Re^2G: Retrieve, Rerank, Generate

NAACL 2022

# of citations 89

Catholic University of Korea / Biomedical Signal Processing Lab
박수형 pshpulip22@catholic.ac.kr

Contents

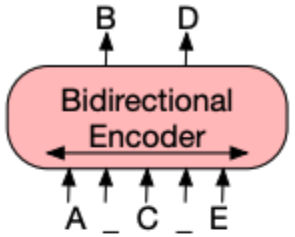
Background

Background

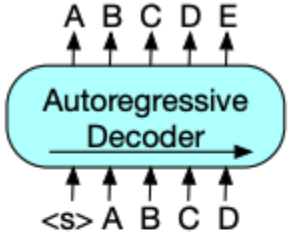
BART

- BART는 문장을 손상 시키고 이를 복원하는 방식으로 학습하여 다양한 텍스트 생성 및 요약 작업에 효과적

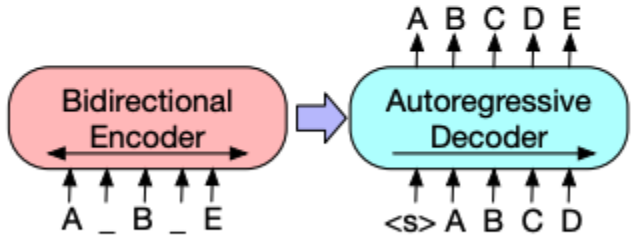
BART = BERT + GPT



(a) BERT: Random tokens are replaced with masks, and the document is encoded bidirectionally. Missing tokens are predicted independently, so BERT cannot easily be used for generation.



(b) GPT: Tokens are predicted auto-regressively, meaning GPT can be used for generation. However words can only condition on leftward context, so it cannot learn bidirectional interactions.

