오픈 도메인 질문 응답을 위한 다중 문서 리더 및 검색기의 End-to-End 학습

초록

1. **엔드 투 엔드 차별화 가능한 훈련 방법**: 연구에서는 여러 개의 검색된 문서에서 정보를 결합하여 답변을 생성하는 오픈 도메인 질문 응답 시스템을 위한 새로운 훈련 방법을 제안합니다. 이 방법은 전체 프로세스를 하나의 통합된 모델로 훈련할 수 있도록 합니다.
2. **잠재 변수 모델링**: 검색 결정은 관련 문서 집합에 대한 잠재 변수로 모델링됩니다. 즉, 어떤 문서가 질문에 대한 답변을 제공할 수 있는지를 결정하는 과정이 잠재 변수로 표현됩니다.
3. **마진화의 어려움**: 검색된 문서 집합에 대해 마진화(marginalization)를 수행하는 것은 계산적으로 어려운 작업입니다. 이를 해결하기 위해 연구에서는 기대-최대화(expectation-maximization, EM) 알고리즘을 사용하여 이 과정을 근사합니다.
4. **반복적 추정**: 연구에서는 주어진 질문에 대한 관련 문서 집합(잠재 변수)의 값을 반복적으로 추정하고, 이 추정치를 사용하여 검색기와 리더의 매개변수를 업데이트합니다.
5. **엔드 투 엔드 훈련의 이점**: 이러한 엔드 투 엔드 훈련 방식은 훈련 신호가 리더에서 검색기로 더 잘 흐르도록 하여, 검색기가 질문에 대해 더 관련성 높은 문서를 선택하고, 리더가 더 정확한 문서를 기반으로 답변을 생성할 수 있도록 합니다.
6. **실험 결과**: 세 가지 벤치마크 데이터셋에서 실험한 결과, 제안된 방법이 기존의 유사한 크기의 접근 방식보다 2-3 포인트 높은 정확도를 달성하며 새로운 최첨단 결과를 기록했습니다.
7. **명시적 감독 없이 학습 가능성**: 마지막으로, 이 연구는 검색 결정에 대한 명시적인 감독 없이도 검색을 통해 답변 생성을 개선할 수 있는 가능성을 보여줍니다. 이는 모델이 스스로 학습하여 더 나은 성능을 발휘할 수 있음을 의미합니다.

서론

1. **오픈 도메인 질문 응답(OpenQA)**: OpenQA는 주어진 질문에 대한 답변을 생성하기 위해 언어 모델을 훈련하는 작업입니다. 이 시스템은 질문만을 입력으로 받고, 답변이 포함된 문서 없이 작동합니다. 이는 전통적인 질문 응답 시스템과의 주요 차별점입니다.
2. **외부 지식 소스의 활용**: OpenQA의 유망한 접근 방식 중 하나는 언어 모델을 Wikipedia와 같은 외부 지식 소스(증거 문서)로 보강하는 것입니다. 이 방법은 모델이 더 많은 정보를 활용하여 질문에 대한 답변을 생성할 수 있도록 합니다.
3. **모델 구성 요소**: 이 모델은 두 가지 핵심 구성 요소로 이루어져 있습니다:
    - **정보 검색 시스템(리트리버)**: 지식 소스에서 유용한 텍스트 조각을 식별하는 역할을 합니다.
    - **답변 생성 시스템(리더)**: 검색된 문서와 질문을 바탕으로 답변을 생성합니다.

4. **잠재 변수 모델**: 이 모델은 잠재 변수 모델로 볼 수 있으며, 여기서 잠재 변수는 질문에 대한 답변을 생성하는 데 사용되는 검색된 문서를 나타냅니다. 즉, 모델이 어떤 문서를 검색할지를 결정하는 과정이 잠재 변수로 표현됩니다.
5. **엔드 투 엔드 훈련의 도전**: 이 모델의 엔드 투 엔드(즉, 통합된) 훈련은 도전적입니다. 왜냐하면 검색된 문서를 기반으로 답변을 생성하는 방법과 어떤 문서를 검색할지를 동시에 학습해야 하기 때문입니다.
6. **이전 연구의 접근 방식**: 이전 연구에서는 두 가지 잠재적 해결책을 고려했습니다. 첫 번째는 단계별 훈련(stage-wise training)으로, 이 방법에서는 리트리버를 훈련할 때 리더를 고정하고, 반대로 리더를 훈련할 때 리트리버를 고정하는 방식입니다. 이는 두 구성 요소를 독립적으로 훈련하는 접근 방식입니다.

			Reader and Retriever Training			
Model	*Multi Doc Reader*	*Retriever Adaptation*	*Disjoint*	*End to End*	*Multi Step*	*Unsupervised Retriever*
REALM (Guu *et al.*, 2020)		✓		✓		✓
DPR (Karpukhin *et al.*, 2020)			✓			
RAG (Lewis *et al.*, 2020b)		✓		✓		
FiD (Izacard and Grave, 2021b)	✓		✓			
FiD-KD (Izacard and Grave, 2021a)	✓	✓			✓	
EMDR^2 (Our Approach)	✓	✓		✓		✓

Table 1: Bird's-eye view of the recent OpenQA approaches. **Multi-Doc reader** indicates whether the reader architecture uses multiple documents or a single document. **Retriever adaptation** shows whether the retriever gets feedback from the reader to update its parameters. **Disjoint** denotes that first the retriever is trained and then the reader is trained. **End-to-end** denotes that the reader and retriever are trained jointly in one cycle. **Multi-step** indicates that the reader and retriever are trained iteratively in multiple cycles. **Unsupervised retriever** indicates whether the retriever is initialized using unsupervised approaches or using supervised data.

TTable 1에서 여러 OpenQA 시스템의 특징을 통해 다양한 모델의 주요 특성을 비교했습니다. 각 항목은 OpenQA 모델의 성능과 기능에 중요한 역할을 합니다:

1. **Multi-Doc Reader (다중 문서 리더)**: 이 항목은 모델의 리더 구성 요소가 여러 문서를 동시에 처리할 수 있는지, 아니면 단일 문서에 제한되는지를 나타냅니다. 여러 문서를 활용하는 모델은 다양한 출처의 정보를 통합하여 더 포괄적인 답변을 제공할 수 있습니다.
2. **Retriever Adaptation (리트리버 적응)**: 이 항목은 리트리버(관련 문서를 검색하는 구성 요소)가 리더로부터 피드백을 받아 성능을 개선하는지를 보여줍니다. 피드백은 리더의 필요에 따라 리트리버의 매개변수를 조정하는 데 도움이 될 수 있습니다.
3. **Disjoint Training (분리 훈련)**: 이 접근 방식은 리트리버와 리더를 별도로 훈련하는 것을 의미합니다. 먼저 리트리버를 훈련한 후, 리더를 독립적으로 훈련하는 방식으로, 두 구성 요소 간의 최적 상호작용을 허용하지 않을 수 있습니다.
4. **End-to-End Training (종단 간 훈련)**: 이 방법에서는 리더와 리트리버가 단일 사이클에서 함께 훈련됩니다. 이러한 공동 훈련은 두 구성 요소 간의 시너지를 향상시켜 더 나은 성능을 이끌어낼 수 있습니다.
5. **Multi-Step Training (다단계 훈련)**: 이 항목은 리더와 리트리버가 여러 반복 사이클에서 훈련된다는 것을 나타내며, 상호작용을 기반으로 점진적인 개선과 정제를 가능하게 합니다.
6. **Unsupervised Retriever (비지도 리트리버)**: 이 항목은 리트리버가 비지도 방법(라벨이 없는 데이터 사용)으로 초기화되는지, 아니면 지도 방법(라벨이 있는 데이터 사용)으로 초기화

되는지를 명시합니다. 비지도 초기화는 모델을 더 유연하게 만들고 라벨 데이터에 대한 의존성을 줄일 수 있습니다.

저자들은 EMDR² 모델을 Natural Questions, TriviaQA, WebQuestions와 같은 대표적인 OpenQA 데이터셋에서 실험하여 최첨단 성능을 입증했습니다. 특히 EMDR²는 리트리버의 초기 설정 방식에 있어 강한 성능을 보여주며, 비지도 방식으로 초기화해도 높은 성과를 내는 모습을 통해 지도 학습이 필수라는 기존의 관념에 도전했습니다.

==이 논문의 핵심 기여 요약:

- EMDR² 모델을 통해 검색 보강 질문 응답 시스템의 종단 간 학습 방법을 제시했습니다.
- 잠재 변수에 대한 별도의 감독 없이도 기존 방법보다 우수한 성능을 실현했습니다.
- 구성 요소별 기여도를 분석하는 절단 연구(ablation studies)를 제공했습니다.
- 재현성과 후속 연구 지원을 위해 코드와 체크포인트를 공개했습니다.

또한 저자들은 EMDR²에서 개발된 기술이 잠재 변수 모델을 포함한 다양한 분야에도 적용 가능하다고 믿으며, 이를 통해 여러 자연어 처리 작업에서 유용한 범용 프레임워크로 활용될 수 있을 것으로 기대하고 있습니다.

2. model

제안된 모델 EMDR²는 두 가지 주요 구성 요소로 이루어져 있습니다:
신경 검색기(neural retriever)와 신경 리더더(neural reader).

이 두 구성 요소는 함께 훈련되어 end-to-end 방식으로 최적화됩니다.
이러한 접근 방식은 두 구성 요소를 동시에 효율적으로 최적화할 수 있게 해줍니다.

2.1 신경 검색기: 이중 인코더

신경 검색기는 주어진 질문에 답하기 위해 큰 문서 집합에서 관련 있는 일부 문서를 선택하는 데 사용됩니다. 이 과정에서는 **이중 인코더 네트워크**라는 구조가 활용됩니다.

이중 인코더는 질문과 문서를 각각 인코딩하는 두 개의 인코더로 구성됩니다. 하나의 인코더(`fq`)는 질문을 벡터 형태로 변환하고, 다른 인코더(`fd`)는 각 문서를 벡터로 변환하여 질문과의 연관성을 계산합니다. 이때, 각 문서에 대한 검색 점수는 인코딩된 질문 벡터와 문서 벡터 간의 내적(dot product)으로 산출됩니다.

점수가 높은 상위 `K`개의 문서가 선택되어 추가적으로 처리되며, 이를 `Z = {z1, ..., zK}` 라고 표현합니다.

주요 구성 요소

- **문서 모음**: 답변을 찾기 위한 문서 집합을 `D = {d1, ..., dM }`라 정의하며, 이 중 특정 질문에 관련 있는 문서를 선택하여 질문에 대한 답을 생성합니다.

