REALM:

Retrieval-Augmented Language Model Pre-Training

REALM

- Google Research 논문
- Kelvin Guu : Stanford Ph.D
- **Kenton Lee**: Washington Ph.D
  - *BERT* 3 저자, *ELMo* 6 저자
  - *Coreference Resolution* 베이스라인 구축
- **Open Domain QA**를 위해 관련 지식을 뽑아올 수 있는 **Retreiver**를 함께 학습!
- 사실상 *Latent Retrieval for Weakly Supervised Open Domain Question Answering (Kenton Lee et al. 2019)* 의 후속 연구

REALM: Retrieval-Augmented Language Model Pre-Training

Kelvin Guu [*1] Kenton Lee [*1] Zora Tung [1] Panupong Pasupat [1] Ming-Wei Chang [1]

Abstract

Language model pre-training has been shown to capture a surprising amount of world knowledge, crucial for NLP tasks such as question answering. However, this knowledge is stored implicitly in the parameters of a neural network, requiring ever-larger networks to cover more facts. To capture knowledge in a more modular and interpretable way, we augment language model pre-training with a latent *knowledge retriever*, which allows the model to retrieve and attend over documents from a large corpus such as Wikipedia, used during pre-training, fine-tuning and inference. For the first time, we show how to pre-train such a knowledge retriever in an unsupervised manner, using masked language modeling as the learning signal and backpropagating through a retrieval step that considers millions of documents. We demonstrate the effectiveness of Retrieval-Augmented Language Model pre-training (REALM) by fine-tuning on the challenging task of Open-domain Question Answering (Open-QA). We compare against state-of-the-art models for both explicit and implicit knowledge storage on three popular Open-QA benchmarks, and find that we outperform all previous methods by a significant margin (4-16% absolute accuracy), while also providing qualitative benefits such as interpretability and modularity.

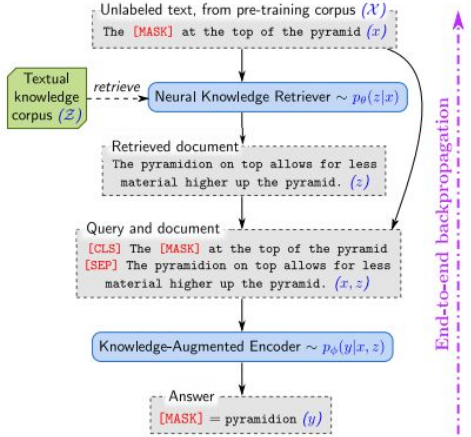


Figure 1. REALM augments language model pre-training with a **neural knowledge retriever** that retrieves knowledge from a **textual knowledge corpus**, $\mathcal{Z}$ (e.g., all of Wikipedia). Signal from the language modeling objective backpropagates all the way through the retriever, which must consider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

correctly predict the missing word in the following sentence: "The ___ is the currency of the United Kingdom" (answer: "pound").

Contents

Abstract

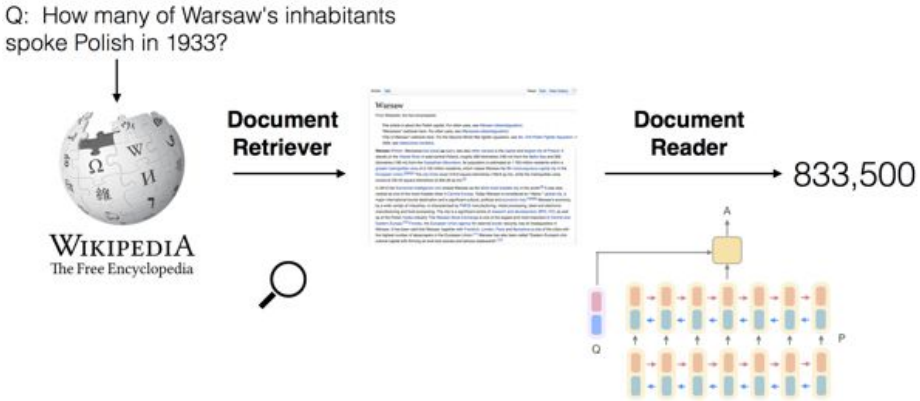
- **Language model** 사전 학습은 QA와 같은 태스크 수행에 있어 필요한 **현실 세계의 지식**들을 학습
- **[Problem]**: 사전 학습을 통해 얻어진 지식들은 파라미터에 **해석할 수 없는 정보**로 내포하게 되며, 더 많은 지식을 주입하고자 한다면 그만큼 **더 많은 파라미터**가 필요하게 됨
- **[Solution]**: 지식 습득을 보다 **해석**하기 쉽고, **모듈화**하기 위해 **Knowledge Retriever**를 제안
  : 모델로 하여금 Wikipedia와 같은 **거대한 코퍼스**에서 관련 지식을 검색하고,
  검색 결과를 활용해 예측 값을 낼 수 있도록 도와줌 *(retrieve-then-predict)*
- **Knowledge Retriever**는 사전 학습, 전이 학습, 추론 등 **모든 단계**에서 활용 가능
- **[Result]**: 현실 세계 지식을 가장 많이 필요로 하는 **Open-domain QA** 태스크에 적용해본 결과, 이전에 제안된 모든 모델들보다 좋은 성능을 보였음