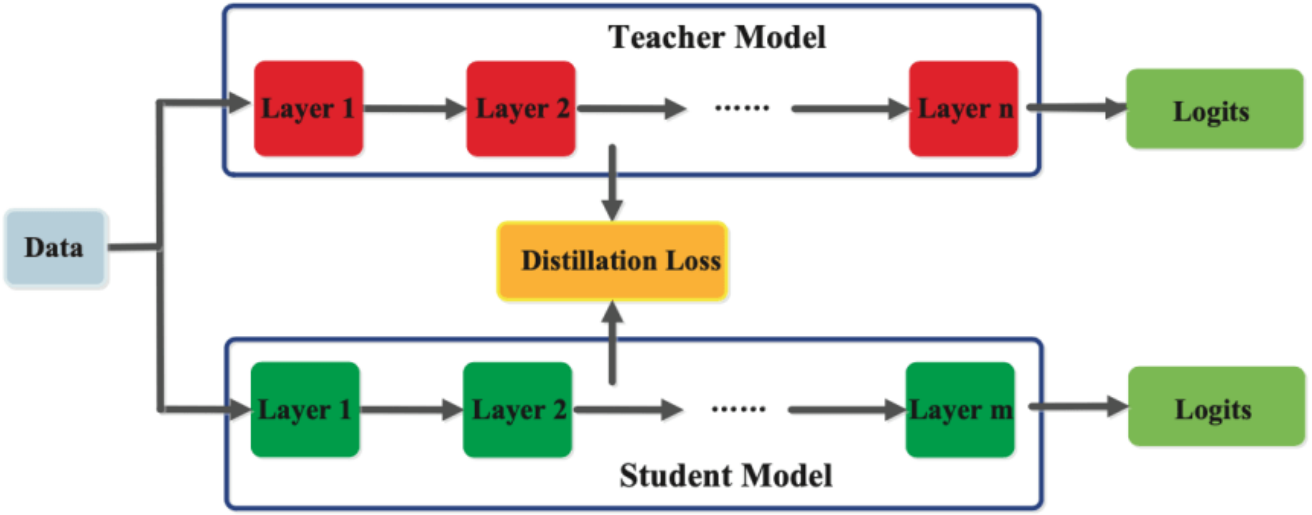


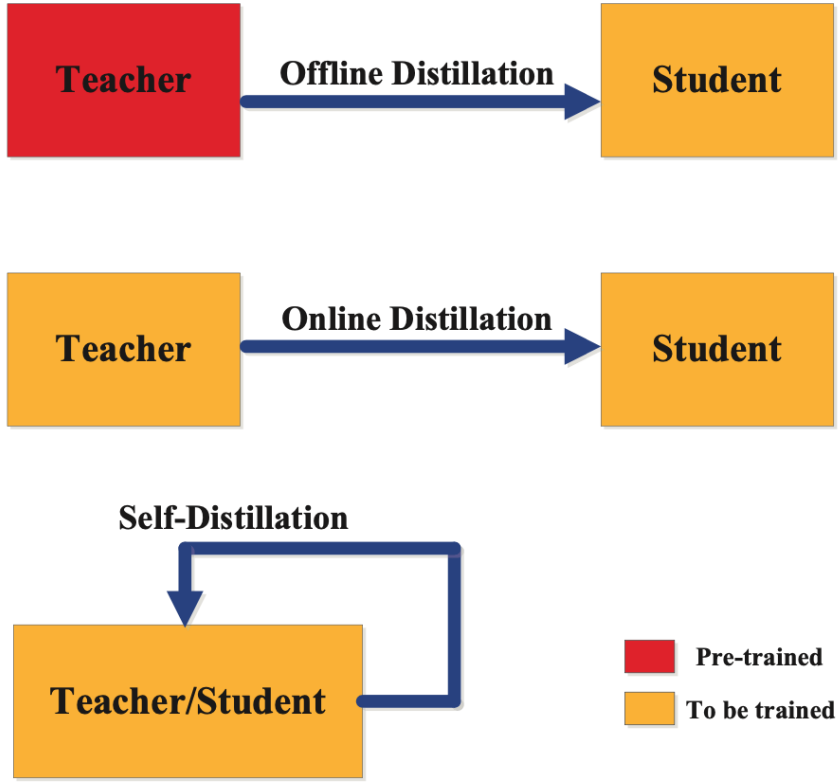
(c) BART: Inputs to the encoder need not be aligned with decoder outputs, allowing arbitary noise transformations. Here, a document has been corrupted by replacing spans of text with mask symbols. The corrupted document (left) is encoded with a bidirectional model, and then the likelihood of the original document (right) is calculated with an autoregressive decoder. For fine-tuning, an uncorrupted document is input to both the encoder and decoder, and we use representations from the final hidden state of the decoder.

Background

Knowledge Distillation

- 성능이 높은 복잡한 모델의 출력을 정답처럼 삼아, 간소화된 모델이 이를 학습하도록 하는 과정으로, 작은 모델이 더 큰 모델의 지식을 증류 받아 유사한 성능을 내도록 돕는 기법

