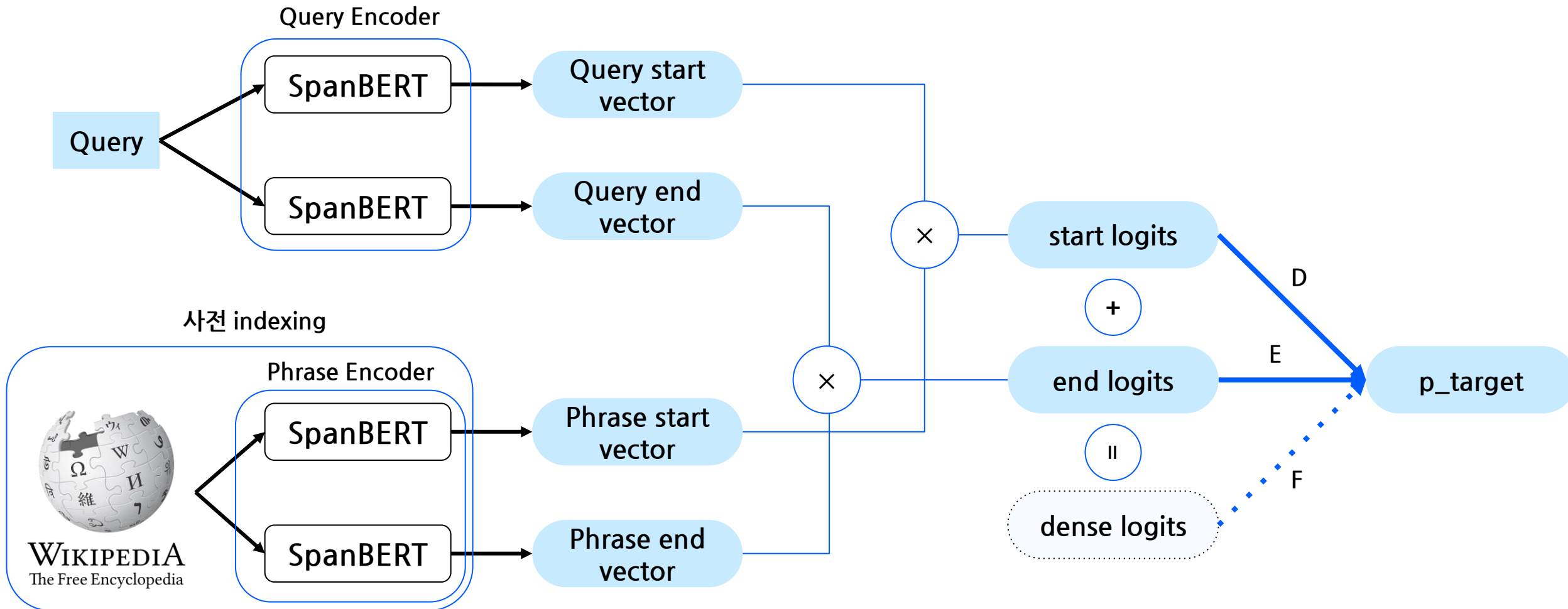


Loss	mAR
L_query	0.6316
L_query + L_sentence	0.6251
L_query + L_context	0.6322
L_query + L_doc	0.6351

## 3.1.2. Loss의 구성 요소 추가



## Loss의 구성 요소 추가





## Loss의 구성 요소 추가

- 결과

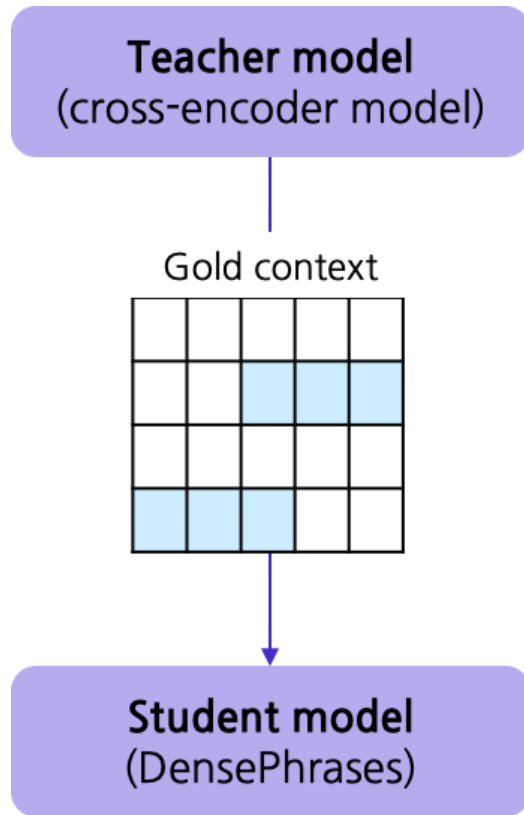
- phrase\_correct: 출력된 sentence에 정답이 포함되어 있고, 그 정답이 해당 sentence를 출력하게 한 phrase일 경우
- sent\_correct: 출력된 sentence에 정답이 포함되어 있지만 그 정답이 해당 sentence를 출력하게 한 phrase는 아닐 경우

	phrase_correct	sent_correct	sum	mAR	Loss의 구성 요소
L_query	2,505	1,773	4,278	0.6316	A + B + C
L_query+L_doc	2,385	1,925	<b>4,310</b>	<b>0.6353</b>	A + B + C + D + E
L_query+L_doc custom	2,302	<b>1,993</b>	4,295	0.6332	A + B + C + D + E + F

	phrase_correct	sent_correct	sum	mAR	Loss의 구성 요소
L_query	2,505	1,773	4,278	0.6316	A + B + C
L_query+L_context	2,105	2,187	<b>4,292</b>	<b>0.6322</b>	A + B + C + D + E
L_query+L_context custom	1,941	<b>2,323</b>	4,264	0.6289	A + B + C + D + E + F

