Dense Passage Retrieval for Open-Domain Question Answering

이번 게시물에서는 현재까지도 다양한 retrieval task에 사용되는 dense retriver인 DPR을 제안한 논문인 Dense Passage Retrieval for Open-Domain Question Answering 논문에 대해 다뤄보겠다

원문 링크는 아래와 같다
Link: https://arxiv.org/abs/2004.04906

Introduction

Open-domain question answering (ODQA) task는 large collection of document로부터 question에 대한 정보를 찾아 답변하는 task이다.

통상적으로, 이러한 ODQA task를 수행하는 framework는 retriever-reader 두 가지로 구성된다.

retriever의 경우, question에 대한 answer가 담겨있는 small subset of passage를 선택하여 retrieve하는 역할이고, reader는 retrieved context를 받아와 알맞은 answer를 산출하는 역할을 한다.

본 연구가 진행되었던 시기에는 retriever 부분에 TF-IDF와 BM25와 같은 방법론들이 주로 사용되었다. 저자들은 이러한 방법론이 아닌, dense encoding에 집중하였다. Dense encoding의 경우, 동음이의어나 의역과 같은 요소들을 잘 식별해낸다는 장점이 존재한다.

예를 들어, 아래와 같은 예제가 있다고 가정해보자

> Question : “Who is the bad guy in lord of the rings?”
> Answer : “Sala Baker is best known for portraying the villain Sauron in the Lord of the Rings trilogy”

위 예제에서, “악역”의 의미로 question에는 “bad guy”가 쓰였지만 answer에서는 “villain”이 사용되었다

TF-IDF, BM25와 같은 term-based 방법론의 경우, 같은 의미에 대한 term이 달라졌기에, 해당 answer가 포함된 passage를 잘 retrieve하지 못할 수 있다. 그러나 dense encoding을 기반으로 한 dense retriever의 경우, term이 달라져도 “bad guy”에 대한 representation과 “villain”에 대한 representation이 embedding space에서 가까운 위치에 있기 때문에 이를 잘 retrieve할 수 있다.

또한, dense encoding의 경우 learnable하다는 특징이 있기에, task-specific representation을 가지기에 충분한 flexibility를 가진다는 장점이 존재한다.

그러나, 좋은 dense vector representation을 학습하기 위해서는 수많은 question-context labeled pair data가 필요하다 여겨져 왔으며, 기존 연구들에서는 TF-IDF와 BM25보다 월등한 성능을 보이지도 않았다.

Inverse cloze task(ICT) objective로 dense retriever에 대해 additional pre-training을 진행한 ORQA에서 dense retriever의 성능이 BM25를 능가하긴 했으나, 여기에도 두 가지의 한계가 존재한다

- ICT pre-training의 경우, 너무 많은 계산량을 필요로 한다
- context encoder가 question-answer pair를 사용하여 fine-tune된것이 아니기에, 해당 encoder를 이용하여 산출되는 representation은 suboptimal하다

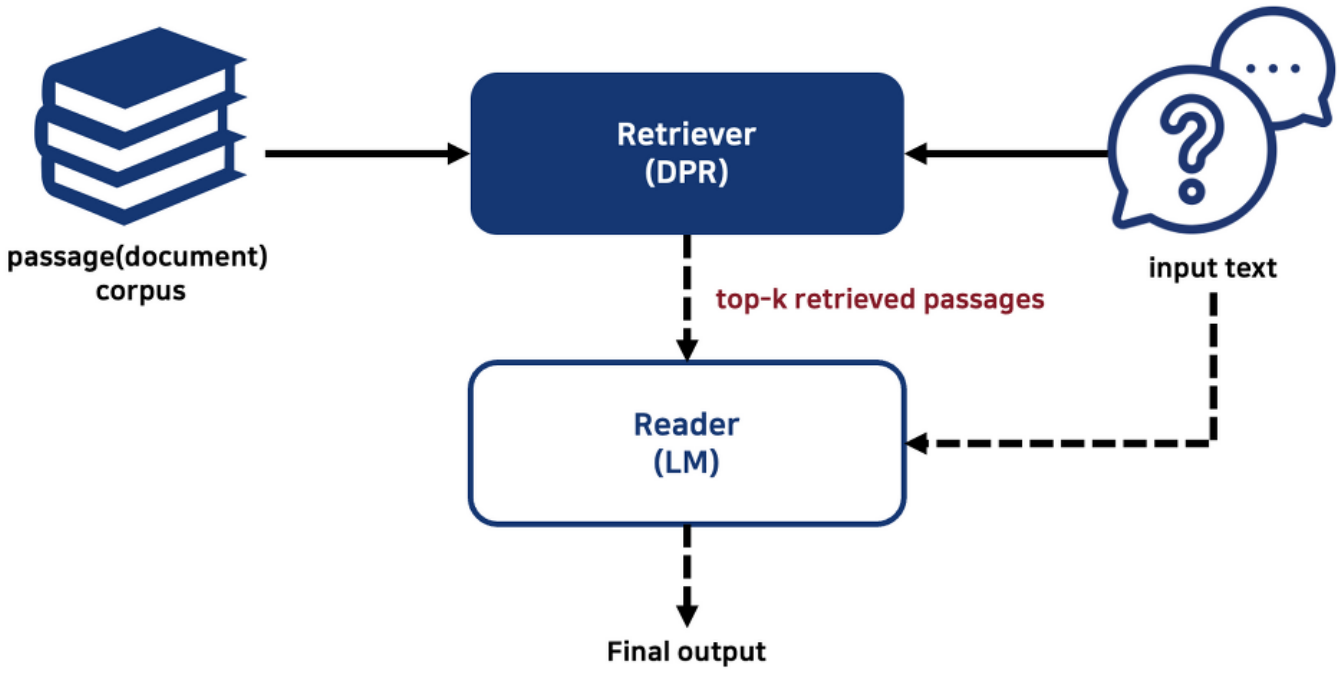
본 연구에서는 이러한 기존 방법론들의 한계를 해결하기 위해, 적은 수의 question-passage pair만을 가지고 학습을 진행하는 것에 초점을 두었다고 하며, 이를 통해 나온 dense retriever를 Dense Passage Retriever(DPR)이라고 명명한다

