Improving Text Embeddings with Large Language Models

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, Furu Wei

Microsoft Corporation

발표자: 송선영

2024/01/16

Introduction

- Information Retrieval (IR) 분야에서 first-stage는 텍스트 임베딩에 의존해 large-scale corpus에서 small set of candidate documents를 찾는 것

- Embedding-based retrieval은 retrieval-augmented generation(RAG)에서 중요한 요소 중 하나
  - RAG: LLM이 모델의 parameter를 수정하지 않고 동적으로 외부 지식에 접근할 수 있도록 함
  - Generated text의 source 정보는 LLM의 해석 가능성과 신뢰성을 향상시킬 수 있는 중요한 요소임

→ 잘 검색하는게 중요하다

Introduction

- 텍스트 임베딩의 performanc와 robustness를 향상시키기 위해 최근 multi-stage training paradigm 사용
  1. 그 다음, 먼저, 수십억 개의 weakly-supervised text pair로 사전학습
  2. labeled dataset으로 미세조정

  - multi-stage approach의 단점
    - 많은 양의 relevance pair를 만드는 데 많은 노력이 들고, 복잡한 구조의 multi-stage pipeline을 사용함
    - Manually하게 수집된 데이터셋에 의존함
      - 이 데이터셋은 Task의 다양성과 language의 적용 범위에 따라 제약을 받음
    - 대부분의 방법이 BERT-style encoder를 백본으로 사용
      - LLM과 context length 확장과 같은 관련 기술들을 사용하지 않음

- 이 논문에서는 LLM을 활용해 텍스트 임베딩을 위한 새로운 방법을 제안
  - Multi-stage가 아닌 single-stage training paradigm 사용
  - BERT-style encoder가 아닌 open-source LLM 파인튜닝

Synthetic Data Generation

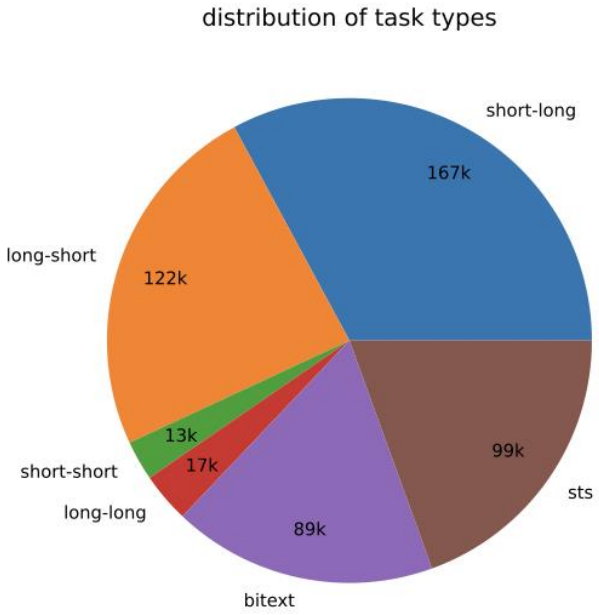
- 다양한 synthetic data 생성하기 위해 아래 두 그룹으로 분류 후, 각 그룹에 다른 prompt template을 적용

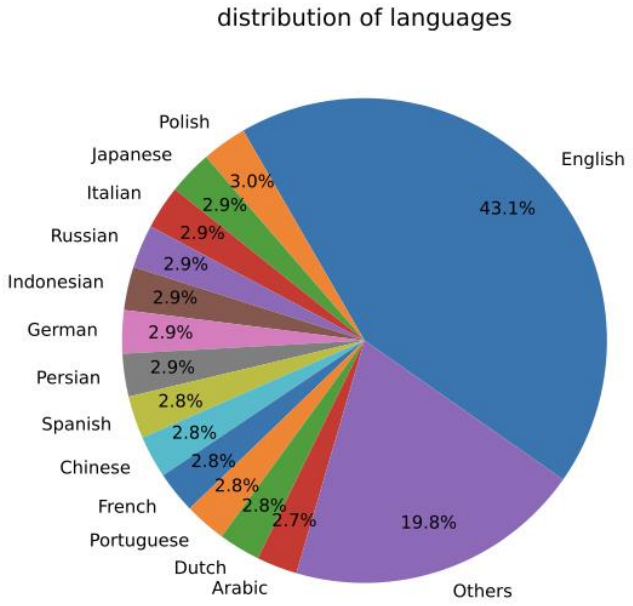
- Asymmetric Tasks
  - Query와 document가 의미적으로는 관련이 있지만 paraphrasing이 아닌 경우
  - 4가지의 subgroup으로 나뉨
    - short-long, long-short, short-short, short-long

