논문 리뷰

Dense Passage Retrieval for Open-Domain Question Answering

정재훈

04

Result

Q&A

03

Experiment setting

Experiment

02

Dense Passage Retrieval

ORQA

01

Dense Passage Retrieval

Sparse Representation

Result

Experiment

Dense Passage retrieval

Introduce

TF- IDF

$$tfidf(t,d,D) = tf(t,d) \times idf(t,D)$$

- 문서내에 등장 빈도가 높을 수록 중요
- 다른 문서에서는 검색어가 출현하지 않을수록

BM-25

$$score(D,Q) = \sum_{i=1}^{n} IDF(q_i) \cdot \frac{f(q_i,D) \cdot (k_1+1)}{f(q_i,D) + k_1 \cdot (1 - b + b \cdot \frac{|D|}{avgdl})}$$

$f(q_i,D)$: 문서 D에서 q_i의 term frequency
$|D|$: 문서 D의 길이
k_1, b: 파라메터
$avgdl$: 문서 집합의 평균 문서 길이

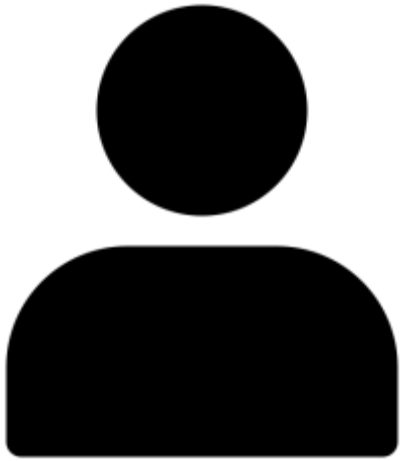
- 문서 내용에 검색어 출현 빈도가 높을수록
- 다른 문서에서는 검색어가 출현하지 않을수록
- 문서 내용이 짧을수록

TF- IDF
BM-25

Dense Passage Retrieval

- BM25 대비 9~19%의 성능이 증가

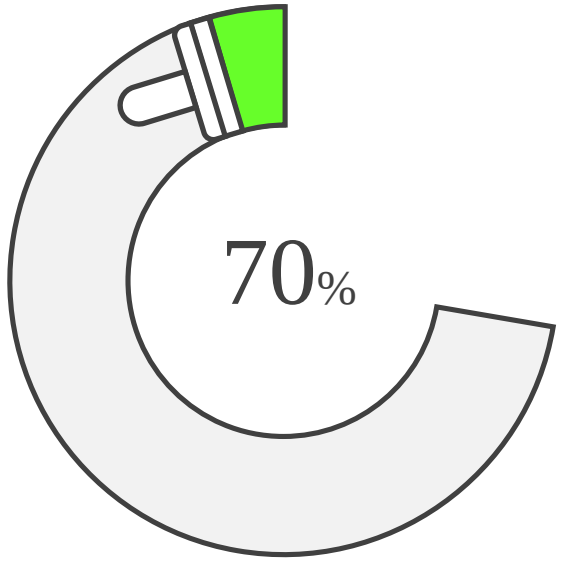
Who is the bad guy in lord of the rings?



Sparse Representation

VS

Dense Passage Retrieval



일반적으로 많은 데이터 필요

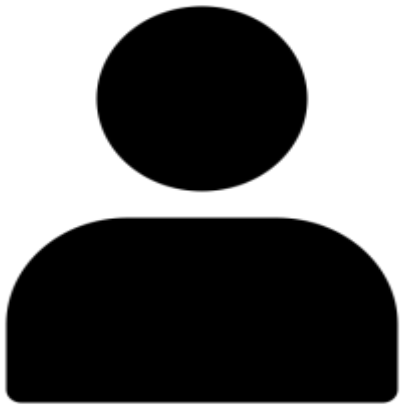
ORQA

ICT – inverse cloze task
BUT two weekness

- Regular sentence가 좋은지 명확하지 않다.
- Fine-tuning 하지 않기 때문에, suboptimal이 될 수 없다.