본 연구의 핵심 contribution은 아래와 같다

- Additional pre-training을 거치지 않고 question encoder와 passage encoder를 fine-tuning 시키기만 한 retriever로 BM25의 성능을 능가함을 보임
- retriever의 높은 성능이 end-to-end QA system의 높은 성능으로 이어진다는 것을 보임

Dense Passage Retriever(DPR)

본 연구에서 제안하는 dense passage retriever(DPR)는 open-domain QA task에서 사용되는 retriever이며, passage(document)들을 low-dimensional and continuous space에서 indexing하고, 이를 바탕으로 효과적으로 input text와 관련있는 top-k개의 passage를 retrieve함이 목적이다. 아래의 figure는 이 과정을 나타낸다.



Overview

DPR이 위의 목적을 달성하기 위해서는, 우선적으로 passage를 low-dimensional and continuous space로 mapping해주는 encoder가 필요하다. 논문에서는 해당 encoder를 dense encoder $E_p()$라고 한다. $E_p()$는 passage의 정보를 잘 담고 있는 passage representation을 만들어낸다

또한, input text도 representation으로 변환해야만 두 representation의 유사도를 측정하여 relevant top-k passage를 골라낼 수 있다. 이를 위해 input text를 input representation으로 만들어주는 encoder $E_Q()$를 둔다

이렇게 두 encoder를 통해 만들어진 두 representation의 유사도를 구하여 relevant top-k passage를 선정하게 된다. 본 연구에서는 아래와 같이 dot product로 두 representation간 유사도를 측정한다

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

저자들은 inner product 외에도 decomposable similarity function인 L2 norm, cosine similarity등에 대해 비교 실험을 진행하였다고 한다. 그 결과, 다른 similarity function들도 잘 작동한다는 것을 확인했지만, inner product가 가장 simple한 function이기에 inner product를 채택하였다고 한다

Encoder로는 BERT-base를 사용하였으며, passage encoder $E_p()$를 통해 산출된 passage representation들에 대해서는 FAISS indexing을 진행했다고 한다

Training

DPR을 학습시킨다는 것은, representation을 잘 만들어내는 encoder가 되게끔 학습을 시키는 것이다.

즉, question과 passage 쌍이 서로 관련있으면 smaller distance를 가지고, 관련이 없으면 bigger distance를 가지는 representation을 산출하는 encoder로 만들어간다는 것인데, 이를 위한 loss function은 아래와 같다

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

또한, m개의 instance로 구성된 Training Dataset $\mathcal{D}$는 $\mathcal{D} = \{\langle q_i, p_i^+, p_{i,1}^-, \ldots, p_{i,n}^- \rangle\}$이라고 하며, 위 수식에서 각각의 요소들은 아래와 같은 뜻을 가진다

- q_i : question
- p_i^+: relevant(positive) passage
- $p_{i,j}^-$: irrelevant(negative) passage

즉, instance 안의 passage중에서, positive passage에 대한 log-likehood가 높아질수록 loss function의 값은 작아지고, 이를 최소화하도록 학습하여 결과적으로는 positive passage에 대한 likelihood를 높이고, negative passage에 대한 likelihood를 낮추도록 학습하는 것이다.