(cf. 이전에 제안된 모델: [Implicit] T5, [Explicit] ORQA, HardEM, …)

Introduction

- **BERT, RoBERTa, T5** 등의 **PLM** 모델들은 **사전 학습에 사용된 거대한 코퍼스**를 통해 많은 양의 현실 세계 지식을 저장하는 것으로 알려짐 *(Language models as knowledge bases?)*
- **BERT**에 *"The [MASK] is the currency of the United Kingdom"* 라는 문장을 입력 값으로 넣어주면, 모델은 마스킹 된 토큰을 ***"pound"*** 로 정확하게 예측할 것
- 그러나 **PLM** 모델들이 학습한 지식들은 파라미터에 **함축적으로** 내포되기 때문에, 어떤 지식들이 정확히 어디에 저장되는지에 대한 해석하기가 어렵다는 단점
- 또한 모델들로 하여금 **더 많은 지식**을 저장하게끔 훈련시키고자 한다면, 모델러는 보다 **더 큰 사이즈의 네트워크**를 구축해야만 하고, 이는 매우 제한적인 개선 방안

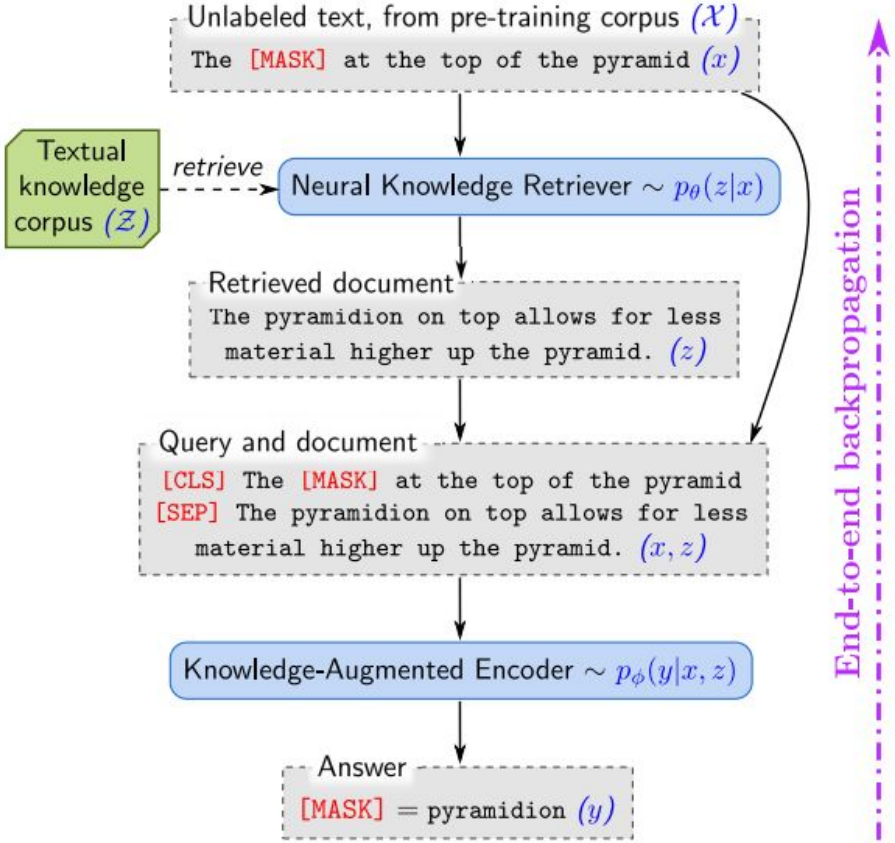
Introduction



- 모델이 학습하는 지식을 보다 **해석** 가능하게 하고, **모듈화**하기 위해 사전 학습 시 *(learnable)* **Textual Knowledge Retriever**를 붙이는 **Retrieval-Augmented Language Model** (REALM) 제안
- 파라미터에 **함축적으로** 지식을 저장하는 이전 모델들과 달리, REALM은 어떠한 문서를 검색해 사용하는지가 **명시적으로** 드러나므로 보다 해석이 쉬워짐
- 즉, 모델은 예측 값을 만들기 전에 **Retriever**를 사용해 Wikipedia와 같은 **거대한 코퍼스**에서 관련 문서를 검색하고, **검색된 문서들**에 어텐션 메커니즘을 적용하는 방법으로 예측 값을 결정

Introduction

- 모델의 **Perplexity** (PPL) 를 개선하는 retrieval에는 **리워드**를, 그렇지 않은 retrieval에는 **페널티**를 부여
- 그림에서 모델이 *"The [MASK] at the top of the pyramid"* 라는 문장 내 마스크 토큰을 제대로 예측하도록 하기 위해서는 **Retriever**가 *"The pyramidion on top allows for less material higher up the pyramid"* 라는 문장을 검색해주는 것이 바람직
- 때문에 REALM의 접근법은 ***retrieve-then-predict*** !