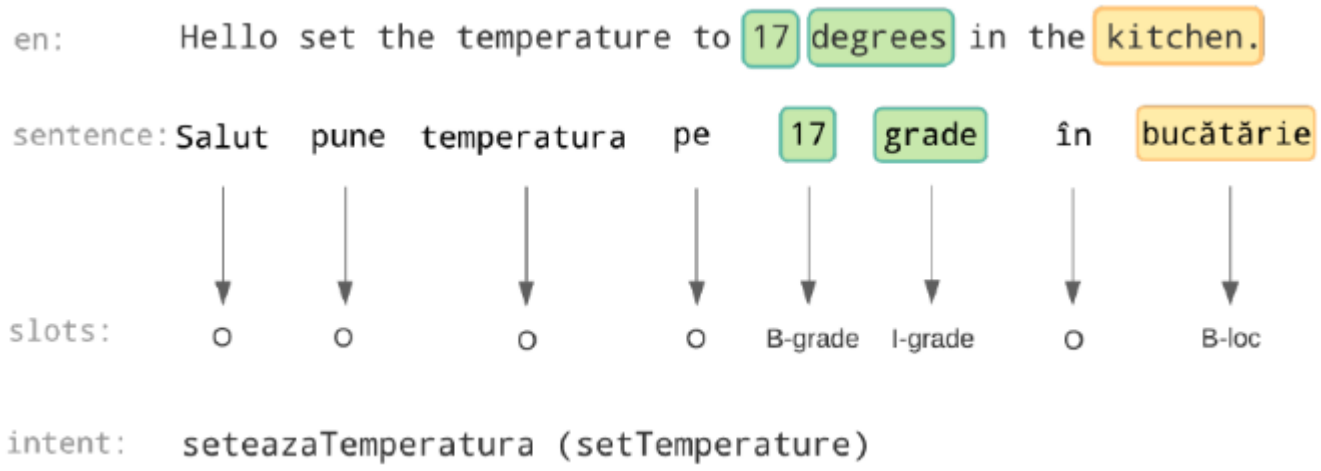




Background

Slot Filling

- Language Model이 특정 단어가 어떤 카테고리에 속하는지 맞추는 것



Sentences

X_1	show	flights	from	baltimore	to	dallas
X_2	show	flights	from	philadelphia	to	boston

Slot Filling

Y	O	O	O	B-FromCity	O	B-ToCity

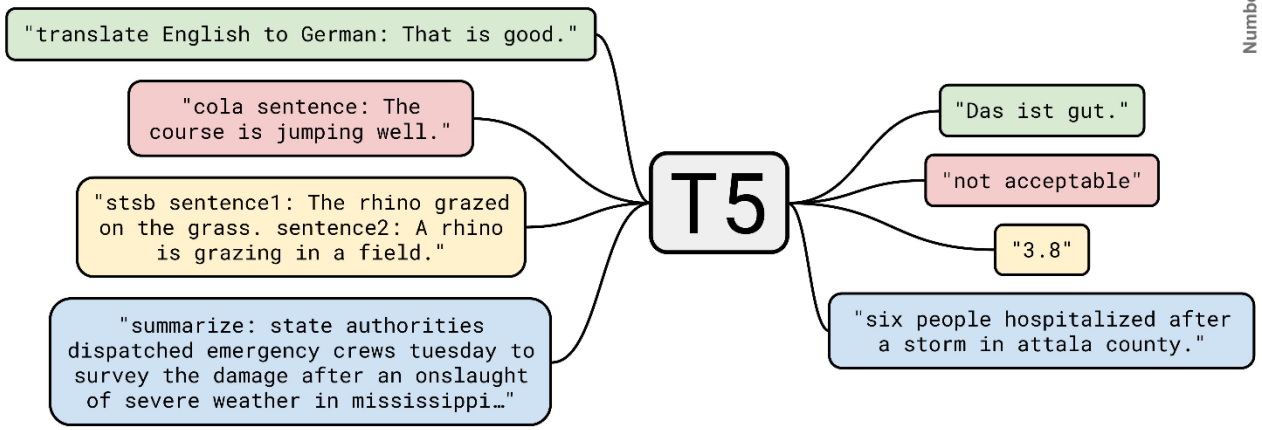
Delexicalization

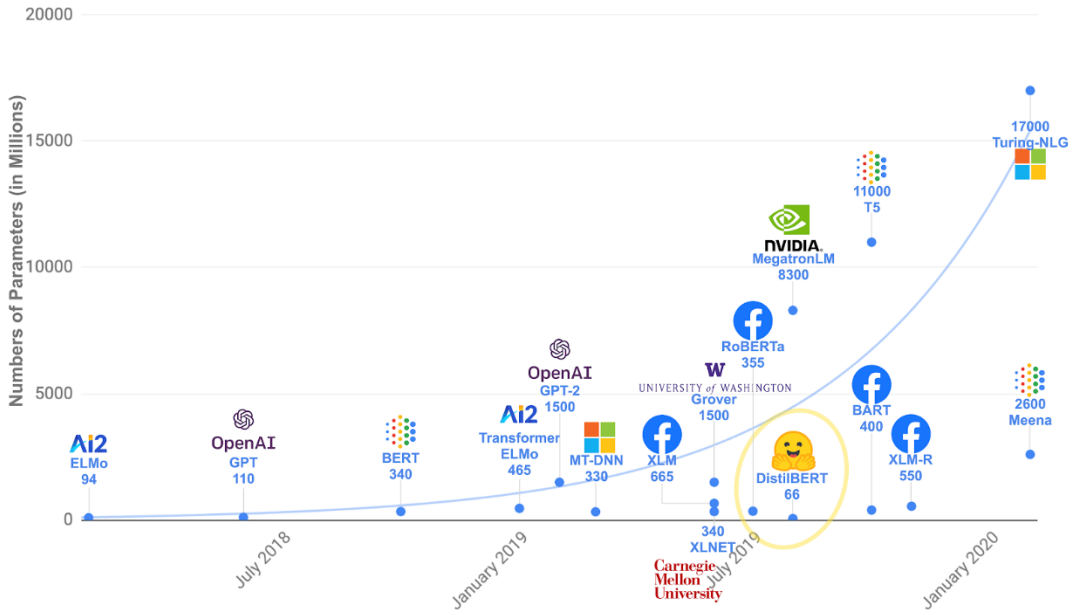
Y^{word}	show	flights	from	B-FromCity	to	B-ToCity