그런데, 이 수식을 보다보면 question에 대한 negative passage를 어떻게 정할지에 대해 의문이 들게 된다.

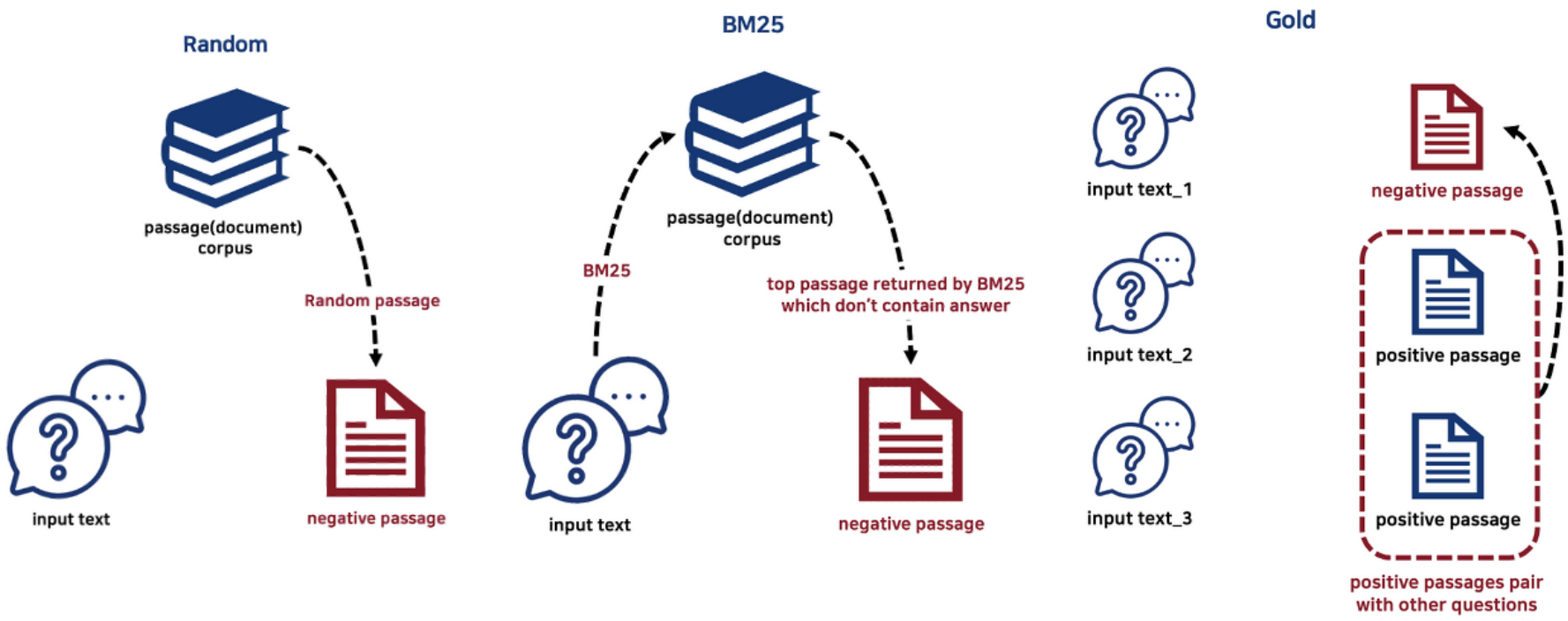
논문에서는 이에 대한 여러 옵션을 제안하고, 뒷부분에서 비교 실험을 진행하였다고 한다. 그렇다면, 지금부터는 논문에서 제안한 옵션들에 대해 살펴보도록 하겠다

Positive and negative passages

우선, 기본적으로 논문에서 제안하는 negative passage sampling 옵션은 아래의 3가지이다.

1. Random : passage corpus로부터 무작위로 추출
2. BM25 : BM25를 사용하여 return된 passage 중에서, answer를 포함하고 있진 않지만, question과 match가 많이 된 passage
3. Gold : training set에서, 다른 question의 positive passage들을 negative passage로 사용

이들에 대한 전반적인 과정을 나타내는 figure는 아래와 같다



이에 대한 비교 실험 결과, 같은 mini-batch 안에 있는 다른 question의 positive passage들을 negative passage로 사용하고(Gold), 하나의 BM25 passage를 더 해준 조합이 가장 성능이 좋았다고 한다.