Introduction

코사인 유사도를 기반으로 Spherical K-Means 클러스터링 적용해 k 개의 클러스터 형성해 활용 *(post)*

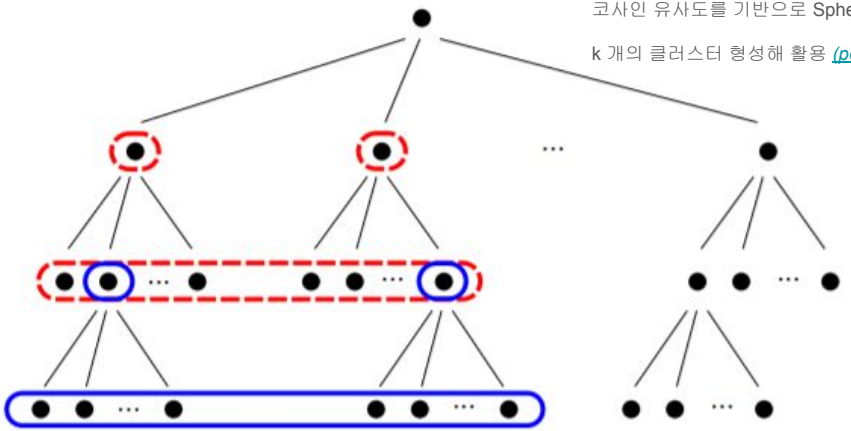


Figure 1: Walk down a hierarchical clustering tree: at each level we have a candidate set for the next level. In the first level, the dashed red boxed represent the p best matches, which gives us a candidate set for the second level, etc.

- 하지만 Wikipedia와 같은 대형 코퍼스에 Retrieval을 적용하는 것은 **Computationally expensive** !
- 해당 문제를 해결하기 위해 각 문서에 대한 연산을 **캐싱**해놓은 후 **비동기적으로 업데이트** 시켰으며, 가장 유관한 문서를 선택하는 작업에는 **Maximum Inner Product Search** (MIPS) 를 적용

Introduction

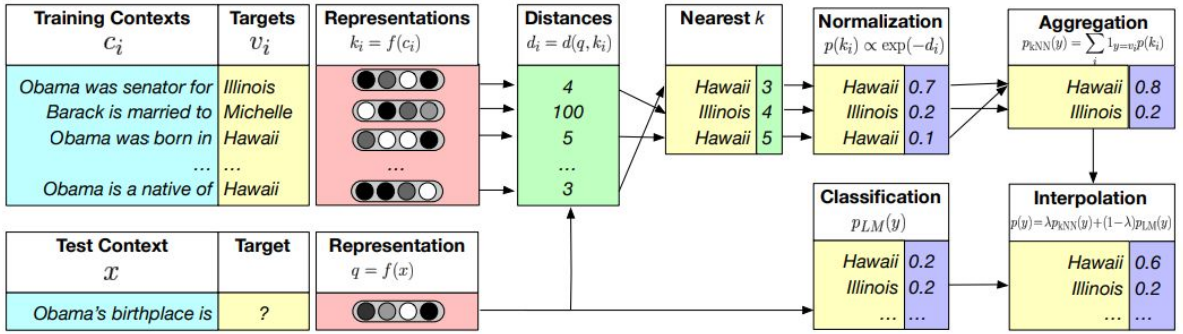


Figure 1: An illustration of kNN-LM. A datastore is constructed with an entry for each training set token, and an encoding of its leftward context. For inference, a test context is encoded, and the k most similar training contexts are retrieved from the datastore, along with the corresponding targets. A distribution over targets is computed based on the distance of the corresponding context from the test context. This distribution is then interpolated with the original model's output distribution.

- 많은 선행 연구들에서 **Retrieval step**의 이점을 증명했으나, 사전 학습에 적용한 사례는 없음
- 언어 모델 연구 중 사전 학습된 PLM에 kNN 알고리즘을 덧붙여 **Inference 개선**을 시도한 **kNN-LM**의 경우, 언어 모델 외 다른 다운스트림 태스크에 적용할 수 없는 프레임워크
- **kNN 알고리즘**의 경우, 라벨링 데이터가 필요하지만 **kNN-LM**은 단순 LM 예제들만을 저장하기 때문

Background: Language Model Pre-training

- 언어 모델 사전 학습의 목적은 Unlabeled 코퍼스에서 유용한 언어 표현을 학습하는 것
- 사전 학습된 언어 모델은 이후 특정 다운스트림 태스크에 Fine-tuning 해 사용 가능
- REALM은 BERT 스타일의 Masked Language Model (MLM) 을 활용
- 좋은 MLM은 의미적 / 문법적 지식, 그리고 현실 세계 지식을 모두 학습할 수 있어야 함

*"The **[MASK]** is the currency **[MASK]** the UK";* *y = ("**pound**", "**of**")*

- 위 예에서 **pound** 는 현실 세계의 지식을 **of** 는 문법적 지식을 학습한 결과

Background: Open-domain Question Answering

- 모델이 현실 세계 지식을 학습했는지 측정하는 가장 좋은 방법은 문제 해결에 있어 현실 세계 지식이 중요하게 작용하는 **Open-domain Question Answering** 태스크를 풀어보게 하는 것
- 여기서 **"Open"** 은 SQuAD와 같은 전통적인 기계 독해 태스크와 달리 답을 내포하고 있는 컨텍스트 문장을 부여받지 않은 채, 문제를 해결해야 함을 의미
- 또한 모든 질문에 대한 답을 해내야 하므로, **수백만 문서에 대한 지식**을 확보하고 있어야 함
- 대부분의 **Open-QA** 모델들은 **Textual Knowledge** 코퍼스 ***Z*** 와 **retrieval-based** 접근법을 활용
  : 질문 ***x*** 가 주어졌을 때, ***x*** 에 대한 답변을 찾을 수 있는 관련 문서 ***z*** 를 코퍼스 ***Z*** 에서 추출한 후, 답변 ***y*** 를 ***x*** 와 ***z*** 를 활용해 추출하는 구조