적은 데이터로도 가능

추가적인 pretraining 없이 (Q,P) pair만을 이용해 더 나은 Dense Embedding Model을 학습시킬 수 있을까?



Dense Passage Retriever

BERT
Dual Encoder Architecture

Question Encoder, Passage Encoder

Embedding vector :
Question vector와 Relevant Passage vector 사이의 내적 최대화

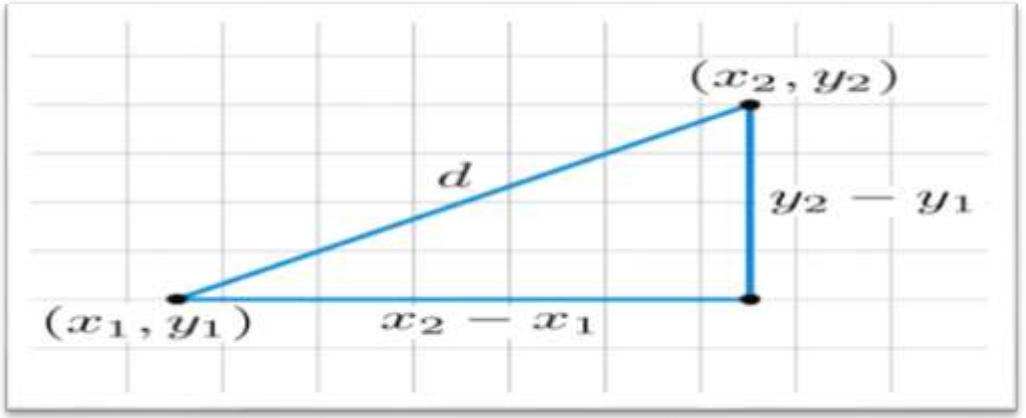
Dense Passage Retriever 특징

$$\text{sim}(q,p) = E_Q(q)^\intercal E_P(p)$$

- 상위 k개의 연관된 Passage를 효율적으로 제공
- K는 20 ~ 100개

FAISS 사용 – vector 유사도 측정

유클리드 거리



$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}.$$

- 가장 간단하다.

모델 학습

$$L(q_i, p_i^+, p_{i,1}^-, \cdots, p_{i,n}^-) = -\log \frac{e^{\text{sim}(q_i, p_i^+)}}{e^{\text{sim}(q_i, p_i^+)} + \sum_{j=1}^{n} e^{\text{sim}(q_i, p_{i,j}^-)}}$$

qi : Question
p+ : Postive passage
P- : Negative passage

Negative Sample

Random : 코퍼스 내의 random한 passage를 선택하는 방법
BM25 : 실제 정답을 포함하고 있지는 않지만 코퍼스 내에서 BM25 기준으로 top-k의 문서를 사용하는 방법
Gold : 학습셋 내의 다른 질의의 positive passage를 선택하는 방법

In – batch negatives

$$S = QP^T \quad S = QP^T \ (\text{BxB})$$

Given a query Q, containing keywords $q_1, \ldots, q_n$, the BM25 score of a document D is:

$$\text{score}(D,Q) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)}$$

S : question와 passage 사이의 유사도 담음
Q : Qusetion matrix
P : Passage matrix

Loss Function은 NLL을 활용
NLL은 **Ground Truth 카테고리에 대해 모델이 예측한 확률값이 작아질수록 <unhappy>한 특성**
Loss를 최소화하도록 모델을 optimize하면, ground truth에 대한 예측치를 높임

Experimental setup

Natural Question	Google 추출, 간단 질문과 스윙을 위해서 설계
Trivia QA	사소한 질문들과 답변으로 구성
Web Questions	Google Suggset API를 사용하여 선택한 질문 구성 답변은 Freebase의 엔터티이다.
Curated TPEC	TPEC QA 트랙에서 질문을 얻는다.
SQuAD	과거 QA연구에 사용, 제공된 단락이 없으면 많은 질문에 내용 부족