- **질문과 문서 인코딩**: 이중 인코더(`fq`, `fd`)가 질문 `q`와 각 문서 `di`를 벡터 형태로 변환합니다. 이 벡터들의 내적(dot product)이 해당 문서의 질문 관련성 점수로 사용되며, 식으로는 다음과 같이 표현됩니다:

$$\text{score}(\boldsymbol{q}, \boldsymbol{d}_i; \Phi) = f_q(\boldsymbol{q}; \Phi_q)^\top f_d(\boldsymbol{d}_i; \Phi_d),$$

여기서 `Φ = [Φq, Φd]`는 리트리버의 학습 매개변수를 나타냅니다.

- **관련 문서 선택**: 주어진 질문 `q`에 대해, 가장 높은 점수를 받은 상위 `K`개의 문서를 선택하여 최종적으로 `Z = {z1, ..., zK}`로 표시합니다.
- **인코더 구성**: 질문과 문서를 인코딩하기 위해 BERT와 유사한 트랜스포머 구조가 사용됩니다. 이 모델은 12개 층과 768차원 히든 레이어를 가지며, 질문 및 문서 벡터의 최종 표현으로 BERT의 [CLS] 토큰을 사용해 대표 벡터를 생성합니다.
- **초기화 문제**: `fq`와 `fd`를 BERT의 사전 학습된 가중치로 초기화할 경우, 검색 정확도가 낮아진다는 문제가 제기되었습니다(Lee et al., 2019; Sachan et al., 2021). 이를 보완하기 위해 리트리버는 비지도 학습 기법으로 초기화되며, 자세한 방법은 **§3.2**에서 설명됩니다.

2.2 신경 리더: Fusion-in-Decoder

신경 리더는 질문 `q`와 검색된 문서 집합 `Z`를 입력받아 답변을 생성하는 역할을 합니다. 이 리더는 T5 모델 위에 구축된 **Fusion-in-Decoder(FiD)** 구조를 기반으로 하고 있습니다.

- **FiD 구조와 T5 모델**: FiD는 T5 모델을 바탕으로 동작합니다. T5는 시퀀스-투-시퀀스 형태의 트랜스포머 모델로, 질문과 문서를 인코딩하는 인코더 `ge`와 답변을 생성하는 디코더 `gd`로 구성되어 있습니다.
- **입력 구성**: 각 검색된 문서 `zk`는 해당 문서의 제목 `tzk`와 질문을 앞에 붙여서 다음과 같은 형태의 입력 `xk`로 변환됩니다: $\boldsymbol{x}_k = \texttt{[CLS]}\,\boldsymbol{q}\,\texttt{[SEP]}\,\boldsymbol{t}_{\boldsymbol{z}_k}\,\texttt{[SEP]}\,\boldsymbol{z}_k\,\texttt{[SEP]}$, 여기서 `[CLS]`는 문서의 시작을 표시하고, `[SEP]`는 질문, 제목, 문서를 구분하는 역할을 합니다.
- **인코더 처리**: 이렇게 구성된 `xk`는 T5 인코더 `ge`에 독립적으로 입력됩니다. 모든 검색된 문서의 인코딩 결과가 함께 연결(concatenation)되어 다음과 같은 형태의 최종 벡터 `XZ`가 만들어집니다: $X_Z = [g_e(x_1); \ldots; g_e(x_K)] \in R^{(N \times K) \times H}$ 여기서 `N`은 각 문서의 토큰 수, `H`는 T5 인코더의 히든 크기입니다. 이 작업에서는 T5-base 설정을 사용하며, `N = 512`와 `H = 768`입니다.
- **디코더 처리 및 답변 생성**: `XZ`는 T5 디코더 `gd`에 입력됩니다. 디코더는 답변 토큰을 생성할 때, 이전에 생성된 토큰에 주의(causal attention)를 주고 `XZ`에 포함된 여러 문서의 정보를 교차 주의(cross-attention)를 통해 함께 참조합니다. 이를 통해 디코더는 여러 문서의 유용한 정보를 통합하여 종합적인 답변을 생성할 수 있습니다.

- **답변 확률 정의**: 생성되는 답변의 확률은 다음과 같이 정의됩니다:

$$p(a|q,Z;\Theta)=\prod_{t=1}^{T}p(a_t|a_{<t},q,Z;\Theta)$$

여기서 `θ`는 리더의 매개변수(즉, T5 인코더와 디코더)를 나타내며, `T`는 답변 토큰 수입니다. 디코더는 특별한 종료 토큰(EOS)을 출력하거나 지정된 최대 답변 길이에 도달할 때까지 답변 토큰을 생성합니다.

2.3 리더와 리트리버의 종단 간 훈련 (End-to-End Training)

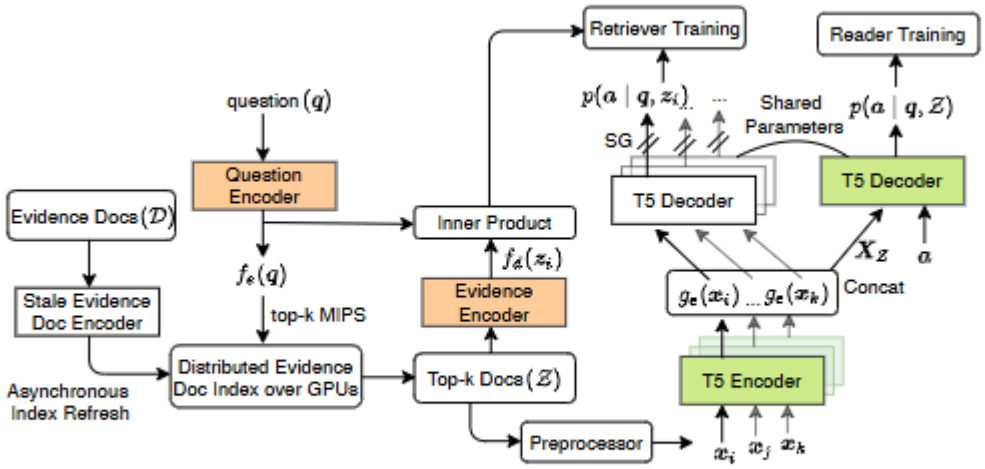

Figure 1: An illustration of the different components of EMDR^2. Colored blocks indicate components which contain trainable parameters.

첨부된 그림 1에서는 EMDR² 모델의 구성 요소가 어떻게 서로 연결되어 있는지 보여줍니다.

- **질문 인코더(Question Encoder)**와 **증거 문서 인코더(Evidence Encoder)**는 각각 질문과 문서를 벡터로 변환하여 검색 점수를 계산합니다.
- **top-K MIPS**는 가장 관련성 높은 상위 `K`개의 문서를 선택합니다.
- **T5 인코더**는 선택된 문서들과 질문을 입력받아 각 문서를 인코딩한 결과를 생성합니다.
- **T5 디코더**는 모든 문서의 정보를 결합해 최종 답변을 생성합니다.

이 과정에서 리더와 리트리버는 함께 학습되며, 각 구성 요소는 **공유 매개변수(shared parameters)**를 사용하여 상호 최적화됩니다.

이 연구에서는 질문 응답 시스템에서 **리더**와 **리트리버**를 동시에 훈련하여 최적의 답변을 생성합니다. 이는 이전의 연구와는 달리, 리더와 리트리버가 상호작용하며 **종단 간(differentiable end-to-end)** 방식으로 최적화되는 접근법입니다.

핵심 개념 및 목표

1. **잠재 변수 Z**: 검색된 문서의 집합을 나타내는 잠재 변수를 `z`로 정의합니다. 가능한 검색 문서 집합의 값 중 하나를 `z`라고 할 때, 모든 가능한 `z` 값에 대해 답변의 **마진 가능도**

(marginal likelihood)를 계산합니다:

$$p(a|q;\Theta,\Phi) = \sum_{Z=Z} p(a|q,Z;\Theta)p(Z|q;\Phi)$$

여기서, `θ`는 리더의 매개변수, `ϕ`는 리트리버의 매개변수입니다. 이 목적 함수를 최대화하는 것이 목표지만, 모든 `z`의 조합을 고려하는 것은 계산적으로 매우 어렵습니다.

2. **로그 가능도 계산**: 단일 문서 집합 `z`에 대해서는 로그 가능도를 단순하게 계산할 수 있습니다:

$$\log p(a|q,Z;\Theta)p(Z|q;\Phi) = \log p(a|q,Z;\Theta) + \log p(Z|q;\Phi)$$

3. **기대-최대화(EM) 알고리즘**: EM 알고리즘을 사용하여 이 잠재 변수 모델을 학습합니다. EM 알고리즘은 잠재 변수 `z`의 사후 확률을 반복적으로 계산하고, 이를 바탕으로 `θ`와 `ϕ`를 업데이트하는 방식으로 학습을 진행합니다.

두 가지 Z 추정치: Zreader와 Zretriever

이 연구에서는 모델의 두 가지 구성 요소인 리더와 리트리버를 업데이트하기 위해 `z`에 대한 두 가지 추정치를 사용합니다:

- **Zreader**: 리더 파라미터 `θ`를 업데이트할 때 사용하는 추정치입니다. `Zreader`는 질문에 대해 리더가 필요한 정보를 갖춘 문서 집합으로 설정됩니다.
- **Zretriever**: 리트리버 파라미터 `ϕ`를 업데이트할 때 사용하는 추정치입니다. 이 경우, 리트리버가 더 관련성이 높은 문서를 찾도록 **근사 사후 확률(approximate posterior)**을 최대화하는 방향으로 설정됩니다.

Zreader와 Zretriever를 이용한 두 가지 추정치로의 분리

리더 매개변수 Θ 업데이트 (첫 번째 항)

리더를 업데이트할 때, `Zreader` 값을 사용하여 Θ를 업데이트합니다. `Zreader`는 상위 `K`개의 문서로 구성되며, 이 문서는 현재 리트리버 매개변수 `ϕ`로 계산된 개별 점수가 가장 높은 문서들입니다. 이러한 설정은 **사전 분포(prior)** `p(Z | q; ϕ)`에 의존하여 `Zreader`를 추정하는 것과 같으며, 여기서 답변 `a`에 대한 정보를 사용하지 않습니다. 훈련 시에 사전 분포를 사용하는 이유는 평가 시에도 사전 점수를 이용하여 상위 `K` 문서를 선택하기 때문입니다. 이를 통해 훈련과 테스트 단계에서 일관된 결과를 보장할 수 있습니다.

