



DSBA Open QA Study

Open-Domain Question Answering Paper Review #1



고려대학교 산업경영공학과

Data Science & Business Analytics Lab

발표자 : 고유경

1. Reading Wikipedia to Answer Open-Domain Questions (Chen et al., ACL 2017, 1154회 인용)
2. R³: Reinforced Ranker-Reader for Open-Domain Question Answering (Wang et al., AAAI 2018, 261회 인용)

Reading Wikipedia to Answer Open-Domain Questions

Danqi Chen*

Computer Science
Stanford University
Stanford, CA 94305, USA
danqi@cs.stanford.edu

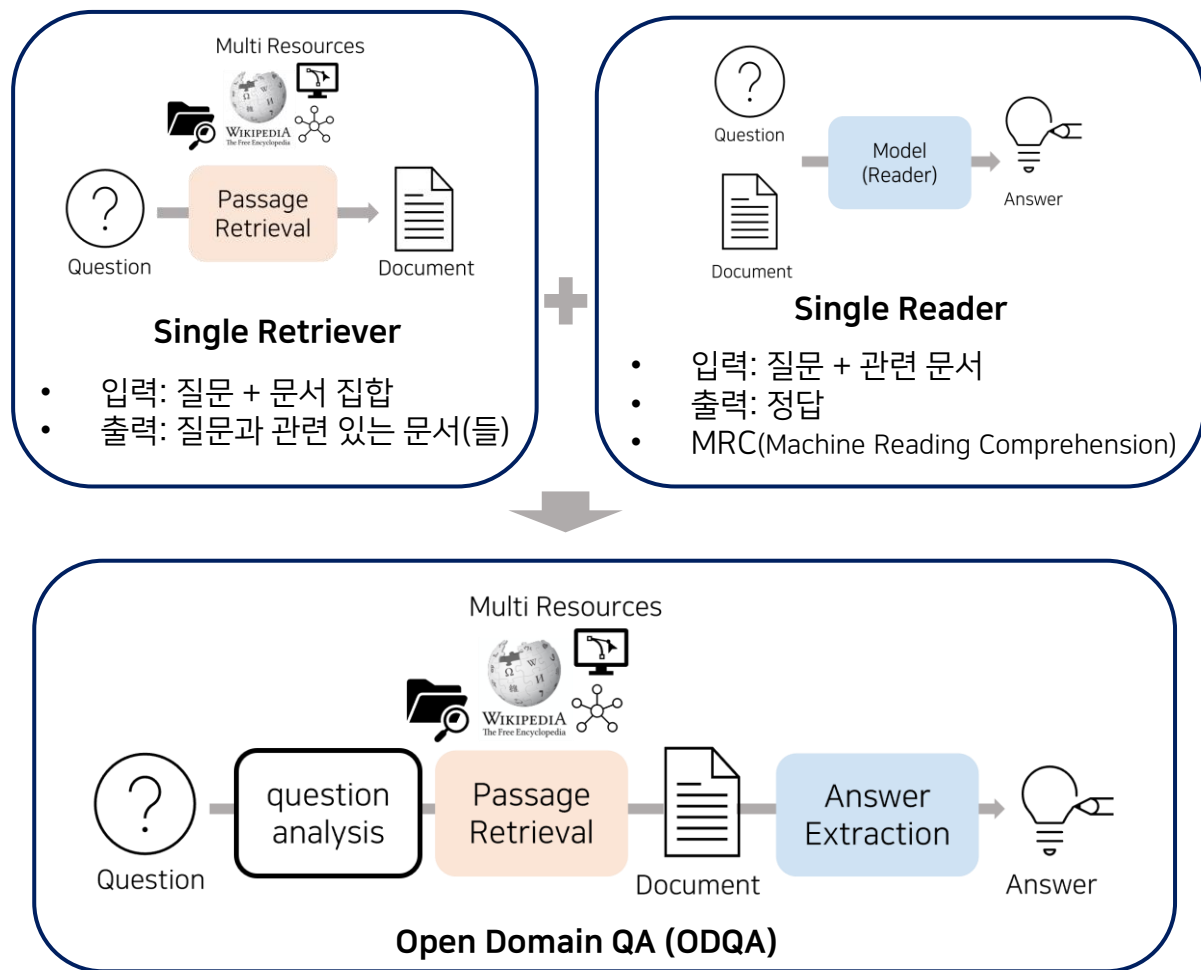
Adam Fisch, Jason Weston & Antoine Bordes