Approach: REALM's generative process

- 사전 학습과 Fine-tuning 시 모두 입력 ***x*** 에 대해 가능한 출력 ***y*** 를 추출하는 분포 ***p(y | x)*** 를 학습
- 사전 학습 시에는 태스크가 **MLM**
  : ***x***는 사전 학습 코퍼스 ***X*** 에서 추출된 문장으로 일부 토큰들이 마스킹, ***y*** 는 마스크 토큰의 원 단어
- Fine-tuning 시에는 태스크가 **Open-QA**: ***x*** 는 질문, ***y*** 는 답변
- REALM은 입력에 대한 출력을 나타내는 분포 ***p(y | x)*** 를 두 단계로 분해
  1. Retrieve: 입력 값 ***x*** 에 대해 도움이 될 수 있는 문장 ***z*** 를 Knowledge 코퍼스 ***Z*** 에서 검색
     해당 분포를 ***p(z | x)*** 로 두고 학습
  2. Predict: 앞서 검색된 ***z*** 와 입력 값 ***x*** 를 활용해 출력 값 ***y*** 를 생성; ***p(y | z, x)***
- 출력 ***y*** 를 생성할 수 있는 전체 likelihood를 계산해야 하므로, ***z*** 를 잠재 변수로 두고 가능한 모든 문서 ***z*** 에 대해 likelihood를 계산: $p(y \mid x) = \sum_{z \in \mathcal{Z}} p(y \mid z, x)\, p(z \mid x).$

Approach: Model architecture

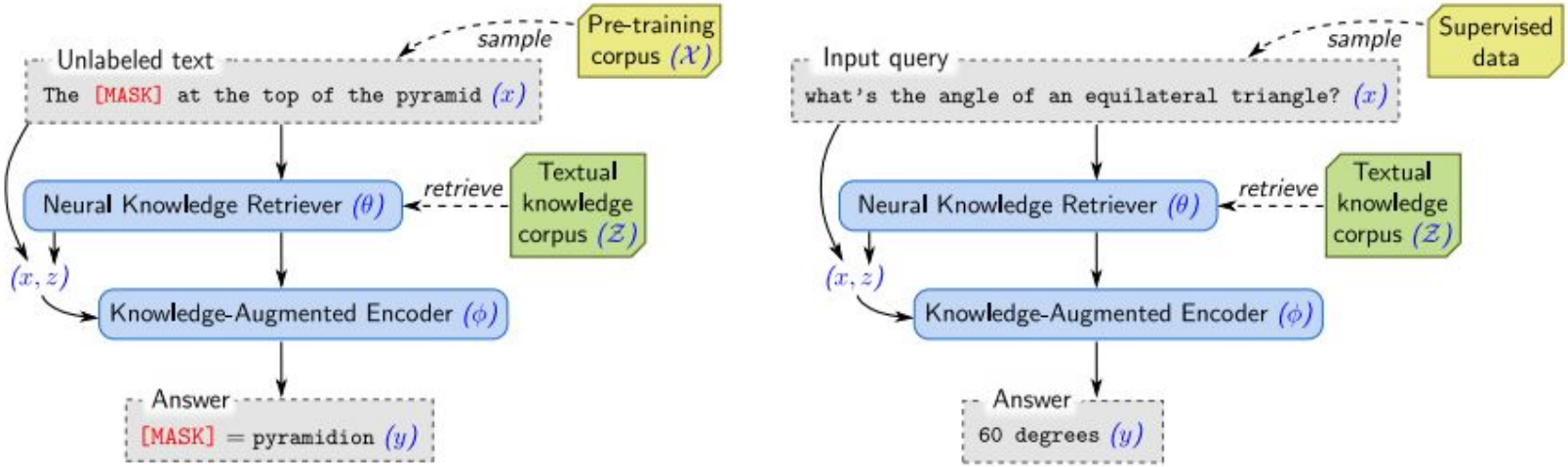


Figure 2. The overall framework of REALM. **Left:** *Unsupervised pre-training.* The knowledge retriever and knowledge-augmented encoder are jointly pre-trained on the unsupervised language modeling task. **Right:** *Supervised fine-tuning.* After the parameters of the retriever (θ) and encoder (ϕ) have been pre-trained, they are then fine-tuned on a task of primary interest, using supervised examples.

