REALM:Retrieval-Augmented Language Model Pre-Training

International Conference on Machine Learning (ICML), 2020
Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, Ming-Wei Chang

집현전 2기 5조 : 김대규, 모윤호, 원혜진

Contents

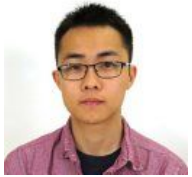
Open-QA Task

(DrQA) Reading Wikipedia to Answer Open-Domain Questions(2017, Danqi Chen)
- https://aclanthology.org/P17-1171/
- Retriever(문서 검색)과 Reader(기계 독해) task를 결합하여 위키피디아만 knowledge 자원으로 사용하여 오픈도메인 QA를 해결하는걸 제안
- Retriever로 TF-IDF IR system을 사용, 정해진 문서 셋에서(closed set) 올바른 답을 찾도록 Reader만 학습하는것에 focus

(ORQA) Latent Retrieval for Weakly Supervised Open Domain Question Answering(Kenton Lee, 2019-06)
- https://arxiv.org/abs/1906.00300
- question-answer 문자열 페어만 가지고 IR system 없이 retriever와 reader를 end-to-end로 jointly learn하는 것을 제안, Retriever와 Reader 모두 학습
- unsupervised 방식으로 Retriever가 적합한 문서를 검색하는 방법을 학습하는 ICT(Inverse Cloze Task) 사전 학습 방법을 도입

REALM: Retrieval-Augmented Language Model Pre-Training (2020. 02)
- https://arxiv.org/abs/2002.08909
- ORQA를 작성했던 Kenton Lee가 제 2 저자 (참고 BERT 3저자)
- ORQA에서 소개된 ICT를 base model로 하여 분석가능한 모델링 방법 제시


Kelvin Guu


Kenton Lee

Abstract

- LM pre-training은 많은 양의 world knowledge을 포착
  - knowledge는 network parameter에 암시적으로 저장
  - 더 큰 network가 더 많은 정보를 포함
- 모듈적 + 해석 가능한 방식
  - latent knowledge retriever 을 통해 문서 검색 + 관찰 가능
- Masked Language Model 사용
- pre-training unsupervised knowledge retriever 제안
- Open-QA task는 fine-tuning을 통해 REALM 성능 검증
  - SOTA와 비교했을 때, 4-16% 성능 차이

Introduction

- world knowledge 는 텍스트 코퍼스에서 얻어지게 됨
  - ex. BERT는 문장에서 마스킹된 단어를 정확하게 예측
  - The __ is the currency of the United Kingdom (answer : pound)
- world knowledge는 암시적으로 네트워크의 파라미터에 저장
- 모듈 + 해석가능 방식
  - textual knowledge retriever 을 통해 LM의 사전학습을 보강
  - REALM
- 어떤 knowledge를 검색할지 retriever model이 결정
- 검색된 문서를 기준으로 정답을 예측

Introduction

- unsupervised text를 사용하여 retriever를 학습시킴
  - LM의 PPL을 **개선**하는 경우 : 보상
  - LM의 PPL을 **악화**시키는 경우 : 패널티
- example
  - input x : “the ___ at the top of the pyramid”
  - retrieved document : “The pyramidion on top allows for less material higher up the pyramid.”
  - result : Correct (보상)
- retrieve-then-predict 모듈

Introduction

- pre-training step마다 retriever는 백 만개의 후보 문서를 고려 + 역전파
  - 단점 : 시간이 오래 걸림
  - 해결 방안 : MIPS(Maximum Inner Product Search)

Introduction

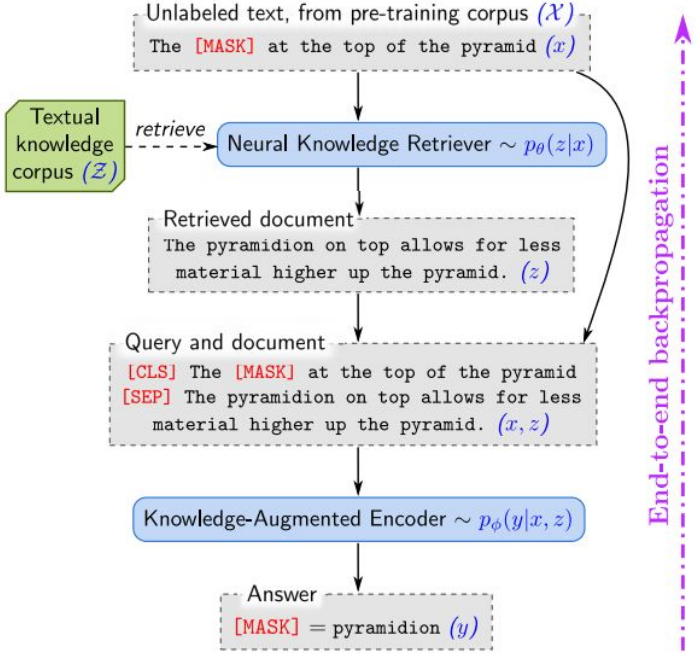


Figure 1. REALM augments language model pre-training with a **neural knowledge retriever** that retrieves knowledge from a **textual knowledge corpus**, $\mathcal{Z}$ (e.g., all of Wikipedia). Signal from the language modeling objective backpropagates all the way through the retriever, which must consider millions of documents in $\mathcal{Z}$—a significant computational challenge that we address.

Background

- Language Model pre-training
  - LM의 목표 : 언어의 표현을 unlabeled corpora를 통해 학습
  - pre-trained 모델을 통해서 task에 대한 fine-tuning도 가능
- Masked Language Model
  - 마스킹된 토큰을 예측하며 학습
  - ex. X : the [MASK] is the currency [MASK] the UK
  - y : (pound, of)

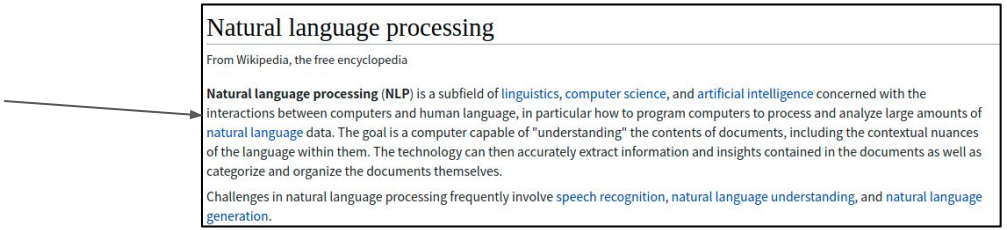
Background

- Open-domain Question Answering
  - world knowledge 모델 성능 평가 task
  - 질문 X에 대한 답변 y를 예측
  - 질문에 대한 특정한 문서가 정해지지 않음 -> (open domain)
  - SQuAD와 같은 task는 특정한 문서가 정해짐
  - 따라서, 더 어려운 task

Inverse Cloze Task

- Retriever가 sentence(question)에서 관련 높은 context(document)를 unsupervised 방식으로 학습하는데 focus
- 기존 IR의 경우 고정된 closed set에서 전통적인 검색 방법(ex. TF-IDF, BM25) document를 찾아 학습하는데 사용하였다
- mask된 문장에서 적절한 query를 찾는 Cloze Task를 활용하여 Retriever를 학습하는 방법을 제안

Which algorithm is used in chatbot?

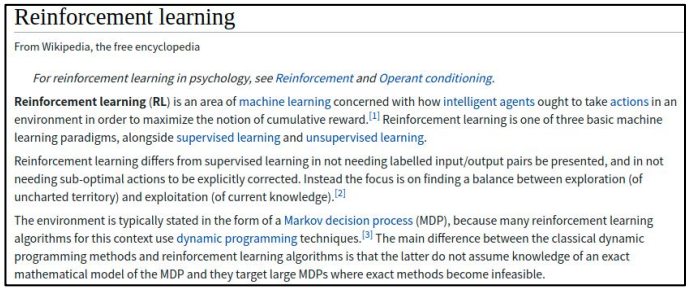


Natural language processing

From Wikipedia, the free encyclopedia

Natural language processing (**NLP**) is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, in particular how to program computers to process and analyze large amounts of natural language data. The goal is a computer capable of "understanding" the contents of documents, including the contextual nuances of the language within them. The technology can then accurately extract information and insights contained in the documents as well as categorize and organize the documents themselves.