Experimental setup

Dataset	Train	Dev	Test
Natural Questions	79,168	58,880	8,757	3,610
TriviaQA	78,785	60,413	8,837	11,313
WebQuestions	3,417	2,474	361	2,032
CuratedTREC	1,353	1,125	133	694
SQuAD	78,713	70,096	8,886	10,570

Train : 훈련 데이터
Dev: 검증 데이터
Test: Test 데이터

Natural Question	*Gold 지문들을 그에 대응하는 지문들로 매칭하거나 교체, 매칭 실패 시 질문 버림*
Trivia QA	*질문 – 답변쌍만 제공하기 때문에 우리는 BM25로 만든, positive passage를 정답으로 가지고 있는 상위권 지문들을 사용한다. 100개의 검색된 지문 중 정답이 없다면 해당 질문은 버려진다.*
Web Questions	
Curated TPEC	
SQuAD	*Gold 지문들을 그에 대응하는 지문들로 매칭하거나 교체, 매칭 실패 시 질문 버림*

Experiments

Training	Retriever	Top-20					Top-100				
		NQ	TriviaQA	WQ	TREC	SQuAD	NQ	TriviaQA	WQ	TREC	SQuAD
None	BM25	59.1	66.9	55.0	70.9	68.8	73.7	76.7	71.1	84.1	80.0
Single	DPR	78.4	79.4	73.2	79.8	63.2	85.4	**85.0**	81.4	89.1	77.2
	BM25 + DPR	76.6	79.8	71.0	85.2	**71.5**	83.8	84.5	80.5	92.7	**81.3**
Multi	DPR	**79.4**	78.8	**75.0**	**89.1**	51.6	**86.0**	84.7	**82.9**	93.9	67.6
	BM25 + DPR	78.0	**79.9**	74.7	88.5	66.2	83.9	84.4	82.3	**94.1**	78.6

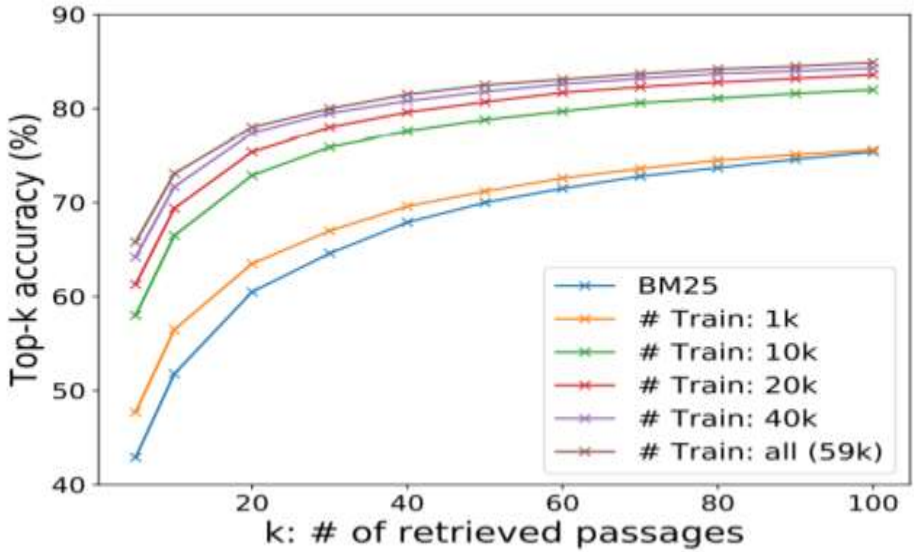
$$\mathrm{BM25}(q,p) + \lambda \cdot \mathrm{sim}(q,p)$$

- Single : 각각의 Dataset에 대하여 학습시킴
- Multiple : SQuAD를 제외한 4개의 Dataset을 합쳐서 학습시킴
- 40 epoch (NQ, TriviaQA, SQuAD)
- 100 epoch (TREC, WQ)
- dropout ratio : 0.1
- optimizer : Adam

- New ranking function을 활용하여 linear combination을 진행
- top-2000개의 passage를 뽑은 후, top-k개의 최종 passage를 선택

Experiments

Why does BM25 perform better in SQuAD dataset?