Facebook AI Research
770 Broadway
New York, NY 10003, USA
{afisch, jase, abordes}@fb.com

1. Reading Wikipedia to Answer Open-Domain Questions (Chen et al., ACL 2017, 1154회 인용)

- Retriever (질문과 관련 있는 문서 검색) + Reader (정답 추출)

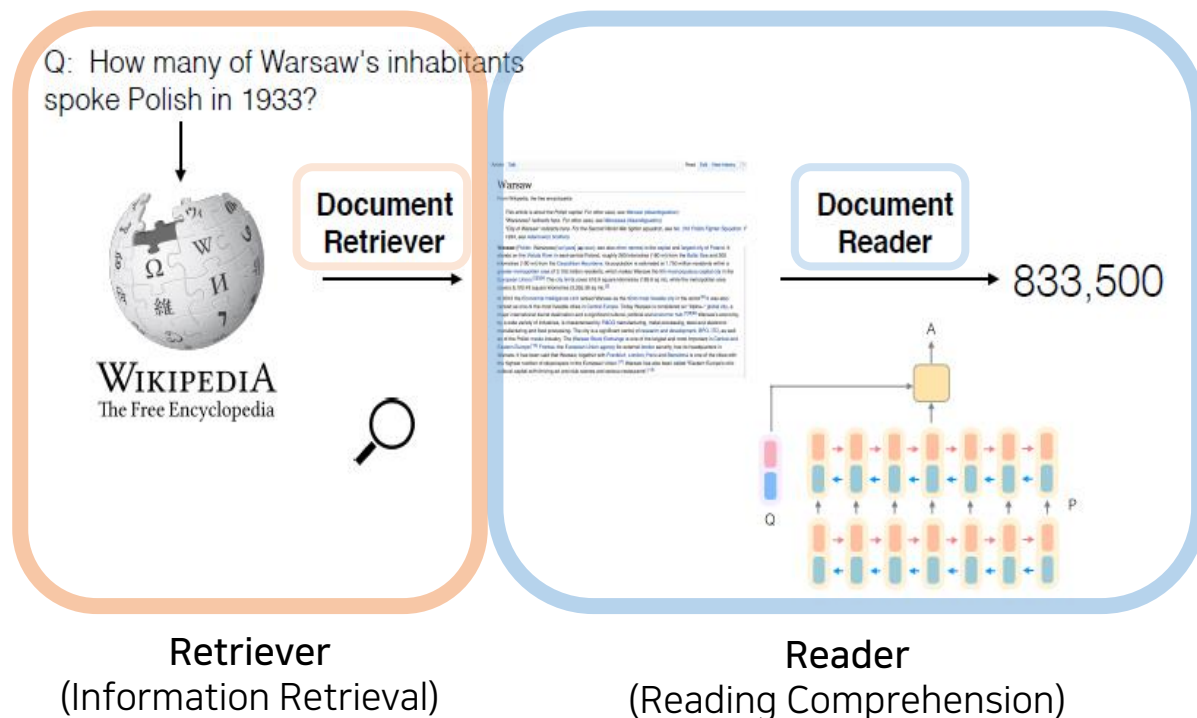
Task Description: Open Domain QA



기존 Open Domain QA 연구

- unstructured documents 집합에서 질문의 정답 추출 (TREC, 1999)
- 질문 분석, 문서 검색(Retriever), 정답 추출(Reader) 순으로 진행
- Unconstrained QA: using multi/redundant resources
 - 컴퓨터가 처리하기 용이한 KB와 같은 structured data도 활용
 - (-) but too sparsely populated -> **single resource!**
- DeepQA(2010): unstructured(text), structured(KB)
- YodaQA(2015): web, database, wikipedia
- DL 기반의 MRC 방법론들의 비약적 발전 -> **ODQA에 적용해보자!**

