고려대학교 KOREA UNIVERSITY

DS·BA Data science & Business analytics

[EMNLP-IJCNLP-MRQA Workshop, 2019]

A Recurrent BERT-based Model for Question Generation

[arXiv, 2019]

Learning to Answer by Learning to Ask: Getting the Best of GPT-2 and BERT Worlds

2020.02.24

고려대학교 산업경영공학과
석사과정 조규원

Question Answering

- [Context, Question] → Find Answer span in the context

> 질병관리본부는 22일 오전 9시 기준 '우한 코로나(코로나19)' 확진자가 142명 추가 발생했다고 밝혔다. 국내 총 확진자 수는 346명으로 늘었다. 지난 18일 31번(여·61) 환자가 확진된 이후 추가 확진자는 △19일 19명 △20일 53명 △21일 100명에서 이날 오전 142명으로 증가 폭이 커지고 있다.

  - Question : 22일 오전 9시 기준 코로나19 확진자 수는?
  - Answer : 346명

Question Generation

- [Context, Answer] → Generate related questions
- Data Augmentation을 통해 Question Answering 성능 향상이 가능

> 질병관리본부는 22일 오전 9시 기준 '우한 코로나(코로나19)' 확진자가 142명 추가 발생했다고 밝혔다. 국내 총 확진자 수는 346명으로 늘었다. 지난 18일 31번(여·61) 환자가 확진된 이후 추가 확진자는 △19일 19명 △20일 53명 △21일 100명에서 이날 오전 142명으로 증가 폭이 커지고 있다.

- Question 1 : 22일 오전 9시 기준 코로나 19 확진자 수는? (Ground Truth)
- Question 2 : 국내 코로나 19 확진자 수는? (Augmented)
- Question 3 : 국내 우한 코로나 확진자 수는? (Augmented)
- Question 4 : 22일 코로나 확진자 수는 총 몇 명으로 늘어났나? (Augmented)

Data Augmentation 효과

Question Generation

- [Context, Answer] → Generate related questions
- 학습 데이터셋에 없던 데이터에 대해서도 Question 생성을 통해 학습이 가능

> 질병관리본부는 22일 오전 9시 기준 '우한 코로나(코로나19)' 확진자가 142명 추가 발생했다고 밝혔다. 국내 총 확진자 수는 346명으로 늘었다. 지난 18일 31번(여·61) 환자가 확진된 이후 추가 확진자는 △19일 19명 △20일 53명 △21일 100명에서 이날 오전 142명으로 증가 폭이 커지고 있다.

- Question 1 : 22일 오전 우한 코로나 추가 확진자 수는? (Ground Truth)
- Question 2 : 19일 코로나 19 추가 확진자 수는?
- Question 3 : 20일 우한 코로나 추가 확진자 수는?
- Question 4 : 21일 우한 코로나 추가 확진자 수는?

Semi-supervised Question Answering

Question Generation

- [Image, Answer] → Generate related questions
- 인풋 데이터의 형태에 따라 다양한 방식으로 Question Generation 이 가능

- Answer : 5명

- Question 1 : 사진 속에 있는 사람의 수는?
- Question 2 : 웃고 있는 사람의 수는?
- Question 3 : 사진 속 여자 수는?

Related Works

- Rule-based --> LSTM-Seq2seq-Attention --> Pre-trained Language Model
- 당연하지만, 위와 같이 연구 방향이 변화하고 있음

① **Rule-based** (Heilman and Smith, 2010)
- ✓ Statement → Question

② **Dual learning** (Tang et al., 2017)
- ✓ Jointly, learn QA, QG task with unlabeled texts

③ **LSTM-Seq2seq-Attention** (Zhao et al., 2018)
- ✓ QG ~ generator, QA ~ discriminator

④ **GAN** (Yang et al., 2018)
- ✓ QG ~ generator, QA ~ discriminator

⑤ **Self-training Cycle** (Sachan and Xing, 2018)v

⑥ **RL-based** (Zhang et al., 2019)
- ✓ QPP, QAP Reward

⑦ **Pre-trained LM**
- ✓ UniLM (Dong et al., 2019)
- ✓ BERT-HLSQG (Chan et al., 2019)
- ✓ GPT2+BERT (Kelin et al., 2019)

Paper Review

- A Recurrent BERT-based Model for Question Generation
- BERT-QG / BERT-SQG / BERT-HLSQG
- 총 3가지 방법에 대한 실험

나는 정답은 잘 맞추는데...
내가 질문을 생성할 수 있을까?