Challenges in natural language processing frequently involve speech recognition, natural language understanding, and natural language generation.



Reinforcement learning

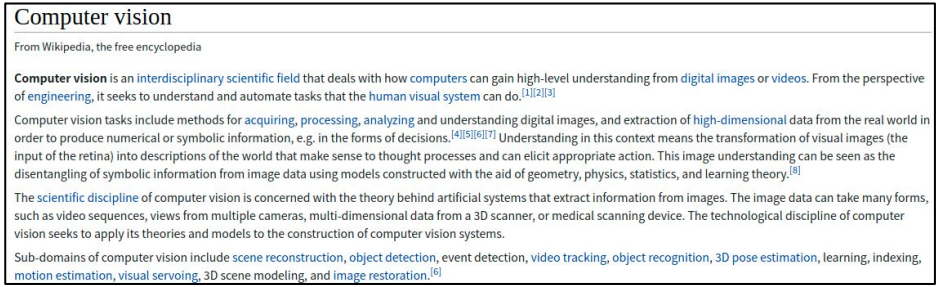
From Wikipedia, the free encyclopedia

For reinforcement learning in psychology, see Reinforcement and Operant conditioning.

Reinforcement learning (**RL**) is an area of machine learning concerned with how intelligent agents ought to take actions in an environment in order to maximize the notion of cumulative reward.[1] Reinforcement learning is one of three basic machine learning paradigms, alongside supervised learning and unsupervised learning.

Reinforcement learning differs from supervised learning in not needing labelled input/output pairs be presented, and in not needing sub-optimal actions to be explicitly corrected. Instead the focus is on finding a balance between exploration (of uncharted territory) and exploitation (of current knowledge).[2]

The environment is typically stated in the form of a Markov decision process (MDP), because many reinforcement learning algorithms for this context use dynamic programming techniques.[3] The main difference between the classical dynamic programming methods and reinforcement learning algorithms is that the latter do not assume knowledge of an exact mathematical model of the MDP and they target large MDPs where exact methods become infeasible.



Computer vision

From Wikipedia, the free encyclopedia

Computer vision is an interdisciplinary scientific field that deals with how computers can gain high-level understanding from digital images or videos. From the perspective of engineering, it seeks to understand and automate tasks that the human visual system can do.[1][2][3]

Computer vision tasks include methods for acquiring, processing, analyzing and understanding digital images, and extraction of high-dimensional data from the real world in order to produce numerical or symbolic information, e.g. in the forms of decisions.[4][5][6][7] Understanding in this context means the transformation of visual images (the input of the retina) into descriptions of the world that make sense to thought processes and can elicit appropriate action. This image understanding can be seen as the disentangling of symbolic information from image data using models constructed with the aid of geometry, physics, statistics, and learning theory.[8]

The scientific discipline of computer vision is concerned with the theory behind artificial systems that extract information from images. The image data can take many forms, such as video sequences, views from multiple cameras, multi-dimensional data from a 3D scanner, or medical scanning device. The technological discipline of computer vision seeks to apply its theories and models to the construction of computer vision systems.

Sub-domains of computer vision include scene reconstruction, object detection, event detection, video tracking, object recognition, 3D pose estimation, learning, indexing, motion estimation, visual servoing, 3D scene modeling, and image restoration.[6]

https://arxiv.org/pdf/1906.00300.pdf

Cloze Task

"Cloze Procedure": A New Tool For Measuring Readability (1953, Wilson L Talyor)
- context를 기반으로 mask된 text를 예측하는 task
- Masked Language Model

ex)

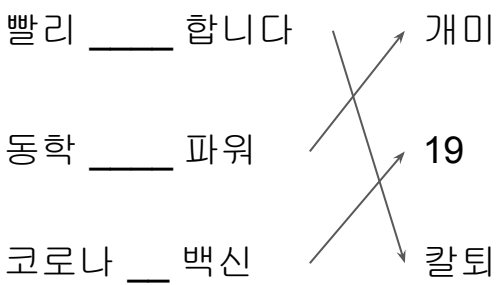
CLOZE TEST

For Lawrence United Washington cut father he his of really slaves very

George Washington was the first President of the United States. He was also the commander in chief of all American forces during the American Revolutionary War. For his central role in the beginning of the United States, he is often called the father of country. His mother was Mary Ball and his was Augustine Washington. They owned a plantation with in Virginia. George studied at local schools. George's died when he was eleven. Then his brother helped train him. There is a story that cut down his father's cherry tree. When asked, did not lie and said that he did down the tree. The story means he was honest. We do not know if the story happened.

SCORE:
3/14

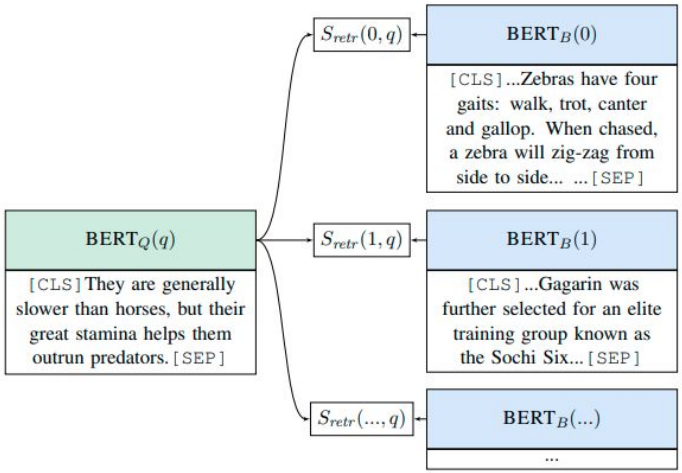
?

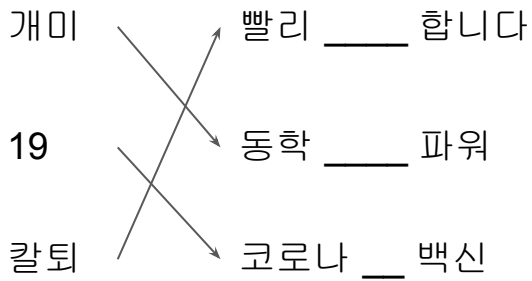


https://www.controlaltachieve.com/2017/04/create-cloze-tests.html

Inverse Cloze Task

(ORQA) Latent Retrieval for Weakly Supervised Open Domain Question Answering(Kenton Lee, 2019-06)
- sentence(question)이 주어질 때 context를 예측
- batch의 candidate 중에 true context를 선택하는 것





https://arxiv.org/pdf/1906.00300.pdf

Inverse Cloze Task - 예시

비트코인

위키백과, 우리 모두의 백과사전.

비트코인(영어: Bitcoin)은 블록체인 기술을 기반으로 만들어진 온라인 암호화폐이다. 비트코인의 화폐 단위는 BTC로 표시한다. 2008년 10월 앤드류 방이라는 가명을 쓰는 프로그래머가 개발하여, 2009년 1월 프로그램 소스를 배포했다. 중앙은행이 없이 전 세계적 범위에서 P2P 방식으로 개인들 간에 자유롭게 송금 등의 금융거래를 할 수 있게 설계되어 있다. 또 중앙은행을 거치지 않아 수수료 부담이 적다. 거래장부는 블록체인 기술을 바탕으로 전 세계적인 범위에서 여러 사용자들의 서버에 분산하여 저장하기 때문에 해킹이 불가능하다. SHA-256 기반의 암호 해시 함수를 사용한다.

2009년 비트코인의 소스 코드가 공개되었고, 이더리움, 이더리움 클래식, 리플, 라이트코인, 에이코인, 대시, 모네로, 제트캐시, 퀀텀 등 다양한 알트코인들이 생겨났다. 알트코인은 비트코인 이후에 등장한 암호화폐를 의미하며[3][4], 비트코인은 여러 알트코인들 사이에서 일종의 기축통화 역할을 하고 있다.

비트코인(영어: Bitcoin)은 블록체인 기술을 기반으로 만들어진 온라인 암호화폐이다.

비트코인의 화폐 단위는 BTC로 표시한다. 2008년 10월 앤드류 방이라는 가명을 쓰는 프로그래머가 개발하여, 2009년 1월 프로그램 소스를 배포했다.