1. Annotation Bias

Question의 token들이 passage에 포함 되어 있을 확률이 높다.

2. Biased Examples

많이 보는 문서 약 500개로부터 만들어진 passage들로부터 가공된 dataset이기에

- **많은 training dataset을 활용할 수록 retrieval accuracy는 증가**
- **top-k 즉, retrieval할 최종 passage의 개수를 늘릴 수록 retrieval accuracy는 증가**

Experiments

Type	#N	IB	Top-5	Top-20	Top-100
Random	7	✗	47.0	64.3	77.8
BM25	7	✗	50.0	63.3	74.8
Gold	7	✗	42.6	63.1	78.3
Gold	7	✓	51.1	69.1	80.8
Gold	31	✓	52.1	70.8	82.1
Gold	127	✓	55.8	73.0	83.1
G.+BM25$^{(1)}$	31+32	✓	65.0	77.3	84.4
G.+BM25$^{(2)}$	31+64	✓	64.5	76.4	84.0
G.+BM25$^{(1)}$	127+128	✓	**65.8**	**78.0**	**84.9**

#N : Nagative Sample 개수
IB : In – Batch 적용여부
Top k – 상위 k개 retriever 정확도를 가지고 측정

- 별 2개 이상을 활용하는 것은 더 이상 도움이 안된다고 판단한다
- In-Batch Negative Sampling이 확실한 성능 개선이 있음을 보여준다.
- Batch_size가 커짐에 따라 성능이 개선이 있음을 보여준다.
- BM25+DPR을 사용하는 경우가 가장 성능이 좋음

Impact of gold passages

	Top-1	Top-5	Top-20	Top-100
Gold	44.9	66.8	78.1	85.0
Dist. Sup.	43.9	65.3	77.1	84.4

- Dist. Sup과 Gold 사이의 차이를 실험하여 진행하였으며 Gold Context를 활용하였을 때 보다 성능이 좋음
- Dist. Sup : BM25를 활용하여 context들 중에서 정답을 포함하면서 가장 확률이 높은 context를 ground-truth passage로 활용

Similarity and Loss

Sim	Loss	Retrieval Accuracy			
		Top-1	Top-5	Top-20	Top-100
DP	NLL	**44.9**	**66.8**	**78.1**	**85.0**
	Triplet	41.6	65.0	77.2	84.5
L2	NLL	43.5	64.7	76.1	83.1
	Triplet	42.2	66.0	**78.1**	84.9

- L2 norm은 DP(dot product)와 비슷한 성능을 내며, 이 둘 모두 cosine 유사도보다 높은 성능을 낸다.
- DP와 NLL을 활용하였을 때, 가장 좋은 성능

Cross – dataset generalization

Fine-tuning 없이도 잘 작동하는가?

Dense Passage Retrieval

- top-20 accuracy : WQ(75% -> 69.9%) / TREC(89.1% -> 86.3%) (Multiple 대비 NQ에서만 학습했을 때의 성능이다.)

Sparse Embedding

- BM25(WQ-55.0/TREC-70.9)

Qualitative Analysis

Dense Passage Retrieval

- 어휘적 변화나 의미적 관계를 더 잘 포착

Sparse Embedding

- BM25와 같은 Term-matching method의 경우, 매우 선택적인 keyword나 구절에 대해서는 매우 민감하게 반응

Run-time Efficiency

Dense Passage Retrieval

- 21백만개의 passage를 처리하는데 8개의 GPU로 병렬 처리하여 8.8시간이 걸린다
- FAISS index를 활용할 경우, 하나의 GPU server로 8.5시간
- 초당 995.0개 처리

