Unsupervised Dense Information Retrieval with Contrastive Learning

Authors: Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, Edouard Grave

Venue: TMLR 2022

1. Introduction

- Document Retriever(DR)란?
  - 특정 query에 답하기 위해 large collection에서 관련 document를 찾는 작업
  - 전통적으로는 TF-IDF 또는 BM25와 같은 어휘 유사성을 사용
    → 이런 방법은 일반화가 어려움
  - 신경망에 기반한 방법은 어휘 유사성 이상의 학습이 가능

- Collection에 수백만 또는 수십억 개의 요소가 포함되어 있으면, query와 관련 document를 일치시키기가 어려움
  - 해결 방법: **dense retriever**
    - 제안 방법 1: zero-shot
      - 하나의 방법으로 large dataset에 학습시키고 새로운 domain에 적용
      - 하지만 이런 방법은 TF와 같은 classic한 방법보다 성능이 낮고, 영어 이외에 annotated large dataset이 없음
    - 제안 방법2: **unsupervised learning**
      - 문서가 주어지면 합성 쿼리를 생성한 다음, 쿼리가 주어진 여러 다른 문서 중에서 원본 문서를 검색하도록 network를 훈련시킴

1. Introduction

- 이 연구의 **contribution**
  - Contrastive learning이 unsupervised retriever에 잘 됨을 보임
  - Few-shot 환경에서도 제안 모델이 좋은 성능을 보임
  - MS MARCO에서 미세조정하기 전 사전학습 방법으로 사용해 BEIR benchmark에서 높은 성능을 보임

  - 최종적으로, contrastive learning을 통해 multilingual dense retriever를 학습하고 SOTA를 달성함

2. Method

- Retrieve의 목적: 주어진 query에 대해 large collection에서 관련 document 찾기

- Document와 query가 독립적으로 인코딩되는 bi-encoder architecture 사용
  - Input: document와 query
  - Output: 각 document에 대한 관련성 점수

  - Query와 document 간의 관련성 점수
    - 인코더를 적용한 후, 두 representation 사이의 dot 곱
      1. Query q와 document d와 동일 모델 f_θ가 있을 때, q와 d는 f_θ 를 사용해 각각을 독립적으로 인코딩함
      2. 그렇게 나온 representation의 dot 곱을 q와 d 의 관련성 점수 $s(q,d)$라고 함

$$s(q,d) = \langle f_\theta(q), f_\theta(d) \rangle$$

→ 유사도라고 생각하면 됨

2. Method

- Unsupervised training 방법으로 대조학습(contrastive learning) 사용
- 앞서 구한 관련성 점수로 대조학습의 loss function을 구현

- Contrastive learning
  - Query q가 주어졌을 때, 연관된 positive document k_+와 negative documents(i=1,..,K) k_i
  - 이 때, loss function은 아래 식과 같음

$$\mathcal{L}(q, k_+) = -\frac{\exp(s(q, k_+)/\tau)}{\exp(s(q, k_+)/\tau) + \sum_{i=1}^{K} \exp(s(q, k_i)/\tau)}$$

  - q와 k_+ pair는 높은 점수를 받도록, q와 k_i는 낮은 점수를 받도록 학습

2. Method

- Single document에서 Positive pair 구축
  - Positive pair는 (view1, view2)로 구성
    - View1는 query, view2는 key

  - 이 연구에서 고려한 positive pair 구축 방법들
    - ICT(Inverse Cloze Task)
      - Text segment에서 토큰의 범위를 random하게 샘플링해서 view1을 얻고, 그 범위의 나머지로 view2를 얻음
        1. 즉, $(w_1, w_2, \dots, w_n)$의 text sequence가 주어졌을 때, ICT는 span $(w_a, \dots, w_b)$를 샘플링함.
        2. 이 때, $1 \leq a \leq b \leq n$
        3. Span $(w_a, \dots, w_b)$ 는 view1이 됨.
        4. 그 외 $(w_1, w_2, \dots, w_{a-1}, w_{b+1}, \dots, w_n)$이 view2가 됨