중앙은행이 없이 전 세계적 범위에서 P2P 방식으로 개인들 간에 자유롭게 송금 등의 금융거래를 할 수 있게 설계되어 있다.

또 중앙은행을 거치지 않아 수수료 부담이 적다. 거래장부는 블록체인 기술을 바탕으로 전 세계적인 범위에서 여러 사용자들의 서버에 분산하여 저장하기 때문에 해킹이 불가능하다.

SHA-256 기반의 암호 해시 함수를 사용한다.

Inverse Cloze Task - 예시

스파게티 알리오 올리오

위키백과, 우리 모두의 백과사전.

스파게티 알리오 올리오(이탈리아어: Spaghetti aglio e olio)는 이탈리아 요리의 파스타 요리이다. 아브루초 주의 전통 요리로 이탈리아 전역에서 널리 먹는다.

방법 [편집]

마늘을 올리브유에 빻아서 으깬 다음 뿌려서 만들며, 홍고춧가루를 흩뿌려서 먹기도 한다. 잘게 썬 파슬리를 위에 장식으로 뿌리면서 파르미자노 레자노 치즈를 같이 먹는 것은 매우 흔하나, 전통 조리법에 따르면 치즈가 들어간 것은 아니다. 주로 해산물이 들어간다.

스파게티 알리오 올리오(이탈리아어: Spaghetti aglio e olio)는 이탈리아 요리의 파스타 요리이다. 아브루초 주의 전통 요리로 이탈리아 전역에서 널리 먹는다.

마늘을 올리브유에 빻아서 으깬 다음 뿌려서 만들며, 홍고춧가루를 흩뿌려서 먹기도 한다.

잘게 썬 파슬리를 위에 장식으로 뿌리면서 파르미자노 레자노 치즈를 같이 먹는 것은 매우 흔하나, 전통 조리법에 따르면 치즈가 들어간 것은 아니다. 주로 해산물이 들어간다.

Inverse Cloze Task - 예시

잉글랜드

위키백과, 우리 모두의 백과사전.

다른 뜻에 대해서는 잉글랜드 (동음이의) 문서를 참고하십시오.

잉글랜드(England, 고대 영어: Englaland)는 영국의 구성국 중 하나이다. 잉글랜드는 그레이트브리튼섬 남부의 대부분을 차지하며, 북쪽으로 스코틀랜드, 서쪽으로 웨일스와 아일랜드해, 남쪽으로 영국 해협, 동쪽으로 북해와 접한다. 지형은 대체로 평탄하며 면적은 130,395 km^2이다.

잉글랜드는 영국 전체 인구의 4/5 이상인 4,913만 8,831명(2001년 기준)을 보유하고 있으며, 잉글랜드에 위치한 런던은 영국의 수도이기도 하다. 잉글랜드는 영국 역사에서 중심적인 역할을 했으며, **영국**이라는 명칭 또한 잉글랜드에서 유래했다. 잉글랜드 왕국의 독자적인 역사는 1707년 연합법으로 잉글랜드, 스코틀랜드, 웨일스가 그레이트브리튼 왕국으로 통합하면서 구성국이 되었다.

잉글랜드(England, 고대 영어: Englaland)는 영국의 구성국 중 하나이다. 잉글랜드는 그레이트브리튼섬 남부의 대부분을 차지하며, 북쪽으로 스코틀랜드, 서쪽으로 웨일스와 아일랜드해, 남쪽으로 영국 해협, 동쪽으로 북해와 접한다. 지형은 대체로 평탄하며 면적은 130,395 km2이다.

잉글랜드는 영국 전체 인구의 4/5 이상인 4,913만 8,831명(2001년 기준)을 보유하고 있으며, 잉글랜드에 위치한 런던은 영국의 수도이기도 하다.

잉글랜드는 영국 역사에서 중심적인 역할을 했으며, **영국**이라는 명칭 또한 잉글랜드에서 유래했다. 잉글랜드 왕국의 독자적인 역사는 1707년 연합법으로 잉글랜드, 스코틀랜드, 웨일스가 그레이트브리튼 왕국으로 통합하면서 구성국이 되었다.

Inverse Cloze Task - 예시

비트코인

비트코인(영어: Bitcoin)은 블록체인 기술을 기반으로 만들어진 온라인 암호화폐이다.

비트코인의 화폐 단위는 BTC로 표시한다. 2008년 10월 앤드류 방이라는 가명을 쓰는 프로그래머가 개발하여, 2009년 1월 프로그램 소스를 배포했다.

중앙은행이 없이 전 세계적 범위에서 P2P 방식으로 개인들 간에 자유롭게 송금 등의 금융거래를 할 수 있게 설계되어 있다.

또 중앙은행을 거치지 않아 수수료 부담이 적다. 거래장부는 블록체인 기술을 바탕으로 전 세계적인 범위에서 여러 사용자들의 서버에 분산하여 저장하기 때문에 해킹이 불가능하다. SHA-256 기반의 암호 해시 함수를 사용한다.

알리오 올리오

스파게티 알리오 올리오(이탈리아어: Spaghetti aglio e olio)는 이탈리아 요리의 파스타 요리이다. 아브루초 주의 전통 요리로 이탈리아 전역에서 널리 먹는다.

마늘을 올리브유에 빻아서 으깬 다음 뿌려서 만들며, 홍고춧가루를 흩뿌려서 먹기도 한다.

잘게 썬 파슬리를 위에 장식으로 뿌리면서 파르미자노 레자노 치즈를 같이 먹는 것은 매우 흔하나, 전통 조리법에 따르면 치즈가 들어간 것은 아니다. 주로 해산물이 들어간다.

잉글랜드

잉글랜드(England, 고대 영어: Englaland)는 영국의 구성국 중 하나이다. 잉글랜드는 그레이트브리튼섬 남부의 대부분을 차지하며, 북쪽으로 스코틀랜드, 서쪽으로 웨일스와 아일랜드해, 남쪽으로 영국 해협, 동쪽으로 북해와 접한다. 지형은 대체로 평탄하며 면적은 130,395 km2이다.

잉글랜드는 영국 전체 인구의 4/5 이상인 4,913만 8,831명(2001년 기준)을 보유하고 있으며, 잉글랜드에 위치한 런던은 영국의 수도이기도 하다.

잉글랜드는 영국 역사에서 중심적인 역할을 했으며, **영국**이라는 명칭 또한 잉글랜드에서 유래했다. 잉글랜드 왕국의 독자적인 역사는 1707년 연합법으로 잉글랜드, 스코틀랜드, 웨일스가 그레이트브리튼 왕국으로 통합하면서 구성국이 되었다.

Inverse Cloze Task

비트코인(영어: Bitcoin)은 블록체인 기술을 기반으로 만들어진 온라인 암호화폐이다.

비트코인

비트코인의 화폐 단위는 BTC로 표시한다. 2008년 10월 앤드류 방이라는 가명을 쓰는 프로그래머가 개발하여, 2009년 1월 프로그램 소스를 배포했다.

중앙은행이 없이 전 세계적 범위에서 P2P 방식으로 개인들 간에 자유롭게 송금 등의 금융거래를 할 수 있게 설계되어 있다.

또 중앙은행을 거치지 않아 수수료 부담이 적다. 거래장부는 블록체인 기술을 바탕으로 전 세계적인 범위에서 여러 사용자들의 서버에 분산하여 저장하기 때문에 해킹이 불가능하다.

SHA-256 기반의 암호 해시 함수를 사용한다.

스파게티 알리오 올리오(이탈리아어: Spaghetti aglio e olio)는 이탈리아 요리의 파스타 요리이다. 아브루초 주의 전통 요리로 이탈리아 전역에서 널리 먹는다.

알리오 올리오

마늘을 올리브유에 빻아서 으깬 다음 뿌려서 만들며, 홍고춧가루를 흩뿌려서 먹기도 한다.

잘게 썬 파슬리를 위에 장식으로 뿌리면서 파르미자노 레자노 치즈를 같이 먹는 것은 매우 흔하나, 전통 조리법에 따르면 치즈가 들어간 것은 아니다. 주로 해산물이 들어간다.