리트리버 매개변수 Φ 업데이트 (두 번째 항)

리트리버의 매개변수 `ϕ`를 업데이트할 때는 **사후 분포(posterior)** 추정치를 사용합니다. 즉, `a`로부터 추가적인 정보를 받아 `Zretriever`를 평가하여 Φ를 훈련시키게 됩니다. 사후 분포를 사용

함으로써 리트리버는 사전 분포에만 의존하는 것보다 더 풍부한 훈련 신호로부터 학습할 수 있습니다.

여기서 `p(Zretriever | q, a; θ, ϕ)`를 계산해야 하지만, 이는 `Z`라는 집합의 확률을 계산하는 것이므로 복잡합니다. 이 문제를 간소화하기 위해, `K`개의 문서로 이루어진 집합 `Ztop-K`에서 각 문서 `zk`의 확률을 더한 값이 최대화되도록 근사합니다. 이를 통해 다음과 같은 단순한 형태로 표현됩니다:

$$\sum_{k=1}^{K} p(zk|q,a;\Theta,\Phi)$$

베이즈 정리를 적용하여 이 확률을 다시 표현하면:

$$p(zk|q,a;\Theta,\Phi) \propto p(a|q,zk;\Theta)p(zk|q;\Phi)$$

여기서 리더는 답변의 확률 `p(a | q, zk; θ)`를 계산할 때 오직 한 문서 `zk`에만 조건화됩니다. 이 단순화된 리더는 더 복잡한 리더와 동일한 매개변수 `θ`를 사용하지만, 다수의 문서가 아닌 단일 문서를 기반으로 확률을 계산합니다.

식 계산 과정

1. 현재 리트리버 매개변수 `ϕ`에 따라 Eq. (1)을 사용하여 상위 `K`개의 문서 `zk`를 선택합니다.
2. 문서 `zk ∈ Ztop-K`의 확률은 다음과 같이 계산됩니다:

$$p(zk|q,Ztop-K;\Phi) \approx \frac{\exp(\text{score}(q,zk)/\tau;\Phi)}{\sum_{j=1}^{K}\exp(\text{score}(q,zj)/\tau;\Phi)}$$

여기서 `τ`는 온도 하이퍼파라미터입니다. 이 근사는 상위 `K` 이외의 문서들은 점수가 매우 낮다고 가정하여 계산을 단순화합니다.

EMDR²의 전체 학습 목표

위에서 설명한 결과를 합치면, 특정 예제에 대한 종단 간 학습 목표는 다음과 같이 됩니다:

$$L = \log p(a|q,Ztop-K;\Theta) + \log \sum_{k=1}^{K} \text{SG}(p(a|q,zk;\Theta))p(zk|q,Ztop-K;\Phi)$$

여기서 `SG`는 **stop-gradient 연산자**로, 리더의 매개변수 `θ`가 단일 문서 `zk`에 대해서도 잘 수행되도록 업데이트되지 않도록 합니다.

stop-gradient 연산자는 여러 가지 이점을 제공합니다:

1. FiD 리더는 모든 검색된 문서에 대해 조건화된 상태에서 훈련되며, 이는 테스트 시 리더 사용 방식과 일관성을 유지합니다.
2. stop-gradient 연산자는 역전파 과정이 필요하지 않아 훈련 속도를 높이고, GPU 메모리 사용량도 줄입니다.

EMDR²의 훈련 알고리즘

훈련 예제에 대해 `θ`와 `ϕ`를 업데이트하기 위해, Eq. (6)의 그래디언트를 사용해 `θ`와 `ϕ`에 대한 최적화를 수행합니다. 이를 통해 리더는 상위 `K` 문서인 `Ztop-K`을 기반으로 정확한 답변을 생성하도록 학습하며, 리트리버는 리더의 피드백을 받아 답변 생성에 높은 점수를 제공하는 `K`개의 문서를 선택하도록 학습됩니다.

알고리즘 1: 다중 문서 리더 및 리트리버의 종단 간 학습 절차

- **입력**: 모델 매개변수 `θ`와 `ϕ`, 근거 문서 `D`.
- **훈련 과정**:
  - 수렴할 때까지 반복하며, 다음 단계를 수행합니다:
    - 현재 리트리버 매개변수 `ϕ`를 사용하여 상위 `K` 문서 `Ztop-K`를 계산합니다. // E-step
    - 현재 리더 매개변수 `θ`를 사용하여 각 `zk`에 대해 `p(a | q, zk)`를 계산합니다. // E-step
    - Eq. (6)의 로그 가능도를 최대화하도록 모델 매개변수 `θ`와 `ϕ`를 업데이트합니다. // M-step

3.1 데이터셋

이 연구에서는 다음 세 가지 대표적인 오픈 도메인 질문 응답 데이터셋을 사용하여 실험을 진행합니다:

1. **Natural Questions (NQ)**: 이 데이터셋은 Google 검색 엔진 사용자들이 실제로 입력한 질문을 기반으로 구성된 질문-답변 쌍을 포함하고 있습니다. 이 연구에서는 `Lee et al. (2019)`의 접근 방식에 따라 짧은 답변이 포함된 하위 집합을 사용합니다.
2. **TriviaQA**: TriviaQA는 웹에서 여러 소스로부터 수집된 퀴즈 형식의 질문-답변 쌍으로 구성된 데이터셋입니다.
3. **WebQuestions (WebQ)**: WebQ 데이터셋의 질문은 Google Suggest API를 사용하여 수집되었으며, Mechanical Turk 작업자들이 답변을 작성했습니다. 이 연구에서는 `Chen et al. (2017)`의 버전을 사용하며, 답변에 포함된 Freebase ID를 실제 엔티티 이름으로 대체한 버전입니다.

증거 문서 D

실험에서 증거 문서로 사용된 데이터는 `Karpukhin et al. (2020)`이 제공한 2018년 12월에 수집된 **영어 위키피디아 덤프**의 전처리된 버전입니다. 이 데이터는 각 Wikipedia 문서를 100단어 단위의 겹치지 않는 세그먼트로 나누었고, 각 세그먼트를 하나의 문서로 간주하여 사용했습니다. 이 방식으로 총 21,015,324개의 문서가 생성되었습니다.

자세한 통계 정보와 추가 전처리 세부 사항은 부록 A에 제공되었습니다.

> **참고**: 연구에서는 각 문서가 독립적으로 기여하는 것으로 가정하여 계산을 단순화했고, 모든 증거 문서를 고려하지 않음으로써 확률 분포의 엄밀성을 유지하지 않았지만, 실험에서 이 방법이 잘 작동함을 확인했습니다.

3.2 구현 세부 사항

하드웨어 및 라이브러리

모든 실험은 96개의 CPU, 1.3TB의 물리적 메모리, 16개의 A100 GPU가 탑재된 머신에서 수행되었습니다. 모델 구현에는 **PyTorch** 라이브러리를 사용했습니다.

모델 구성

리트리버와 리더 모두 12개의 레이어, 768차원의 히든 사이즈, 12개의 어텐션 헤드를 갖춘 **베이스 구성**을 사용했습니다. 모든 실험에서는 별도의 언급이 없는 한 50개의 문서를 검색했으며, GPU 메모리 제약으로 인해 베이스 구성만 사용했지만, 이 구성으로 얻은 결과는 더 큰 모델에서도 일반화될 것으로 예상됩니다.

검색

빠른 검색을 지원하기 위해 모든 증거 문서의 임베딩을 사전 계산하여 **분산 저장**했습니다. 이 임베딩들은 **문서 인덱스**라고 하며, 각 질문에 대해 **온라인** 방식으로 최대 내적 검색(MIPS)을 수행해 실시간으로 관련 문서를 검색합니다. 이는 분산 행렬 곱셈을 통해 구현되었습니다.

검색된 문서는 BERT의 토크나이저를 통해 서브워드로 변환되고, T5 리더에 입력됩니다. 만약 토크나이즈된 문서가 512 토큰보다 짧으면, 인접 문서의 토큰을 추가해 최대 토큰 길이에 맞춰 패딩합니다. 이러한 패딩은 답변 생성 시 더 넓은 문맥을 제공하는 데 도움을 줍니다.

초기화 및 훈련 세부 사항

모델 매개변수는 **비지도 사전 훈련**으로 초기화한 후, 지도 학습으로 훈련됩니다. 비지도 사전 훈련은 리트리버를 "웜 스타트(warm-start)"하는 데 중요한 역할을 하며, 질문에 적절한 문서를 출력하도록 합니다.

1. **비지도 역클로즈 작업(Inverse Cloze Task)**: 리트리버 매개변수를 100,000 스텝 동안 비지도 역클로즈 작업으로 사전 훈련합니다.

2. **마스크된 중요한 스팬(Masked Salient Spans, MSS)**: 증거 문서에서 이름 있는 엔티티가 포함된 문장을 추출한 뒤, 엔티티의 15%를 마스킹하여 모델 훈련에 사용합니다. 마스크된 문장은 질문으로 간주하고, 마스크된 엔티티는 정답으로 간주하여 학습을 진행합니다. 이 작업은 82,000 스텝 동안 배치 크기 64로 수행되며, **Adam 옵티마이저**를 사용합니다.
3. **초기화 및 파인튜닝**: MSS 훈련 이후, EMDR²를 사용해 데이터셋 별 질문-답변 훈련 예제로 파인튜닝합니다. NQ와 TriviaQA는 배치 크기 64로 10 에포크 동안, WebQ는 배치 크기 16으로 20 에포크 동안 훈련합니다. 훈련 중에는 매 500 스텝마다 체크포인트를 저장하고, 개발 셋 성능을 기반으로 최상의 체크포인트를 선택합니다.

문서 임베딩의 업데이트

종단 간 훈련 중에는 문서 인코더(`fd`)의 매개변수도 매 스텝마다 업데이트되므로, 사전 계산된 문서 임베딩이 시간이 지나면서 구형(stale)이 될 수 있습니다. 이를 방지하기 위해 가장 최근의 문서 인코더 체크포인트를 사용해 새로운 문서 임베딩을 비동기적으로 계산하고, 500 스텝마다 문서 인덱스를 업데이트하여 최신 상태를 유지합니다.