    - Independent cropping
      - 문서에서 두 개의 span을 독립적으로 샘플링해서 positive pair를 생성
      - View1, view2 사이의 overlap을 만듦으로써 network가 query와 document 사이의 정확한 match를 학습하도록 함.
        - 이런 방식은 BM25의 어휘 매칭 방법과 유사

    - Additional data augmentation
      - 무작위 단어 삭제, 교체, 마스킹과 같은 데이터 증강 방법

2. Method

- **많은 수의 negative pair 구축**
  - 이 연구에서 고려한 negative pair 구축 방법들
    - **In-batch Negatives**
      - 같은 batch 에 있는 다른 example을 negative로 사용
      - Gradient는 query와 key 둘 다의 representation을 통해 back-propagation됨
      - Batch size가 클 수록 잘 됨
    - **MoCo**
      - 이전 batch에서 나온 representation을 queue에 저장하고 그걸 negative로 사용
      - 작은 batch로도 많은 negative 사용이 가능
      - Gradient는 query로만 back-propagation되고, key의 representation은 고정
        - Key의 representation은 느리게 업데이트 되는 두번째 network(momentum encoder)에서 생성
      - Query network의 parameter는 in-batch negative와 유사하게 back-propagation과 stochastic gradient descent로 업데이트 됨
      - Key network(momentum encoder)의 parameter는 아래 식으로 업데이트 됨

$$\theta_k \leftarrow m\theta_k + (1-m)\theta_q$$

3. Experiments

- 이 연구에서는 **Contriever** 모델을 제안
  1. Random cropping을 사용해 positive pair 샘플링, pair의 각 요소의 token을 10% 확률로 제거
  2. MoCo 방식 사용
  3. Wikipedia CCNet data를 학습에 사용

- **학습 데이터**
  - Wikipedia와 CCNet data가 배치에 각각 반반씩 들어가게 구성

- **평가 데이터**
  - 두 개의 QA dataset: **NaturalQuestions, TriviaQA**
    - 검색 document는 2018년 12월 20일의 English Wikipedia 사용
    - recall@k를 평가 척도로 사용

  - **BEIR benchmark**
    - nDCG@10, Recall@100을 평가 척도로 사용
    - nDCG@10
      - Top 10개의 검색 문서 중 ranking에 초점을 맞춤
      - 검색 엔진 등 사람에게 return되는 순위를 평가하는 데에 적합함
    - recall@100
      - 100개의 검색 문서들 중 얼마나 많은 관련 문서가 있는지에 대한 비율
      - QA와 같은 머신러닝 시스템에 사용되는 retriever를 평가하는 데에 적합함

3. Experiments

3-1. fully unsupervised model의 성능 비교

- MS MARCO 또는 다른 annotated data로 미세조정하지 않음

- **QA dataset에서 성능 비교**
  - Contriever는 baseline인 BM25와 비교했을 때, NaturalQuestions의 R@100에서 약 3점정도 더 높음
  - ICT, Masked Salient Spans 보다 높은 성능을 보임