잉글랜드는 영국 전체 인구의 4/5 이상인 4,913만 8,831명(2001년 기준)을 보유하고 있으며, 잉글랜드에 위치한 런던은 영국의 수도이기도 하다.

잉글랜드

잉글랜드(England, 고대 영어: Englaland)는 영국의 구성국 중 하나이다. 잉글랜드는 그레이트브리튼섬 남부의 대부분을 차지하며, 북쪽으로 스코틀랜드, 서쪽으로 웨일스와 아일랜드해, 남쪽으로 영국 해협, 동쪽으로 북해와 접한다. 지형은 대체로 평탄하며 면적은 130,395 km2이다.

잉글랜드는 영국 역사에서 중심적인 역할을 했으며, **영국**이라는 명칭 또한 잉글랜드에서 유래했다. 잉글랜드 왕국의 독자적인 역사는 1707년 연합법으로 잉글랜드, 스코틀랜드, 웨일스가 그레이트브리튼 왕국으로 통합하면서 구성국이 되었다.

Inverse Cloze Task

비트코인(영어: Bitcoin)은 블록체인 기술을 기반으로 만들어진 온라인 암호화폐이다.

스파게티 알리오 올리오(이탈리아어: Spaghetti aglio e olio)는 이탈리아 요리의 파스타 요리이다. 아브루초 주의 전통 요리로 이탈리아 전역에서 널리 먹는다.

question

잉글랜드는 영국 전체 인구의 4/5 이상인 4,913만 8,831명(2001년 기준)을 보유하고 있으며, 잉글랜드에 위치한 런던은 영국의 수도이기도 하다.

비트코인

비트코인의 화폐 단위는 BTC로 표시한다. 2008년 10월 앤드류 방이라는 가명을 쓰는 프로그래머가 개발하여, 2009년 1월 프로그램 소스를 배포했다.

중앙은행이 없이 전 세계적 범위에서 P2P 방식으로 개인들 간에 자유롭게 송금 등의 금융거래를 할 수 있게 설계되어 있다.

또 중앙은행을 거치지 않아 수수료 부담이 적다. 거래장부는 블록체인 기술을 바탕으로 전 세계적인 범위에서 여러 사용자들의 서버에 분산하여 저장하기 때문에 해킹이 불가능하다.

SHA-256 기반의 암호 해시 함수를 사용한다.

알리오 올리오

마늘을 올리브유에 빻아서 으깬 다음 뿌려서 만들며, 홍고춧가루를 흩뿌려서 먹기도 한다.

잘게 썬 파슬리를 위에 장식으로 뿌리면서 파르미자노 레자노 치즈를 같이 먹는 것은 매우 흔하나, 전통 조리법에 따르면 치즈가 들어간 것은 아니다. 주로 해산물이 들어간다.

document

잉글랜드

잉글랜드(England, 고대 영어: Englaland)는 영국의 구성국 중 하나이다. 잉글랜드는 그레이트브리튼섬 남부의 대부분을 차지하며, 북쪽으로 스코틀랜드, 서쪽으로 웨일스와 아일랜드해, 남쪽으로 영국 해협, 동쪽으로 북해와 접한다. 지형은 대체로 평탄하며 면적은 130,395 km2이다.

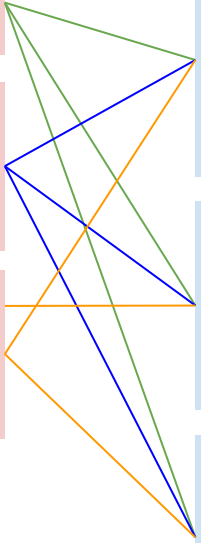
잉글랜드는 영국 역사에서 중심적인 역할을 했으며, 영국이라는 명칭 또한 잉글랜드에서 유래했다. 잉글랜드 왕국의 독자적인 역사는 1707년 연합법으로 잉글랜드, 스코틀랜드, 웨일스가 그레이트브리튼 왕국으로 통합하면서 구성국이 되었다.

Inverse Cloze Task

[CLS] 비트코인(영어: Bitcoin)은 블록체인 기술을 기반으로 만들어진 온라인 암호화폐이다. [SEP]

[CLS] 스파게티 알리오 올리오(이탈리아어: Spaghetti aglio e olio)는 이탈리아 요리의 파스타 요리이다. 아브루초 주의 전통 요리로 이탈리아 전역에서 널리 먹는다. [SEP]

[CLS] 잉글랜드는 영국 전체 인구의 4/5 이상인 4,913만 8,831명(2001년 기준)을 보유하고 있으며, 잉글랜드에 위치한 런던은 영국의 수도이기도 하다. [SEP]



[CLS] 비트코인의 화폐 단위는 BTC로 표시한다. 2008년 10월 앤드류 방이라는 가명을 쓰는 프로그래머가 개발하여, 2009년 1월 프로그램 소스를 배포했다. 중앙은행이 없이 전 세계적 범위에서 P2P 방식으로 …
SHA-256 기반의 암호 해시 함수를 사용한다. [SEP]

[CLS] 마늘을 올리브유에 빻아서 으깬 다음 뿌려서 만들며, 홍고춧가루를 흩뿌려서 먹기도 한다. …
파르미자노 레자노 치즈를 같이 먹는 것은 매우 흔하나, 전통 조리법에 따르면 치즈가 들어간 것은 아니다. 주로 해산물이 들어간다. [SEP]

[CLS] 잉글랜드(England, 고대 영어: Englaland)는 영국의 구성국 중 하나이다. 잉글랜드는 그레이트브리튼섬 남부의 대부분을 차지하며, … 스코틀랜드, 웨일스가 그레이트브리튼 왕국으로 통합하면서 구성국이 되었다. [SEP]

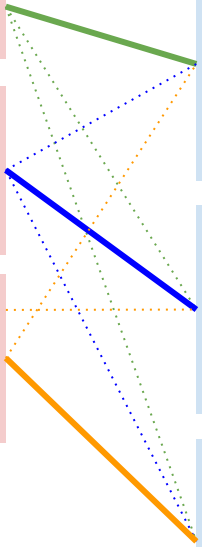
[batch_size, embedding_dim] x **[batch_size, embedding_dim]** ^{T} = **[batch_size, batch_size]**

Inverse Cloze Task

[CLS] 비트코인(영어: Bitcoin)은 블록체인 기술을 기반으로 만들어진 온라인 암호화폐이다. [SEP]

[CLS] 스파게티 알리오 올리오(이탈리아어: Spaghetti aglio e olio)는 이탈리아 요리의 파스타 요리이다. 아브루초 주의 전통 요리로 이탈리아 전역에서 널리 먹는다. [SEP]

[CLS] 잉글랜드는 영국 전체 인구의 4/5 이상인 4,913만 8,831명(2001년 기준)을 보유하고 있으며, 잉글랜드에 위치한 런던은 영국의 수도이기도 하다. [SEP]



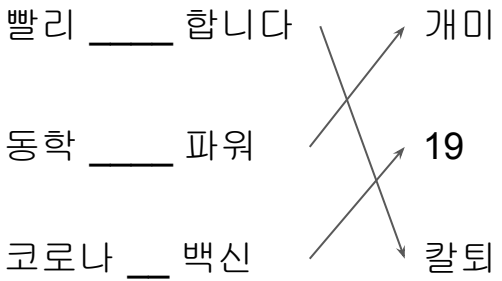
[CLS] 비트코인의 화폐 단위는 BTC로 표시한다. 2008년 10월 앤드류 방이라는 가명을 쓰는 프로그래머가 개발하여, 2009년 1월 프로그램 소스를 배포했다. 중앙은행이 없이 전 세계적 범위에서 P2P 방식으로 …
SHA-256 기반의 암호 해시 함수를 사용한다. [SEP]

[CLS] 마늘을 올리브유에 빻아서 으깬 다음 뿌려서 만들며, 홍고춧가루를 흩뿌려서 먹기도 한다. …
파르미자노 레자노 치즈를 같이 먹는 것은 매우 흔하나, 전통 조리법에 따르면 치즈가 들어간 것은 아니다. 주로 해산물이 들어간다. [SEP]