Introduction

Introduction

- GPT-3와 T5는 Transformer 딥러닝 NLP 모델 계열에서 가장 강력한 NLG 모델이며, 이들은 많은 양의 세계 지식 저장
  →텍스트를 생성함에 있어서 강력한 성능 보임
- 이들은 trainable parameter의 크기가 증가함에 따라 성능 또한 점차 발전함



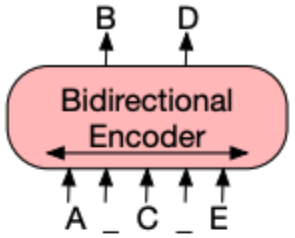


Introduction

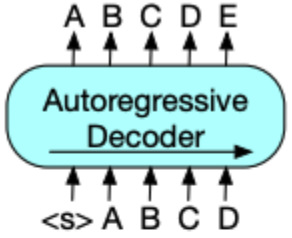
- 최근 non-parametric knowledge를 사용하는 Transformer에 대한 연구가 이루어짐
- REALM, RAG는 모두 조건부 생성을 지원하기 위해 indexing된 passage set을 사용
- 이를 통해 모델들은 지식의 원천으로 corpus를 사용하여 모델이 사용할 수 있는 정보를 수십 또는 수백 기가바이트로 확장 및 computational cost의 부분적은 선형 증가로 확장

Introduction

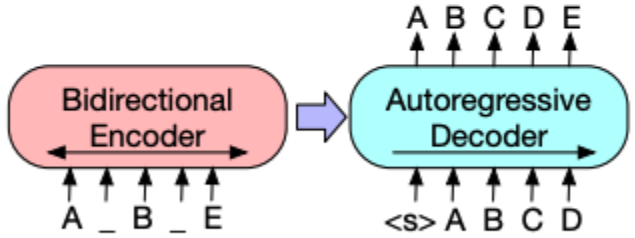
- 최근의 발전은 BART에 의해 영감을 받았으며, 이는 Bidirectional encoder와 auto regressive decoder를 결합하여 하나의 seq2seq 모델을 만듦



(a) BERT: Random tokens are replaced with masks, and the document is encoded bidirectionally. Missing tokens are predicted independently, so BERT cannot easily be used for generation.



(b) GPT: Tokens are predicted auto-regressively, meaning GPT can be used for generation. However words can only condition on leftward context, so it cannot learn bidirectional interactions.



(c) BART: Inputs to the encoder need not be aligned with decoder outputs, allowing arbitary noise transformations. Here, a document has been corrupted by replacing spans of text with mask symbols. The corrupted document (left) is encoded with a bidirectional model, and then the likelihood of the original document (right) is calculated with an autoregressive decoder. For fine-tuning, an uncorrupted document is input to both the encoder and decoder, and we use representations from the final hidden state of the decoder.

Introduction

- 해당 연구는 이전 작업과 다른 두가지 측면 존재
  - reranking 방식은 비교할 수 없는 점수의 검색 결과를 병합할 수 있도록 함
  - Knowledge Distillation을 사용하여 최종 출력 sequence의 정답만을 사용하여 initial retrieval, reranker, generator 학습
- KILT benchmark는 최근 외부 지식 접근이 필요한 NLP task를 해결하기 위해
  pretrain된 language model의 능력을 평가하기 위해 도입 →4가지의 작업인 Slot filling, QA, fact checking, conversation 평가