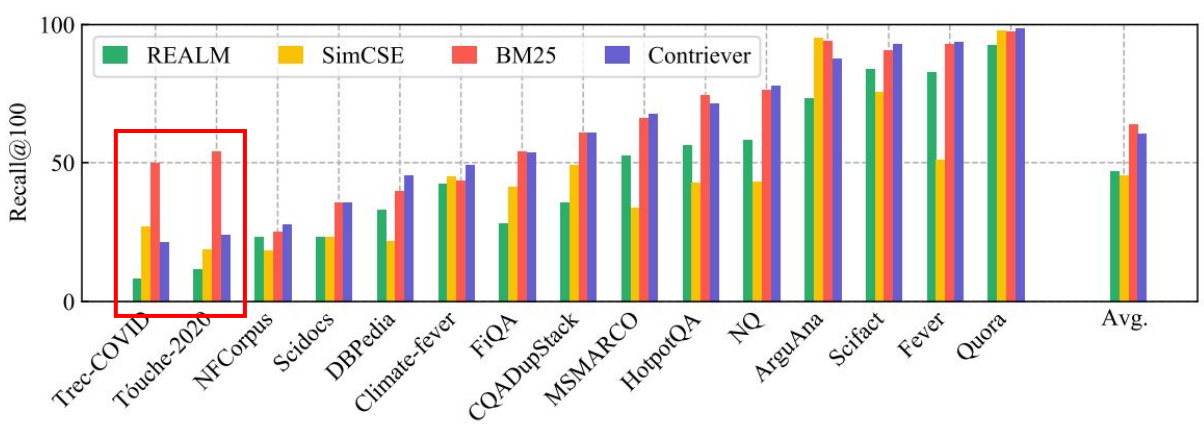
	NaturalQuestions			TriviaQA		
	R@5	R@20	R@100	R@5	R@20	R@100
Inverse Cloze Task (Sachan et al., 2021)	32.3	50.9	66.8	40.2	57.5	73.6
Masked salient spans (Sachan et al., 2021)	41.7	59.8	74.9	53.3	68.2	79.4
BM25 (Ma et al., 2021)	-	62.9	78.3	-	**76.4**	**83.2**
Contriever	**47.8**	**67.8**	**82.1**	**59.4**	74.2	**83.2**
supervised model: DPR (Karpukhin et al., 2020)	-	78.4	85.4	-	79.4	85.0
supervised model: FiD-KD (Izacard & Grave, 2020a)	73.8	84.3	89.3	77.0	83.6	87.7

3. Experiments

3-1. fully unsupervised model의 성능 비교

- **BEIR benchmark 에서 성능 비교**
  - Contriever는 대부분의 dataset에서 BM25와 비슷
    - 특히, 15개의 dataset 중 11개의 dataset에서는 BM25보다 높음
    - Trec-COVID, Touche-2020에서는 BM25가 월등히 높음
      - Trec-COVID
        - 코로나19와 관련된 정보 검색 dataset
        - Contreiver 훈련에 사용한 data는 코로나19가 발생하기 전에 수집되었기 때문에 적용되지 않았을 수 있음
      - Touche-2020
        - 긴 문서가 포함되어 있어 dense neural retriever에서는 잘 안 될 수 있음

- SimcSE, RE



3. Experiments

3-2. MS MARCO로 학습해 BEIR benchmark에서 성능 비교

- MS MARCO로 학습한 ML-based retriever를 비교
  - 비교 대상인 ML-based retrieve의 세 가지 타입: sparse, dense, late-interaction
    - Sparse
      - Splade v2: BERT 사전학습 모델로 document의 sparse representation을 계산
    - Dense
      - DPR, ANCE: unsupervised data로 학습한 bi-encoder
      - TAS-B: cross-encoder에서 bi-encoder로 증류 기법 사용
      - GenQ: 생성 모델로 합성 query-document의 pair 생성
    - Late-Interaction
      - Late-interaction은 미리 document에 대한 모든 임베딩을 구축해둬 적은 계산량을 가진다는 특징이 있음
      - ColBERT: query와 document의 맥락화된 표현 사이의 쌍별 점수를 계산

3. Experiments

3-2. MS MARCO로 학습해 BEIR benchmark에서 성능 비교

- "CE"는 cross-encoder를 의미
- 대조학습을 사전학습으로 사용할 경우 높은 성능을 보임
- Contriever는 dense bi-encoder 중 가장 좋은 성능을 보였고, recall@100에서는 SOTA 달성