Model	top-K	NQ		TriviaQA		WebQ		*# of*
		dev	test	dev	test	dev	test	*params*
Closed-Book QA Models								
T5-*base* (Roberts *et al.*, 2020)	0	-	25.7	-	24.2	-	28.2	220M
T5-*large* (Roberts *et al.*, 2020)	0	-	27.3	-	28.5	-	29.5	770M
T5-*XXL* (Roberts *et al.*, 2020)	0	-	32.8	-	42.9	-	35.6	11B
GPT-3 (Brown *et al.*, 2020)	0	-	29.9	-	-	-	41.5	175B
Open-Book QA Models								
BM25 + BERT (Lee *et al.*, 2019)	5	24.8	26.5	47.2	47.1	27.1	21.3	220M
ORQA (Lee *et al.*, 2019)	5	31.3	33.3	45.1	45.0	36.8	30.1	330M
REALM (Guu *et al.*, 2020)	5	38.2	40.4	-	-	-	40.7	330M
DPR (Karpukhin *et al.*, 2020)	25	-	41.5	-	56.8	-	34.6	330M
RECONSIDER (Iyer *et al.*, 2021)†	30	-	43.1	-	59.3	-	44.4	440M
RAG-Sequence (Lewis *et al.*, 2020b)†	50	44.0	44.5	55.8	56.8	44.9	45.2	626M
Individual Top-K (Sachan *et al.*, 2021)	-	-	45.9	-	56.3	-	-	440M
Joint Top-K (Sachan *et al.*, 2021)	50	-	49.2	-	64.8	-	-	440M
FiD (Izacard and Grave, 2021b)	100	-	48.2	-	65.0	-	-	440M
FiD-KD (Izacard and Grave, 2021a)	100	48.0	49.6	68.6	68.8	-	-	440M
Our Implementation (Base Configuration)								
FiD / T5-*base*	0	26.0	25.1	26.7	27.8	31.0	32.4	220M
FiD (DPR retriever, T5 reader)	1	37.3	38.4	50.8	50.4	40.2	38.3	440M
FiD (DPR retriever, T5 reader)	50	47.3	48.3	65.5	66.3	46.0	45.2	440M
FiD (MSS + DPR retriever, T5 reader)	50	48.8	50.4	68.0	68.8	43.5	46.8	440M
FiD (MSS retriever, MSS reader)	50	38.5	40.1	60.0	59.8	39.1	40.2	440M
EMDR^2 (MSS retriever, MSS reader)	50	**50.4**	**52.5**	**71.1**	**71.4**	**49.9**	**48.7**	440M

Table 2: Exact match scores on three evaluation datasets. Top-K denotes the number of retrieved documents that are used by the reader to produce an answer. To provide a fair comparison with our reimplementations, we show results from other papers with the base configuration, except for RAG-Sequence that uses BART-*large* (Lewis *et al.*, 2020a). † indicates that their results on WebQ use NQ training data to pretrain the model.

Inference. We use greedy decoding for answer generation at inference time.

이 표는 여러 모델을 사용하여 세 가지 주요 오픈 도메인 질문 응답 데이터셋(Natural Questions, TriviaQA, WebQuestions)에서의 **정확 일치 점수(Exact Match Score)**를 비교한 것입니다.
표는 모델의 성능과 사용된 구성, 그리고 각 데이터셋에서의 결과를 나누어 보여줍니다.

주요 설명

- **Model**: 사용된 모델의 이름과 참고 문헌을 나타냅니다. 크게 두 가지 유형으로 나뉩니다:
  - **Closed-Book QA Models**: 질문에 대한 답을 외부 문서 없이 생성하는 모델.
  - **Open-Book QA Models**: 외부 문서에서 검색한 정보를 바탕으로 답을 생성하는 모델.
- **Top-K**: 리더가 답변을 생성할 때 사용한 문서의 개수를 의미합니다. `K`가 클수록 더 많은 문서를 참조하여 답변을 생성합니다.
- **NQ, TriviaQA, WebQ**: 각 데이터셋에서의 정확 일치 점수(EM)를 dev와 test 세트로 구분하여 보여줍니다. 높은 점수가 더 높은 정확도를 의미합니다.
- **# of params**: 각 모델의 파라미터 수를 나타냅니다. 파라미터가 많을수록 더 복잡하고, 일반적으로 성능이 높은 모델일 가능성이 큽니다.

Closed-Book QA Models

- **T5-base**에서 **GPT-3**까지 다양한 크기의 사전 훈련된 모델들이 사용되었으며, 이 모델들은 외부 문서 없이 질문에 답변을 생성합니다.
- 파라미터 수는 T5-base의 220M에서 GPT-3의 175B까지 다양합니다.
- 이 모델들은 open-book 모델에 비해 정확도는 낮지만, 외부 문서 의존성이 없어 빠르게 답변을 생성할 수 있다는 장점이 있습니다.

Open-Book QA Models

- **BM25 + BERT**에서 **FiD-KD**까지 여러 개의 open-book 모델들이 있으며, 이들은 검색된 문서를 바탕으로 답변을 생성합니다.
- **FiD-KD**와 같은 최신 모델은 50개의 문서를 참조하여 높은 정확도를 기록하고 있습니다. 특히, FiD-KD는 NQ, TriviaQA, WebQ에서 뛰어난 성능을 보입니다.
- 모델마다 파라미터 수가 다양하며, RAG-Sequence의 경우 BART-large를 사용했기 때문에 파라미터 수가 더 큽니다(426M).

Our Implementation (Base Configuration)

이 섹션은 본 연구에서 구현한 모델의 결과를 보여줍니다. **FiD** 기반으로 다양한 리트리버와 리더 설정을 조합하여 성능을 평가했습니다.

- **FiD / T5-base**: T5-base를 사용한 기본 구현으로, 비교적 낮은 정확도를 보입니다.
- **DPR 리트리버 + T5 리더** 및 **MSS 리트리버 + T5 리더**: 각 리트리버와 리더 조합의 성능을 보여주며, 리트리버와 리더를 MSS 기반으로 모두 설정한 **EMDR²**가 가장 높은 정확도를 보

입니다. 특히 EMDR²는 NQ, TriviaQA에서 각각 50.4와 71.1의 높은 점수를 기록했습니다.

Inference

추론 단계에서 **Greedy Decoding**을 사용하여 답변을 생성했습니다.

3.3 기준선 (Baselines)

본 연구에서 제안한 모델을 다른 OpenQA(오픈 도메인 질문 응답) 접근 방식들과 비교하며, 이러한 접근 방식은 다음 두 가지 범주로 나뉩니다:

1. **Closed-book QA 모델**:
   - 이 모델들은 대규모 언어 모델을 사용하여, 학습에 사용된 코퍼스에 내장된 세계 지식을 바탕으로 질문에 답변을 생성합니다. Petroni et al. (2019)의 연구에 따르면, 이런 언어 모델들은 상당한 양의 세계 지식을 내장하고 있습니다.
   - Roberts et al. (2020)의 연구를 기준으로, 더 큰 T5 모델이 질문-답변 쌍으로 미세 조정(finetuning)될 경우 뛰어난 성능을 보일 수 있음을 확인했습니다.
   - 또한, GPT-3 (Brown et al., 2020)의 few-shot 결과와도 비교했습니다.
2. **Open-book QA 모델**:
   - 본 연구와 유사하게, 이러한 모델들은 리트리버와 리더 컴포넌트로 구성되어 있으며, 검색 후 예측(retrieve then predict) 접근 방식을 채택하여 증거 문서 모음을 기반으로 질문에 답변을 생성합니다.
   - 모델들은 주로 리트리버 초기화 방법(예: ORQA; Lee et al., 2019, DPR; Karpukhin et al., 2020), 리더가 단일 문서를 처리하는지(예: ORQA, DPR, RAG; Lewis et al., 2020b) 또는 다중 문서를 처리하는지(예: FiD; Izacard and Grave, 2021b), 그리고 리더와 리트리버가 공동으로 또는 다단계로 학습되는지(예: REALM; Guu et al., 2020, FiD-KD; Izacard and Grave, 2021a)에 따라 차별화됩니다.

3.4 결과 (Results)

본 연구는 각 데이터셋에 포함된 정답을 참조하여 **정확 일치(EM) 점수**를 기준으로 성능을 평가하였으며, 표 2에 주요 결과가 나와 있습니다.

- **구성**: 표는 세 가지 섹션으로 나누어져 있습니다. 첫 번째 섹션은 다른 논문에서 가져온 closed-book QA 모델의 결과, 두 번째 섹션은 open-book QA 모델의 결과, 세 번째 섹션은 본 연구에서 제안한 모델과 관련 기준선의 재구현 결과입니다.
- **재구현 성능**: T5-base를 재구현하여 검색 문서 개수를 0과 1로 설정해 강력한 기준선을 만들었습니다. 표 2에 따르면, 검색 문서가 없는 경우보다 상위 1개의 문서를 참조하는 설정이 성능을 크게 향상시킴을 알 수 있으며, 이는 OpenQA 과제에서 검색의 중요성을 의미합니다. 상위 k 문서 개수를 50으로 늘리면 FiD 모델의 성능이 크게 향상되어, 검색 문서를 집합

으로 모델링하는 것이 중요함을 확인할 수 있습니다(Izacard and Grave, 2021b의 관찰을 검증).

- **EMDR²와 FiD 비교**: EMDR²와 FiD의 재구현을 비교한 결과, 종단 간 훈련 방식인 EMDR²가 더 우수한 성능을 보입니다. FiD는 리트리버를 먼저 학습하고 리더를 학습하는 2단계 접근 방식을 채택합니다. FiD의 세 가지 변형을 사용했습니다:
  1. 리더와 리트리버를 MSS(Masked Salient Spans) 방식으로 초기화한 경우,
  2. DPR 훈련으로 리트리버를 초기화한 경우 (원 논문에서 사용한 설정),
  3. MSS + DPR 훈련으로 리트리버를 초기화한 경우(Sachan et al., 2021).

  EMDR²는 모든 데이터셋에서 이러한 변형들을 크게 능가했습니다.
- **FiD-KD와의 비교**: FiD-KD(Izacard and Grave, 2021a)는 다단계 훈련과 지식 증류를 활용하는 복잡한 접근 방식입니다. EMDR²는 NQ와 TriviaQA에서 FiD-KD를 각각 2.5점 이상 앞서며, 새로운 최첨단 성능을 기록했습니다. 또한 EMDR²는 다음과 같은 세 가지 장점을 갖고 있습니다:
  1. FiD-KD는 100개의 문서를 사용하는 반면, EMDR²는 50개의 문서만 사용하여 더 효율적입니다.
  2. FiD-KD는 리트리버와 리더의 다중 훈련 사이클이 필요한 반면, EMDR²는 종단 간 훈련 사이클 한 번으로 완료됩니다.
  3. FiD-KD는 리트리버의 지도 초기화가 필요하지만, EMDR²는 비지도 초기화에서도 우수한 성능을 발휘하며, 초기화에 더 강인한 특성을 가지고 있습니다.
- **WebQ 데이터셋 성능**: WebQ 데이터셋은 다른 데이터셋에 비해 학습 데이터 크기가 훨씬 작습니다. 이전 접근 방식(RAG)은 감독된 전이 학습을 통해 좋은 성능을 얻었지만, EMDR²는 이러한 감독 전이 없이도 RAG보다 3.5점 더 높은 성능을 보이며, 소규모 데이터셋에서도 적용 가능성을 보여주었습니다.