## 3.1.3. Knowledge Distillation (with cross-encoder model)

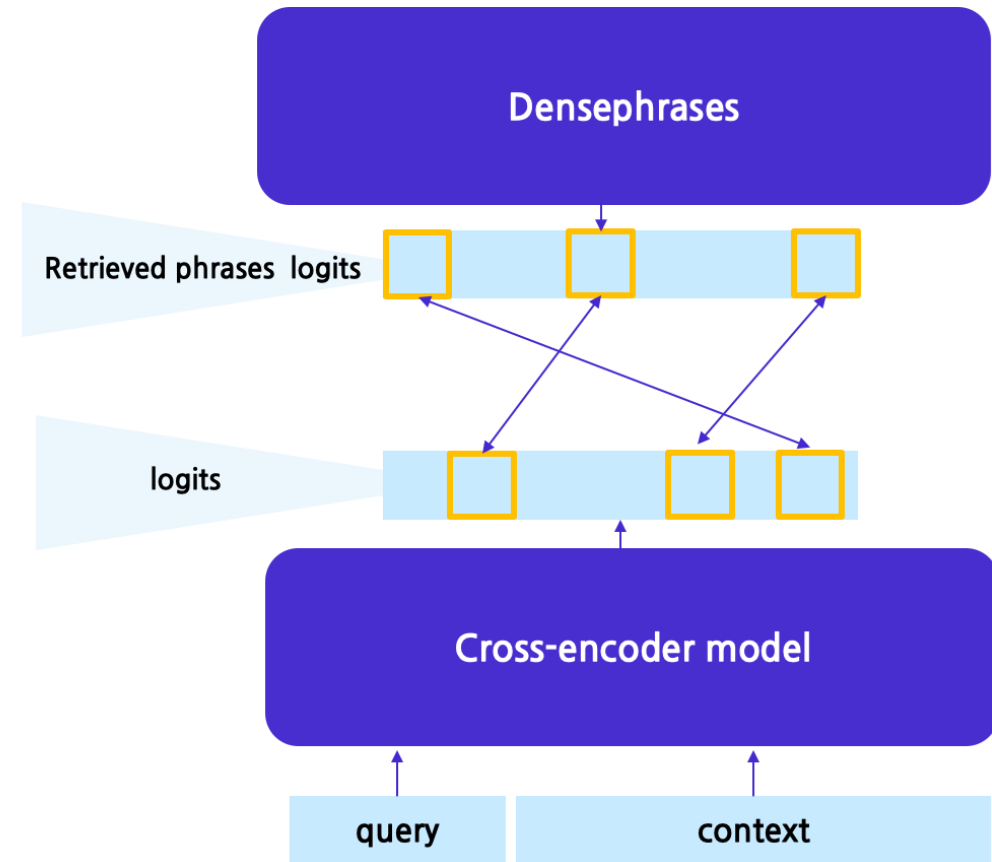
: cross encoder model에서 생성한 logit과 densephrase에서 생성한 logit의 차이를 loss로 사용



- Cross Encoder 를 Teacher model 로, densephrases model 을 Student model 로 사용
- QSFT 환경에서 knowledge distillation 을 활용하여 context 내에서 **gold phrase** 를 보다 잘 찾을 수 있도록 학습
- 보다 정확한 phrase retrieval 로 인한 mAR 향상 목표
- pre-train 된 cross-encoder (spanbert-base-cased-nq) 활용

# Knowledge Distillation (with cross-encoder model)

1. 정답 context와 query를 cross-encoder model에 input으로 넣어 logit 을 생성
1. Densephrase에서 retrieve된 top-k개의 phrase들중 정답 context에 포함된 phrase들의 위치를 추출
1. 2에서 추출 된 phrase token들의 logit값과 1에서 생성된 logit값들을 일치하는 token 의 값을 비교하여 그 차이를 distillation loss로 사용



# Knowledge Distillation (with cross-encoder model)

## ● 결과

- phrase\_correct: 출력된 sentence에 정답이 포함되어 있고, 그 정답이 해당 sentence를 출력하게 한 phrase일 경우
- sent\_correct: 출력된 sentence에 정답이 포함되어 있지만 그 정답이 해당 sentence를 출력하게 한 phrase는 아닐 경우

	phrase_correct	sent_correct	sum	mAR
L_query	2,505	1,773	4,278	0.6316
L_query & Distillation	2,532	1,758	4,290	0.6311

	phrase_correct	sent_correct	sum	mAR
L_query+L_doc	2,385	1,925	4,310	0.6353
L_query+L_doc & Distillation	2,504	1,821	4,425	0.6362