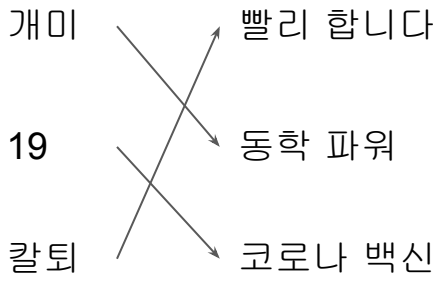
[CLS] 잉글랜드(England, 고대 영어: Englaland)는 영국의 구성국 중 하나이다. 잉글랜드는 그레이트브리튼섬 남부의 대부분을 차지하며, … 스코틀랜드, 웨일스가 그레이트브리튼 왕국으로 통합하면서 구성국이 되었다. [SEP]

[batch_size, embedding_dim] x **[batch_size, embedding_dim]** ^{T} = **[batch_size, batch_size]**

Inverse Cloze Task



Cloze Task



Inverse Cloze Task

"개미"가 "빨리 합니다", "동학 파워" 코로나 백신" 중에 어느 문장과 관련이 높은지를 스스로 학습하도록 모델링

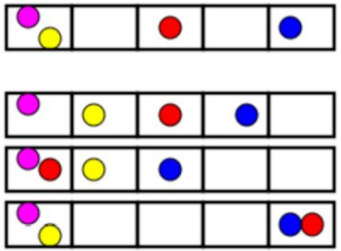
MIPS(Maximum Inner Product Search) 알고리즘

- 결론적으로 **Approximate Nearest Neighbor** Search의 방법론 중 하나

- 목표는 매우 빠르게 쿼리의 **어느정도** 근접한 이웃(nearest neighbors)을 찾는 것

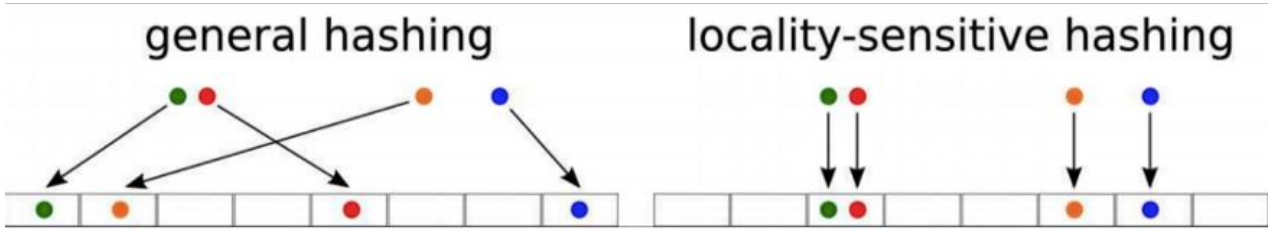
- 가장 근접한 이웃을 찾는 것 대신 **대략적으로 근접한 이웃(**Approximate Nearest Neighbor**)**을 찾는 것으로 계산적인 이득을 가져온다.

- 대략적인 근접 이웃을 보는 방법 → **Locality Sensitive Hashing (LSH)**



- 큰 차수를 지닌 벡터 값을 Hashing을 하여 작은 벡터 값으로 치환한다. 이 때, 동일한 표현을 지닌 벡터끼리는 근접한 bucket에 들어가도록 Hashing하는 것이 핵심 point

- 따라서 query와 근접한 Nearest Neighbor를 뽑을 때 전체를 보는 대신 bucket만 보면 되기 때문에 계산량이 줄어든다



https://whytin.github.io/cs584_s16/slides/lsh-11.pdf

MIPS - Locality Sensitive Hashing (LSH)

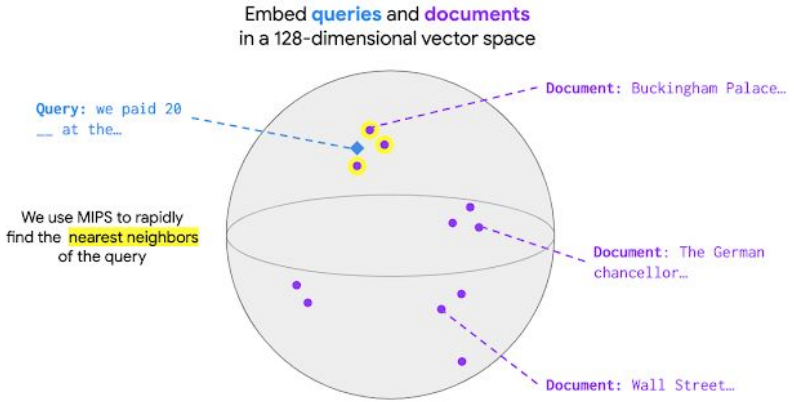
근접한 이웃을 구하는 방법은 ?

- Euclidean distance
- Cosine similarity
- Inner product

Asymmetric LSH (ALSH) for Sublinear Time Maximum Inner Product Search (MIPS)(2014, Anshumali Shrivastava, Ping Li) https://arxiv.org/abs/1405.5869

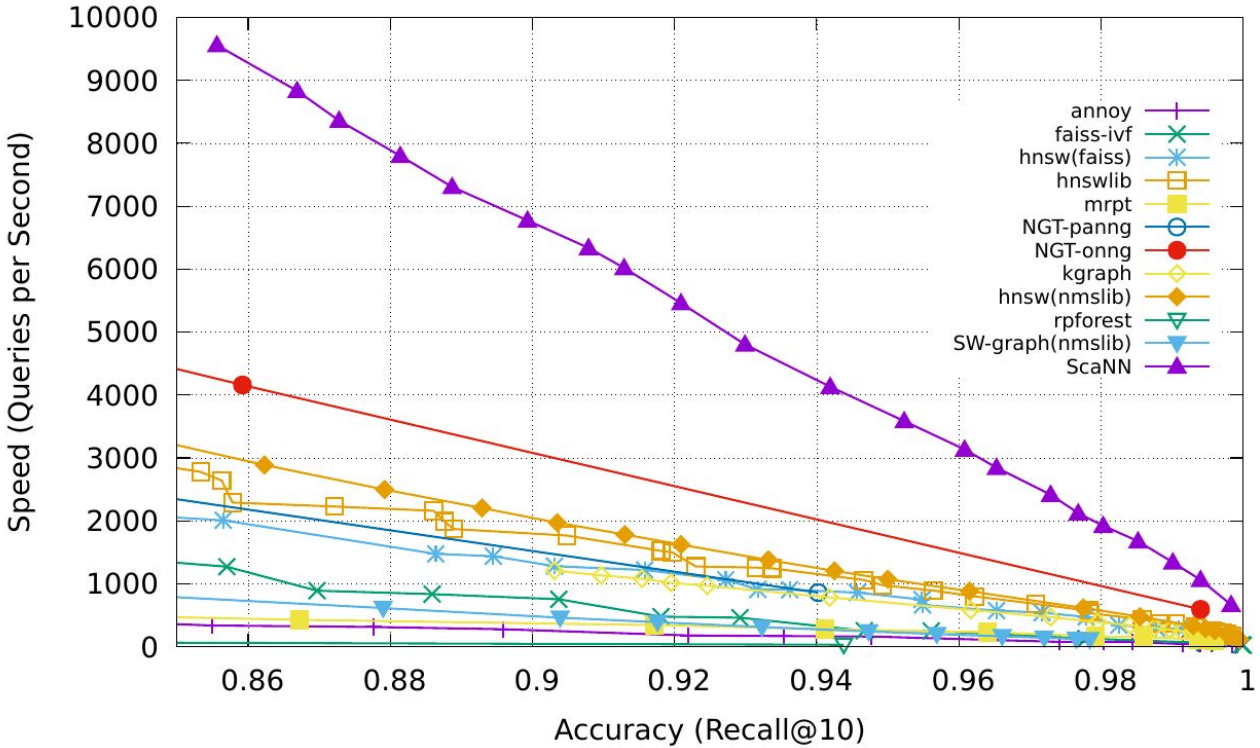
- 기존에는 LSH가 Euclidean distance와 Cosine similarity와 같은 similarity score만 적용 가능
- 이 논문에서 최초로 Inner product에도 적용하는데 성공