연구에서는 모델 출력에 대한 **정성적 분석**도 수행했으며, 이는 부록 E에 포함되어 있습니다.

3.5 절단 연구 (Ablations)

이 연구에서는 EMDR²와 FiD 모델의 성능을 분석하기 위해 다음 두 가지 변수를 변경하여 실험했습니다: **검색된 문서 수(K)**와 **리트리버 초기화 방법**.

검색된 문서 수의 영향

- **Figure 2**에서, 검색된 문서 수 K를 증가시키면 EMDR²와 FiD 모두 성능이 향상됨을 볼 수 있습니다.
- K가 작을 때 EMDR²가 FiD보다 성능 격차가 큰데, 이는 EMDR²가 메모리 제한 등의 이유로 검색할 수 있는 문서 수가 적을 때도 더 효과적으로 작동함을 의미합니다.
- K가 증가하면 FiD와 EMDR² 모두 성능이 점진적으로 개선되며, 상위 K 문서에서 다양한 정보를 통합할 수 있는 모델의 중요성이 드러납니다.

리트리버 초기화 방법의 영향

Table 3에서는 EMDR² 모델 훈련 시 다양한 리트리버 초기화 전략이 성능에 미치는 영향을 보여줍니다. 사용된 초기화 전략은 다음과 같습니다:

1. **비지도 MSS 사전 훈련**: 마스크된 중요한 스팬(MSS) 방식을 통해 사전 훈련된 리트리버.
2. **지도 리트리버 훈련 (DPR)**: DPR 방식으로 사전 훈련된 리트리버.
3. **MSS + DPR 사전 훈련**: MSS로 초기화한 후 DPR로 추가 훈련한 리트리버 (Sachan et al., 2021).

각 초기화 방식의 성능 분석

- **비지도 MSS 사전 훈련**: NQ 데이터셋에서, MSS 사전 훈련을 통해 초기 `R@50`이 66.4로 낮게 시작되지만, EMDR² 훈련 후 `R@50`이 86.3으로 20%가량 개선되었습니다. 이는 노란색 셀로 강조되어 있습니다.
- **DPR 훈련**: DPR로 리트리버를 초기화한 경우, 초기 `R@50`이 더 높게 시작되지만, 최종 `R@50`은 MSS와 동일한 수준에 도달합니다. 이로 인해 OpenQA 과제에서 높은 성능을 위해 꼭 DPR 초기화가 필요한 것은 아님을 시사합니다.
- **MSS + DPR**: MSS와 DPR을 조합한 초기화 방법은 초기 `R@50`이 가장 높았지만, 최종 답변 성능에서 MSS 단독 초기화보다 특별한 개선을 보이지 않았습니다.

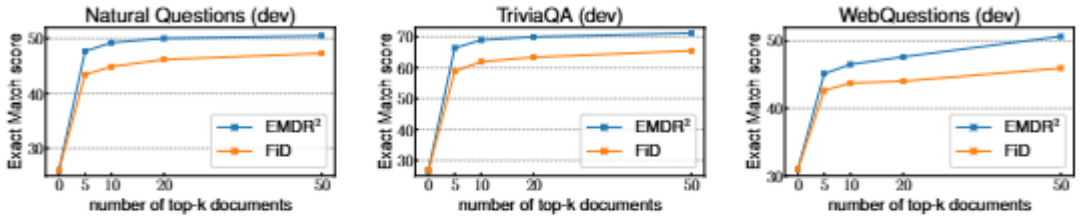

Figure 2: Performance on NQ, TriviaQA, and WebQ as we vary the number of retrieved documents.

Retriever Initialization	Reader Initialization	NQ (dev) R@50 B.T.	NQ (dev) R@50 A.T.	NQ (dev) EM	TriviaQA (dev) R@50 B.T.	TriviaQA (dev) R@50 A.T.	TriviaQA (dev) EM	WebQ (dev) R@50 B.T.	WebQ (dev) R@50 A.T.	WebQ (dev) EM
MSS pre-training	MSS pre-training	66.4	86.3	50.4	74.8	86.2	71.1	59.8	88.6	49.9
MSS pre-training	T5	66.4	86.3	50.3	74.8	86.3	70.9	59.8	88.6	47.7
DPR training	T5	82.3	86.3	50.0	83.2	86.2	70.5	84.2	88.6	49.0
MSS + DPR	MSS pre-training	84.5	86.3	50.5	85.3	86.3	71.2	85.0	88.6	49.9

Table 3: R@50 denotes the retrieval recall from the top-50 retrieved documents. B.T. and A.T. indicates R@50 score Before Training and After Training the model, respectively.

또한, WebQ 데이터셋에서 **MSS 초기화**가 T5 리더를 사용할 때보다 답변 추출 성능을 2포인트 개선했음을 확인했습니다(주황색 셀). 이는 **소규모 데이터셋**이거나 **리소스가 적은 환경에서 OpenQA 작업**에 중요하다는 것을 강조합니다.

3.6 대체 종단 간 학습 목표 (Alternative End-to-End Training Objectives)

본 연구에서는 EMDR²의 학습 목표(식 6)를 두 가지 대체 학습 목표와 비교하여 성능을 평가했습니다. **Table 4**는 개발 셋에서의 EM(Exact Match) 점수를 보여주며, 이를 통해 각 학습 목표가 NQ, TriviaQA, WebQ 데이터셋에서 미치는 영향을 확인할 수 있습니다.

대체 학습 목표 1 (`Lalt-1`)

첫 번째 대체 학습 목표에서는 리트리버 매개변수 `ϕ`를 훈련할 때 `p(z | q; ϕ)`를 문서별 확률의 곱으로 단순화합니다:

$$L_{\text{alt-1}} = \log p(a|q, Z; \Theta) + \sum_{k=1}^{K} \log p(z_k|q, Z; \Phi)$$

이 식에서 두 번째 항은 리트리버가 문서 간 차별을 하지 않도록 하여, 모든 문서를 동일하게 선택하는 경향이 생깁니다. 즉, **균일한 검색(uniform retrieval)**을 목표로 하는 것이며, 이는 **대립적 목표(adversarial objective)**의 영향을 보여주기 위해 포함되었습니다.

- **결과**: `Lalt-1`은 예상대로 성능이 크게 떨어지며, NQ, TriviaQA, WebQ 데이터셋에서 모두 낮은 점수를 기록했습니다. 이는 훈련 과정에서 리트리버가 차별성을 잃게 되면, 질문 응답 시스템의 성능이 크게 저하됨을 의미합니다.

대체 학습 목표 2 (`Lalt-2`)

두 번째 대체 학습 목표에서는 각 검색된 문서의 사후 확률을 **균일한 사전 확률**을 가정하여 근사합니다:

$$\tilde{p}(z_k|q, a, Z_{\text{top-K}}; \Theta) \propto p(a|q, z_k; \Theta) \times \frac{1}{K}$$

이 근사치를 사용하여 리더와 리트리버 매개변수를 다음과 같이 훈련합니다:

$$L_{\text{alt-2}} = \log p(a|q, Z; \Theta) + \text{KL}(\text{SG}(\tilde{p}(z_k|q, a, Z_{\text{top-K}}; \Theta)) || p(z_k|q, Z; \Phi))$$

이 방법은 리트리버가 문서를 검색할 확률과, 해당 문서가 답변 생성에 기여하는 정도를 일치시키는 것을 목표로 합니다.

- **결과**: `Lalt-2`는 NQ와 TriviaQA에서 FiD 기준선보다 성능이 향상되었지만, 여전히 EMDR² 보다는 낮은 성능을 보였습니다. WebQ 데이터셋에서는 `Lalt-2`가 수렴하지 않아 성능이 저하되었습니다.

Method	top-k	NQ	TriviaQA	WebQ
FiD	50	47.3	65.5	46.0
EMDR^2	50	**50.4**	**71.1**	**49.9**
$\mathcal{L}_{\text{alt-1}}$	50	14.1	11.9	28.0
$\mathcal{L}_{\text{alt-2}}$	50	49.9	69.6	28.8

Table 4: EM scores on the development set for alternative training objectives.

종합 평가

- **EMDR²**: NQ, TriviaQA, WebQ 데이터셋에서 가장 높은 성능을 기록하며, 안정적인 수렴과 높은 성능을 보여줍니다.
- `Lalt-1`: 리트리버의 선택성을 저해하는 대립적 목표로 인해 성능이 크게 떨어집니다. 이는 QA 시스템에서 리트리버가 문서 간의 차별적 선택을 유지하는 것이 중요함을 시사합니다.
- `Lalt-2`: FiD 기준선을 초과하는 성능을 보이지만, EMDR²보다 낮은 성능을 나타냈으며, 특히 WebQ에서 수렴 문제가 발생했습니다. 이는 추가적인 수렴 분석이 필요함을 시사하며, 향후 연구 과제로 남겨졌습니다.

4. 관련 연구 (Related Work)

EMDR² 연구는 신경 리더와 리트리버의 **종단 간 학습(end-to-end training)**에 기반을 두고 있으며, 이에 대한 설명은 §1, §2, §3에서 다루고 있습니다. 여기서는 신경 리트리버와 신경 리더에 대한 개별 연구와 이들이 다른 자연어 처리 과제에서 어떻게 활용되는지를 중심으로 관련 연구를 논의합니다.

신경 리트리버 (Neural Retrievers)

신경 리트리버는 문서당 계산되는 임베딩 수에 따라 크게 두 가지로 나눌 수 있습니다:

1. **이중 인코더(Dual Encoders)**: 각 문서당 하나의 임베딩을 저장합니다. 이 접근 방식은 효율적이며, Lee et al. (2019)와 같은 연구에서 사용되었습니다.
2. **다중 벡터 인코더(Multivector Encoders)**: 여러 개의 임베딩을 사용하여 검색 성능을 높이지만, 대규모 검색에 있어 계산 비용이 더 많이 듭니다 (예: Khattab and Zaharia, 2020).