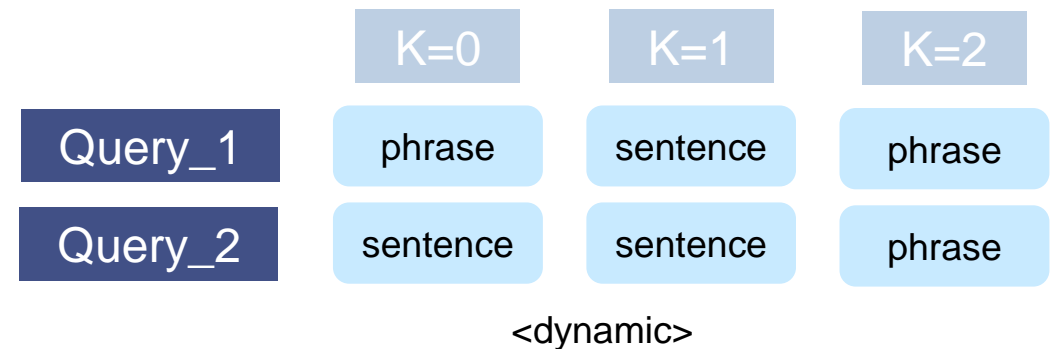
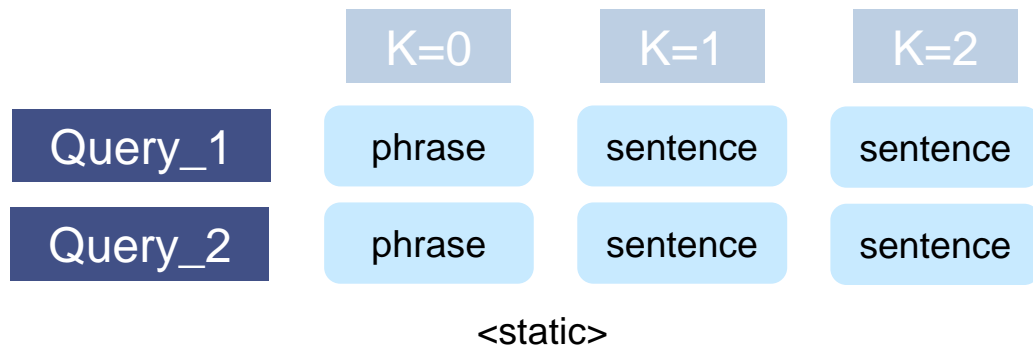
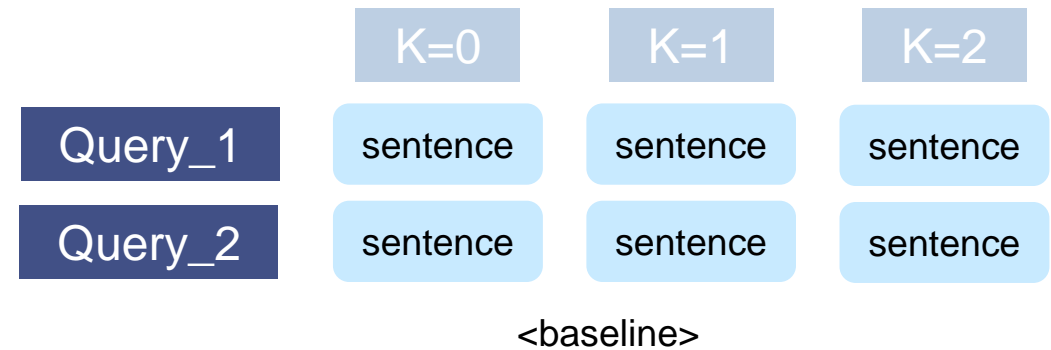
Loss	mAR
L_query	0.6316
L_query+L_sentence	0.6251
L_query+L_context	0.6322
L_query+L_doc	0.6351
L_query+L_doc & Distillation	0.6362

## 3.2 Multi Unit Retrieval

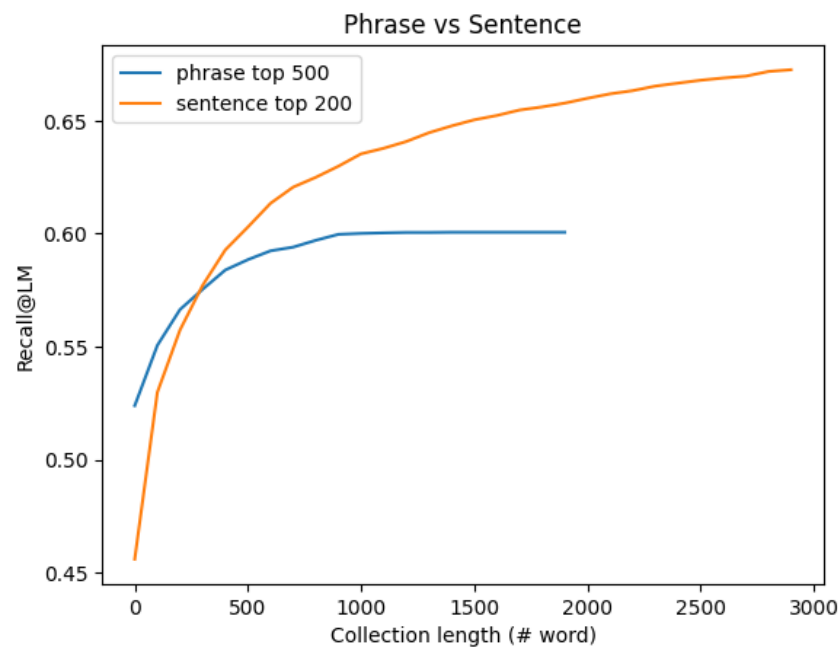
- Static
- Dynamic
- Optimization
- Result