- 주로 Faiss, Annoy(tmi. 회사에서 사용 중) 등이 있고 논문에서는 ScaNN을 사용하였다



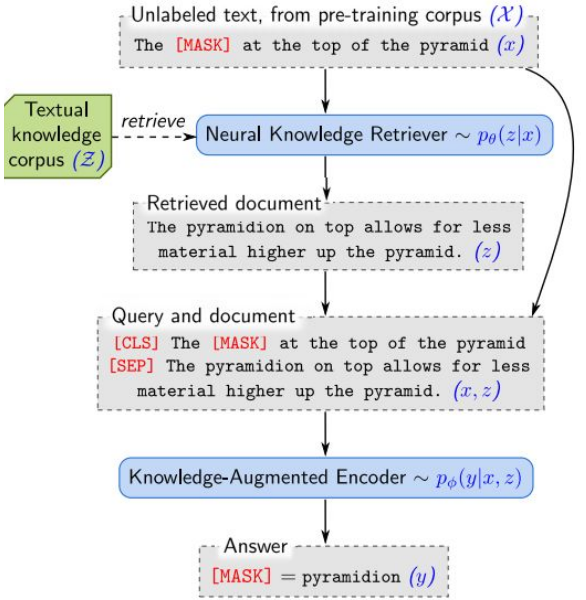
https://whytin.github.io/cs584_s16/slides/lsh-11.pdf

ScaNN



https://github.com/google-research/google-research/tree/master/scann

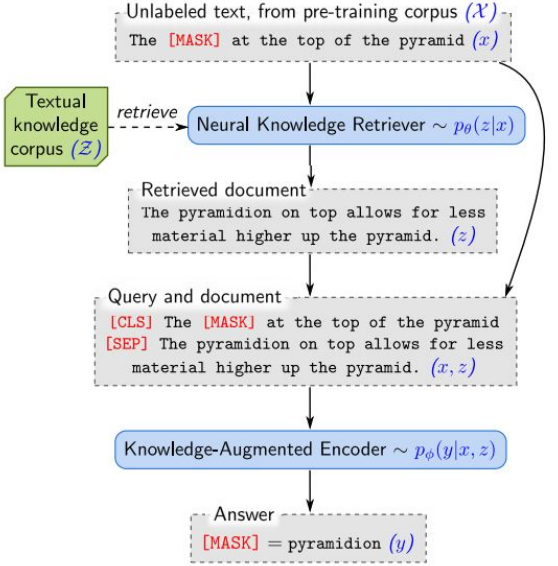
REALM's generative process



Open QA Model = Retriever + Reader
- Retriever: question과 적합한 문서셋을 검색
- Reader(=Ranker): question과 Retriever에서 검색된 문서에서 적합한 답변을 search(start index, end index)

REALM's generative process

- input x로 부터 가능한 output y의 분포 $p(y|x)$ 학습
  - pre-training에서 x는 마스킹된 문장 y는 마스킹 된 토큰 예측값
  - fine-tuning에서는 Open-QA task의 질문 - 답변 쌍
- retrieve
  - knowledge corpora Z 로 부터 input x 에 연관된 문서 z를 검색
  - $p(z|x)$
- predict
  - retrieved z와 input x에 대해 모두 y를 예측하는데 사용
  - $p(y|z,x)$
- $p(y|x) = \sum_{z \in Z} p(y|z,x)p(z|x)$



Model architecture

- Knowledge Retriever
  - 문서 z와 입력 x 의 관련성 측정
  - **f(x,z) - relevance score**
  - Embed_input, _doc 는 d 차원 벡터
  - 이 둘을 내적(inner product)
  - x와 관련된 z를 선별 가능
  - **p(z|x) - retrieval distribution**
  - softmax

$$p(z \mid x) = \frac{\exp f(x, z)}{\sum_{z'} \exp f(x, z')},$$

$$f(x, z) = \mathbf{Embed}_{\mathbf{input}}(x)^\top \mathbf{Embed}_{\mathbf{doc}}(z),$$

Model architecture

- Knowledge Retriever
  - BERT -style
  - wordPiece tokenizer

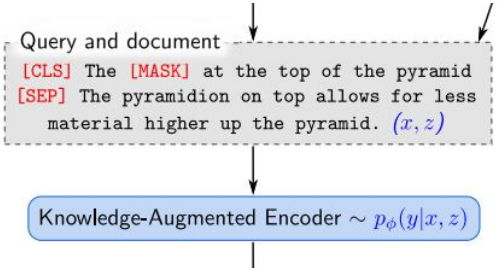
$$\texttt{join}_{\text{BERT}}(x) = \texttt{[CLS]}\, x\, \texttt{[SEP]}$$
$$\texttt{join}_{\text{BERT}}(x_1, x_2) = \texttt{[CLS]}\, x_1\, \texttt{[SEP]}\, x_2\, \texttt{[SEP]}$$

  - Transformer Embedding + [CLS] 토큰 포함 → BERT_cls
  - 벡터 차원 축소를 위해 행렬 W와 linear projection

$$\text{Embed}_{\text{input}}(x) = \mathbf{W}_{\text{input}}\text{BERT}_{\text{CLS}}(\texttt{join}_{\text{BERT}}(x))$$
$$\text{Embed}_{\text{doc}}(z) = \mathbf{W}_{\text{doc}}\text{BERT}_{\text{CLS}}(\texttt{join}_{\text{BERT}}(z_{\text{title}}, z_{\text{body}}))$$

Model architecture

- Knowledge-Augmented Encoder
  - x와 z를 하나의 sequence로 결합하여 사용
    - x와 z 사이에서 cross-attention을 수행
  - pre-training
    - BERT의 MLM loss 사용(cross entropy loss)
    - 마스킹된 원래 토큰을 예측
  - fine-tuning
    - 일반적인 MRC에서 reader 형식
    - output y를 생성하길 원함
    - span의 시작과 끝을 연결해서 MLP 통과



$$p(y \mid z, x) = \prod_{j=1}^{J_x} p(y_j \mid z, x)$$

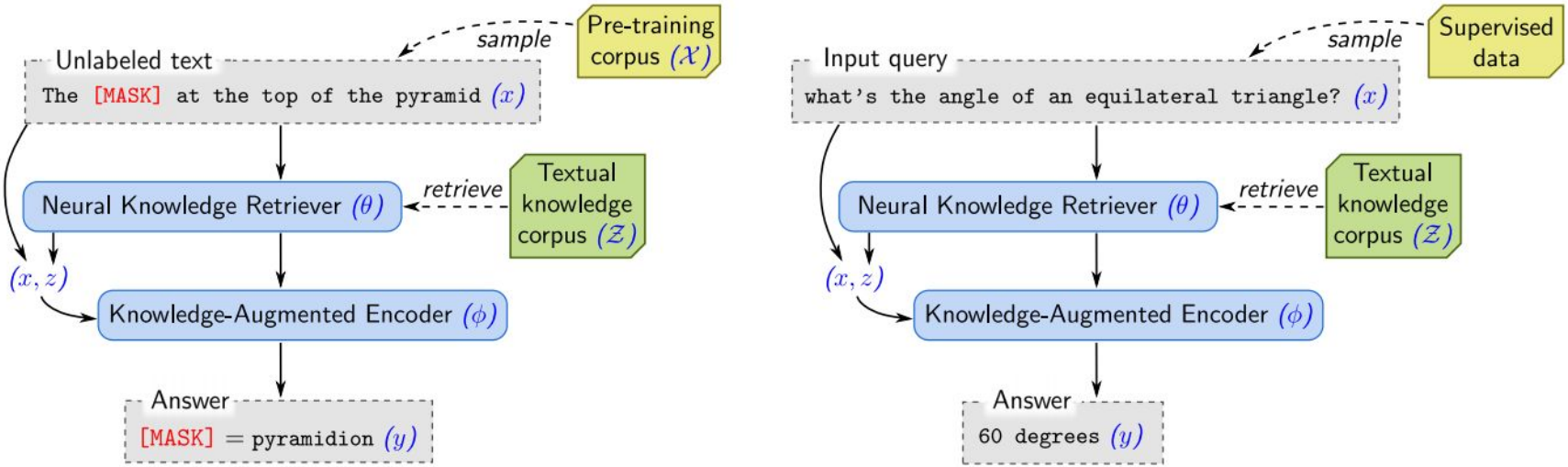
$$p(y_j \mid z, x) \propto \exp\left(w_j^\top \text{BERT}_{\text{MASK}(j)}(\text{join}_{\text{BERT}}(x, z_{\text{body}}))\right)$$

$$p(y \mid z, x) \propto \sum_{s \in S(z,y)} \exp\left(\text{MLP}\left(\left[h_{\text{START}(s)}; h_{\text{END}(s)}\right]\right)\right)$$

$$h_{\text{START}(s)} = \text{BERT}_{\text{START}(s)}(\text{join}_{\text{BERT}}(x, z_{\text{body}})),$$

$$h_{\text{END}(s)} = \text{BERT}_{\text{END}(s)}(\text{join}_{\text{BERT}}(x, z_{\text{body}})),$$