BERT

질문을 생성은 할 수 있겠는데
한 번에 생성하니까 힘드네...

BERT-QG

앞 토큰을 보면서 질문을 생성하니
비교적 잘 되는군... 그런데 정답이
문서에 자주 등장해서 어떤 질문을
생성할지 헷갈린다...

BERT-SQG

정답 부분에 하이라이트를
해주니까 헷갈리지도 않고
질문이 아주 잘 생성 되는군!

BERT-HLSQG

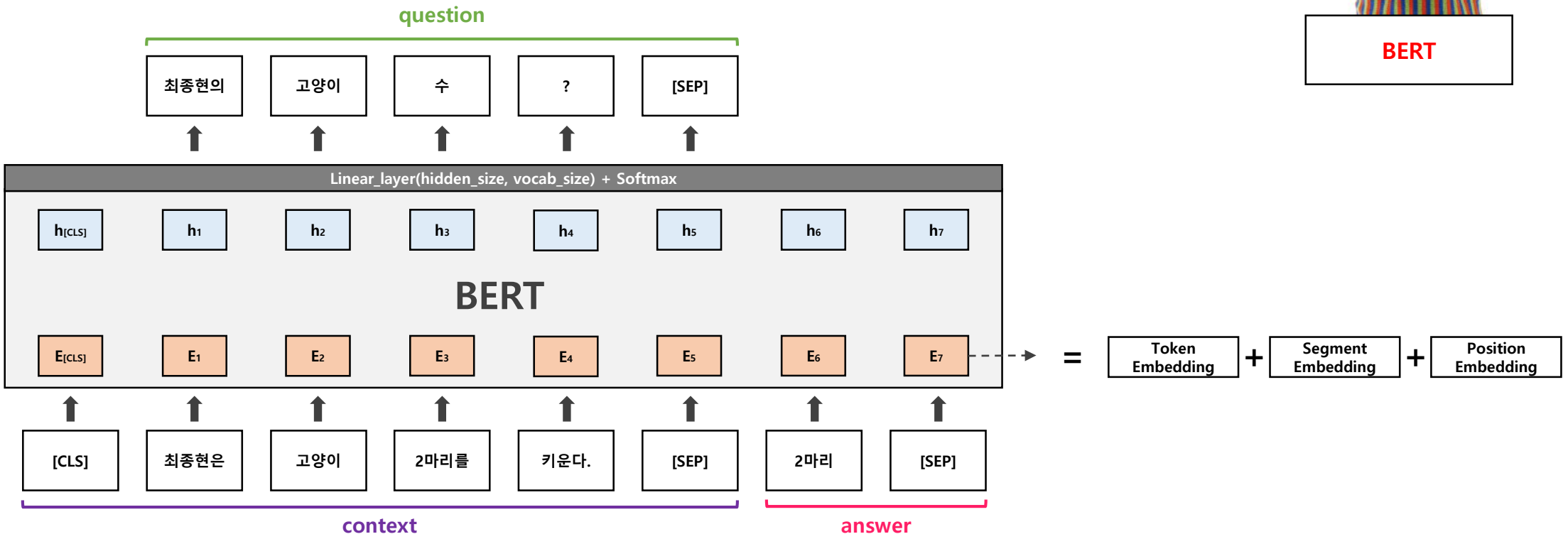
Paper Review

< A Recurrent BERT-based Model for Question Generation >

나는 정답은 잘 맞추는데...
내가 질문을 생성할 수 있을까?

BERT

- BERT-QG(Question Generation)
  - ✓ Input : [[CLS], t_1, t_2, …, t_n, [SEP], a_1, a_2, …, a_k, [SEP]]
  - ✓ Label : [q_1, q_2, …, q_j, [SEP]]