# Multi Unit Retrieval

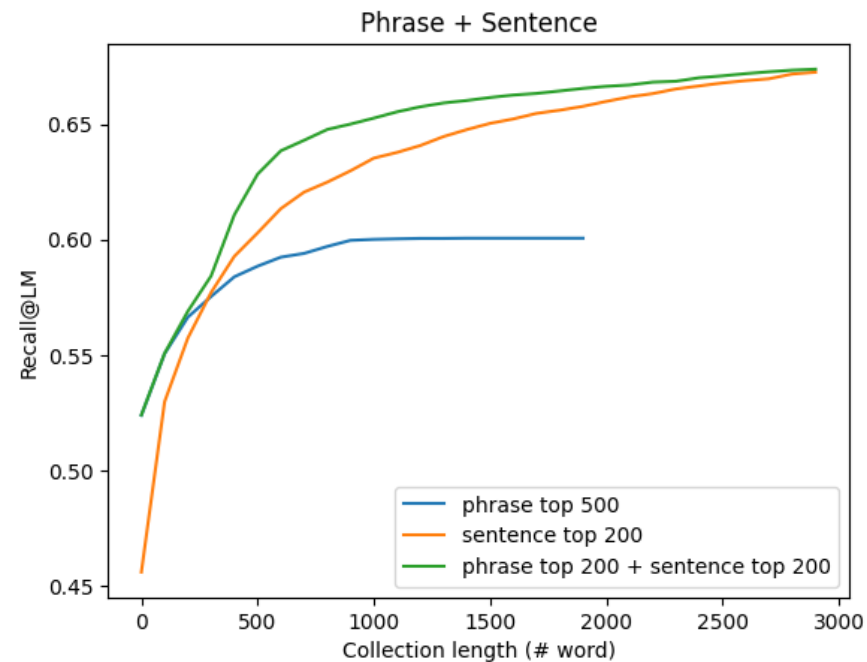
- Baseline : retrieval unit = sentence
- Multi Unit Retrieval
  - Static : phrase와 sentence의 순서를 고정
  - Dynamic : phrase와 sentence의 순서를 고정하지 않음



## 3.2.1. Multi Unit Retrieval - Static



두 방법의 장점을 모두 활용하기 위해  
처음에 **phrase**를 주고 이어서  
**sentence**를 주는 방식 사용



	phrase	sentence
장점	초반에 높은 Recall	높은 Recall 상승폭 높은 최종 Recall
단점	낮은 Recall 상승폭 낮은 최종 Recall	낮은 초반 Recall

	mAR
sentence top 200	0.6316
phrase top 200 + sentence top 200	0.6450

## 3.2.2. Multi Unit Retrieval - Dynamic

- Question과 phrase의 관계에 따라 유동적으로 unit을 결정하고자 각 unit에 적합한 query encoder를 찾고자 함
- L\_query : 정답 phrase가 일치하는지를 구분
- L\_context or L\_doc : 정답 phrase뿐만 아니라 정답의 문맥도 같아야 함

Example

	gold answer	['apple', 'banana']	gold context	'I like apple and banana'
X	phrase	'apple'	source sentence	'I ate an apple'
Y	phrase	'apple'	source sentence	'I like apple and banana'

- X, Y는 하나의 쿼리에 대한 서로 다른 retrieval

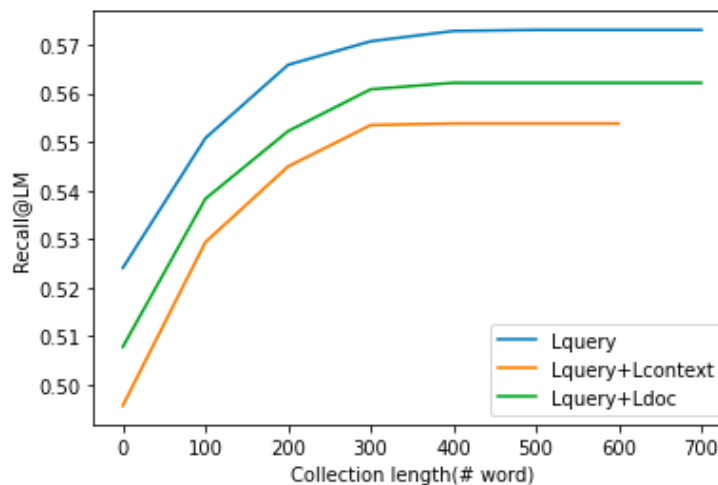
- X는 phrase로 retrieve하는 것이 경제적
- Y는 sentence로 retrieve하는 것이 recall 면에서 좋음
- L\_query encoder는 phrase X와 Y를 구분하지 못함
- L\_query+L\_context는 X보다 Y를 찾을 경향이 높음



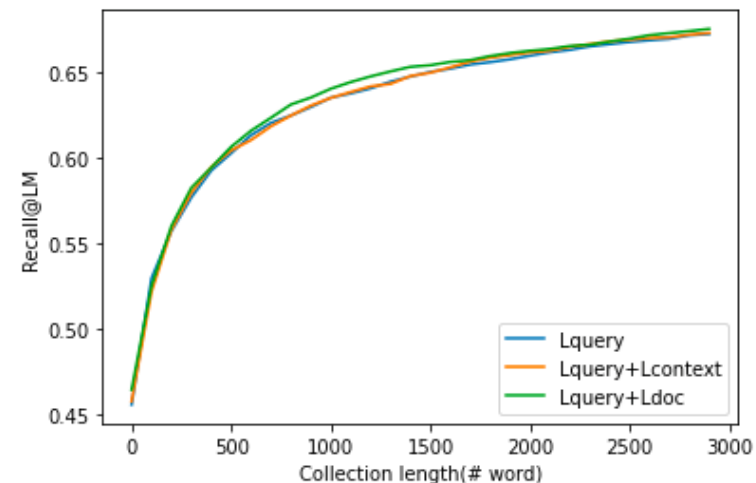
## Dual Query Encoder : unit 별로 다른 encoder 사용