Introduction

Contribution

- Re^2G를 도입하여 검색을 통합하는 Generation Language Model에서 reranker의 효과를 입증
- Neural과 traditional keyword 기반 접근을 결합하여 초기 검색 방법을 Ensemble로 확장함으로 써 Re^2G를 확장
- Re^2G는 T-REx (Slot Filling), Natural Questions (QA), TriviaQA (QA), FEVER (Fact Checking), Wizard of Wikipedia (Conversation) 에서 각각 9%, 31%, 34%, 22%, 10%의 상대적인 성능 향상을 통해 당시의 SOTA 달성
- 지속적인 개발을 지원하기 위해 코드를 오픈 소스로 배포

Methodology

Methodology

- RAG, Multi-DPR, KGI 접근법은 neural IR (Information Retrieval) 구성 요소를 학습하고 이를 통해 올바른 출력 생성되도록 end-to-end로 학습
- 초기 검색 결과는 rerank를 통해 크게 개선될 수 있음이 이전에 입증
  → 검색을 통합하는 NLG 시스템이 rerank로 부터 혜택 있음 가정
- DPR에서 반환된 passage의 순위 뿐만이 아니라 서로 비교가 불가능한 점수의 경우에도 rerank 가능 (e.g. BM25 & ANN idx)

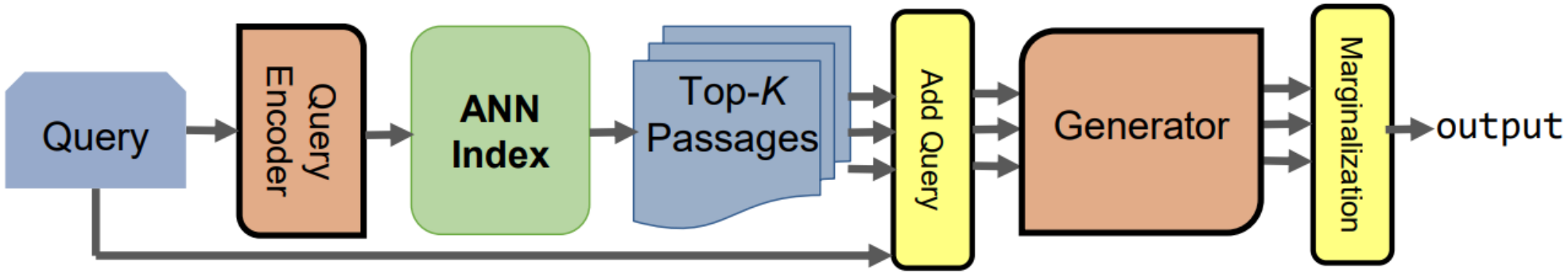


Figure 2: RAG Architecture

Methodology

Reranker

- reranker는 sequence-pair classification에 기반하여 query와 passage가 BERT에 함께 입력
  cross attention은 두 sequence의 토큰에 함께 적용 (interaction model)
- interaction model 구조는 initial retrieval에 사용된 representation model과 대조