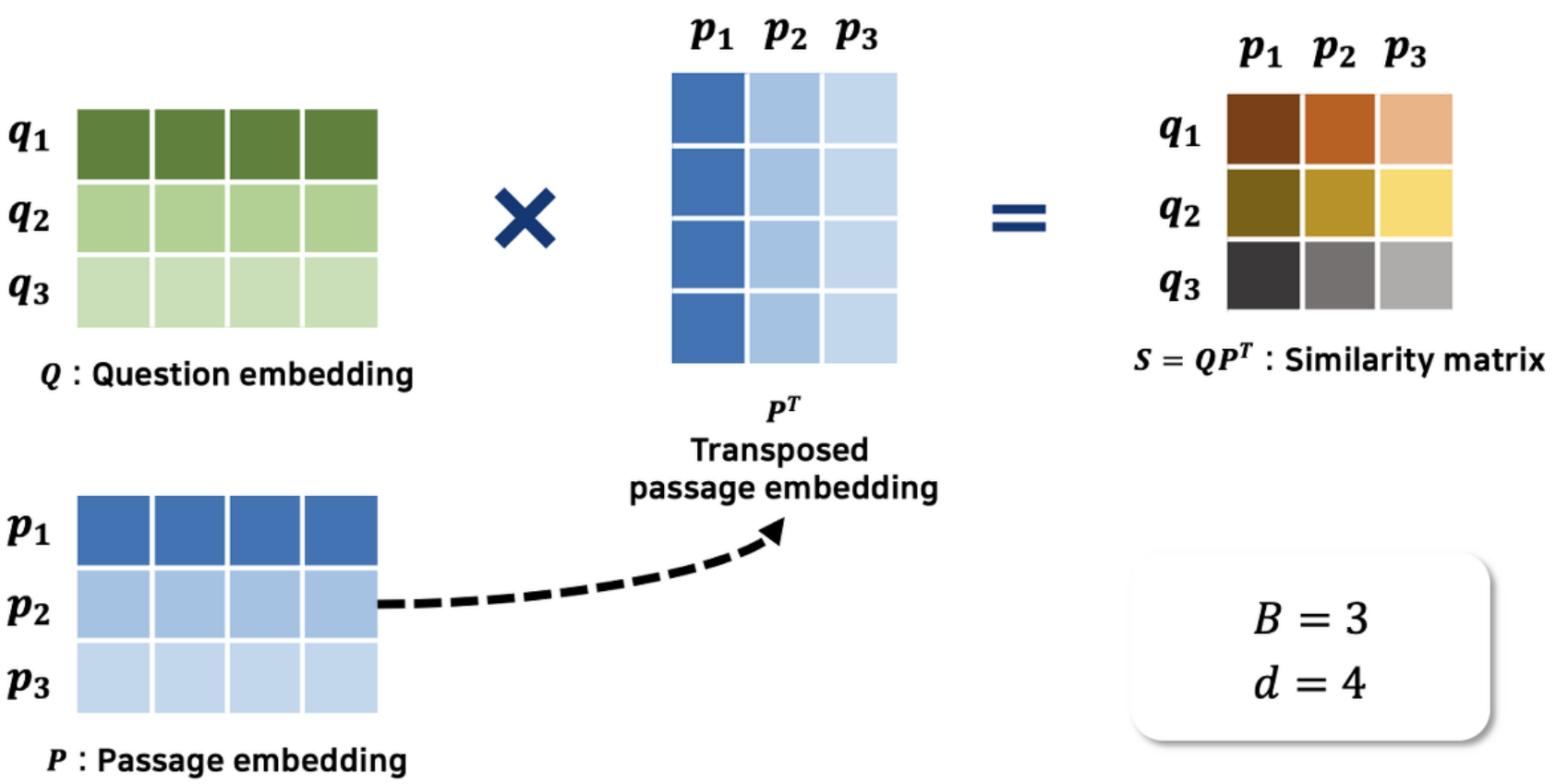
In-batch negatives

하나의 mini-batch 안에 B개의 question이 있다고 가정할 때, 각각의 question은 positive(relevant) passage를 가진다.

이때, Q와 P를 각각 $(B \times d)$ 의 question embedding matrix, passage embedding matrix라고 하자.