Model architecture



Training

- $p(y|x) = \sum_{z \in Z} p(y|z,x)p(z|x)$
  - 모든 Z에 대해서 확률을 구하기가 어려움
  - Maximum Inner Product Search (MIPS)알고리즘을 통해 top k 문서 찾아냄
- $f(x,z) = \mathbf{Embed}_{\mathbf{input}}(x)^\top \mathbf{Embed}_{\mathbf{doc}}(z),$
  - MIPS 사용하려면 Embed_doc(z)를 미리 계산해야함
  - 즉, 모든 문서에 대해서 미리 계산
  - 그러나, 새로운 문서가 추가되면 $p(z|x)$가 더이상 맞지 않음
- 해결방안
  - 비 동기적으로 index를 새로고침하는 방법 제안

Training

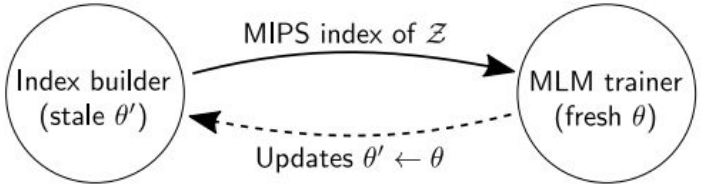
- 

Figure 3. REALM pre-training with asynchronous MIPS refreshes.

- 연구에서는 pre-training 과정에서만 사용
- fine-tuning에서는 pre-trained된 파라미터를 통해 MIPS index **한번만** 만듦

What does the retriever learn?

- $$\nabla \log p(y \mid x) = \sum_{z \in \mathcal{Z}} r(z) \nabla f(x, z)$$

$$r(z) = \left[\frac{p(y \mid z, x)}{p(y \mid x)} - 1 \right] p(z \mid x).$$

- retrieval signal을 주는 방법 제안
- r(z)에 따라서 retriever gradient score f(x,z)의 값이 변경
- p(y|z,x) > p(y|x) : 양수 -> 보상
- p(y|z,x) < p(y|x) : 음수 -> 패널티

y when using document z. The term $p(y \mid x)$ is the expected value of $p(y \mid x, z)$ when randomly sampling a document from $p(z \mid x)$. Hence, document z receives a positive update whenever it performs better than expected.

Injecting inductive biases into pre-training

- Salient span masking
  - Named Entity & 날짜 마스킹
  - ex. "United Kingdom", "July 1969"
- Null document
  - top k 문서에 공문서(empty document) 후보 추가
- Prohibiting trivial retrievals
  - pre-training 말뭉치와 knowledge 말뭉치를 겹치지 않게 조정
- Initialization
  - Embed_input(x)와 Embed_doc(z)를 warm-start하는 Inverse Cloze Task(ICT) 도입
  - Lee et al ,. (2019)

Experiments

- Open-QA Benchmarks
  - 연구에서는 Answer 작성자가 정답을 알지 못하는 상태로 만들어진 dataset을 원했음
    - 보다 현실적인 information-seeking needs를 반영할 수 있기 때문에
- 실험에 사용된 Open-QA 셋
  - NaturalQuestion-Open
    - Google search engine에서 생성된 질문과 답변
  - WebQuestions
    - Google Suggest API로 부터 수집한 쌍
  - CuratedTrec
    - 실제 사이트(MSNSearch, AskJeeves)에서 사용자의 질문과 답변

Approaches compared

- Retrieval-based Open-QA
  - non-learned heuristic retrieval approaches
  - ex. DrQA, HardEM, GraphRetriever, PathRetriever
- Generation-based Open-QA
  - 주어진 context와 명시적인 추출없이 직접적으로 답변 생성
  - ex. GPT-2, **T5**

Implementation Details

- fine-tuning
  - hyperparameters : All of them are same with ORQA (Lee et al ,. 2019)
  - knowledge corpus : 2018.12.20 영어 위키피디아
    - 문서는 BERT WordPiece 288 로 tokenizing
    - 그 결과 13 million retrieval 후보가 생성됨
  - inference : top 5 문서 후보 설정
  - 전체 모델은 single GPU 12GB 머신에서 모델 실행 가능

Implementation Details

- pre-training
  - Optimizer : BERT default
  - batch size : 512
  - learning rate : 3e-5
  - 200k step은 Google Cloud TPU 64개
  - retrieve와 marginalize는 공문서 포함해서 8개의 문서 후보 설정(top 8)
- pre-training corpus 선택
  1. pre-training 말뭉치 == knowledge 말뭉치
     a. 영어 wikipedia
  2. pre-training 말뭉치 != knowledge 말뭉치
     a. CC-News

Main results

Name	Architectures	Pre-training	NQ (79k/4k)	WQ (3k/2k)	CT (1k /1k)	# params
BERT-Baseline (Lee et al., 2019)	Sparse Retr.+Transformer	BERT	26.5	17.7	21.3	110m
T5 (base) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	27.0	29.1	-	223m
T5 (large) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	29.8	32.2	-	738m
T5 (11b) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	34.5	37.4	-	11318m
DrQA (Chen et al., 2017)	Sparse Retr.+DocReader	N/A	-	20.7	25.7	34m
HardEM (Min et al., 2019a)	Sparse Retr.+Transformer	BERT	28.1	-	-	110m
GraphRetriever (Min et al., 2019b)	GraphRetriever+Transformer	BERT	31.8	31.6	-	110m
PathRetriever (Asai et al., 2019)	PathRetriever+Transformer	MLM	32.6	-	-	110m
ORQA (Lee et al., 2019)	Dense Retr.+Transformer	ICT+BERT	33.3	36.4	30.1	330m
Ours ($\mathcal{X}$ = Wikipedia, $\mathcal{Z}$ = Wikipedia)	Dense Retr.+Transformer	REALM	39.2	40.2	**46.8**	330m
Ours ($\mathcal{X}$ = CC-News, $\mathcal{Z}$ = Wikipedia)	Dense Retr.+Transformer	REALM	**40.4**	**40.7**	42.9	330m

Main results

Name	Architectures	Pre-training	NQ (79k/4k)	WQ (3k/2k)	CT (1k /1k)	# params
BERT-Baseline (Lee et al., 2019)	Sparse Retr.+Transformer	BERT	26.5	17.7	21.3	110m
T5 (base) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	27.0	29.1	-	223m
T5 (large) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	29.8	32.2	-	738m
T5 (11b) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	34.5	37.4	-	11318m
DrQA (Chen et al., 2017)	Sparse Retr.+DocReader	N/A	-	20.7	25.7	34m
HardEM (Min et al., 2019a)	Sparse Retr.+Transformer	BERT	28.1	-	-	110m
GraphRetriever (Min et al., 2019b)	GraphRetriever+Transformer	BERT	31.8	31.6	-	110m
PathRetriever (Asai et al., 2019)	PathRetriever+Transformer	MLM	32.6	-	-	110m
ORQA (Lee et al., 2019)	Dense Retr.+Transformer	ICT+BERT	33.3	36.4	30.1	330m
Ours ($\mathcal{X}$ = Wikipedia, $\mathcal{Z}$ = Wikipedia)	Dense Retr.+Transformer	REALM	39.2	40.2	**46.8**	330m
Ours ($\mathcal{X}$ = CC-News, $\mathcal{Z}$ = Wikipedia)	Dense Retr.+Transformer	REALM	**40.4**	**40.7**	42.9	330m