- Symmetric Tasks
  - Query와 document가 의미적으로 유사하지만 표면적인 형태는 다름
  - 2개의 scenario로 나뉨
    - monolingual STS, bitext retrieval

Statistics of the Synthetic Data

- 150k instruction으로 500k의 example 생성
  - Azure OpenAI Service 사용 (GPT-4 사용가능한 service)
  - 25%는 GPT-3.5-Turbo로 생성, 나머지는 GPT-4로 생성
- 93개의 언어
  - 주요 언어는 영어
  - 하위 75개의 low-resource 언어는 언어당 평균 약 1k개의 example 존재





Fine-tuning

- 관련된 query-document pair (q^+, d^+) 가 주어질 때, 먼저 아래 instruction template을 q^+에 적용해 q^+_{inst} 를 생성

$$q^+_{\text{inst}} = \text{Instruct: } \{\text{task\_definition}\} \; \backslash n \; \text{Query: } \{q^+\}$$

Brainstorm a list of potentially useful text retrieval tasks.
Here are a few examples for your reference:
- Provided a scientific claim as query, retrieve documents that help verify or refute the claim.
- Search for documents that answers a FAQ-style query on children's nutrition.

Please adhere to the following guidelines:
- Specify what the query is, and what the desired documents are.
- Each retrieval task should cover a wide range of queries, and should not be too specific.

Your output should always be a python list of strings only, with about 20 elements, and each element corresponds to a distinct retrieval task in one sentence. Do not explain yourself or output anything else. Be creative!

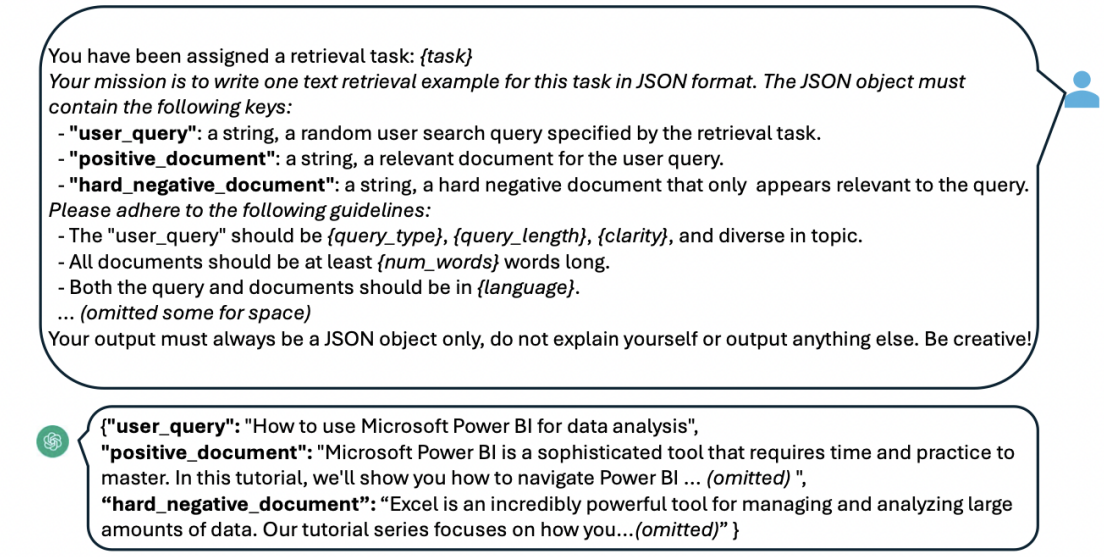
["Retrieve company's financial reports for a given stock ticker symbol.",
"Given a book name as a query, retrieve reviews, ratings and summaries of that book.",
"Search for scientific research papers supporting a medical diagnosis for a specified disease."
... *(omitted for space)*]

Table 7: Instructions for each training dataset.