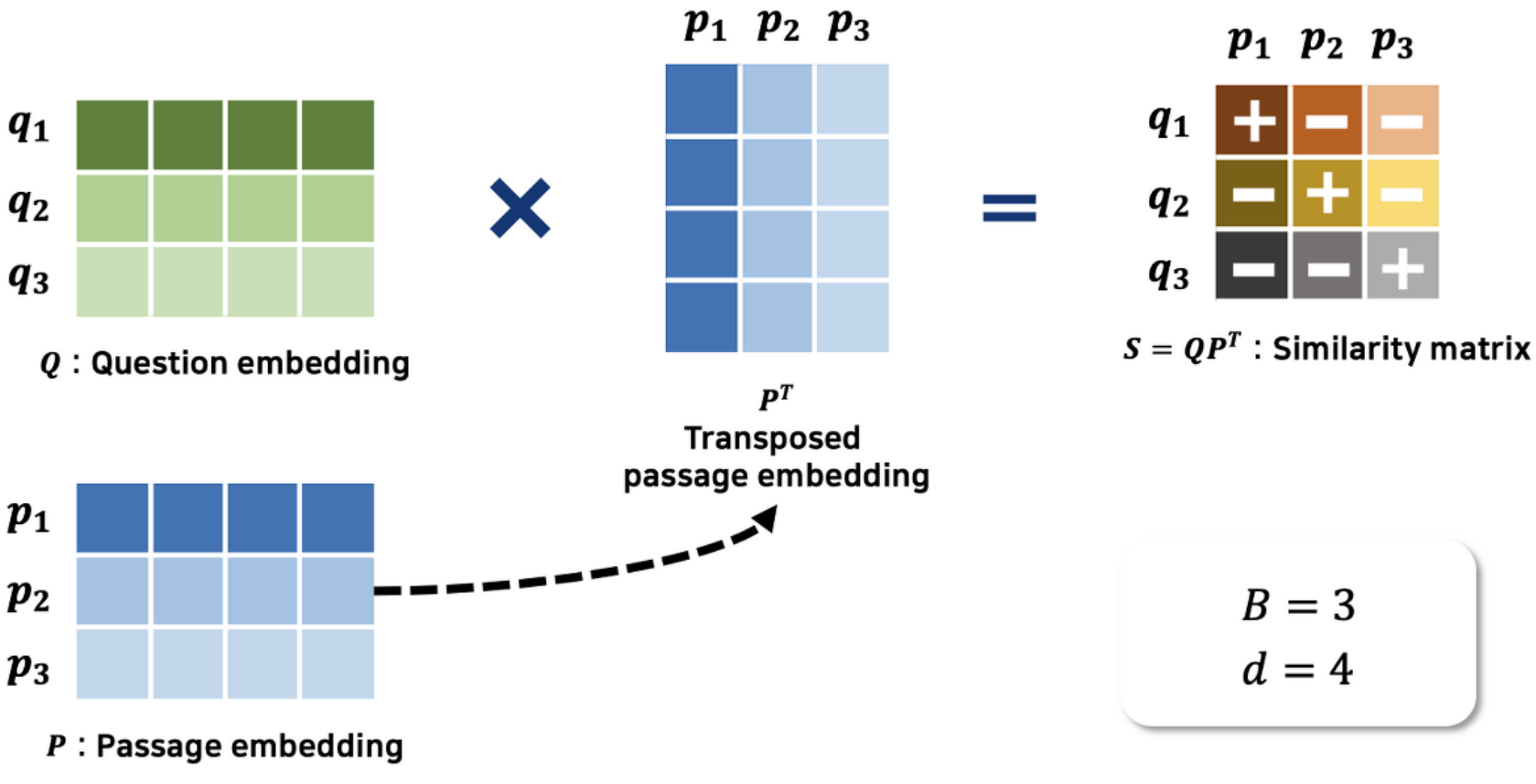
그렇다면 $S = QP^T$는 $(B \times B)$의 similarity score matrix라고 볼 수 있게 된다.

아래는 해당 과정을 나타내는 figure이다



이때, 각각의 question과 passage를 q_i, p_i라고 할 때, $i = j$이면 positive(relevant) passage, $i \neq j$이면 negative(irrelevant) passage라고 할 수 있다

이를 적용하여 figure를 다시 만들어보면 아래와 같은 결과가 나온다



저자들은 이렇게 구해진 similarity matrix를 학습 시 재사용하여, 효율적으로 mini-batch 안에 있는 question-passage pair에 대해 학습하였다고 한다.

이러한 in-batch training은 각각의 question에 대해 $B-1$개의 negative passage를 제공할 수 있다

Experimental Setup

그럼 이제, 위의 DPR을 어떠한 setup으로 실험을 진행했는지 알아보자

Wikipedia Data Pre-processing

우선, passage는 wikipedia data를 기반으로 한다. Dec. 20, 2018 버전의 Wikipedia dump를 기반으로 preprocessing을 진행하여 passage set을 구축하였다

저자들은 DrQA에서 사용한 pre-processing code를 사용하여 semi-structured data와 disambiguation page들을 제거했다고 한다.

이후, 100 words를 기준으로 passage를 구축하여 총 21,015,324개의 passage를 구축하였다고 한다

각각의 passage들은 해당 passage가 기인한 wikipedia 문서의 title이 prepend되었으며, 이때 title과 passage 본문은 [SEP] token으로 구분했다고 한다.

Question Answering Dataset

저자들은 아래와 같은 dataset을 사용했다고 한다

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

Table 1: Number of questions in each QA dataset. The two columns of **Train** denote the original training examples in the dataset and the actual questions used for training DPR after filtering. See text for more details.