EMDR²는 OpenQA에서 방대한 증거 문서 집합을 다루기 때문에 효율성을 위해 **이중 인코더**를 사용합니다. Sachan et al. (2021)은 OpenQA에서 이중 인코더의 성능이 **고자원 환경에서는 Inverse Cloze Task**, **저자원 환경에서는 마스크된 중요한 스팬(MSS)**으로 사전 훈련할 때 향상될 수 있음을 보여주었습니다.

신경 리더 (Neural Readers)

신경 리더는 검색된 문서를 입력으로 받아 답변을 생성하는 역할을 합니다. 주요 방식은 다음과 같습니다:

1. **추출형 리더(Extractive Readers)**: 검색된 문서에서 특정 스팬을 추출하여 답변을 생성합니다 (예: Clark and Gardner, 2018).
2. **생성형 리더(Generative Readers)**: 검색된 문서를 바탕으로 조건부 생성하여 답변을 만듭니다 (예: Izacard and Grave, 2021b).

EMDR²는 생성형 리더 방식을 사용하여, 검색된 문서에서 적절한 정보를 통합해 답변을 생성합니다.

기타 응용 분야

질문 응답 이외에도, **검색 보강 방식**은 다음과 같은 자연어 처리 과제에서도 활용되고 있습니다:

- **왼쪽-오른쪽 언어 모델링**: 외부 메모리에서 유사한 단어를 검색해 모델의 당혹도를 줄이는 데 사용되었습니다 (예: Khandelwal et al., 2020).
- **기계 번역**: 도메인에 특화된 타겟 언어 토큰을 검색해 성능을 개선하는 데 사용되었습니다 (예: Khandelwal et al., 2021).
- **대화 모델링**: 지식을 기반으로 한 텍스트를 검색하여 생성된 대화의 사실성을 높이는 데 기여했습니다 (예: Fan et al., 2021).

EMDR²와 이전 연구 간의 상세한 비교는 부록 C와 D에 제공됩니다.

5. 토론 (Discussion)

기여 요약

EMDR²는 **검색 보강 질문 응답 시스템**을 위한 종단 간 학습 방법을 제안했습니다. 기대-최대화(EM) 알고리즘을 사용하여 학습 목표를 설정하였고, 세 가지 주요 OpenQA 데이터셋에서 최첨단 성능을 달성했습니다.

기술적 한계

EMDR²는 다른 검색 보강 질문 응답 모델들과 유사한 몇 가지 한계를 가집니다:

- **높은 비용**: 증거 문서가 압축되지 않은 상태로 저장되기 때문에 메모리와 계산 비용이 높습니다.
- **작업 범위 제한**: 본 연구에서는 오픈 도메인 질문 응답에 초점을 맞췄으며, 다른 텍스트 생성 모델에서도 EMDR²의 성능을 검토해볼 필요가 있습니다.
- **훈련 비용**: 16개의 GPU가 필요할 정도로 리소스 소모가 크며, 환경적 영향도 고려해야 합니다.

잠재적 사회적 영향

EMDR²는 저자원 환경에서 언어 모델의 성능을 개선할 가능성이 있지만, 대규모 언어 모델이 가진 **편향성**을 그대로 가질 수 있습니다. 또한, 검색 보강 방식이기 때문에 증거 문서에 대한 접근이 허용된 공격자가 문서를 조작하면 **가짜 답변**을 생성할 가능성도 있습니다.

부록

Dataset	Train	Filtered Train	Dev	Test
WebQuestions (WebQ)	3,417	2,474	361	2,032
Natural Questions (NQ)	79,168	58,880	8,757	3,610
TriviaQA	78,785	60,413	8,837	11,313

Table 5: OpenQA 데이터셋 통계

이 표는 **OpenQA 데이터셋**의 세부 통계를 보여줍니다. 각 데이터셋의 **훈련(Train)**, **필터링된 훈련(Filtered Train)**, **개발(Dev)**, **테스트(Test)** 세트의 데이터 수를 나타내며, EMDR² 모델의 학습 과정에 대한 정보를 제공합니다.

- **WebQuestions (WebQ)**: 총 3,417개의 질문-답변 쌍을 포함한 훈련 세트가 있으며, 필터링 후 2,474개로 줄어듭니다. 개발 셋은 361개, 테스트 셋은 2,032개입니다.
- **Natural Questions (NQ)**: 원래 훈련 세트에는 79,168개의 질문-답변 쌍이 포함되어 있으며, 필터링 후 58,880개로 줄어듭니다. 개발 셋은 8,757개, 테스트 셋은 3,610개입니다.
- **TriviaQA**: 78,785개의 질문-답변 쌍을 포함한 훈련 세트가 있으며, 필터링된 훈련 세트에는 60,413개가 남습니다. 개발 셋은 8,837개, 테스트 셋은 11,313개입니다.

필터링된 훈련 세트 (Filtered Train)

- **필터링 기준**: 필터링된 훈련 세트는 **DPR(Dense Passage Retrieval)** 실험에 사용되며, 질문에 대한 BM25 검색 결과가 제공된 정답 문맥과 일치하지 않는 질문-답변 쌍을 제외한 데이터로 구성됩니다. 즉, 질문에 대한 BM25 검색 결과가 정답이 포함된 문서와 일치하지 않는 경우는 필터링하여, **정확한 맥락을 가진 데이터**만 사용하게 됩니다.
- **출처**: 이 필터링된 훈련 세트는 **Karpukhin et al. (2020)**이 제공한 데이터를 활용했습니다.

이 표의 데이터셋 통계는 질문 응답 모델의 **종단 간 학습**과 **리트리버의 지도 학습**에 사용되며, 필터링된 데이터셋을 사용하여 리트리버가 더 정확한 문서를 선택하도록 돕습니다.

A. 데이터셋 세부 정보

데이터셋 통계

- **검증 데이터 구성**: 훈련 세트에서 약 10%를 무작위로 선택하여 검증용으로 사용했습니다. 모든 데이터셋은 **Lee et al. (2019)**의 데이터셋 분할 방식을 따랐으며, **Table 5**에서 각 데이터셋의 훈련, 개발, 테스트 셋 크기를 확인할 수 있습니다.

전처리 (Pre-processing)

- **TriviaQA**: QA 모델 훈련을 위해 인간이 직접 주석을 단 답변을 선택하고, 답변 길이가 5 단어를 초과하는 질문은 제외했습니다. 이 과정을 통해 훈련 세트에서 2,362개의 예시가 필터링되었습니다.

데이터셋 라이선스와 URL

- 모든 데이터셋은 오픈 소스로 커뮤니티에서 널리 사용됩니다.
  - **NQ**: 데이터셋 URL - Natural Questions, 라이선스 - LICENSE
  - **TriviaQA**: 데이터셋 URL - TriviaQA, 라이선스 - LICENSE
  - **WebQ**: 데이터셋 URL - WebQ, 라이선스 - LICENSE
  - **전처리된 버전**: Karpukhin et al. (2020)이 오픈 소스로 제공한 NQ, TriviaQA, 그리고 증거 데이터셋을 사용했습니다 - Download data

Hyperparameter	BERT	ICT	T5	MSS
Dataset	Wikipedia, BookCorpus	Wikipedia	C4, Wikipedia, OpenWebText	Wikipedia
Num. Parameters	110M	220M	220M	440M
Hidden Size	768	768	768	768
Attention heads	12	12	12	12
Dropout	0.1	0.1	0.1	0.1
Optimizer	Adam	Adam	Adam	Adam
Batch Size	256	4096	2048	64
Training Steps	1M	100K	1M	82K
Warmup Ratio	0.01	0.01	0.01	0.05
Max. Learning Rate	1e-4	1e-4	1e-4	2e-5
Weight Decay	1e-2	1e-2	1e-2	1e-1
Learning Rate Decay	Linear	Linear	Linear	Linear
Gradient Clipping	1.0	1.0	1.0	1.0

Table 6: Hyperparameters for training BERT, ICT, T5, and MSS models.

Hyperparameter	NQ	TriviaQA	WebQ
Num. Parameters	440M	440M	440M
Hidden Size	768	768	768
Attention heads	12	12	12
Dropout	0.1	0.1	0.1
Optimizer	Adam	Adam	Adam
Batch Size	64	64	16
Epochs	10	10	20
Warmup Ratio	0.01	0.01	0.01
Max. Learning Rate	2e-5	2e-5	2e-5
Weight Decay	1e-1	1e-1	1e-1
Learning Rate Decay	Linear	Linear	Linear
Gradient Clipping	1.0	1.0	1.0
Temperate (τ)	27.7	27.7	27.7

Table 7: Hyperparameters for finetuning on NQ, TriviaQA, and WebQ datasets.

B. 추가 훈련 세부 정보

BERT와 Inverse Cloze Task (ICT)

- **BERT 및 ICT 구현**: BERT (Devlin et al., 2019)와 ICT (Lee et al., 2019)의 구현은 오픈 소스 **Megatron-LM** 도구에서 가져왔습니다.
- **훈련 설정**: BERT 가중치로 이중 인코더 리트리버를 초기화하고, Wikipedia 단락을 사용해 최대 256 토큰 길이로 잘라 훈련을 진행했습니다. BERT와 ICT 훈련에 사용된 하이퍼파라미터는 **Table 6**에 나와 있습니다.

T5

- **T5 구현**: T5 언어 모델(Raffel et al., 2020)의 구현도 Megatron-LM 도구에서 가져왔으며, 원 논문과 동일한 스텝 수와 배치 크기로 훈련했습니다. BERT 소문자화 토크나이저를 T5와 BERT 모두에 사용했습니다.

비지도 MSS 사전 훈련

- **MSS 훈련 초기화**: 리트리버는 ICT 가중치로, 리더는 T5 가중치로 초기화합니다. Stanza 도구(Qi et al., 2020)를 사용해 증거 문서를 문장 단위로 분할하고, OntoNotes-5.0 데이터셋으로 훈련된 NER 모델을 통해 이름 엔티티를 마스크 토큰으로 대체했습니다.
- **MSS 학습 과정**: 마스킹된 문장은 질문으로 간주하여 관련 증거 문서를 검색하고, 리더는 검색된 문서를 바탕으로 마스킹된 이름 엔티티를 복원하도록 학습됩니다. 이 과정의 하이퍼파라미터는 **Table 6**에 나와 있습니다.