L\_query encoder로 retrieve한 phrase, L\_query+L\_context 또는 L\_query+L\_context로 retrieve한 sentence 사용

	phrase	sentence
L_query	0.5629	0.6316
L_query + L_context	0.5407	0.6322
L_query + L_doc	0.5509	0.6351

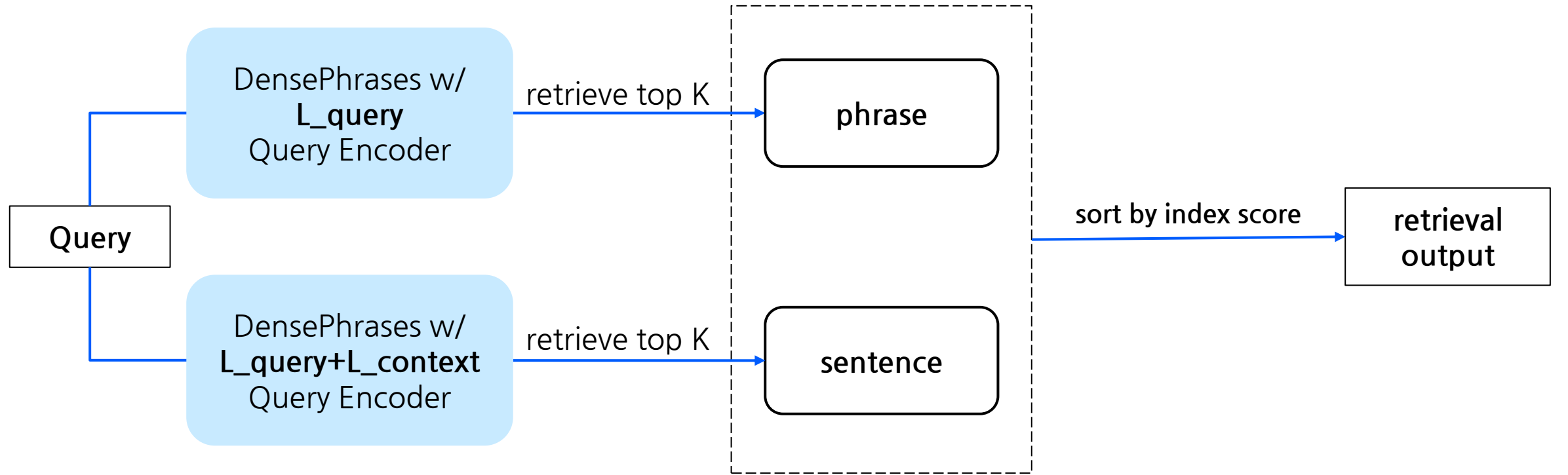


retrieval unit = phrase



retrieval unit = sentence

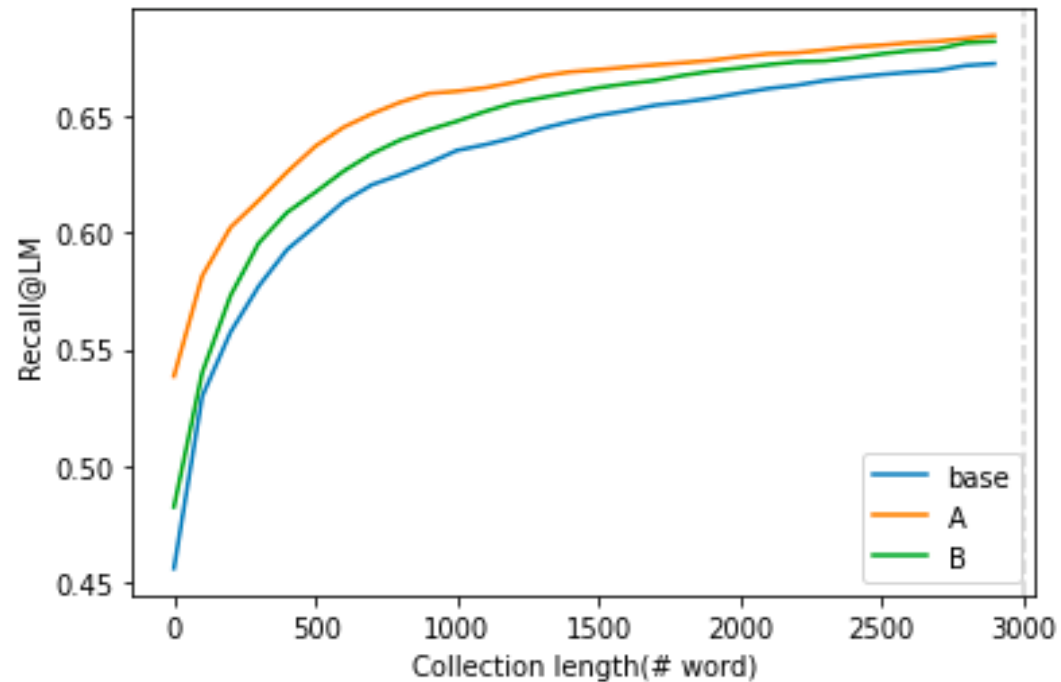
## Dynamic Retrieval : Overview



Dual Query Encoder

index score : Query vector와 phrase vector의 inner product 값

# Multi Unit Retrieval - Dynamic



	Encoder	mAR
base	sentence top 200 : L_query	0.6316
A	phrase top 200 : L_query, sentence top 200 : L_query+L_context	<b>0.6564</b>
B	phrase top 200 : L_query, sentence top 200 : L_query+L_doc	0.6441