Sparse Embedding

- 23.7개의 질문을 처리 Building and inverted index를 하는데 30분
- 초당 23.7개 처리

Experiments: Question Answering

End- to-end QA System

1. Retriever로부터 주어진 top-k passage들에 대하여 reader model이 최종 answer를 추출
2. reader는 각 passage들에 대하여 passage selection score를 부여
3. 각 passage로부터 answer span을 추출하고 span score를 부여
4. 가장 높은 passage selection score와 best span score를 가진 span이 최종 정답으로 선택
5. passage selection model : question과 passage 사이의 cross-attention을 통해 re-rank

How can you get the answer span?

$$P_{\text{start},i}(s) = \text{softmax}\left(\mathbf{P}_i \mathbf{w}_{\text{start}}\right)_s, \quad (3)$$

$$P_{\text{end},i}(t) = \text{softmax}\left(\mathbf{P}_i \mathbf{w}_{\text{end}}\right)_t, \quad (4)$$

$$P_{\text{selected}}(i) = \text{softmax}\left(\hat{\mathbf{P}}^{\intercal} \mathbf{w}_{\text{selected}}\right)_i, \quad (5)$$

where $\hat{\mathbf{P}} = [\mathbf{P}_1^{[\text{CLS}]}, \ldots, \mathbf{P}_k^{[\text{CLS}]}] \in \mathbb{R}^{h \times k}$ and $\mathbf{w}_{\text{start}}, \mathbf{w}_{\text{end}}, \mathbf{w}_{\text{selected}} \in \mathbb{R}^h$ are learnable vectors.

- Pi Pi는 i번째 passage에 대한 BERT representation 이며, (Lxh)를 가지며 L은 maximum length of the passage, h는 hidden dimension을 의미한다
- 정답 span의 처음에 위치할 확률과 마지막에 위치할 확률은 위의 그림과 같이 계산된다.
- Passage가 선택될 확률은 모든 passage들의 [CLS] token에 대한 hidden embedding vector와 학습가능한 vector w selected w selected와의 연산 통해 계산
- P selected P selected 에 의해서 어떤 passage가 선택될 것이니 구한 후, P start P start x P end P end 에 의해 span score가 계산

Reader Training

- Retriever로 부터 주어진 100개의 passage들 중에서 1개의 positivie passage와 m-1개의 negative passage를 sampling
- m은 hypter-parameter로서 m = 24로 활용하였다.
- Reader에서의 학습은 선택된 positive passage의 log-likehood와 함께 positive passage에서의 모든 정답 span의 Marginal log-likelihood를 최대화하는 방향으로 학습
- batch_size : 16(NQ, TriviaQA, SQuAD), 4(TREC, WQ)

ODQA 성능 비교

Training	Model	NQ	TriviaQA	WQ	TREC	SQuAD
Single	BM25+BERT (Lee et al., 2019)	26.5	47.1	17.7	21.3	33.2
Single	ORQA (Lee et al., 2019)	33.3	45.0	36.4	30.1	20.2
Single	HardEM (Min et al., 2019a)	28.1	50.9	-	-	-
Single	GraphRetriever (Min et al., 2019b)	34.5	56.0	36.4	-	-
Single	PathRetriever (Asai et al., 2020)	32.6	-	-	-	**56.5**
Single	$\text{REALM}_{\text{Wiki}}$ (Guu et al., 2020)	39.2	-	40.2	46.8	-
Single	$\text{REALM}_{\text{News}}$ (Guu et al., 2020)	40.4	-	40.7	42.9	-
Single	BM25	32.6	52.4	29.9	24.9	38.1
	DPR	**41.5**	56.8	34.6	25.9	29.8
	BM25+DPR	39.0	57.0	35.2	28.0	36.7
Multi	DPR	**41.5**	56.8	**42.4**	49.4	24.1
	BM25+DPR	38.8	**57.9**	41.1	**50.6**	35.8

- **retriever accuracy가 높을 수록 전체 ODQA 결과 역시 더 높다.**
- **여러 종류의 데이터셋을 혼합하여 학습을 진행한 결과 성능이 더 좋음**

감사합니다.