질문-답변 쌍을 통한 지도 학습

- 지도 학습 세부 사항은 **§3.2**에 제공되어 있으며, 사용된 하이퍼파라미터는 **Table 7**에 나와 있습니다. 온도 파라미터(τ)는 숨겨진 크기의 제곱근으로 설정했습니다.

훈련 시간

- **하드웨어 구성**: 96개의 CPU, 1.3TB 메모리, 16개의 A100 GPU가 장착된 머신에서 모든 실험을 진행했습니다.
- **훈련 소요 시간**: NQ와 TriviaQA 실험에는 약 25시간, WebQ 실험에는 약 8시간이 걸렸습니다. 지도 학습 전에 한 번의 비지도 MSS 사전 훈련을 82,000 스텝 동안 수행했으며, 약 1주일이 소요되었습니다.

Method	R@5 after ICT	R@5 after MSS
REALM (Guu *et al.*, 2020)	13.9	38.5
EMDR^2	28.0	38.6

Table 8: Retrieval recall on the NQ development set after ICT and MSS pre-training.

Method	Evidence Size	Evidence Dimension	GPU Memory (in FP16)
REALM (Guu *et al.*, 2020)	13M	128	3 GB
EMDR^2	21M	768	30 GB

Table 9: Comparison of evidence embeddings storage for retrieval.

C. 비지도 사전 훈련과 REALM과의 비교

EMDR²에서는 REALM(Guu et al., 2020)에서 소개된 몇 가지 훈련 기술을 사용했습니다. 특히 **마스크된 중요한 스팬(Masked Salient Spans, MSS) 사전 훈련**과 **비동기 증거 임베딩 업데이트** 기법이 이에 해당하며, 이 기술들을 EMDR² 훈련 방식에 맞게 적용했습니다.

C.1 ICT와 MSS 사전 훈련

- **Inverse Cloze Task (ICT)**와 **MSS**는 모두 비지도 기법으로, 리트리버의 초기 검색 성능을 높이는 데 사용됩니다.

- **ICT 사전 훈련**: 리트리버를 ICT로 초기화했으며, REALM과 유사하게 ORQA(Lee et al., 2019)의 설정을 따랐습니다. EMDR²의 Recall@5 성능이 REALM 논문에서 보고된 것보다 높았는데, 이는 REALM이 128차원의 임베딩을 사용하는 반면, EMDR²는 768차원 임베딩을 사용하여 더 나은 성능을 얻었기 때문으로 분석됩니다.
- **MSS 사전 훈련**: ICT로 초기화한 리트리버를 바탕으로 MSS 사전 훈련을 추가로 수행했습니다. EMDR²는 배치 크기 64로 82,000 스텝 동안 훈련했으며, REALM은 배치 크기 512로 200,000 스텝 동안 훈련했습니다. 적은 배치 크기와 훈련 스텝에도 불구하고 EMDR²는 REALM과 비슷한 Recall@5 성능을 달성했습니다.
- **추가 구현 세부 사항**: EMDR²는 REALM에서 사용된 **null 문서**를 필요로 하지 않는 점도 차별화됩니다.
- **저자원 데이터셋(WebQ)**: WebQ와 같은 저자원 데이터셋에서는 MSS 사전 훈련이 FiD 리더의 성능을 높여줍니다. 예를 들어, WebQ에서 MSS 사전 훈련된 리더는 T5 리더보다 1 EM 포인트 이상 성능이 개선되었습니다(Table 3 참조).

C.2 비동기 증거 임베딩 업데이트

- **훈련 중 비동기 임베딩 업데이트**는 500 스텝마다 수행되며, REALM과 유사하지만 몇 가지 차이점이 있습니다.
  - EMDR²는 **MSS 사전 훈련**과 **지도 학습** 모두에서 비동기 임베딩 업데이트를 수행하는 반면, REALM은 MSS 사전 훈련에서만 수행합니다.
  - EMDR²는 2,100만 개의 증거 문서를 다루고, REALM은 1,300만 개의 문서를 다뤄야 하므로, 계산량이 더 많습니다.
  - 훈련 시 8개의 GPU가 모델 훈련에, 다른 8개의 GPU가 비동기 방식으로 증거 임베딩 계산에 사용되었습니다.

C.3 검색을 위한 사전 계산된 증거 임베딩 저장

- **REALM과 EMDR²의 비교**: Table 9에 REALM과 EMDR²의 증거 임베딩 저장 방식의 차이가 나와 있습니다.
  - REALM은 각 Wikipedia 문서를 288개의 워드피스로 나누어 1,300만 개의 증거를 생성하는 반면, EMDR²는 100개 단어 단위로 나누어 2,100만 개의 증거를 생성합니다.
  - 임베딩 차원도 REALM은 128차원, EMDR²는 768차원으로, EMDR²가 훨씬 고차원의 임베딩을 사용합니다.
  - 이로 인해 REALM의 증거 임베딩을 FP16으로 저장할 때 약 3GB 메모리가 필요한 반면, EMDR²는 약 30GB 메모리가 필요합니다.
  - **GPU 메모리 제약**: A100 GPU의 최대 메모리가 40GB이므로 EMDR²는 단일 GPU에 30GB의 임베딩을 모두 저장할 수 없습니다. 이를 해결하기 위해 16개의 GPU에 걸쳐 임베딩을 분산 저장하고, **분산 비동기 MIPS**를 사용해 빠르게 검색합니다.

D. 이전 연구와의 비교 (Comparison with Previous Work)

여기서는 EMDR²가 이전의 몇 가지 연구와 어떻게 다른지에 대해 설명합니다. 이전 연구와의 차이점은 **Hard EM 및 강화 학습 기반 모델**, 그리고 **개별 Top-K 및 Joint Top-K 모델**을 중심으로 논의됩니다.

D.1 Hard EM 및 강화 학습 기반 리더-랭커 모델과의 비교

- **Hard EM (Min et al., 2019)**과 **R3 (Reinforced Reader-Ranker; Wang et al., 2018)**는 EMDR² 및 대체 학습 목표 `Lalt-2`와 개념적으로 유사하지만, 구조적으로 동일하지는 않습니다.
- EMDR²는 결정론적 접근 방식을 사용하여 상위 K개의 문서를 선택하며, Hard EM 및 R3의 강화 학습(REINFORCE) 방식과는 다릅니다.

Hard EM과의 차이점:

- Min et al. (2019)은 **추출형 리더** 모델을 위한 Hard EM 접근 방식을 제안했습니다. 이 모델에서는 정답이 여러 번 등장하는 문맥을 기반으로 리더를 훈련합니다.
- Hard EM 접근 방식은 TF-IDF와 BM25를 리트리버로 사용하며, 이는 훈련 가능한 모델이 아닙니다. 반면, EMDR²는 **종단 간 훈련이 가능한 밀집 리트리버(dense retriever)**를 사용합니다.
- Hard EM은 리트리버를 훈련하지 않으며, EMDR²처럼 리더와 리트리버를 모두 훈련하는 방식과는 다릅니다. 따라서 Hard EM 방식은 EMDR²에 직접 적용할 수 없습니다.

R3와의 차이점:

- R3 모델은 리트리버, 랭커, 리더의 세 가지 단계로 구성되어 있으며, 리트리버는 BM25 기반으로 훈련이 불가능한 방식입니다.
- R3에서 랭커는 리트리버로부터 100개의 문서를 받아 하나의 문서를 선택해 리더에 전달합니다. 반면, EMDR²는 다중 문서를 선택해 리더에 전달하므로, R3의 접근 방식과는 차별화됩니다.
- R3는 강화 학습을 사용하여 비분화 가능한 선택 작업을 훈련하지만, EMDR²는 **종단 간 미분 가능**한 학습 목표를 사용합니다.

D.2 개별 Top-K 및 Joint Top-K 모델과의 비교

Individual Top-K와의 비교 (Sachan et al., 2021):

- Individual Top-K는 종단 간 학습 접근법 중 하나지만, 단일 문서 리더를 사용하는 반면 EMDR²는 **다중 문서 리더**로 구성됩니다.
- Individual Top-K는 여러 문서를 사용해 성능을 최적화하는 목적 함수가 있으며, EMDR²는 이보다 더 높은 성능을 보여줍니다.

Joint Top-K와의 비교 (Sachan et al., 2021):

- Joint Top-K와 EMDR²는 FiD 모델을 기반으로 한 종단 간 학습 접근 방식이지만, 다음과 같은 주요 차이점이 있습니다:
    1. **목적 함수의 차이**: Joint Top-K는 상위 K 문서의 검색 확률 점수를 인터어텐션 점수에 더해 리더가 상위 K 문서를 더 중요하게 여기도록 유도합니다. 하지만 리트리버에 대한 피드백은 없습니다. 반면, EMDR²는 학습 목표의 두 번째 항을 통해 리트리버가 리더의 답변 생성 확률에 동의하도록 개선하도록 유도합니다.
    2. **성능 차이**: EMDR²는 FiD 기준선보다 훨씬 더 높은 성능 향상을 제공합니다. NQ와 TriviaQA에서 EMDR²는 각각 4.3점과 6.4점의 EM 점수 향상을 보였지만, Joint Top-K는 NQ에서 1점 향상에 그쳤고 TriviaQA에서는 개선이 없었습니다.

이러한 비교 결과는 EMDR²가 다중 문서 리더-리트리버 접근 방식에서 Joint Top-K보다 훨씬 더 나은 종단 간 학습 알고리즘임을 입증합니다.

E. 정성적 분석 (Qualitative Analysis)

Table 10에서는 MSS 사전 훈련 모델과 MSS 사전 훈련 후 NQ에서 EMDR²로 미세 조정된 모델의 리트리버 출력 예시를 제시합니다. 이 분석을 통해 리트리버가 어떤 방식으로 질문에 관련된 문서를 선택하는지와 초기화 방식에 따른 성능 차이를 살펴봅니다.

MSS 사전 훈련과 미세 조정의 효과

- **MSS 사전 훈련**만 수행된 리트리버는 질문과 관련된 문서를 상위 결과로 출력하지만, 정답을 제공하기에는 충분히 관련성이 높은 문서가 아닙니다.
- **MSS 사전 훈련 후 NQ에서 EMDR²로 미세 조정**된 모델은 상위 1위 문서가 질문에 더욱 직접적으로 답변할 수 있을 정도로 높은 관련성을 보이며, 리트리버의 정확도가 크게 향상됩니다.
- 미세 조정 후 상위 1위 문서의 **신뢰도 점수**도 향상되었음을 관찰할 수 있습니다.