Task Description: Open Domain QA



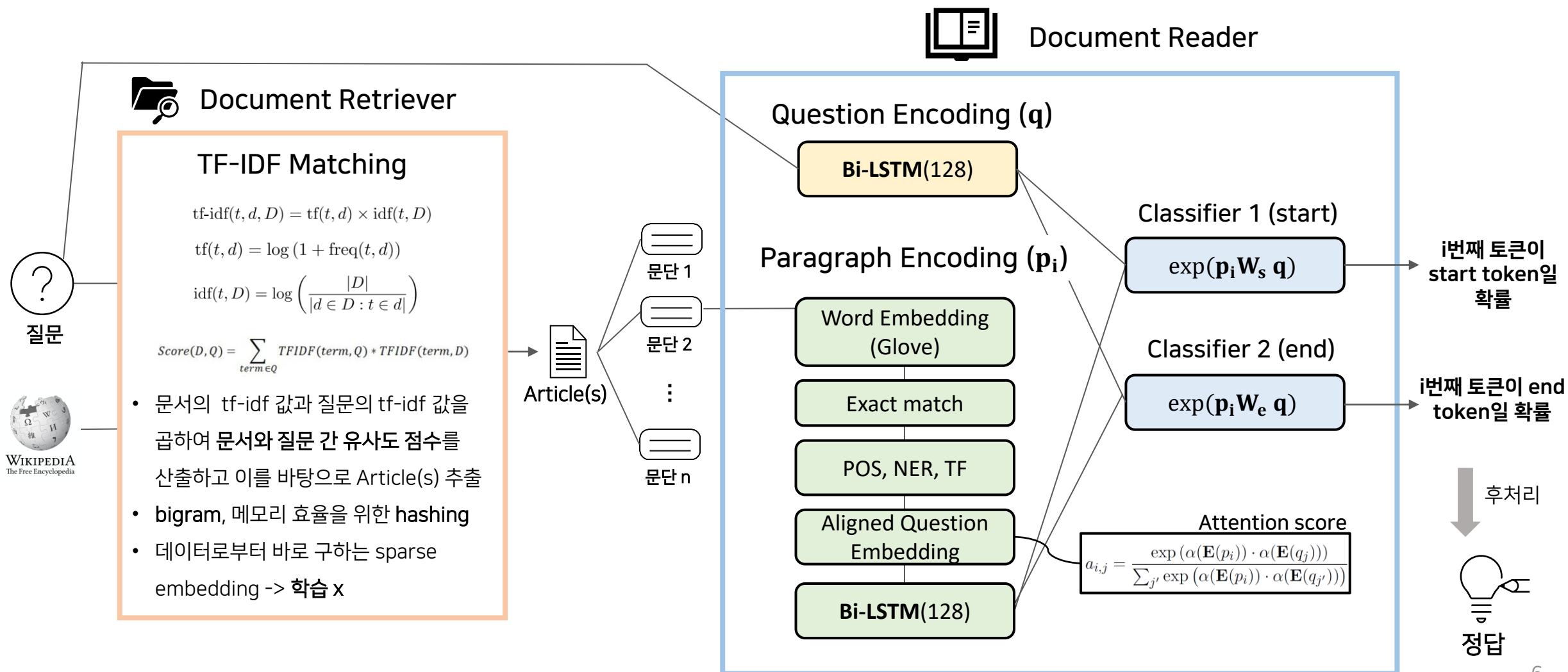
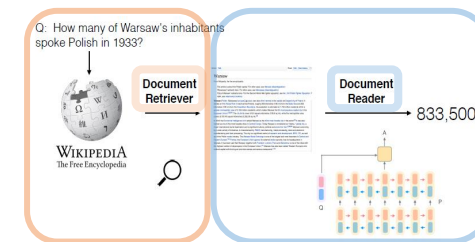
논문이 제안하는 방법

DrQA: 2-Stage Retriever-Reader

- 규모가 큰 single resource(only Wikipedia)만을 가지고 더 정교하고 완전한 ODQA 시스템을 구축해보자!
- (+) generic: 위키피디아 내부 그래프 구조에 의존하지 않아(only text) 다른 문서 집합에도 적용 가능
- more challenging - MRC(reader) 중요
 - distant supervision (MRC용 데이터 생성)
 - multitask learning (여러 데이터셋으로 학습)