	BM25	BM25+CE	DPR	ANCE	TAS-B	Gen-Q	ColBERT	Splade v2	Ours	Ours+CE
MS MARCO	22.8	41.3	17.7	38.8	40.8	40.8	40.1	43.3	40.7	**47.0**
Trec-COVID	65.6	**75.7**	33.2	65.4	48.1	61.9	67.7	71.0	59.6	70.1
NFCorpus	32.5	**35.0**	18.9	23.7	31.9	31.9	30.5	33.4	32.8	34.4
NQ	32.9	53.3	47.4	44.6	46.3	35.8	52.4	52.1	49.8	**57.7**
HotpotQA	60.3	70.7	39.1	45.6	58.4	53.4	59.3	68.4	63.8	**71.5**
FiQA	23.6	34.7	11.2	29.5	30.0	30.8	31.7	33.6	32.9	**36.7**
ArguAna	31.5	31.1	17.5	41.5	42.9	**49.3**	23.3	47.9	44.6	41.3
Touche-2020	**36.7**	27.1	13.1	24.0	16.2	18.2	20.2	36.4	23.0	29.8
CQADupStack	29.9	37.0.	15.3	29.6	31.4	34.7	35.0	-	34.5	**37.7**
Quora	78.9	82.5	24.8	85.2	83.5	83.0	85.4	83.8	**86.5**	82.4
DBPedia	31.3	40.9	26.3	28.1	38.4	32.8	39.2	43.5	41.3	**47.1**
Scidocs	15.8	16.6	7.7	12.2	14.9	14.3	14.5	15.8	16.5	**17.1**
FEVER	75.3	**81.9**	56.2	66.9	70.0	66.9	77.1	78.6	75.8	**81.9**
Climate-FEVER	21.3	25.3	14.8	19.8	22.8	17.5	18.4	23.5	23.7	**25.8**
Scifact	66.5	68.8	31.8	50.7	64.3	64.4	67.1	**69.3**	67.7	69.2
Avg. w/o CQA	44.0	49.5	26.3	41.3	43.7	43.1	45.1	50.6	47.5	51.2
Avg.	43.0	48.6	25.5	40.5	42.8	42.5	44.4	-	46.6	50.2
Best on	1	3	0	0	0	1	0	1	1	9

nDCG@10

	BM25	DPR	ANCE	TAS-B	Gen-Q	ColBERT	Splade v2	Ours
MS MARCO	65.8	55.2	85.2	88.4	88.4	86.5	-	**89.1**
Trec-COVID	**49.8**	21.2	45.7	38.7	45.6	46.4	12.3	40.7
NFCorpus	25.0	20.8	23.2	28.0	28.0	25.4	27.7	**30.0**
NQ	76.0	88.0	83.6	90.3	86.2	91.2	**93.0**	92.5
HotpotQA	74.0	59.1	57.8	72.8	67.3	74.8	**82.0**	77.7
FiQA	53.9	34.2	58.1	59.3	61.8	60.3	62.1	**65.6**
ArguAna	94.2	75.1	93.7	94.2	**97.8**	91.4	97.2	97.7
Touche-2020	**53.8**	30.1	45.8	43.1	45.1	43.9	35.4	29.4
CQADupStack	60.6	40.3	57.9	62.2	65.4	62.4	-	**66.3**
Quora	97.3	47.0	98.7	98.6	98.8	98.9	98.7	**99.3**
DBPedia	39.8	34.9	31.9	49.9	43.3	46.1	**57.5**	54.1
Scidocs	35.6	21.9	26.9	33.5	33.2	34.4	36.4	**37.8**
Fever	93.1	84.0	90.0	93.7	92.8	93.4	**95.1**	94.9
Climate-fever	43.6	39.0	44.5	53.4	45.0	44.4	52.4	**57.4**
Scifact	90.8	72.7	81.6	89.1	89.3	87.8	92.0	**94.7**
Avg. w/o CQA	63.6	48.3	60.1	65.0	64.2	64.5	64.8	67.1
Avg.	63.4	47.7	60.0	64.8	64.2	64.3	-	67.0
Best on	2	0	0	0	1	0	4	7

recall@100

3. Experiments

3-3. few-shot 환경에서 성능 비교

- 적은 수의 example 사용
  - 이럴 경우, BM25와 같은 lexical based method는 작은 training set을 활용해 가중치를 조정할 수 없음

- BEIR benchmark에서 성능 측정
  - 작은 dataset에서 Contriever 방식의 사전학습이 BERT 사전학습보다 높은 성능을 보임
  - MS MARCO dataset으로 미세조정 했을 때도, Contreiver 가 더 높은 성능을 보임
  - Contriever는 BM25보다도 높은 성능을 보임
    - few-shot 환경에서는 lexical method보다 dense retriever가 더 좋다는 것을 보임