Dataset	Instruction
ELI5	Provided a user question, retrieve the highest voted answers on Reddit ELI5 forum
HotpotQA	Given a multi-hop question, retrieve documents that can help answer the question
FEVER	Given a claim, retrieve documents that support or refute the claim
MIRACL / MrTyDi / NQ / SQuAD / TriviaQA	Given a question, retrieve Wikipedia passages that answer the question Retrieve Wikipedia passages that answer the question
NLI	Given a premise, retrieve a hypothesis that is entailed by the premise Retrieve semantically similar text
MS-MARCO	Given a web search query, retrieve relevant passages that answer the query Given a web search query, retrieve relevant documents that answer the query
Quora Duplicates	Given a question, retrieve questions that are semantically equivalent to the given question Find questions that have the same meaning as the input question
DuReader / T2Ranking	Given a Chinese search query, retrieve web passages that answer the question

Fine-tuning

- Query와 document 마지막에 [EOS] token을 추가해, LLM의 입력으로 주어 마지막 layer의 [EOS] vector를 가지고 각각의 embedding인 $(h_{q^+_{inst}}, h_{d^+})$ 를 얻을 수 있음
- InfoNCE loss를 사용



$$\min \quad \mathbb{L} = -\log \frac{\phi(q^+_{\text{inst}}, d^+)}{\phi(q^+_{\text{inst}}, d^+) + \sum_{n_i \in \mathbb{N}} (\phi(q^+_{\text{inst}}, n_i))}$$

$$\phi(q, d) = \exp(\frac{1}{\tau} \cos(\mathbf{h}_q, \mathbf{h}_d))$$

Training & Evaluation

- 사전학습된 Mistral-7b checkpoint를 fine-tuning
  - Epoch:1, batch size: 2048, learning rate: 10^{-4}, weight decay: 0.1
  - LoRA 사용 (rank 16)
  - GPU 메모리 효율성을 위해 gradient checkpointing, mixed precision 등의 방법 적용

- 학습 데이터
  - 생성된 합성 데이터 + 13개의 public dataset을 활용해 총 180만 개의 example 생성

Evaluation

- MTEB benchmark 에서 평가
  - MTEB (Massive Text Embedding Benchmark)
    - Multiple sources, multiple languges dataset

# of datasets →	Class. 12	Clust. 11	PairClass. 3	Rerank 4	Retr. 15	STS 10	Summ. 1	Avg 56
Unsupervised Models								
Glove [34]	57.3	27.7	70.9	43.3	21.6	61.9	28.9	42.0
$\text{SimCSE}_{\text{bert-unsup}}$ [12]	62.5	29.0	70.3	46.5	20.3	74.3	31.2	45.5
Supervised Models								
$\text{SimCSE}_{\text{bert-sup}}$ [12]	67.3	33.4	73.7	47.5	21.8	79.1	23.3	48.7
Contriever [17]	66.7	41.1	82.5	53.1	41.9	76.5	30.4	56.0
GTR_{xxl} [31]	67.4	42.4	86.1	56.7	48.5	78.4	30.6	59.0
$\text{Sentence-T5}_{\text{xxl}}$ [30]	73.4	43.7	85.1	56.4	42.2	82.6	30.1	59.5
$\text{E5}_{\text{large-v2}}$ [44]	75.2	44.5	86.0	56.6	50.6	82.1	30.2	62.3
$\text{GTE}_{\text{large}}$ [22]	73.3	46.8	85.0	59.1	52.2	83.4	31.7	63.1
$\text{BGE}_{\text{large-en-v1.5}}$ [46]	76.0	46.1	87.1	60.0	54.3	83.1	31.6	64.2
Ours								
$\text{E5}_{\text{mistral-7b}}$ + full data	**78.5**	50.3	**88.3**	**60.2**	**56.9**	**84.6**	31.4	**66.6**
w/ synthetic data only	78.2	**50.5**	86.0	59.0	46.9	81.2	31.9	63.1
w/ synthetic + msmarco	78.3	49.9	87.1	59.5	52.2	81.2	**32.7**	64.5