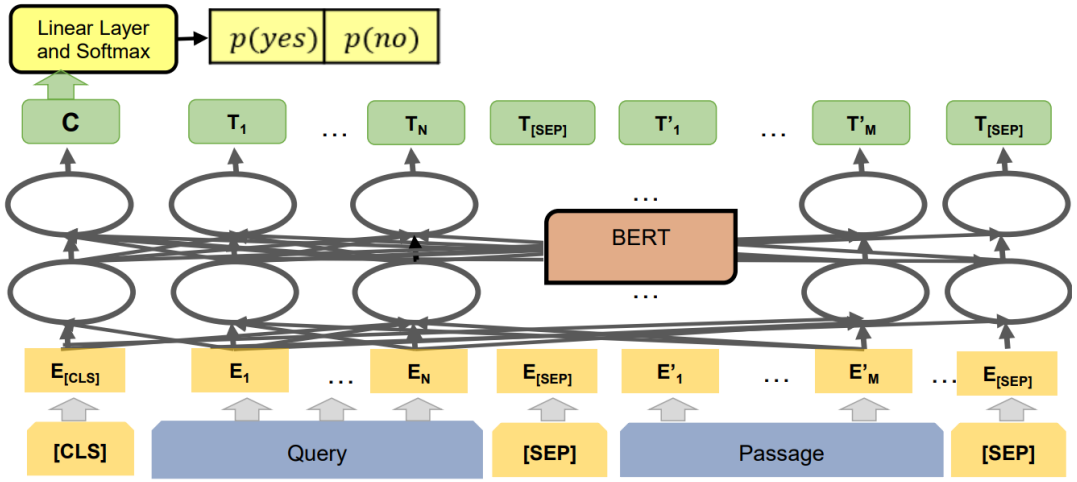


Figure 4: Interaction Model Reranker

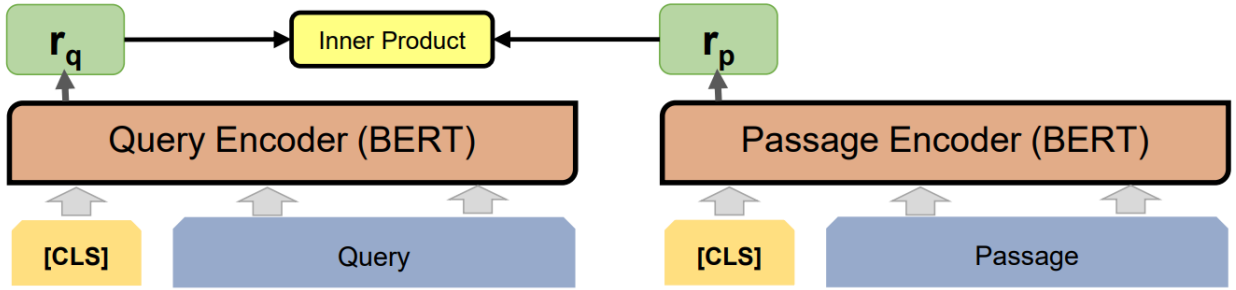


Figure 5: Representation Model for Initial Retrieval

Methodology

Training

- KILT 작업에는 두가지 유형의 정답이 제공: 목표 출력 sequence와 corpus 내에서 목표 출력을 지원하는 구문 또는 구문에 대한 출처
- 4단계로 나누어 training
  - DPR training
  - generation training
  - reranker training
  - end-to-end training
- 초기 DPR & reranker에서 출처 정답 사용 / generation과 end-to-end에서 목표 출력만을 사용

Methodology

Training

- 4-step training instances
  - 원본 KILT instance는 $< q, t, \text{Prov} >$의 tuple; q: $query$, t: $target$, Prov: $Passage\ set$
  - DPR training은 $< q, p^+, p^- >$의 tuple로 여기서 $p^+ \in \text{Prov},\ p^- \notin \text{Prov}$
  - reranker training은 DPR과 BM25의 적용으로 시작하여 $< q, P >$ tuple 생성; $P = BM25(q) \cup DPR(q)$
  - generation 및 end-to-end training instance는 $< q, t >$의 query와 target 쌍

T-REx	**Dracula (7923)**	Wizard of Wikipedia
Input: Dracula [SEP] narrative location **Output:** Transylvania **Provenance:** 7923-2	Dracula is an 1897 Gothic horror novel by Irish author Bram Stoker. It introduced the character of Count Dracula, and established many conventions of subsequent vampire fantasy. The novel tells the story of Dracula's attempt to move from Transylvania to England so that he may find new blood and spread the undead curse, and of the battle between Dracula and a small group of men and a woman led by Professor Abraham Van Helsing.	**Input:** • I really like vampires!! • Vampires are intense and based on European folklore. Do you have any favorite vampires? • I think dracula is the best one!!! **Output:** He's one of the best! He's based on the character from the 1897 horror book of the same name. **Provenance:** 7923-1
Natural Questions		
Input: when did bram stoker's dracula come out **Output:** 1897 **Provenance:** 7923-1		
FEVER		
Input: Dracula is a novel by a Scottish author. **Output:** REFUTES **Provenance:** 7923-1		

Figure 1: KILT tasks of slot filling, question answering, fact checking and dialog