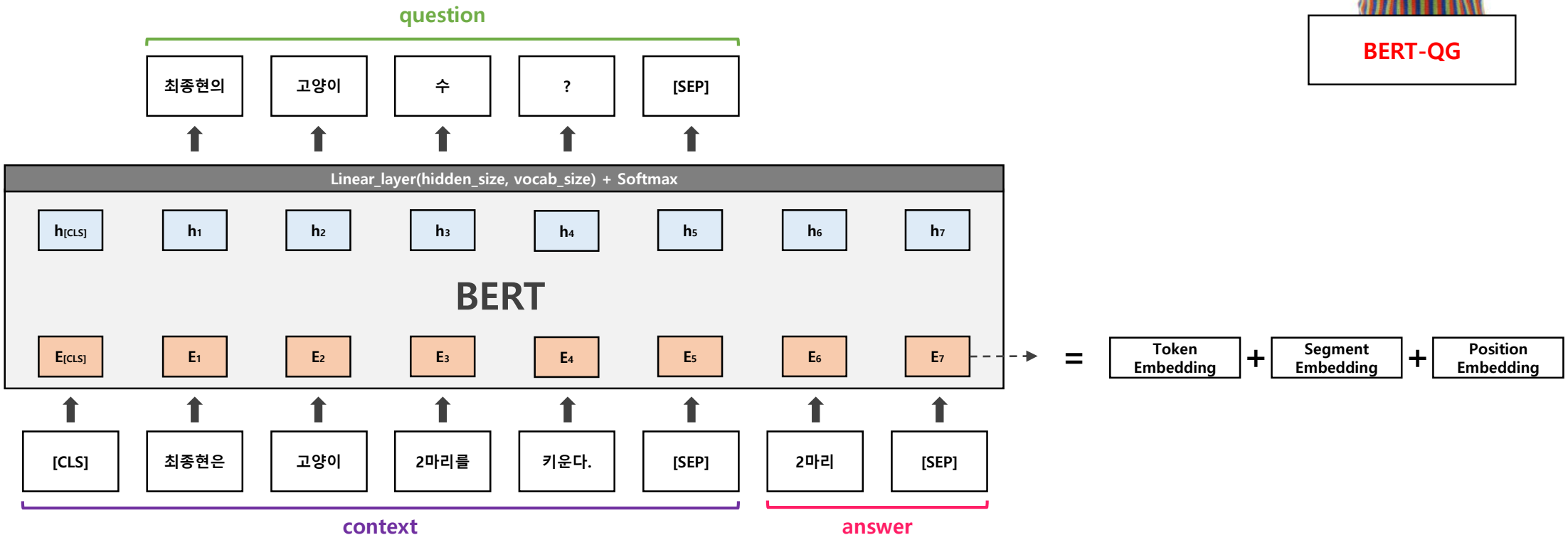
Paper Review

< A Recurrent BERT-based Model for Question Generation >

질문을 생성은 할 수 있겠는데
한 번에 생성하니까 힘드네...

BERT-QG

- BERT-QG 방법의 문제점
  - ✓ 디코딩 단계에서 이전 Step의 단어에 대한 고려 없이 한 번에 문장 생성
  - ✓ 제대로 된 문장이 생성이 되지 않는 문제가 발생



Paper Review

< A Recurrent BERT-based Model for Question Generation >

질문을 생성은 할 수 있겠는데
한 번에 생성하니까 힘드네...

BERT-QG

- BERT-SQG(Sequential Question Generation)
  - BERT-QG는 디코딩 단계에서 이전 Step의 token에 대한 고려 없이 한 번에 문장 생성
  - 제대로 된 문장이 생성이 되지 않는 문제가 발생 → Poor Readability
  - 이전 Step의 디코딩 된 token에 대해 고려해주기 위해 모델 input을 Sequential 하게 구성
  - Input : [[CLS], t_1, t_2, ⋯, t_n, [SEP], a_1, a_2, ⋯, a_k, [SEP], q_1, q_2, ⋯, q_j, [MASK]]
  - Label : [q_1, q_2, ⋯, q_j, [SEP]]
  - [MASK] token의 hidden_state인 $h_{[MASK]}$ 만 사용해서 해당 Step의 디코딩 수행 → 반복