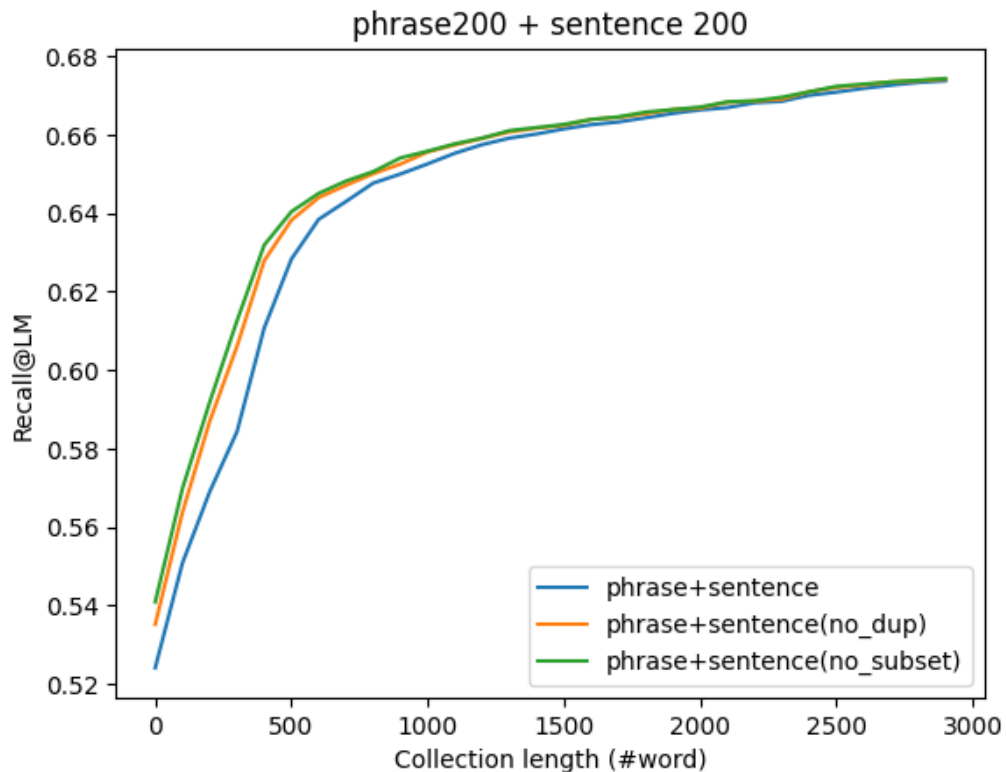
## 3.2.3. Multi Unit Retrieval - Optimization

- 중복 및 Subset 제거
  - k 번째 반환되는 phrase가 k 번째 이전에 등장했을 경우 최적화 가능
- 최적화 이후 9개의 결과가 4개로 줄어든 것을 볼 수 있음 → 더 적은 # word로 더 다양한 정답을 포함



## 3.2.3. Multi Unit Retrieval - Optimization

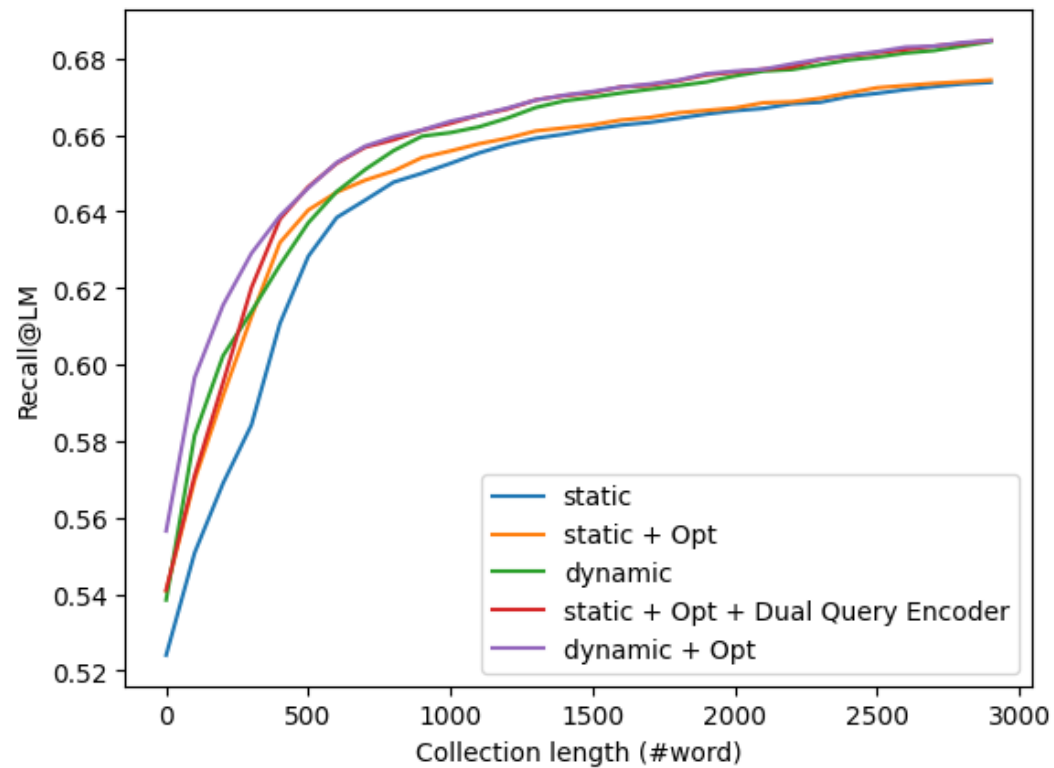
- 중복 및 Subset 제거
  - k 번째 반환되는 phrase가 k 번째 이전에 등장했을 경우 최적화 가능
- 최적화 이후 9개의 결과가 4개로 줄어든 것을 볼 수 있음 → 더 적은 # word로 더 다양한 정답을 포함