Approach: Model architecture (Knowledge Retriever)

- ***p(z | x)*** 를 모델링 하는 Neural Knowledge Retriever

- *join* 함수는 우측과 같이 정의

  : *Wordpiece tokenization 활용*

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

- 임베딩 함수는 우측과 같이 정의

  : *[CLS] 토큰의 차원을 줄이기 위해 W에 프로젝션*

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

- 이제 Retriever는 우측 식으로 정의될 수 있음

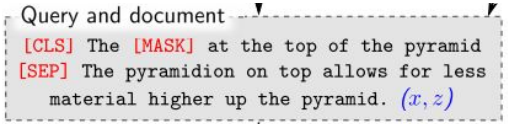
  : *입력 값 **x**와 문서 **z** 의 유사도를 구하는 함수 **f**는 내적 값,*

  retrieval distribution은 모든 유사도에 대한 Softmax 결과

$$p(z \mid x) = \frac{\exp f(x, z)}{\sum_{z'} \exp f(x, z')},$$
$$f(x, z) = \text{Embed}_{\text{input}}(x)^\top \text{Embed}_{\text{doc}}(z),$$

- Retriever는 BERT **스타일**의 입력 값을 취하는 **Transformer !**

Approach: Model architecture (Knowledge-augmented Encoder)



- ***p(y | z, x)*** 를 모델링 하는 Knowledge-augmented Encoder
- 입력 ***x*** 와 문서 ***z*** 내 본문을 하나의 시퀀스로 join 해 Transformer에 입력 *(cf. retriever와 별개의 모델)*
  이는 ***y*** 를 예측하기 전 ***x*** 와 ***z*** 사이에서 충분한 어텐션을 추출하기 위함
- 사전 학습과 Fine-tuning 시의 학습은 다른 방향으로 진행
  - 사전 학습: **MLM**을 수행하므로, BERT와 같은 Loss func 사용:

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

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

  - Fine-tuning: ***y*** 를 ***z*** 내에 존재하는 스팬으로 간주하며,
    S(z, y) 는 ***z*** 내 ***y*** 와 매치되는 스팬의 집합
    (cf. START와 END는 각각 스팬의 시작 토큰과 마지막 토큰의 벡터를 의미)

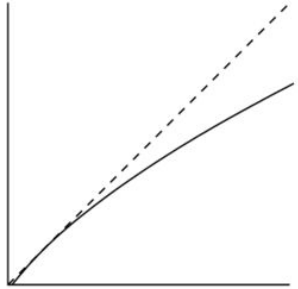
$$p(y \mid z, x) \propto \sum_{s \in S(z,y)} \exp\left(\mathtt{MLP}\left(\left[h_{\mathtt{START(s)}}; h_{\mathtt{END(s)}}\right]\right)\right)$$

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

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

- 즉, 정답이 되는 스팬의 **시작 토큰**과 **마지막 토큰**을
  이어붙인 후 **MLP**를 태운 값들의 합과 ***p(y | z, x)*** 가 비례

Approach: Training



- 사전 학습과 Fine-tuning 시 모두 학습 목표는 ***log p(y | x)*** 를 최대로 키우는 것
- **[Problem]**: ***p(y | x)*** 를 구하기 위해 코퍼스 ***Z*** 내 모든 문서 ***z*** 와의 연산을 행해야 하는 것
- **[Solution]**: 모든 연산 대신, ***p(z | x)*** 에서 도출된 **top-k** 개 문서만을 활용해 ***p(y | x)*** 를 구하자

  대부분의 문서가 ***p(z | x)*** 에서 0에 가까운 확률 값을 가진다면 이는 타당한 근사 방법
- 위 근사에도 불구하고, **top-k** 개 문서를 추출하기 위한 효율적인 방법이 필요
- 앞서 ***p(z | x)*** 는 **내적** 값에 의해 결정되도록 정의했으므로, 우리는 **MIPS** 를 활용해

  top-k 개 문서를 찾을 수 있음: **MIPS** 는 문서 개수에 따라 시간과 메모리가 **Sub-linearly** 하게 증가

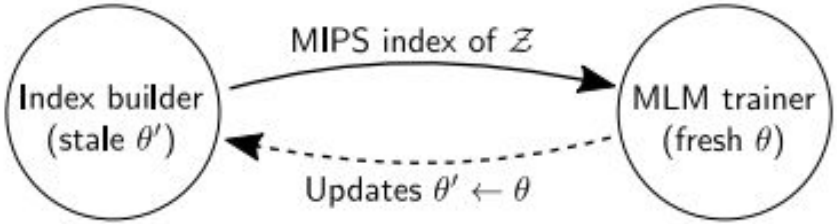
Approach: Training

- **MIPS**를 활용하기 위해서는 모든 문서 ***z*** 에 대한 임베딩 값을 사전에 계산해 놓아야 함
- 하지만 해당 데이터는 **문서 임베딩 매트릭스**가 업데이트 되는 순간 ***p(z | x)*** 와 일치하지 않게 됨
- 즉, **MIPS** 가 활용하는 인덱스가 더 이상 사용 불가능해지는 것
- 이러한 문제를 해결하기 위해 학습 스텝을 **수 백번** 진행한 후,

  비동기적으로 **MIPS** 인덱스를 재설정하고 문서를 다시 임베딩하는 방식을 취하기로 함
- 물론 이 과정에서도 업데이트 사이에 **MIPS** 인덱스가 벗어나는 현상이 발생할 수 있으나,

  해당 과정은 단순히 **top-k** 개 문서를 추출하기 위함이므로 전체 프로세스에 큰 영향을 주지는 않음
- 또한 학습 과정 동안 새로이 업데이트 된 파라미터를 활용해 ***p(z | x)*** 를 다시 계산하기도 함