	X (context / answer / question)	x_i
iter0	[CLS] The Super Bowl 50 was played at Santa Clara, California. [SEP] Santa Clara, California. [SEP] [MASK]	Where
iter1	[CLS] The Super Bowl 50 was played at Santa Clara, California. [SEP] Santa Clara, California. [SEP] Where [MASK]	did
iter2	[CLS] The Super Bowl 50 was played at Santa Clara, California. [SEP] Santa Clara, California. [SEP] Where did [MASK]	Super
iter3	[CLS] The Super Bowl 50 was played at Santa Clara, California. [SEP] Santa Clara, California. [SEP] Where did Super [MASK]	Bowl
iter4	[CLS] The Super Bowl 50 was played at Santa Clara, California. [SEP] Santa Clara, California. [SEP] Where did Super Bowl [MASK]	50
iter5	[CLS] The Super Bowl 50 was played at Santa Clara, California. [SEP] Santa Clara, California. [SEP] Where did Super Bowl 50 [MASK]	take
iter6	[CLS] The Super Bowl 50 was played at Santa Clara, California. [SEP] Santa Clara, California. [SEP] Where did Super Bowl 50 take [MASK]	place?
iter7	[CLS] The Super Bowl 50 was played at Santa Clara, California. [SEP] Santa Clara, California. [SEP] Where did Super Bowl 50 take place [MASK]	[SEP]
iter8	[CLS] The Super Bowl 50 was played at Santa Clara, California. [SEP] Santa Clara, California. [SEP] Where did Super Bowl 50 take place [SEP] [MASK]	

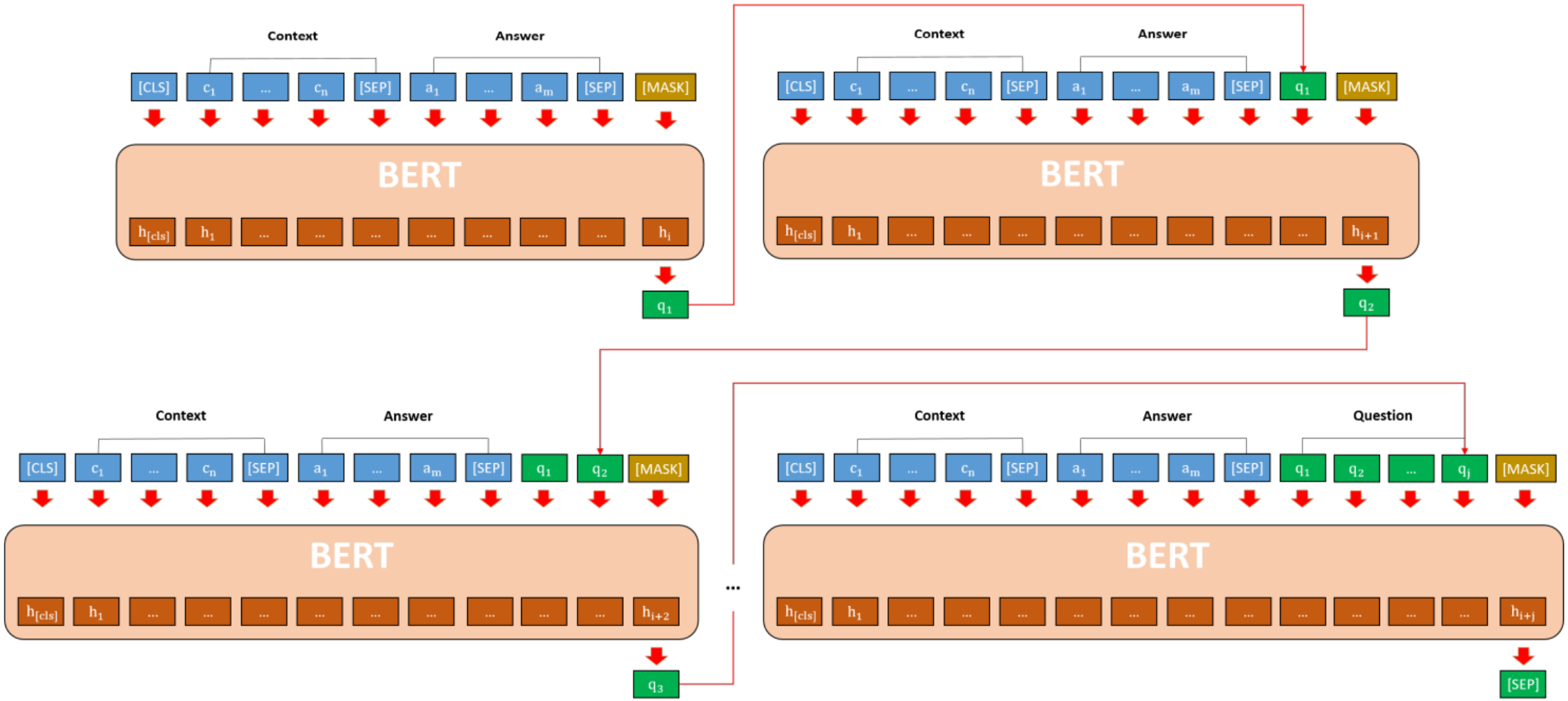


Figure 3: The BERT-SQG architecture

Paper Review

< A Recurrent BERT-based Model for Question Generation >

- BERT-SQG 문제점
  - Context 내에 정답과 같은 Span이 자주 등장할 경우 발생
  - 정답 부분이 어디인지에 따라 생성되는 문장이 달라져야 한다.
  - BERT-SQG는 이런 상황에 적절히 대응하기 어렵다.