Selection of positive passage

그런데, 여기에서 TREC, WebQuestions, TriviaQA dataset에는 passage가 주어지지 않고, 오로지 question과 answer만 주어지기에, 저자들은 BM25를 통해 positive passage labeling을 진행하였다.

question과 구축한 wikipedia passage에 대해 BM25를 적용시켜, answer를 포함하고 있는 highest-ranked passage를 positive passage로 labeling하였다.

다만, answer를 포함하고 있는 passage가 나오지 않는 경우도 있는데, 저자들은 이러한 경우엔 top 100 retrieved passage 모두에서 answer가 없을 경우, 해당 question은 폐기하였다고 한다.

SQuAD와 NaturalQuestions의 경우, 기본적으로 passage까지 주어지지만, 해당 dataset들에서 passage를처리한 과정이 본 연구와는 달랐기에, 두 dataset에 대해서도 positive passage에 대해 match and replace를 진행하였다고 한다.

그 결과, 위의 table에서 확인할 수 있는것처럼 preprocessing 전과 후의 question 개수가 달라진 것을 확인할 수 있다.

Experiments: Passage Retrieval

저자들은 먼저 기존 BM25와 같은 traditional retrieval method에 비해 DPR이 얼마나 성능 향상을 이끌어내는지에 대한 실험을 진행하였다.

해당 실험에서 사용된 DPR model에 대한 정보는 아래와 같다

- trained using the in-batch negative setting
- batch size : 128
- trained both encoder for up to 40 epochs for large dataset(NQ, TriviaQA, SQuAD)
- trained both encoder for up to 100 epochs for small dataset(TREC, WQ)
- learning rate 10^{-5}
- optimizer : Adam
- dropout : 0.1

또한, 개별 dataset에 fine-tuning을 시킨 이후 해당 dataset에 대해 성능 평가를 하는 것 이외에도, 여러 dataset을 혼합하고, 해당 mixed dataset으로 훈련시킨 encoder 하나만으로 모든 dataset에 대해 성능 평가를 진행하였다고 한다. (단, SQuAD는 training dataset에서 제외됨)

이러한 training schema들에 대해 전자는 single, 후자는 multi라고 명명되었다

추가적으로, 각 setting에 대해 DPR 단독으로 사용했을때와, BM25와 DPR을 같이 사용하는 setting에 대해서도 성능 측정을 하였다. 이런 경우네는 우선 DPR과 BM25 각각 top-2000 passage를 retrieve한 다음, 아래의 수식을 이용하여 rerank를 진행하여 최종적으로 retrieve될 passage를 산출하였다고 한다.

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

(본 연구에서는 $\lambda = 1.1$을 사용했다고 한다)

Main Result

이에 대한 결과는 아래와 같다

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

Table 2: Top-20 & Top-100 retrieval accuracy on test sets, measured as the percentage of top 20/100 retrieved passages that contain the answer. *Single* and *Multi* denote that our Dense Passage Retriever (DPR) was trained using individial or combined training datasets (all the datasets excluding SQuAD). See text for more details.

SQuAD dataset을 제외한 모든 dataset에서 DPR이 BM25에 비해 더 좋은 성능을 내는 것을 확인할 수 있다.

조금 더 자세히 들여다보자. Multiple dataset으로 학습한 경우, TREC과 같이 작은 크기의 dataset에서 성능 향상의 효과가 나타나는 것을 확인할 수 있다. 그와는 반대로, NQ나 WebQuestion과 같이 보다 큰 크기의 dataset에서는 Single setting에 비해 성능이 하락하는 것을 확인할 수 있다.

또한, BM25와 DPR 혼합 setting의 경우, 몇몇 dataset에서만 성능 향상이 있었음을 확인할 수 있다.

그런데, DPR이 유독 SQuAD dataset에 대해서는 성능이 좋지 않다. 저자들은 이에 대해 아래의 두 가지 이유일것이라 추측한다

- SQuAD의 annotator들이 passage를 본 뒤에 question을 작성했기 때문에, passage와 question 사이의 어휘 중복이 많이 일어났다. 이러한 점이 BM25에게 더 유리하다
- SQuAD dataset은 단지 500개 가량의 wikipedia article들에서 수집되었기 때문에, training example들이 extremely biased되어서이다

Ablation Study on Model Training

이어서, 저자들은 model 학습에 대해 비교 실험을 진행하였다.