리트리버 초기화 비교

- **MSS 초기화**와 **MSS + DPR 초기화**를 비교하여 리더의 학습 손실을 분석했습니다. Table 3에 따르면, MSS 초기화(비지도 학습)는 정확도가 낮은 반면, **MSS + DPR 초기화**는 더 높은 정확도를 보입니다.
- **Figure 4**에서, 초기 학습 단계에서는 MSS + DPR 초기화된 리트리버가 MSS 초기화된 리트리버보다 **작은 학습 손실**을 보이지만, 약 1200 스텝 이후부터는 두 초기화 방식 간의 학습 손실 차이가 줄어듭니다. 이는 **종단 간 학습 과정이 진행되면서 MSS 리트리버의 정확도가 개선**되었기 때문입니다.
- TriviaQA와 WebQ에서도 비슷한 경향이 있지만, 그 효과는 NQ보다 적게 나타났습니다.

리더와 리트리버 손실 시각화

- **Figure 3**에서는 MSS 사전 훈련으로 초기화한 모델에서 리더와 리트리버의 학습 손실 경로를 보여줍니다. 이 그래프는 학습 과정 중 리더와 리트리버의 성능이 어떻게 변화하는지 시각적으로 확인할 수 있도록 도와줍니다.

Questions from NQ test	Answer	MSS Pre-training	EMDR^2 finetuned on NQ
what type of reaction occurs to form a dipeptide	peptide bond	probability=0.39 ...Bornyl diphosphate synthase In enzymology, bornyl diphosphate synthase (BPPS) () is an enzyme that catalyzes the chemical reaction Bornyl diphosphate synthase is involved in the biosynthesis of the cyclic monoterpenoid bornyl diphosphate. As seen from the reaction above, BPPS takes geranyl diphosphate as its only substrate and isomerizes into the product, (+)- bornyl diphosphate. This reaction comes from a general class of enzymes called terpene synthases that ...	probability=0.78 ...Subsequent to this coupling reaction, the amine protecting group P and the ester are converted to the free amine and carboxylic acid, respectively. For many amino acids, the ancillary functional groups are protected. The condensation of the amine and the carboxylic acid to form the peptide bond generally employs coupling agents to activate the carboxylic acid. The Bergmann azlactone peptide synthesis is a classic organic synthesis for the preparation of dipeptides. ...
when was the japanese videogame company nintendo founded	23 September 1889	probability=0.37 ...contributed to the development of the following games. Creatures (company) Ape, Inc. was founded in March 1989 and Shigesato Itoi became its chief executive officer. Nintendo president Hiroshi Yamauchi had wanted to support new talent in game design. Liking Itoiś work, he proposed the idea of the company to Itoi and invested in it. Apeś staff included Tsunekazu Ishihara, who later became the Pokémon Companyś CEO, and Ashura Benimaru Itoh, a renowned illustrator. They began work on "Mother", which released in July. Its music was composed by Hip Tanaka, who later became the second CEO of Creatures ...	probability=0.61 ...Nintendo Co., Ltd. is a Japanese multinational consumer electronics and video game company headquartered in Kyoto. Nintendo is one of the world's largest video game companies by market capitalisation, creating some of the best-known and top-selling video game franchises, such as "Mario", "The Legend of Zelda", and "Pokémon". Founded on 23 September 1889 by Fusajiro Yamauchi, it originally produced handmade hanafuda playing cards. By 1963, the company had tried several small niche businesses, such as cab services and love hotels. Abandoning previous ventures in favour of toys in the 1960s ...

Table 10: Examples of top-1 retrieved documents from the NQ test when the model is pre-trained with Masked Salient Spans (MSS) or finetuned on NQ data. If the answer exists in the document it is highlighted in blue color, and the probability of the document (Eq. 5) is indicated in orange color.

Table 10: NQ 테스트 질문에 대한 리트리버 출력 예시

Table 10은 MSS 사전 훈련과 NQ에서 EMDR²로 미세 조정한 모델에서 각각 질문에 대해 리트리버가 선택한 상위 문서(top-1)를 예시로 보여줍니다.

- **첫 번째 질문 ("What type of reaction occurs to form a dipeptide?")**에 대해:
  - **MSS 사전 훈련 모델**에서는 질문과 관련된 내용이 나오지만, 정답과는 직접적인 관련이 없는 문서가 선택되었습니다 (확률: 0.39).
  - **EMDR²로 미세 조정된 모델**에서는 질문에 대해 더 적합한 문서가 선택되며, 펩타이드 결합 형성에 필요한 정보를 포함하고 있습니다 (확률: 0.78).
- **두 번째 질문 ("When was the Japanese video game company Nintendo founded?")**에 대해:
  - **MSS 사전 훈련 모델**에서는 닌텐도와 관련된 정보를 포함하나, 정확한 창립 연도에 대한 언급이 부족한 문서가 선택되었습니다 (확률: 0.37).
  - **EMDR²로 미세 조정된 모델**에서는 닌텐도가 1889년에 설립된 정확한 정보를 포함하는 문서가 선택되었습니다 (확률: 0.61).

이를 통해 미세 조정된 EMDR² 모델이 질문에 대해 더욱 관련성이 높은 문서를 선택할 수 있음을 알 수 있으며, 리트리버의 정확도와 신뢰도 점수가 모두 향상되었습니다.

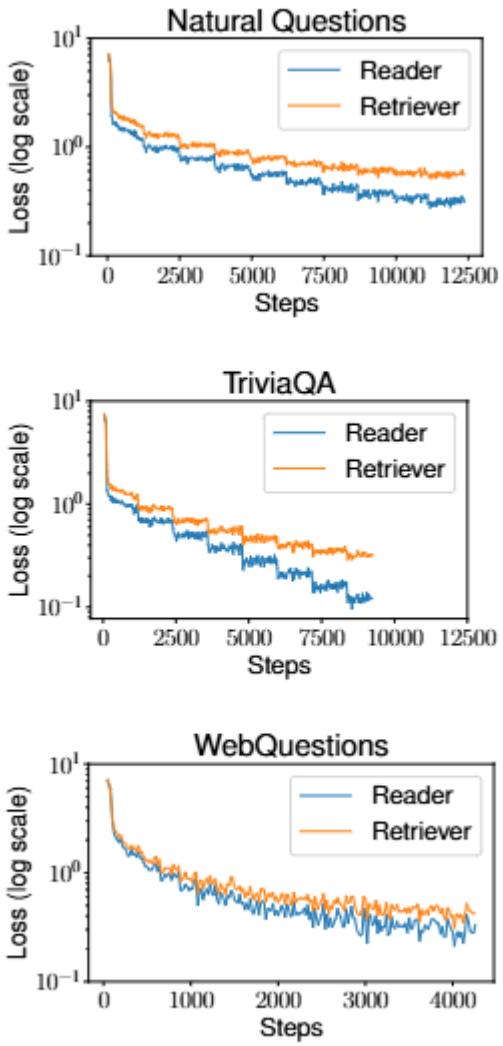

Figure 3: Reader and retriever training losses when the model is initialized with MSS pre training.

Figure 3: 리더와 리트리버 학습 손실의 변화

Figure 3은 MSS로 초기화된 모델에서 **리더와 리트리버의 학습 손실 경로**를 보여줍니다.

- **Natural Questions (NQ)**, **TriviaQA**, **WebQ** 데이터셋에서 리더와 리트리버의 손실이 점진적으로 감소하며, 리더와 리트리버가 함께 최적화되는 모습을 보여줍니다.
- 리트리버는 초기에 손실이 크지만 시간이 지남에 따라 리더의 손실과 비슷한 수준으로 낮아지며, 이는 **리더와 리트리버가 함께 종단 간 학습을 통해 성능이 개선됨**을 나타냅니다.

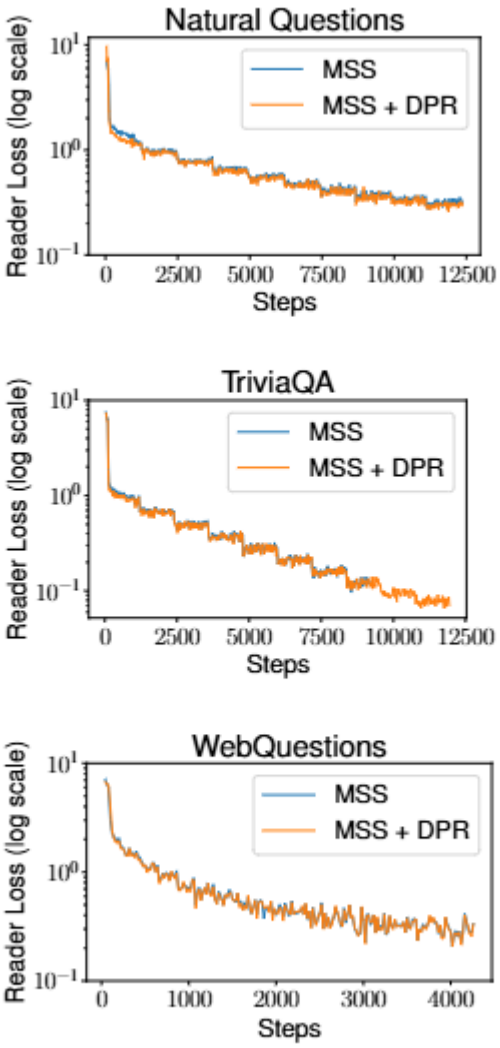

Figure 4: Reader training loss vs steps for NQ, TriviaQA, and WebQ when the retriever is either initialized by MSS pre-training or by MSS followed by supervised DPR training (MSS + DPR).

Figure 4: 리더 학습 손실과 리트리버 초기화 방식의 비교

이 그래프는 **Natural Questions (NQ)**, **TriviaQA**, **WebQuestions (WebQ)** 데이터셋에서 **MSS 초기화**와 **MSS + DPR 초기화**로 학습을 시작했을 때의 **리더 학습 손실 변화**를 보여줍니다.

- **MSS**와 **MSS + DPR** 초기화의 학습 손실 경향을 비교하면, 초기 단계에서 **MSS + DPR** 초기화가 **더 낮은 손실**을 보여줍니다. 이는 supervised DPR 학습이 리트리버의 초기 검색 성능을 개선하는 효과가 있음을 나타냅니다.
- 하지만 학습이 진행될수록 두 초기화 방식의 손실 차이는 점차 감소하며, **종단 간 학습을 통해 MSS 리트리버의 성능이 개선**됩니다. 이러한 경향은 NQ에서 가장 뚜렷하게 나타나며, TriviaQA와 WebQ에서는 그 정도가 덜하지만 유사한 경향이 확인됩니다.