밥을 지을 때는 소금을 넣어서는 절대 안된다.
그렇지만 스테이크를 구울 때는 후추와 함께 소금을 살짝 뿌리면 맛있다.

  - Answer : 소금
  - Question : 밥을 지을 때 절대 넣어서 안될 것은?
  - Question : 스테이크 구울 때 후추와 함께 넣으면 좋은 것은?

앞 토큰을 보면서 질문을 생성하니 비교적 잘 되는군... 그런데 정답이 문서에 자주 등장해서 어떤 질문을 생성할지 헷갈린다...

BERT-SQG

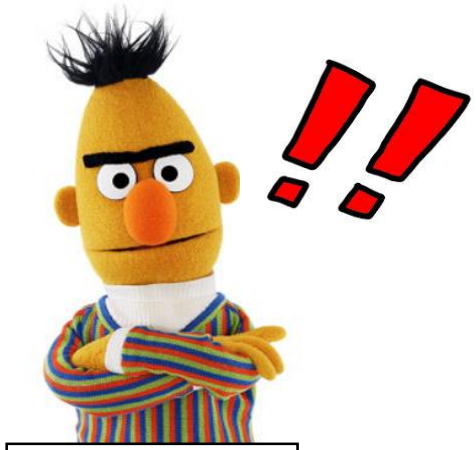
Paper Review

< A Recurrent BERT-based Model for Question Generation >

- BERT-SQG 문제점
  - 이를 반영해주기 위해 Answer 를 따로 분리해서 넣지 않고 Context 내에서 위치를 표기 → [HL] Token 활용
  - Input : [[CLS], t_1, t_2, [HL], t_{answer_start}, ⋯, t_{answer_end}, [HL], ⋯, t_n, [SEP], q_1, q_2, ⋯, q_j, [MASK]]
  - Label : [q_1, q_2, ⋯, q_j, [SEP]]
  - 이후, 동작 방식은 BERT-SQG와 동일

	X	context	question	x_i
iter0	[CLS] The Super Bowl 50 was played at [HL] Santa Clara, California [HL] . [SEP]		[MASK]	Where
iter1	[CLS] The Super Bowl 50 was played at [HL] Santa Clara, California [HL] . [SEP]		Where [MASK]	did
iter2	[CLS] The Super Bowl 50 was played at [HL] Santa Clara, California [HL] . [SEP]		Where did [MASK]	Super
iter3	[CLS] The Super Bowl 50 was played at [HL] Santa Clara, California [HL] . [SEP]		Where did Super [MASK]	Bowl
iter4	[CLS] The Super Bowl 50 was played at [HL] Santa Clara, California [HL] . [SEP]		Where did Super Bowl [MASK]	50
iter5	[CLS] The Super Bowl 50 was played at [HL] Santa Clara, California [HL] . [SEP]		Where did Super Bowl 50 [MASK]	take
iter6	[CLS] The Super Bowl 50 was played at [HL] Santa Clara, California [HL] . [SEP]		Where did Super Bowl 50 take [MASK]	place?
iter7	[CLS] The Super Bowl 50 was played at [HL] Santa Clara, California [HL] . [SEP]		Where did Super Bowl 50 take place [MASK]	[SEP]
iter8	[CLS] The Super Bowl 50 was played at [HL] Santa Clara, California [HL] . [SEP]		Where did Super Bowl 50 take place [SEP] [MASK]	

정답 부분에 하이라이트를
해주니까 헷갈리지도 않고
질문이 아주 잘 생성 되는군!