	Additional data	SciFact	NFCorpus	FiQA
# queries		729	2,590	5,500
BM25	-	66.5	32.5	23.6
BERT	-	75.2	29.9	26.1
Contriever	-	84.0	33.6	36.4
BERT	MS MARCO	80.9	33.2	30.9
Contriever	MS MARCO	**84.8**	**35.8**	**38.1**

4. Multilingual Retrieval

- 영어로 제공되는 large labeled dataset은 있지만, low-resource language에서는 그렇지 않음
- 이런 경우에는 unsupervised training 방법이 좋음

- 이 연구에서 **multilingual model인 mContriever**를 제안함
  - Contriever와 방법은 동일하지만 pre-training data와 hyperparameter가 다름
  - 104개의 언어로 학습된 BERT인 mBERT로 초기화
  - 29개의 언어로 사전학습
    - Pre-training data로는 CCNet에 포함된 언어를 고려함
    - 사전 학습 중 example은 언어에 걸쳐 균일하게 샘플링함
    - Queue size는 32768

  - 사전학습 후 미세조정
    - MS MARCO에 대해 사전학습된 mContriever 모델을 미세조정함

4. Multilingual Retrieval

- 평가
  - 두 개의 벤치마크에서 영어 데이터에 대해 미세조정한 것과 하지 않은 것의 성능을 평가
    - 두 개의 벤치마크: Mr.TyDi, MKQA
    - Mr.TyDi
      - TyDi QA에서 파생된 multilingual 정보검색 벤치마크
      - 질문이 주어지면 같은 언어의 Wikipedia 문서 pool에서 관련 문서를 찾는 것을 목표로 함
    - MKQA
      - 언어 간 검색 성능 평가 목적
      - 특정 언어로 된 질문이 주어지면 영어 위키백과에서 검색하고, 검색된 문서에 영어 답변이 있는지를 평가
      - MKQA dataset은 MKQA에서 파생된 cross-lingual retrieval benchmark로, 26개의 언어로 동일한 질문과 답변을 제공
        - 답변할 수 없는 질문, 예/아니오 로만 대답할 수 있는 질문, 답변이 긴 질문은 dataset에서 제거했음
        - 그 결과, 6619개의 query 평가 셋이 생성됨
  - contrastive pre-training 후 미세조정한 mContriever+MS MARCO와 contrastive pre-training을 하지 않고 mBERT로 초기화된 mBERT+MS MARCO를 비교
    - Contrastive pre-training의 효과를 확인하기 위해

4. Multilingual Retrieval

- 평가 - Mr.TyDi
  - MRR@100은 첫 번째 문서가 실제 중 몇 번째에 위치하는지를 나타내는 척도로, 첫번째 문서의 품질에 중점을 두는 metric
  - mBERT+MS MARCO보다 mContriever+MS MARCO가 더 좋은 성능을 보임
    - Multilingual pre-training의 효과를 입증함
  - 오직 영어 데이터로만 학습했는데 모든 언어에서 성능 개선을 보임
  - recall@100에서는 BM25의 성능을 능가함
    - MS MARCO로 미세조정한 후에는 더 높은 성능 향상을 보임
  - Mr.TyDi로 미세조정했을 때 성능이 더욱 향상되었음
    - 특히 MRR@100에서 그 차이가 더 큼