	mAR
phrase+sentence	0.6450
phrase+sentence(no_dup)	0.6493
phrase+sentence(no_subset)	0.6505

- no\_dup : 중복 제거
- no\_subset : subset 제거

## 3.2.4. Multi Unit Retrieval : Result



	mAR
static	0.6450
static + Opt	0.6505
dynamic	0.6564
static + Opt + Dual	0.6583
dynamic + Opt	0.6609

- Opt : 최적화(중복제거, subset 제거) 적용
- Dual : Dual query encoder 적용

---

# 4. Conclusion

---

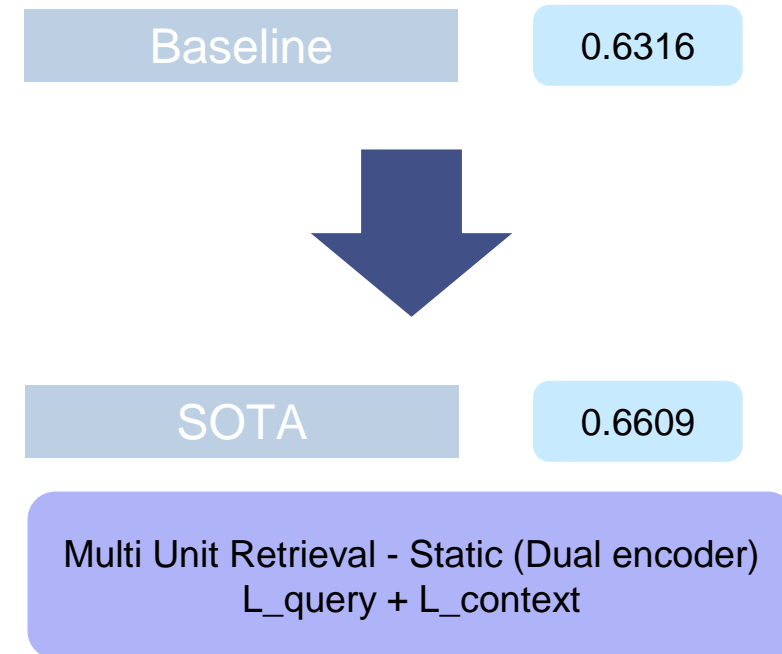
Result

의의 및 한계

추후 발전 계획

## 4.1. Result

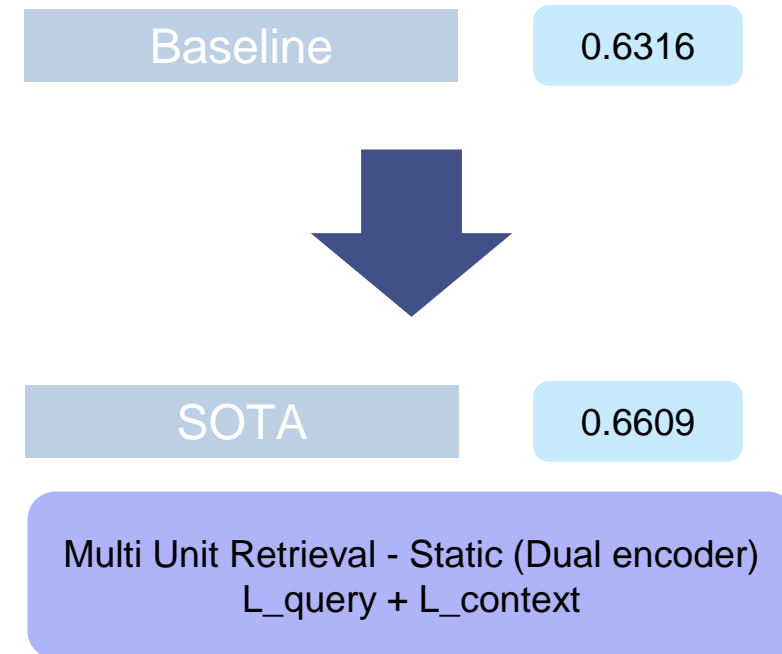
- QSFT
  - Query loss의 단위 변경
    - $L_{\text{query}} + L_{\text{doc}}$
    - $L_{\text{query}} + L_{\text{sentence}}$
    - $L_{\text{query}} + L_{\text{context}}$
  - Loss 구성 요소 추가
  - Knowledge Distillation
- Multi Unit Retrieval
  - Static
    - Single Encoder
    - Dual Encoder
  - Dynamic
    - Dual Encoder
  - 최적화





## 4.1. Result

- QSFT
  - Query loss의 단위 변경
    - $L_{\text{query}} + L_{\text{doc}}$
    - $L_{\text{query}} + L_{\text{sentence}}$
    - $L_{\text{query}} + L_{\text{context}}$
  - Loss 구성 요소 추가
  - Knowledge Distillation
- Multi Unit Retrieval
  - Static
    - Single Encoder
    - Dual Encoder
  - Dynamic
    - Dual Encoder
  - 최적화