Main results

Name	Architectures	Pre-training	NQ (79k/4k)	WQ (3k/2k)	CT (1k /1k)	# params
BERT-Baseline (Lee et al., 2019)	Sparse Retr.+Transformer	BERT	26.5	17.7	21.3	110m
T5 (base) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	27.0	29.1	-	223m
T5 (large) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	29.8	32.2	-	738m
T5 (11b) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	34.5	37.4	-	11318m
DrQA (Chen et al., 2017)	Sparse Retr.+DocReader	N/A	-	20.7	25.7	34m
HardEM (Min et al., 2019a)	Sparse Retr.+Transformer	BERT	28.1	-	-	110m
GraphRetriever (Min et al., 2019b)	GraphRetriever+Transformer	BERT	31.8	31.6	-	110m
PathRetriever (Asai et al., 2019)	PathRetriever+Transformer	MLM	32.6	-	-	110m
ORQA (Lee et al., 2019)	Dense Retr.+Transformer	ICT+BERT	33.3	36.4	30.1	330m
Ours ($\mathcal{X}$ = Wikipedia, $\mathcal{Z}$ = Wikipedia)	Dense Retr.+Transformer	REALM	39.2	40.2	**46.8**	330m
Ours ($\mathcal{X}$ = CC-News, $\mathcal{Z}$ = Wikipedia)	Dense Retr.+Transformer	REALM	**40.4**	**40.7**	42.9	330m

Main results

Name	Architectures	Pre-training	NQ (79k/4k)	WQ (3k/2k)	CT (1k /1k)	# params
BERT-Baseline (Lee et al., 2019)	Sparse Retr.+Transformer	BERT	26.5	17.7	21.3	110m
T5 (base) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	27.0	29.1	-	223m
T5 (large) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	29.8	32.2	-	738m
T5 (11b) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	34.5	37.4	-	11318m
DrQA (Chen et al., 2017)	Sparse Retr.+DocReader	N/A	-	20.7	25.7	34m
HardEM (Min et al., 2019a)	Sparse Retr.+Transformer	BERT	28.1	-	-	110m
GraphRetriever (Min et al., 2019b)	GraphRetriever+Transformer	BERT	31.8	31.6	-	110m
PathRetriever (Asai et al., 2019)	PathRetriever+Transformer	MLM	32.6	-	-	110m
ORQA (Lee et al., 2019)	Dense Retr.+Transformer	ICT+BERT	33.3	36.4	30.1	330m
Ours ($\mathcal{X}$ = Wikipedia, $\mathcal{Z}$ = Wikipedia)	Dense Retr.+Transformer	REALM	39.2	40.2	**46.8**	330m
Ours ($\mathcal{X}$ = CC-News, $\mathcal{Z}$ = Wikipedia)	Dense Retr.+Transformer	REALM	**40.4**	**40.7**	42.9	330m

Main results

Name	Architectures	Pre-training	NQ (79k/4k)	WQ (3k/2k)	CT (1k /1k)	# params
BERT-Baseline (Lee et al., 2019)	Sparse Retr.+Transformer	BERT	26.5	17.7	21.3	110m
T5 (base) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	27.0	29.1	-	223m
T5 (large) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	29.8	32.2	-	738m
T5 (11b) (Roberts et al., 2020)	Transformer Seq2Seq	T5 (Multitask)	34.5	37.4	-	11318m
DrQA (Chen et al., 2017)	Sparse Retr.+DocReader	N/A	-	20.7	25.7	34m
HardEM (Min et al., 2019a)	Sparse Retr.+Transformer	BERT	28.1	-	-	110m
GraphRetriever (Min et al., 2019b)	GraphRetriever+Transformer	BERT	31.8	31.6	-	110m
PathRetriever (Asai et al., 2019)	PathRetriever+Transformer	MLM	32.6	-	-	110m
ORQA (Lee et al., 2019)	Dense Retr.+Transformer	ICT+BERT	33.3	36.4	30.1	330m
Ours ($\mathcal{X}$ = Wikipedia, $\mathcal{Z}$ = Wikipedia)	Dense Retr.+Transformer	REALM	39.2	40.2	**46.8**	330m
Ours ($\mathcal{X}$ = CC-News, $\mathcal{Z}$ = Wikipedia)	Dense Retr.+Transformer	REALM	**40.4**	**40.7**	42.9	330m

Analysis

- Encoder or Retriever
  - REALM pre-training의 retriever나 encoder 또는 둘다, 성능 개선을 하는지 분석
  - 최고의 결과는 둘다 사용해야함

Table 2. Ablation experiments on NQ's development set.

Ablation	Exact Match	Zero-shot Retrieval Recall@5
REALM	38.2	38.5
REALM retriever+Baseline encoder	37.4	38.5
Baseline retriever+REALM encoder	35.3	13.9
Baseline (ORQA)	31.3	13.9
REALM with random uniform masks	32.3	24.2
REALM with random span masks	35.3	26.1
30× stale MIPS	28.7	15.1

Analysis

- Masking scheme
  - BERT-base의 random token masking & SpanBERT의 random span masking과 비교
  - REALM에서는 중요

Table 2. Ablation experiments on NQ's development set.

Ablation	Exact Match	Zero-shot Retrieval Recall@5
REALM	38.2	38.5
REALM retriever+Baseline encoder	37.4	38.5
Baseline retriever+REALM encoder	35.3	13.9
Baseline (ORQA)	31.3	13.9
REALM with random uniform masks	32.3	24.2
REALM with random span masks	35.3	26.1
30× stale MIPS	28.7	15.1

Analysis

- MIPS index refresh rate
  - index refresh을 빈번하게 하는 것에 대한 중요성을 파악
  - 빈번하게 하는 것이 중요

Table 2. Ablation experiments on NQ's development set.

Ablation	Exact Match	Zero-shot Retrieval Recall@5
REALM	38.2	38.5
REALM retriever+Baseline encoder	37.4	38.5
Baseline retriever+REALM encoder	35.3	13.9
Baseline (ORQA)	31.3	13.9
REALM with random uniform masks	32.3	24.2
REALM with random span masks	35.3	26.1
30× stale MIPS	28.7	15.1

Analysis

- Examples of retrieved documents
  - (a) BERT보다 (c)REALM의 확률이 더 높음
  - (c)보다 (b)의 확률은 더 확실히 증가
  - REALM이 unsupervised learning으로 학습되지만, 마스킹된 부분을 잘 채워넣기 위해 document 검색을 잘함

Table 3. An example where REALM utilizes retrieved documents to better predict masked tokens. It assigns much higher probability (0.129) to the correct term, "Fermat", compared to BERT. (Note that the blank corresponds to 3 BERT wordpieces.)

	x:	An equilateral triangle is easily constructed using a straightedge and compass, because 3 is a ___ prime.
(a) BERT	$p(y = \text{"Fermat"} \mid x)$	$= 1.1 \times 10^{-14}$ (No retrieval.)
(b) REALM	$p(y = \text{"Fermat"} \mid x, z)$	$= 1.0$ (Conditional probability with document z ="257 is … a Fermat prime. Thus a regular polygon with 257 sides is constructible with compass …")
(c) REALM	$p(y = \text{"Fermat"} \mid x)$	$= 0.129$ (Marginal probability, marginalizing over top 8 retrieved documents.)

Appendix

x: "Jennifer ___ formed the production company Excellent Cadaver."	
BERT	also (0.13), then (0.08), later (0.05), . . .
REALM ($\mathcal{Z}$ =20 Dec 2018 corpus)	smith (0.01), brown (0.01), jones (0.01)
REALM ($\mathcal{Z}$ =20 Jan 2020 corpus)	**lawrence** (0.13), brown (0.01), smith (0.01), . . .

Table 4. An example where REALM adapts to the updated knowledge corpus. The Wikipedia page "Excellent Cadaver" was added in 2019, so the model was not about to recover the word when the knowledge corpus is outdated (2018). Interestingly, the same REALM model pre-trained on the 2018 corpus is able to retrieve the document in the updated corpus (2020) and generate the correct token, "Lawrence".

Conclusion

- Inference에서 knowledge corpus를 통해 reasoning 가능한 pre-training 학습 방법 제시
- MLM 학습시 retrieval 모델까지 back-prob되도록 모델 수정
- pre-training 에서 지속적으로 retrieval 모델도 업데이트
  - MIPS index refresh
- 문서 후보 (top k)를 매 step마다 뽑음
- retriever + knowledge-augmented encoder 최고 성능
- Open-QA task에서 SOTA!!

감사합니다