	ar	bn	en	fi	id	ja	ko	ru	sw	te	th	avg
						MRR@100						
BM25 (Zhang et al., 2021)	36.7	41.3	15.1	28.8	38.2	21.7	28.1	32.9	39.6	42.4	41.7	33.3
mDPR (Zhang et al., 2021)	26.0	25.8	16.2	11.3	14.6	18.1	21.9	18.5	7.3	10.6	13.5	16.7
Hybrid (Zhang et al., 2021)	49.1	53.5	28.4	36.5	45.5	35.5	36.2	42.7	40.5	42.0	49.2	41.7
mBERT + MS MARCO	34.8	35.1	25.7	29.6	36.3	27.1	28.1	30.0	37.4	39.6	20.3	31.3
XLM-R + MS MARCO	36.5	41.7	23.0	32.7	39.2	24.8	32.2	29.3	35.1	54.7	38.5	35.2
mContriever	27.3	36.3	9.2	21.1	23.5	19.5	22.3	17.5	38.3	22.5	37.2	25.0
+ MS MARCO	43.4	42.3	27.1	35.1	42.6	32.4	34.2	36.1	51.2	37.4	40.2	38.4
+ Mr. Tydi	**72.4**	**67.2**	**56.6**	**60.2**	**63.0**	**54.9**	**55.3**	**59.7**	**70.7**	**90.3**	**67.3**	**65.2**
						Recall@100						
BM25 (Zhang et al., 2021)	80.0	87.4	55.1	72.5	84.6	65.6	79.7	66.0	76.4	81.3	85.3	74.3
mDPR (Zhang et al., 2021)	62.0	67.1	47.5	37.5	46.6	53.5	49.0	49.8	26.4	35.2	45.5	47.3
Hybrid (Zhang et al., 2021)	86.3	93.7	69.6	78.8	88.7	77.8	70.6	76.0	78.6	82.7	87.5	80.9
mBERT + MS MARCO	81.1	88.7	77.8	74.2	81.0	76.1	66.7	77.6	74.1	89.5	57.8	76.8
XLM-R + MS MARCO	79.9	84.2	73.1	81.6	87.4	70.9	71.1	74.1	73.9	91.2	89.5	79.7
mContriever	82.0	89.6	48.8	79.6	81.4	72.8	66.2	68.5	88.7	80.8	90.3	77.2
+ MS MARCO	88.7	91.4	77.2	88.1	89.8	81.7	78.2	83.8	91.4	96.6	90.5	87.0
+ Mr. Tydi	**94.0**	**98.6**	**92.2**	**92.7**	**94.5**	**88.8**	**88.9**	**92.4**	**93.7**	**98.9**	**95.2**	**93.6**

4. Multilingual Retrieval

- 평가 – MKQA
  - CORA를 비교 대상 중 하나로 고려함
    - CORA: Cross-lingual Open-Retrieval Answer Generation

  - mContriever+MS MARCO가 CORA retriever보다 높은 성능을 보임
  - Mr.TyDi의 결과와 유사하게 mContriever+MS MARCO가 mBERT+MS MARCO보다 높은 성능을 보임

	Avg. R@20	Avg. R@100	en	ar	ja	ko	es	he	de
CORA (Asai et al., 2021)	49.0	59.8	**75.6**	44.5	47.0	45.5	69.2	48.3	**68.1**
mBERT + MS MARCO	45.3	57.9	74.2	44.0	51.7	48.2	63.9	46.8	59.6
XLM-R + MS MARCO	46.9	59.6	73.4	42.5	53.2	49.6	63.4	46.9	61.1
mContriever	31.4	49.2	65.3	43.0	47.1	44.8	37.2	44.7	49.0
+ MS MARCO	**53.9**	**65.6**	**75.6**	**53.3**	**60.4**	**55.4**	**70.0**	**59.6**	66.6

5.
Conclusion

- 이 연구에서는 MoCo를 사용한 unsupervised contrastive learning 방식의 Contriever를 제안함

- 이 연구에서는 몇 가지 흥미로운 점을 발견했음
  1. Contrastive learning을 사용해 supervision없이 학습한 신경망은 BM25와 경쟁할 수 있을 만큼 우수한 검색 성능을 보였음
  2. MS MARCO dataset으로 미세조정함으로써 성능을 더 개선할 수 있음. 특히 recall@100의 경우 높은 성능을 얻을 수 있음
  3. Cross-encoder를 사용해 모델로 검색된 문서의 순위를 재조정해 BEIR benchmark의 nDGG@10에서 SOTA를 달성

Thank You

감사합니
다.