BERT-HLSQG

Experiments

< A Recurrent BERT-based Model for Question Generation >

- Datasets
  - QG task on SQuAD (Rajpurkar et al., 2016)
  - 기존 논문들과 비교를 위해, Data split ratio를 맞추어 실험 진행

Dataset	Train	Dev	Test	비교 논문
SQuAD 73K	73,240 (80%)	10,570 (10%)	11,877 (10%)	Du et al., 2017
SQuAD 81K	81,577 (100%)	8,964 (50%)	8,964 (50%)	Zhao et al., 2017

- Implementation Details
  - BERT-base model (12 layers, 768 hidden dimensions, 12 attention heads)
  - 2 RTX TITAN – 5 epochs
  - Beam Search – beam_size=3

Experiments

< A Recurrent BERT-based Model for Question Generation >

- Evaluation Metrics & Results
  - NLG Evaluation Metrics : {BLEU-1, BLEU-2, BLEU-3, BLEU-4, METEOR, ROUGE-L}
  - 아래 주소에서 코드를 받아서 쉽게 사용 가능
  - https://github.com/Maluuba/nlg-eval (Sharma et al., 2017)

	Model	BLEU 1	BLEU 2	BLEU 3	BLEU 4	METEOR	ROUGE-L
SQuAD 73K	NQG-RC	43.09	25.96	17.50	12.28	16.62	39.75
	PLQG	43.47	28.23	20.40	15.32	19.29	43.91
	BERT-QG	34.17	15.52	8.36	4.47	14.78	37.60
	BERT-SQG	48.38	**33.15**	24.75	19.08	22.43	46.94
	BERT-HLSQG	**48.29**	33.12	**24.78**	**19.14**	**22.89**	**47.07**
SQuAD 81K	PLQG	44.51	29.07	21.06	15.82	19.67	44.24
	BERT-QG	34.18	15.51	8.57	4.97	14.57	37.65
	BERT-SQG	50.18	35.03	26.60	20.88	23.84	48.37
	BERT-HLSQG	**50.71**	**35.44**	**26.95**	**21.20**	**24.02**	**48.68**

Table 5: Comparison between our model and the published methods using sentence level context

Experiments

< A Recurrent BERT-based Model for Question Generation >

- ## Evaluation Metrics & Results
  - 대부분의 경우에서 BERT-HLSQG가 BERT-SQG 보다 정량적인 Generation 평가에서 미묘하지만 더 좋은 성능을 보임
  - 추가로 실험한 Reading Comprehension Task 실험
  - 인간이 생성한 질문으로만 학습한 모형 대비 EM, F1-score가 3~5% 차이가 나고 질문의 길이가 비슷함
  - 비교적 인간이 생성한 Question에 가까운 문장 생성했다고 주장
  - 별도의 정성적 평가에 대한 자료 제시가 없는 것은 아쉬움

	Exact Match	F1 score	Question avg. tokens
RC	79.09	86.82	12.29
RC-SQG	74.07	82.91	12.09
RC-HLSQG	74.36	83.07	12.06

Table 4: Reading comprehension evaluation results

RC : 인간이 생성한 질문 100%로 RC Task 학습

RC-SQG : BERT-SQG로 생성한 질문 50% + 인간 생성 질문 50% RC Task 학습

RC-HLSQG : BERT-HLSQG로 생성한 질문 50% + 인간 생성 질문 50% RC Task 학습

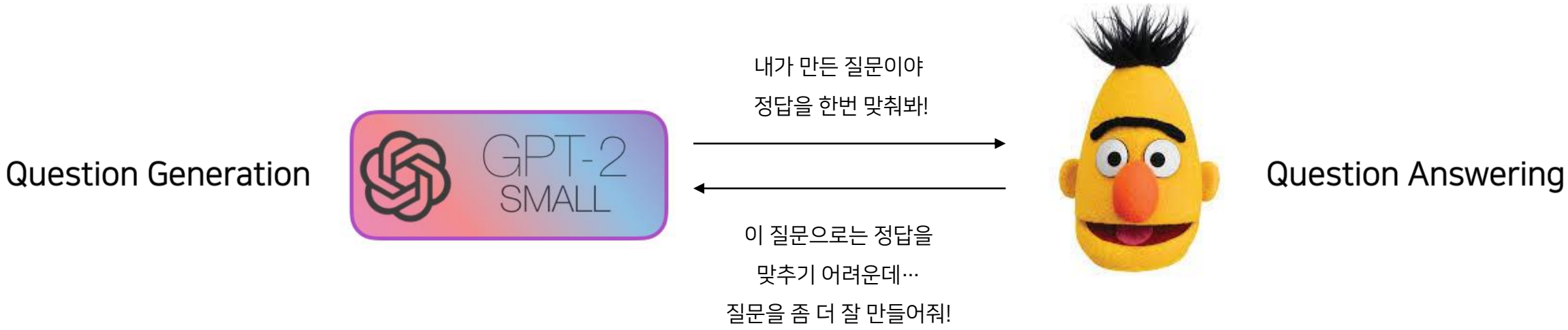
	Model	BLEU 1	BLEU 2	BLEU 3	BLEU 4	METEOR	ROUGE-L
SQuAD 73K	NQG-RC	42.54	25.33	16.98	11.86	16.28	39.37
	PLQG	45.07	29.58	21.60	16.38	20.25	44.48
	BERT-QG	37.49	18.32	10.47	6.10	16.80	41.01
	BERT-SQG	**50.00**	34.54	25.98	20.11	23.88	48.12
	BERT-HLSQG	49.73	**34.60**	**26.13**	**20.33**	**23.88**	**48.23**
SQuAD 81K	PLQG	45.69	30.25	22.16	16.85	20.62	44.99
	BERT-QG	32.61	14.50	7.70	4.08	14.18	37.94
	BERT-SQG	50.89	35.49	26.87	21.04	24.25	48.66
	BERT-HLSQG	**51.54**	**36.45**	**27.96**	**22.17**	**24.80**	**49.68**