Approach: Training (Implementing asynchronous MIPS refreshes)



- 우리는 2개의 작업을 병렬적으로 실행해 **MIPS** 인덱스를 비동기적으로 업데이트할 수 있음
  - **Trainer**: 파라미터에 대한 기울기 업데이트
  - **Index builder**: 업데이트된 파라미터로 문서 다시 임베딩한 후, MIPS 인덱스 재설정
- 해당 비동기 프로세스는 사전 학습과 Fine-tuning 시 모두 적용 가능한 작업이지만,

  사전 학습 시에만 해당 작업을 수행하였고 Fine-tuning 시에는 **고정된 MIPS 인덱스**와

  문서 임베딩 매트릭스 활용: 쿼리 임베딩 매트릭스는 Fine-tuning 되므로 Retrieval 함수는 계속해서 개선

Approach: Training (What does the retriever learn?)

- REALM이 문서를 **Retrieval** 하는 과정은 **latent** 하기 때문에 목적 함수가 어떻게 올바른 문서 **retrieval**에 도움을 주는지는 명확하지 않음
- 그러나 모델이 Gradient Descent를 통해 어떻게 올바른 **retrieval**에 보상을 하는지는 설명이 가능

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

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

- 질문 ***x*** 와 문서 ***z*** 가 주어졌을 때 ***f(x, z)*** 는 **Retriever**가 부여한 두 임베딩의 **유사도**
- 문서 ***z*** 에 대해 기울기는 **Retriever**로 하여금 유사도 ***f(x, z)*** 를 ***r(z)*** 만큼 변화하게 함: 커지게 혹은 작아지게
- ***r(z)*** 는 ***p(y | z, x)*** 가 ***p(y | x)*** 보다 클때만 양수가 될 수 있음
  - ***p(y | z, x)*** 는 문서 ***z*** 를 활용해 ***y*** 를 예측할 확률, ***p(y | x)*** 는 임의의 문서 ***z*** 를 ***p(z | x)*** 에서 샘플링한 ***p(y | x, z)*** 의 예상 값
  - 그러므로 문서 ***z*** 는 예상 값보다 좋은 값을 내놓는 경우에만 ***x*** 와 가까워지도록 업데이트

Approach: Injecting Inductive bias into Pre-training

REALM을 연구하며 발견한 **Retrieval**에 더 좋은 효과를 보일 수 있는 트릭들을 소개 !

- **Salient Span Masking**: 사전 학습 시, 문장 내 컨텍스트만 필요로 하는 예제가 아닌 **현실 세계 지식**을 필요로 하는 예제를 많이 학습하도록 하고자 했음. 이를 위해 사전 학습된 **BERT Tagger**와 **정규표현식**을 활용해 문장 내 존재하는 엔티티 및 날짜를 감지해 이들 중 하나를 의도적으로 마스킹하도록 하였고, 이는 실제로 많은 성능 개선을 가져다줌

- **Null document**: (A)의 처리에도 불구하고 모든 마스크 토큰이 현실 세계 지식을 요하지는 않음. 이처럼 Retrieval이 필요하지 않은 경우를 모델링하기 위해 **top-k** 개 문서에 빈 문서 *null document* 를 추가해 해당 예외를 처리하였음

Approach: Injecting Inductive bias into Pre-training

- **Prohibiting trivial retrievals**:
  - 사전학습에 사용되는 코퍼스 ***X*** 와 지식 코퍼스 ***Z*** 가 동일할 경우, 훈련 문장 ***x*** 가 문서 ***z*** 에서 추출되는 경우가 발생
  - 이러한 경우 모델은 단순히 마스킹 되지 않은 버전의 ***x*** 를 ***z*** 에서 찾아 문제를 해결하게 되고, 이는 ***p(z | x)*** 에 과도한 긍정 학습을 유발 및 단순 스트링 매치로 문제를 해결하고자 하게 됨
  - 이를 방지하기 위해 ***x*** 가 ***z*** 에서 추출될 수 있는 데이터는 배제

- **Initialization**:
  - 입력 값과 문서에 대한 **임베딩 매트릭스**가 좋은 가중치를 가지지 않는다면, **Retriever** 는 ***x*** 와 무관한 문서 ***z*** 를 추출하게 되고 **Encoder**는 ***z*** 를 무시하도록 학습
  - 해당 문제를 해결하기 위해 특정 문장이 주어졌을 때, 해당 문장이 어떤 문서에서 추출된 것인지를 맞추는*Inverse Cloze Task* (Lee et al. 2019) 를 **Retriever**의 사전학습 태스크로 활용
  - **Encoder**는 *BERT-base* 사전 학습을 그대로 따름