1. **Document Retriever:** 통계 기반 전통 IR 모델 (학습 x)
 - 입력: 질문 + Wikipedia articles 500만개
 - 출력: 질문과 관련 있는 문서(들)
2. **Document Reader:** RNN 기반 DL 모델 (학습 o)
 - 입력: 질문 + retrieved된 문서(들)
 - 출력: 문서/문단 내 정답 위치 (start, end)

Method: DrQA



Dataset Description

- ✓ **Wikipedia**: 정답을 찾는데 활용하는 Knowledge source, 500만개 Articles
- ✓ **SQuAD**: (문단, 질문, 정답) 쌍으로 구성, Reader를 학습시키는데 활용하는 main resource
- ✓ **CuratedTREC, WebQuestions, WikiMovies**
 - 질문-정답 쌍으로만 구성되어 있음 -> Reader 모델 학습에 활용하기 위해 passage 데이터 추가
 - **Distant Supervision**: Retriever를 통해 top5 Wikipedia article 추출하고 이 중 정답과 가장 많이 겹치는 top5 passage 추출

Evaluation

평가지표: EM(Exact Match) / F1

1. Document Retriever

- 평가지표: 데이터 중 retrieved 문서가 정답을 포함하는 비율
- bigram: 위키피디아 검색엔진(62.7)보다 좋은 결과(77.8)

2. Document Reader

- baseline: R-net, BiDAF 등
- SQuAD로 학습한 모델 SOTA 기록 (EM: 69.5, F1: 78.8)

3. Full Wikipedia ODQA setting (EM)

Dataset	e	YodaQA	DrQA		
			SQuAD	+Fine-tune (DS)	+Multitask (DS)
SQuAD	(All Wikipedia)	n/a	27.1	28.4	29.8
CuratedTREC		31.3	19.7	25.7	25.4
WebQuestions		39.8	11.8	19.5	20.7
WikiMovies		n/a	24.5	34.3	36.5

- SQuAD: SQuAD로 reader 학습
- Fine-tune(DS): SQuAD로 reader 학습 + DS 데이터셋으로 fine-tuning
- **Multitask(DS)** 모든 데이터셋으로 jointly training

Best "complete" single model

The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

R³: Reinforced Ranker-Reader for Open-Domain Question Answering

**Shuohang Wang,^{1*} Mo Yu,² Xiaoxiao Guo,² Zhiguo Wang,² Tim Klinger,²
Wei Zhang,² Shiyu Chang,² Gerald Tesauro,² Bowen Zhou,³ Jing Jiang¹**

¹School of Information System, Singapore Management University

²AI Foundations - Learning, IBM Research AI, Yorktown Heights NY, USA

³JD.COM, Beijing, China

shwang.2014@smu.edu.sg, yum@us.ibm.com, xiaoxiao.guo@ibm.com

2. R³: Reinforced Ranker-Reader for Open-Domain Question Answering (Wang et al., AAAI 2018, 261회 인용)

- Retriever(문서 검색) + Ranker(문서 선택) + Reader(정답 추출)

Task Description: Open Domain QA with Ranker

기존 연구의 한계: 부실한 Retriever

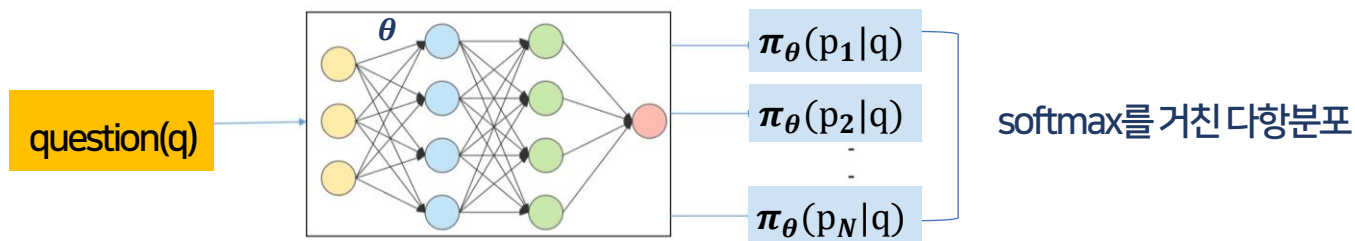
- tf-idf를 활용한 sparse embedding은 n-gram matching에 의존함 (P1)
- semantic한 정보를 파악하기 어려움 (P3)
- Retriever이 부실하면 뒷단 Reader의 학습이 어려워짐

Q:	What is the largest island in the Philippines?
A:	Luzon
P1	Mindanao is the second largest and easternmost island in the Philippines.
P2	As an island, Luzon is the Philippine's largest at 104,688 square kilometers, and is also the world's 17th largest island.
P3	Manila, located on east central Luzon Island, is the national capital and largest city.