## 4.2. 의의 및 한계

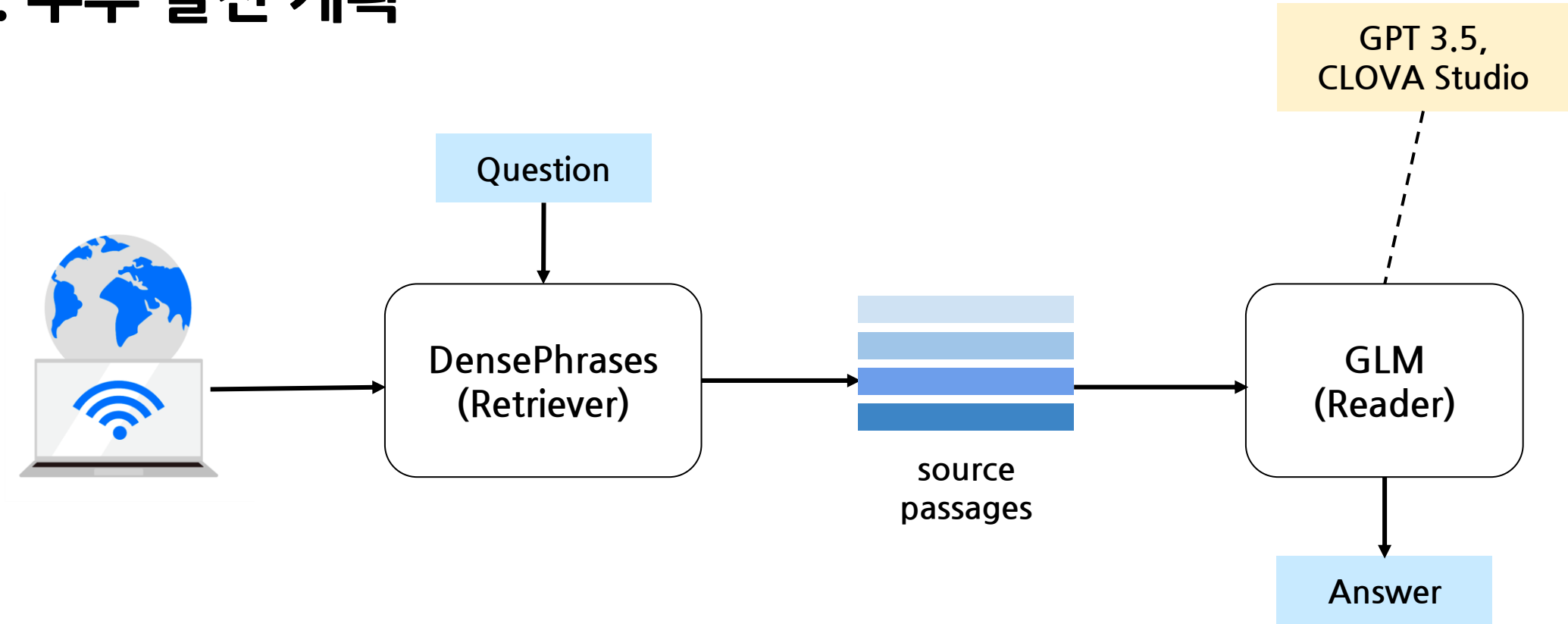
### 의의

- Retriever에서 recall과 length의 trade-off를 고려하며 retrieval model의 성능을 향상시키기 위해 다양한 시도들을 적용
- 프롬프트의 경량화를 통해 reader model의 계산 비용을 최소화하며 성능은 유지할 수 있음

### 한계

- 실험에 사용된 NQ Dataset의 한계
  - NQ Dataset은 **단답형 질문**으로만 구성되어 있고, 질문의 근거가 되는 **source document가 하나**로 정해져 있음
  - 그러나, 실제 상황에서는 1) 단답으로 해결 불가능한 **complex query**들이 포함될 가능성이 있고, 2) 질문의 근거가 되는 **source document가 여러개** 필요할 수도 있음
  - 따라서 실제 상황을 반영하는 data는 아니므로, 추후에 이러한 한계점들을 보완하는 여러 data들로 실험을 진행할 필요가 있음
- Indexing 파일의 크기가 너무 큼 → QSFT 시 소요되는 시간이 너무 큼 (1epoch 당 평균 3시간)
  - 전체 가설을 통합한 결과를 내지 못함
  - 각 가설들을 더 구체화하지 못함

## 4.3. 추후 발전 계획



## 지식기반 챗봇 시연

### 지식기반 챗봇

사용자의 지식베이스에 기반해서 대화하는 챗봇입니다. 본 예시에서는 wikipedia dump에서 검색한 후 이를 바탕으로 답변을 생성합니다.  
retriever: densePhrase, generator: gpt-3.5-turbo-16k-0613 (API)

질문

클리어 제출하기

생성된 답변

prompt에 첨부된 검색 결과들

플래그

---

# 5. Reference

---

## 5. Reference

1. Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. 2021. [Learning Dense Representations of Phrases at Scale](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6634-6647, Online. Association for Computational Linguistics.
2. Jinhyuk Lee, Alexander Wettig, and Danqi Chen. 2021. [Phrase Retrieval Learns Passage Retrieval, Too](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3661-3672, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

---

# End of Document Thank You.