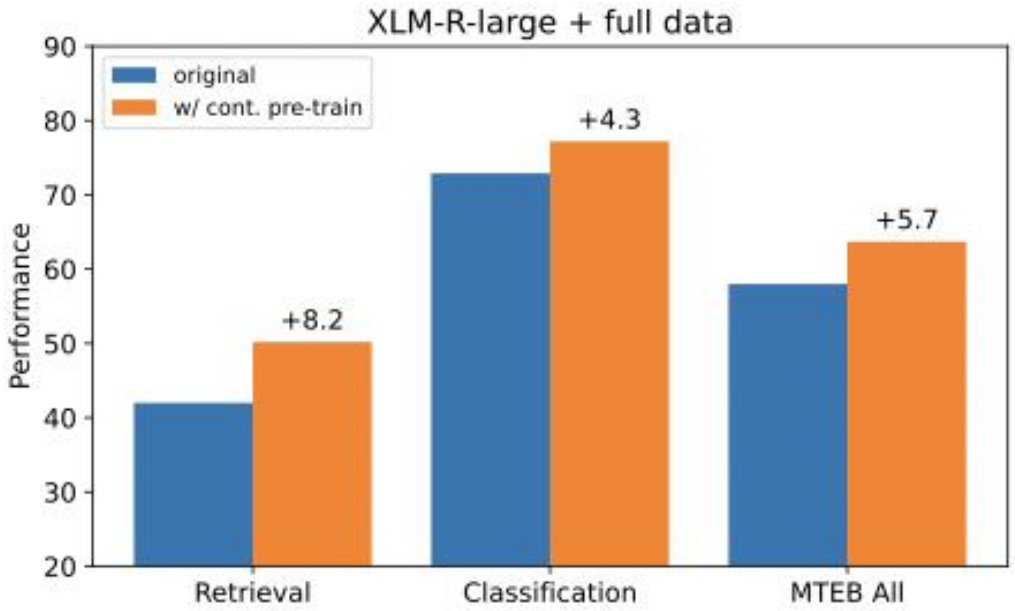
Evaluation

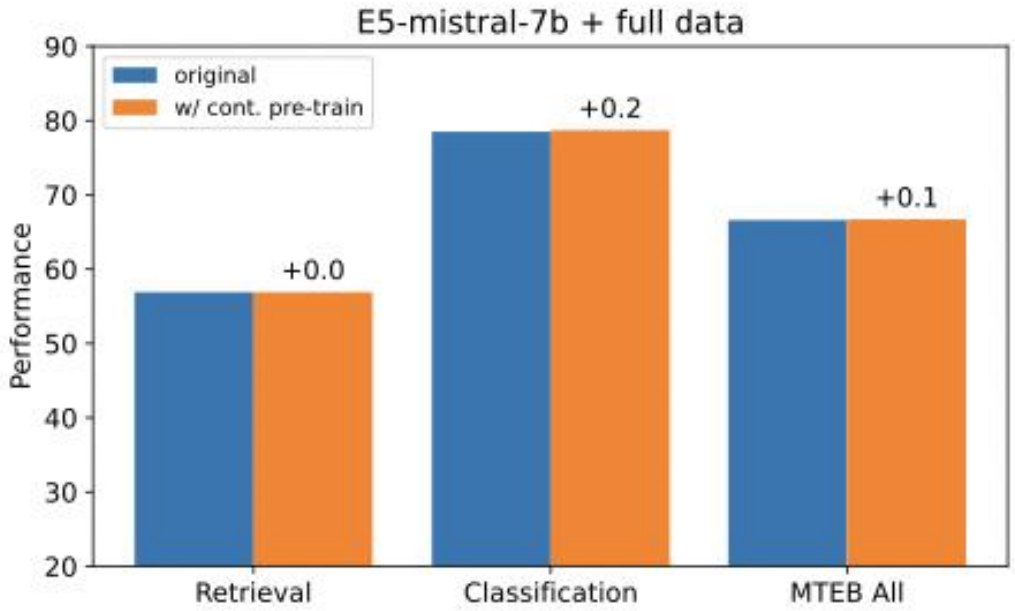
- Multilingual Retrieval 평가
  - 평가 dataset: MIRACL dataset (Multilingual Information Retrieval Across a Continuum of Languages)

	High-resource Languages				Low-resource Languages			
	en	fr	es	ru	te	hi	bn	sw
BM25 [51]	35.1	18.3	31.9	33.4	49.4	45.8	50.8	38.3
mDPR [51]	39.4	43.5	47.8	40.7	35.6	38.3	44.3	29.9
$mE5_{base}$ [44]	51.2	49.7	51.5	61.5	75.2	58.4	70.2	71.1
$mE5_{large}$ [44]	52.9	54.5	**52.9**	67.4	**84.6**	**62.0**	**75.9**	**74.9**
$E5_{mistral\text{-}7b}$ + full data	**57.3**	**55.2**	52.2	**67.7**	73.9	52.1	70.3	68.4

Evaluation

- Contrastive Pre-training이 필수적인가?





Conclusion

- LLM을 활용해 다양한 synthetic data를 생성해 텍스트 임베딩의 품질을 크게 향상시킬 수 있음을 보임
- 최소한의 fine-tuning 만으로도 효과적인 임베딩을 가질 수 있음

Thank You

감사합니다.