논문이 제안하는 개선 방법: Ranker

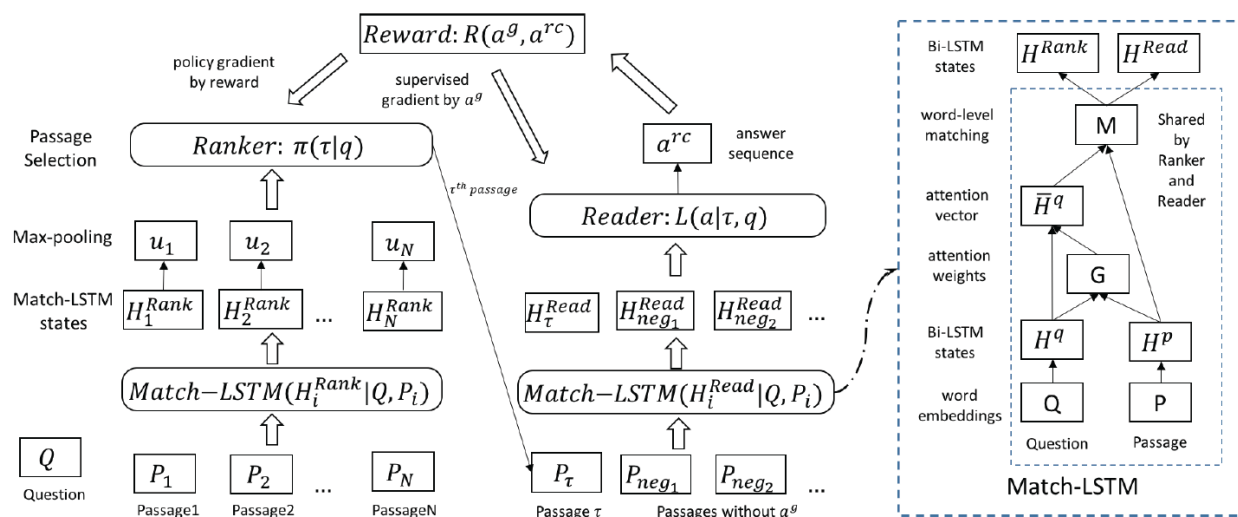
- retrieved된 문서들 중에 중요한(정답을 포함할 것 같은) passage를 뽑아내는 ranker를 추가하자
- 어떻게? 강화학습(Policy Gradient)을 활용!

Policy Gradient: state(질문)를 입력으로 받아 딥러닝 모형으로 action을 취할(passage를 선택할) 확률인 Policy $\pi_{\theta}(a|s)$ 를 추정



보상(reward)을 통해 해당 질문에서는 어떤 passage를 선택하는 것이 최적의 policy인지 스스로 배워나감(try and error)

- ranker와 reader를 jointly하게 학습시킴 (end-to-end)

Method: Reinforced Ranker-Reader(R³)

✓ Match-LSTM

- Ranker와 Reader를 거치기 전 단계
- 주어진 question과 각 passage 간의 관계를 바탕으로 passage의 representation을 도출하는 attention mechanism

$$\mathbf{G} = \text{SoftMax}((\mathbf{W}^g \mathbf{H}^q + \mathbf{b}^g \otimes \mathbf{e}_Q)^T \mathbf{H}^p)$$

Algorithm (batch_size = 1)