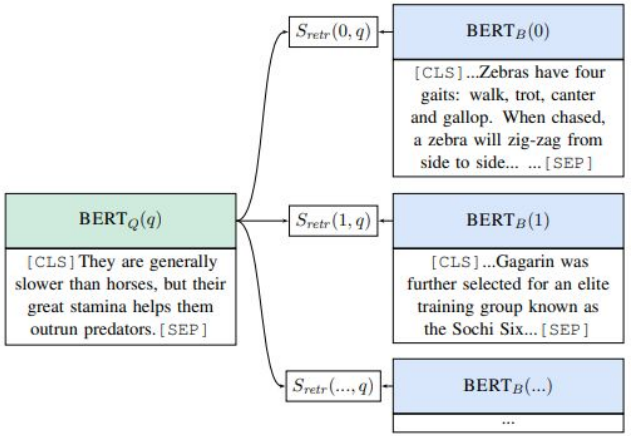


Figure 2: Example of the Inverse Cloze Task (ICT), used for retrieval pre-training. A random sentence (pseudo-query) and its context (pseudo evidence text) are derived from the text snippet: *"...Zebras have four gaits: walk, trot, canter and gallop.* ***They are generally slower than horses, but their great stamina helps them outrun predators.*** *When chased, a zebra will zig-zag from side to side..."* The objective is to select the true context among candidates in the batch.

Experiment: Open-QA Benchmarks

- Open-QA 태스크에 대해서는 여러 벤치마크가 고안되었지만,
  본 실험에서는 질문 제작자가 정답을 모른 채 질문을 작성한 벤치마크만을 다룰 것
- 이는 보다 정보를 찾고자 하는 **필요성**에 의해 작성된 질문이고,
  특정 답을 염두에 둔 채로 작성되지 않았기 때문에 **편향이 최소한**으로 반영된 데이터셋

- **NaturalQuestions-Open**: Google에서 발생한 자연스러운 쿼리와 질문으로 구성된 데이터셋으로,
  5 토큰 이하로 구성된 “short answer type”에 대해서만 실험을 진행
- **WebQuestions**: Google Suggest API 에서 추출한 데이터셋
- **CuratedTrec**: MSNSearch, AskJeeves와 같은 실제 사이트에서 사용자가 발행한 질문과 답변에 대한 쌍으로 구성된 데이터셋으로, 여러 개의 가능 답변과 스펠링 에러를 고려하기 위해 정답을 정규 표현식으로 제공. 생성 모델에 대해서 테스트하기 어렵기 때문에 생성 모델은 본 데이터셋에 대한 실험을 진행하지 않음

Experiment: Approaches compared

- **Retrieval-based**:
  - 대부분의 선행 연구는 학습 가능하지 않은 BoW, 엔티티 링킹 등의 **휴리스틱한 Retrieval** 기법을 활용
  - 이후, 학습 모델을 통해 **Re-rank**를 적용하기는 하나 애초에 Retrieval에서 성능 감소가 발생
  - 저자 선행 연구인 **ORQA** (Lee et al. 2019) 에서 학습 가능한 Retrieval 제안했으나,
    MIPS 인덱스 업데이트 등의 **세밀한 개선**을 더한 것이 REALM !

- **Generation-based**:
  - 컨텍스트 없이 질문 정보만으로 답변을 생성하는 것이 가능할 것인가에 대한 의구심이 학계에 존재
  - **GPT-2** 가 해당 가능성을 보여주었으나, 엄청나게 좋은 성능을 보이지는 않았음
  - **T5** 는 그 가능성을 더 키워주었지만, 현재까진 컨텍스트가 주어진 상황에서만 실험되었음
    T5의 가능성을 실험해보기 위해 Open-QA 태스크 역시 수행하도록 실험

Experiment: Implementation Details

- **Fine-tuning:**
  - **ORQA** (Lee et al. 2019) 와의 직접적인 비교를 위해 **동일한 하이퍼 파라미터** 사용
  - 지식 코퍼스는 **2018년 12월 10일** 기준의 영어 위키피디아 스냅샷 사용
  - 각 문서는 **288개**의 Wordpiece 토큰으로 구성되게끔 나누어 총 **1,300만 개**의 검색 가능 문서 생성
  - Fine-tuning Inference 동안에는 **top-5** 개의 문서를 후보군을 추출하도록 설정
- **Pre-training**:
  - **64개 TPU**를 활용해 **200k** 스텝 동안 사전 학습
  - lr은 **3e-5**로 설정하였으며, BERT의 기본 옵티마이저 사용
  - MIPS 인덱스를 설정하는 문서 임베딩 작업은 **16개 TPU**에서 병렬적으로 처리
  - *null document* 를 포함한 **8개**의 top-k 문서를 검색해 활용
  - 사전 학습 코퍼스로는 지식 코퍼스와 동일한 **위키피디아**와 **CC-News** 중 하나를 선택해 사용

Experiment: Main results

Table 1. Test results on Open-QA benchmarks. The number of train/test examples are shown in paretheses below each benchmark. Predictions are evaluated with exact match against any reference answer. Sparse retrieval denotes methods that use sparse features such as TF-IDF and BM25. Our model, REALM, outperforms all existing systems.