Sample efficiency

먼저, 저자들은 good passage retrieval performance를 얻기 위해 얼마나 많은 training example이 필요한지에 대한 실험을 진행하였다. 해당 실험의 결과는 아래와 같다

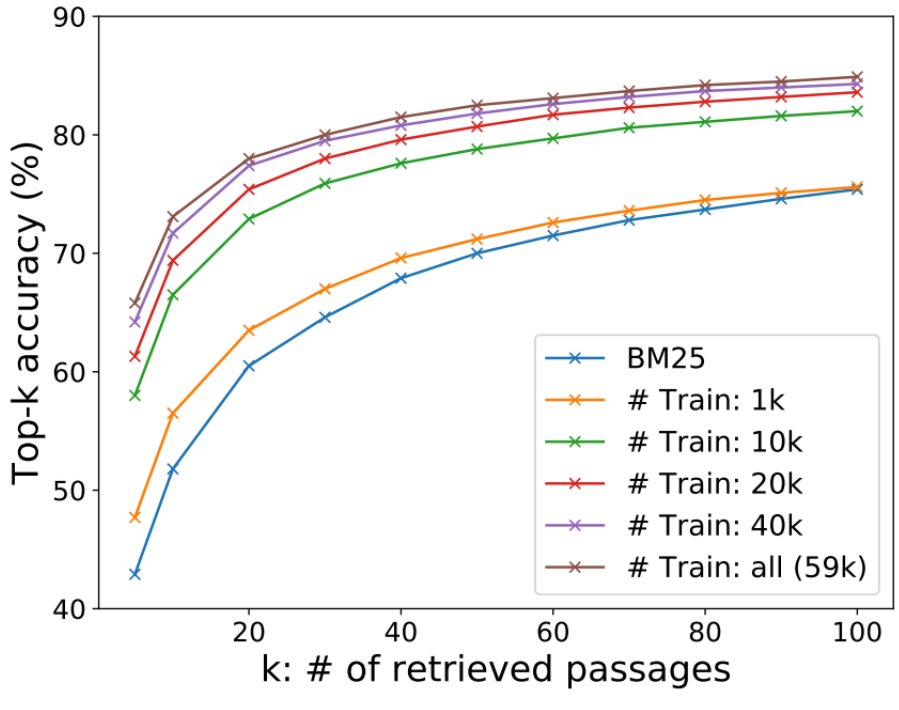


Figure 1: Retriever top-k accuracy with different numbers of training examples used in our dense passage retriever vs BM25. The results are measured on the development set of Natural Questions. Our DPR trained using 1,000 examples already outperforms BM25.

단지 1000개의 training example을 사용한 순간부터도 DPR이 BM25의 성능을 능가함을 확인할 수 있다. 이러한 결과는 작은 수의 question-passage pair로도 high-quality dense retriever를 학습시킬 수 있다는 점을 시사한다.

또한, training example을 추가할수록 성능을 계속 증가하는 것을 확인할 수 있다.

In-batch negative training

이어서, 저자들은 각각 다른 training schemes들에 대해서 NQ dataset의 dev set으로 성능 측정 및 비교를 진행하였다. 해당 결과는 아래와 같다

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**

Table 3: Comparison of different training schemes, measured as top-k retrieval accuracy on Natural Questions (development set). #N: number of negative examples, IB: in-batch training. G.+BM25$^{(1)}$ and G.+BM25$^{(2)}$ denote in-batch training with 1 or 2 additional BM25 negatives, which serve as negative passages for all questions in the batch.

결과 표를 보면, 3개의 block으로 나눠져있음을 확인할 수 있다.

이는 다음과 같은 기준으로 분류되었다

- Top block : standard 1-of-N training setting
- Middle block : Gold with In-batch negative setting

- Bottom block : Gold with In-batch negative setting + 1 or 2 BM25 passage

먼저 Top block 결과부터 살펴보겠다. Top-5에서는 BM25 sampling이 가장 좋은 결과를 산출했지만, Top-20부터는 세 방법론 모두 비슷한 성능을 내는 것을 확인할 수 있다.

그러나, middle block을 보면, top-block의 gold #7 setting에 비해 in-batch training을 적용한 gold #7 setting의 성능이 매우 좋아진 것을 확인할 수 있다

이어서, in batch이기때문에 batch size가 늘어날수록 전체 negative passage의 개수도 늘어난다. 따라서, batch size, 즉 negative example의 개수를 늘릴수록 성능 향상이 이루어지는 것도 확인할 수 있다.

마지막으로, bottom block을 살펴보도록 하겠다. 해당 block은 batch에 존재하는 각각의 question마다 1개 혹은 2개의 BM25 negative passage(hard negative passage)를 추가해준 block이다.

이러한 hard negative를 추가하는 것이 성능 향상에 도움을 줬지만, 2개를 추가했을때는 오히려 1개만 추가했을때보다 성능 하락이 있음을 확인할 수 있다. 또한 middle block과 마찬가지로 batch size, 즉 negative example의 개수를 늘릴수록 성능 향상이 이루어지는 것도 확인할 수 있다.