1. question q 에 대해 Retriever 모델로 추출된 top N 개의 passages 중 10개 샘플링 (최소 2개는 negative sample)
2. Ranker policy $\pi(\tau|q)$ 에 따라 passage 1개 선택(action)
3. Reader 모델을 통해 정답 a^{rc} 추출, reward(r) 획득
4. policy gradient로 Ranker 모델 업데이트
5. Reader 파라미터 업데이트(SGD)

$$L(a^g|\tau, q) = -\log(\beta_{a_\tau^g}^s) - \log(\beta_{a_\tau^e}^e)$$

$$\frac{\partial}{\partial \Theta} L(a^g|\tau, q)$$

✓ Reward

- reader가 추출한 정답이 실제 정답과 얼마나 잘 매칭되는가

$$R(a^g, a^{rc}|\tau) = \begin{cases} 2, & \text{if } a^g == a^{rc} \\ f1(a^g, a^{rc}), & \text{else if } a^g \cap a^{rc} \neq \emptyset \\ -1, & \text{else} \end{cases}$$

Dataset Description

- **Quasar-T**: 43000개의 퀴즈 (질문-답) 쌍
- **SQuAD(OPEN)**: (질문-답-문단) 쌍, passage 제거
- **WikiMovies**: 영화 관련 (질문-답) 쌍, 질문(from OMDb/MovieLens), 정답(from 위키피디아)
- **CuratedTREC**: based on TREC, for ODQA를 위한 TREC 기반 데이터셋, (질문-답) 쌍
- **WebQuestions**: Knowledge-base QA 데이터셋, (질문-답) 쌍

Distant Supervision: 아래 4개 데이터셋에 대해 sentence-level로 적용, top 200 문서 추출 및 tf-idf 점수 기반 top 200 문장 추출

Evaluation

평가지표: EM(Exact Match) / F1

	Quasar-T		SQuAD _{OPEN}		WikiMovies		CuratedTREC		WebQuestions	
	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM
GA (Dhingra et al. 2017)	26.4	26.4	-	-	-	-	-	-	-	-
BiDAF (Seo et al. 2017)	28.5	25.9	-	-	-	-	-	-	-	-
DrQA (Chen et al. 2017a)	-	-	-	28.4	-	34.3	-	25.7	-	19.5
Single Reader (SR)	38.5 ^{·2}	31.5 ^{·2}	35.4 ^{·2}	26.9 ^{·2}	38.8 ^{·1}	37.7 ^{·1}	33.6 ^{·6}	27.4 ^{·4}	22.0 ^{·2}	15.2 ^{·3}
Simple Ranker-Reader (SR ²)	38.8 ^{·2}	31.9 ^{·2}	35.8 ^{·2}	27.2 ^{·2}	39.3 ^{·1}	38.1 ^{·1}	33.4 ^{·6}	27.7 ^{·5}	22.5 ^{·3}	15.6 ^{·4}
Reinforced Ranker-Reader (R ³)	40.9 ^{·3}	34.2 ^{·3}	37.5 ^{·2}	29.1 ^{·2}	39.9 ^{·1}	38.8 ^{·1}	34.3 ^{·6}	28.4 ^{·6}	24.6 ^{·3}	17.1 ^{·3}
DrQA-MTL (Chen et al. 2017a)	-	-	-	29.8	-	36.5	-	25.4	-	20.7
YodaQA (Baudiš and Šedivý 2015)	-	-	-	-	-	-	-	31.3	-	39.8

● Baseline

- SR(Single Reader): passage 랜덤 선택
- SR²(Single Ranker-Reader): Ranker, Reader 각각 학습

● Result

- 모든 데이터셋 baseline에 비해 성능 향상
- WebQuestion 데이터셋 제외 DrQA 성능 증가

1. Reading Wikipedia to Answer Open-Domain Questions (Chen et al., ACL 2017, 1154회 인용)

- 규모가 큰 single resource(only Wikipedia)만을 가지고 complete ODQA 시스템 구축
- Retriever (질문과 관련 있는 문서 검색) + Reader (정답 추출)
- 발전가능성 1) sparse embedding retriever -> dense embedding retriever
- 발전가능성 2) single passage reader -> multi-passage reader

2. R³: Reinforced Ranker-Reader for Open-Domain Question Answering (Wang et al., AAAI 2018, 261회 인용)

- Retriever (질문과 관련 있는 문서 검색) + Ranker(문서 재평가 후 선택) + Reader (정답 추출)
- ranker와 reader jointly training (end-to-end)
- 발전가능성 1) QA에 다양한 강화학습 방법론 적용