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

- REALM이 모든 모델들을 앞서긴 했으나, **T5의 약진**이 눈에 띔
- T5-11b는 Retrieval 기반 선행 연구들을 모두 뛰어넘음: 다만 너무 큰 크기 + 사전 학습 시 SQuAD 데이터셋 사용
- 적게는 20개, 많게는 80개의 문서를 활용하는 선행 연구들에 비해 우리는 **단 5개의 문서**만 활용

Experiment: Analysis (ablation study)

- **Encoder or Retriever**:
  - 처음으로 보고자 했던 것은 사전 학습이 **Retriever**와 **Encoder**를 개선시키는지 여부
  - 둘 다 사전 학습이 성능 향상에 영향을 미침을 표를 통해 알 수 있음

- **Masking scheme**:
  - Salient Span 마스킹 전략을 BERT 식 및 SpanBERT 식 마스킹과 비교
  - Salient Span 마스킹이 SpanBERT에서는 좋은 성능을 보이지 않았지만, REALM에서는 좋은 성능을 보이는 것으로 기록됨
  - 직관적으로 생각해보았을 때, Latent variable은 학습할 때 Retrieval의 영향을 많이 받게 되는데 동일한 형태의 학습 시그널을 계속 주는 것이 더 낫기 때문

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

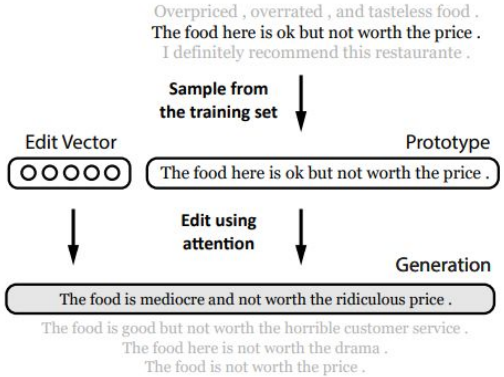
Discussion and Related Work

- **Language modeling with corpus as context**

  : 그동안 언어 표현 모델들은 문제 해결을 위해 더 많은 컨텍스트를 참조할 수 있도록 발전해왔음

  : Word2Vec (단어), Skip-Thought Vector, ELMo (문장), GPT, BERT (문단), ….

  : REALM은 **코퍼스 전체**에 대한 맥락을 볼 수 있는 모델로 볼 수 있음



- **Retrieve-and-edit with learned retrieval**

  : 의미는 같지만 **다르게 작성된 문장**을 보다 잘 표현하고, **Controllable 한 생성**을 하기 위한 *retrieve-and-edit* (Guu et al. 2018) 연구가 수행된 바 있으며, 이는 어휘적으로 유사한 **Edit Vector**를 검색 및 후처리에 활용하는 형태

  : REALM도 비슷한 방식을 취하지만 자기 자신의 PPL을 줄이기 위해 **스스로** 학습한다는 것이 차이

  : Retriever의 Jointly training을 통해 REALM은 단순한 어휘 정보를 넘어 실제 세계의 정보를 학습하는 모델이 되었음

Discussion and Related Work

- **Scalable grounded neural memory**

  : 문서 인덱스는 **문서 임베딩**이 **Key** 값인 **메모리**로 볼 수 있음

  : 인덱스를 통해 Sub-linear 하게 메모리에 접근할 수 있는 **Memory Network** !

  : 이를 활용한다면 큰 언어 모델들에 **Scalable 한 메모리 레이어**를 덧붙일 수도 있을 것

  : 다른 방식과 차별점이 있다면 REALM의 메모리는 단순한 수치 값을 지니는 벡터가 아니라,

  문서 자체이므로 정답 예측에 사용된 문서가 어느 것인지를 요하는 Open-QA와 같은 태스크에 매우 적합

- **Unsupervised Corpus Alignment**

  : 어텐션 기반의 Seq2seq 모델들에서 출력 문장은 관련 토큰에 대한 **Latent selection**에 의해 생성되었음

  : 그리고 이로 인해 **소스 토큰**과 **타겟 토큰** 간의 **Alignment**가 모델 중심으로 이루어질 수 있었음

  : REALM에서도 마찬가지로 사전 학습 코퍼스 ***X*** 와 지식 코퍼스 ***Z*** 가 모델 중심으로 **Alignment** 될 수 있음

Experiment: Analysis (ablation study)

- **MIPS index refresh rate**:
  - 기존에는 대략 500번의 훈련 스텝 진행 후, MIPS 인덱스 업데이트
  - 업데이트 빈도를 늦추게 되면 더 안좋은 성능을 보이는 것을 확인
  - 컴퓨테이션이 허용하는 한 더 빈번한 업데이트를 진행할 경우, 보다 좋은 성능을 기록할 여지가 있음

- **Examples of retrieved documents**:
  - 아래 표는 REALM의 예측 값을 나타냄
  - REALM은 (b)와 같이 관련 문서를 검색할 수 있으므로, 정답을 맞출 수 있는 Marginalized Prob이 BERT에 비해 월등히 높아짐

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

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.)

Future Work

- 본 논문에서는 단순히 큰 지식 코퍼스에 대해 **Reasoning을 수행**할 수 있도록

  사전 학습을 시키는 것에 주로 초점이 맞추어져 있었음
- 이뿐만 아니라, **구조화된 지식**을 함께 넣어 어떤 엔티티가 도움이 되는지에 대한 학습을 한다던가,
- Low-resource 언어의 표현력을 높이기 위해 High-resource 언어에서 지식을 Retrieve 해 활용하는

  Multi-lingual 의 연구를 수행한다던가,
- 텍스트에서는 얻을 수 없는 지식들을 찾기 위해 **이미지**나 **비디오** 등을 Retrieval 하는

  Multi-modal 의 연구를 수행해볼 수 있을 것 !