Methodology

Reranker Training

- reranker를 독립적으로 훈련하기 위해 training set에서 DPR과 BM25의 초기 검색 결과 수집

  → 병합하여 reranker의 training data로 사용
- 일부 dataset에서 여러 개의 positive passage 존재 → positive passage에 대한 log-likelihood의 합계에 negative 사용
- reranker에 의해 제공된 logit: z_r
- 올바른 passage idx: Prov

$$loss = -\sum_{i \in \mathrm{Prov}} \log(softmax(z_r)_i)$$

Methodology

End-to-End Training

- RAG에서 gradient는 query encoder로 전파 → query vector와 passage vector 간의 내적이 각 sequence의 weight로 사용
- 이는 RAG marginalization으로 이어짐
- BART 모델의 입력은 $s_j = p_j[SEP]q$로 구성된 sequence
  - 각 sequence에 대한 확률은 검색 또는 reranker score의 softmax를 통해 결정
  - BART의 token predict logit을 사용하여 각 target token t_i에 대한 확률을 구함
  - loss는 모든 target token 및 sequence에 대해 계산된 negative log-likelihood의 합

$$P(s_j) = softmax(z_r)_i$$
$$P(t_i|s_j) = softmax\left(BART(s_j)_i\right)_{t_i}$$
$$loss = -\sum_{i,j} \log\left(P(t_i|s_j) \cdot P(s_j)\right)$$

Methodology

End-to-End Training

- Re^2G에서 initial retrieval이 아닌 reranker의 score가 sequence의 영향을 가중치로 사용되며,

  이는 query encoder의 gradient가 0이 되며, 이는 marginalization이 더 이상 query 및 passage vector의 내적에 의존 X
- 3가지의 해결 책 고려 가능
  - DPR 및 reranker score 결합
  - query encoder 고정
  - online knowledge distillation

Methodology

End-to-End Training

- 첫번째 solution: DPR 및 reranker에 log softmax를 추가하여 두 system이 생성에 미치는 영향으로 통해 훈련 가능성 확보
  →reranker가 과소평가할 가능성이 있는 구문에 대해 DPR이 가장 높은 점수를 부여하게 되어 성능 저하 발생
- 두번째 solution: query encoder의 parameter를 고정하고 reranker와 generator 구성 요소만을 훈련하는 것
  Wizard of Wikipedia dataset에서 최적의 해결책
- 세번째 solution: Knowledge Distillation을 응용하여 reranker를 teacher model로 사용하여 DPR student 모델에 label 제공
  이때 knowledge distillation은 online으로 발생 → KL-divergence를 사용하여 initial retrieval의 loss로 작용

$$loss = D_{KL}\left(softmax\left(\frac{z_s}{T}\right) \parallel softmax\left(\frac{z_t}{T}\right)\right) \cdot T^2$$

Methodology

Inference

- inference 시 query는 DPR query encoder를 사용하여 encoding 되며, HNSW index로 부터 상위 12개 passage 반환
- query는 또한 BM25로 전달되어 상위 12개 BM25 결과가 수집
- 이 두 세트의 passage가 reranker로 전달되어 scoring

  → 상위 5개의 구문이 query와 결합되어 $BART_{LARGE}$에 전달되어 output 생성
- 5개의 output sequence는 reranker의 softmax를 통해 weight를 두어 최종 output 생성

Experiments

Experiments

Retrieval

	T-REx		NQ		TriviaQA		FEVER		WoW	
	R-Prec	**R@5**	**R-Prec**	**R@5**	**R-Prec**	**R@5**	**R-Prec**	**R@5**	**R-Prec**	**R@5**
BM25	46.88	69.59	24.99	42.57	26.48	45.57	42.73	70.48	27.44	45.74
DPR Stage 1	49.02	63.34	56.64	64.38	60.12	64.04	75.49	84.66	34.74	60.22
KGI_0 DPR	65.02	75.52	64.65	69.60	60.55	63.65	80.34	86.53	**48.04**	**71.02**
Re^2G DPR	**67.16**	**76.42**	**65.88**	**70.90**	**62.33**	**65.72**	**84.13**	**87.90**	47.09	69.88
KGI_0 DPR+BM25	60.48	80.06	36.91	66.94	40.81	64.79	65.95	90.34	35.63	68.47
Reranker Stage 1	81.22	87.00	70.78	73.05	**71.80**	**71.98**	87.71	92.43	55.50	**74.98**
Re^2G Reranker	**81.24**	**88.58**	**70.92**	**74.79**	60.37	70.61	**90.06**	**92.91**	**57.89**	74.62