Cross-dataset generalization

저자들은 DPR이 학습된 dataset이 아닌, unseen dataset에서도 잘 작동하는지, 즉, generalization 성능이 어떤지에 대해서도 실험을 진행하였다.

이를 위해 DPR을 NQ dataset으로 학습한 뒤 WebQuestion, CuratedTREC dataset에 대해 성능 측정을 하였다

해당 실험 결과, 해당 dataset으로 fine-tuned된 DPR에 비해 top-20 retrieval accuracy에서 3~5 point정도 성능 하락이 있었다고 한다.

- WebQuestions : 75.0 → 69.9
- CuratedTREC : 89.1 → 86.3

논문에서는 이에 대해 성능 하락이 있었지만, 어느정도 잘 동작하며, 무엇보다 BM25의 성능보다 우월한 성능을 가진다고 이야기한다

Qualitative Analysis

저자들은 BM25와 DPR이 retrieve하는 passage의 성격 차이에 대해서도 이야기한다.

BM25는 selective keyword에 더 민감하게 반응하고, DPR은 의미적 관계나 어휘적 변형을 더 잘 포착했다고 한다. 아래는 두 retriever가 같은 question에 대해 retrieve한 passage의 예시이다

Question	Passage received by BM25	Passage retrieved by DPR
What is the body of water between England and Ireland?	Title:British Cycling . . . **England** is not recognised as a region by the UCI, and there is no English cycling team outside the Commonwealth Games. For those occasions, British Cycling selects and supports the **England** team. Cycling is represented on the Isle of Man by the Isle of Man Cycling Association. Cycling in Northern **Ireland** is organised under Cycling Ulster, part of the all-Ireland governing **body** Cycling **Ireland**. Until 2006, a rival governing **body** existed, . . .	Title: Irish Sea . . . Annual traffic between Great Britain and **Ireland** amounts to over 12 million passengers and of traded goods. **The Irish Sea** is connected to the North Atlantic at both its northern and southern ends. To the north, the connection is through the North Channel between Scotland and Northern **Ireland** and the Malin Sea. The southern end is linked to the Atlantic through the St George's Channel between Ireland and Pembrokeshire, and the Celtic Sea. . . .
Who plays Thoros of Myr in Game of Thrones?	Title: No One (Game of Thrones) . . . He may be "no one," but there's still enough of a person left in him to respect, and admire who this girl is and what she's become. Arya finally tells us something that we've kind of known all along, that she's not no one, she's Arya Stark of Winterfell." "No One" saw the reintroduction of Richard Dormer and **Paul Kaye**, who portrayed Beric Dondarrion and **Thoros** of **Myr**, respectively, in the third season, . . .	Title: Pål Sverre Hagen Pål Sverre Valheim Hagen (born 6 November 1980) is a Norwegian stage and screen actor. He appeared in the Norwegian film "Max Manus" and played Thor Heyerdahl in the Oscar-nominated 2012 film "Kon-Tiki". Pl Hagen was born in Stavanger, Norway, the son of Roar Hagen, a Norwegian cartoonist who has long been associated with Norways largest daily, "VG". He lived in Jtten, a neighborhood in the city of Stavanger in south-western Norway. . . .

Table 7: Examples of passages returned from BM25 and DPR. Correct answers are written in **blue** and the content words in the question are written in bold.

첫 번째 행은 DPR이 더 잘 retrieve한 경우이고, 두 번째는 BM25가 더 잘 retrieve한 경우이다.

첫 번째 예시의 경우, BM25는 England나 Ireland와 같은 keyword에만 집중하여 잘못된 passage를 가져온 반면, DPR은 body of water의 의미적인 관계를 포착하여 Irish Sea passage를 잘 가져왔다

두 번째 예시의 경우, Thoros of Myr라는 키워드가 중요한 문제였는데, DPR은 이것을 잘 포착하지 못한 반면 BM25는 이러한 요소를 잘 포착함을 확인할 수 있다.

Experiments: Question Answering

마지막으로, DPR을 포함한 다양한 retriever를 사용하여 ODQA task를 진행하였다

결과는 아래와 같다

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

Table 4: End-to-end QA (Exact Match) Accuracy. The first block of results are copied from their cited papers. $\text{REALM}_{\text{Wiki}}$ and $\text{REALM}_{\text{News}}$ are the same model but pretrained on Wikipedia and CC-News, respectively. *Single* and *Multi* denote that our Dense Passage Retriever (DPR) is trained using individual or combined training datasets (all except SQuAD). For WQ and TREC in the *Multi* setting, we fine-tune the reader trained on NQ.

DPR을 사용한 ODQA system이 SQuAD를 제외한 모든 dataset에서 가장 좋은 성능을 낸 것을 확인할 수 있다.