Table 2: Development Set Results for Retrieval

Experiments

Ablations

	T-REx (Slot Filling)					
	R-Prec	**Recall@5**	**Accuracy**	**F1**	**KILT-AC**	**KILT-F1**
Re^2G	81.24±1.08	88.58±0.84	86.60±0.94	89.20±0.81	75.66±1.19	77.08±1.15
Re^2G-KD	81.08±1.09	88.84±0.83	87.00±0.93	89.46±0.80	75.72±1.19	77.00±1.15
Re^2G-BM25	71.92±1.25	78.67±1.10	79.48±1.12	82.52±1.00	66.58±1.31	67.93±1.28
KGI_0	65.02±1.32	75.52±1.16	77.52±1.16	80.91±1.03	60.18±1.36	61.38±1.34

	Natural Questions (Question Answering)					
	R-Prec	**Recall@5**	**Accuracy**	**F1**	**KILT-AC**	**KILT-F1**
Re^2G	70.92±1.67	74.79±1.27	46.70±1.84	62.44±1.65	39.23±1.80	50.90±1.76
Re^2G-KD	69.72±1.69	73.73±1.30	46.56±1.84	61.68±1.67	38.24±1.79	49.93±1.76
Re^2G-BM25	70.88±1.67	74.39±1.28	46.70±1.84	61.98±1.66	39.41±1.80	50.91±1.76
KGI_0	64.65±1.76	69.60±1.39	40.50±1.81	55.07±1.71	32.96±1.73	42.87±1.75

	TriviaQA (Question Answering)					
	R-Prec	**Recall@5**	**Accuracy**	**F1**	**KILT-AC**	**KILT-F1**
Re^2G	72.01±1.20	73.16±0.98	74.01±1.17	80.86±0.99	56.04±1.33	60.91±1.27
Re^2G-KD	72.01±1.20	73.16±0.98	73.80±1.18	80.62±1.00	56.04±1.33	60.84±1.28
Re^2G-BM25	71.10±1.21	68.60±1.03	68.59±1.24	76.68±1.08	52.85±1.34	58.37±1.29
KGI_0	61.13±1.31	63.12±1.08	60.68±1.31	66.61±1.20	44.00±1.33	47.35±1.31

Experiments

Ablations

	FEVER (Fact Checking)			
	R-Prec	**Recall@5**	**Accuracy**	**KILT-AC**
Re^2G	90.06±0.53	92.91±0.47	91.05±0.55	80.56±0.76
Re^2G-KD	89.85±0.54	92.48±0.48	90.78±0.55	80.14±0.77
Re^2G-BM25	88.36±0.57	88.46±0.59	90.63±0.56	78.74±0.78
KGI_0	80.34±0.73	86.53±0.63	87.84±0.63	70.06±0.88

	Wizard of Wikipedia (Dialog)					
	R-Prec	**Recall@5**	**Rouge-L**	**F1**	**KILT-RL**	**KILT-F1**
Re^2G	56.48±1.76	74.00±1.56	17.29±0.52	19.35±0.57	11.37±0.58	12.75±0.63
Re^2G-KD	57.89±1.75	74.62±1.54	17.26±0.52	19.39±0.57	11.61±0.58	13.14±0.64
Re^2G-BM25	55.83±1.76	72.72±1.58	17.15±0.51	19.17±0.56	11.13±0.57	12.52±0.63
KGI_0	48.04±1.77	71.02±1.61	16.75±0.48	19.04±0.53	9.48±0.53	10.74±0.59

Table 3: Development Set Results for Re^2G Variations

Conclusion

Conclusion

- KILT dataset의 각종 task에서 SOTA 달성
- BM25와 DPR의 검색 결과를 통합해 정확도 향상 및 reranker 단독으로도 성능 향상에 기여
- Online knowledge distillation을 사용하여 DPR의 성능을 향상시켜 다섯 개의 dataset 중 4개에서 좋은 성능 달성
- 추가 연구와 실제 응용 분야에 유용한 통찰 제공을 위해 소스 코드 공개