Table 6: Comparison between our model and the published methods using paragraph level context

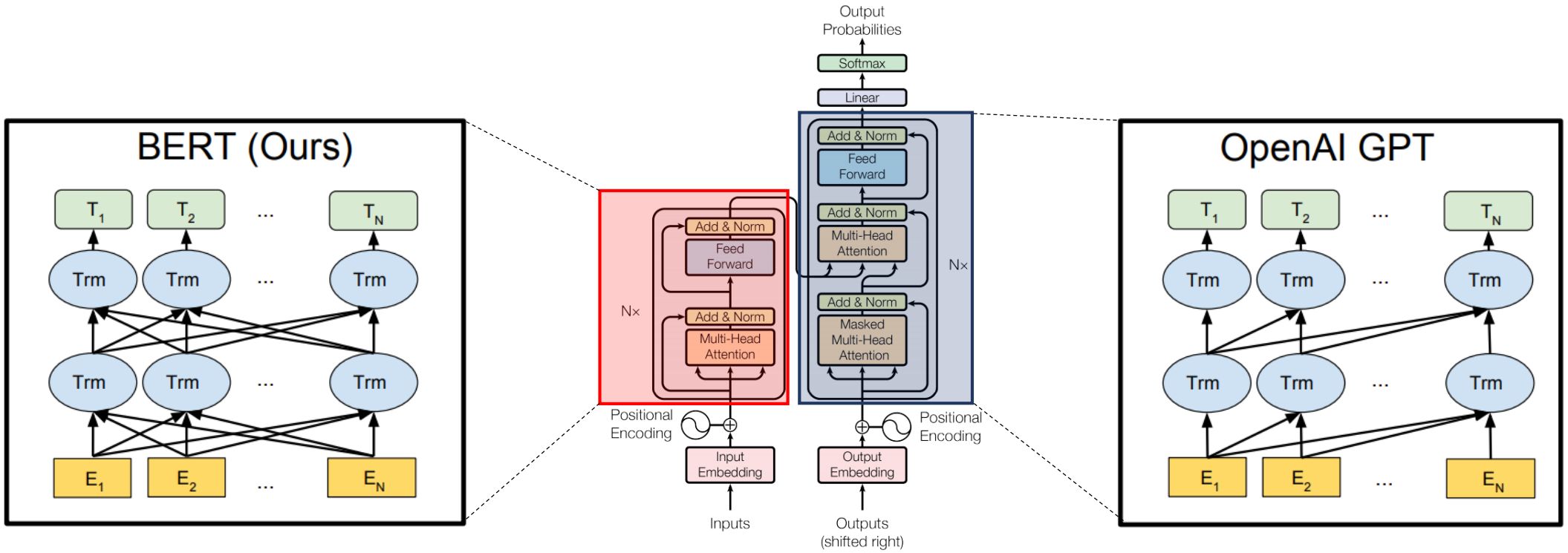
Paper Review

- Learning to Answer by Learning to Ask: Getting the Best of GPT-2 and BERT Worlds
  - ✓ BERT의 Feedback을 통해 GPT2의 QG 성능을 높이려 시도 : Collaborative Learning → Semi-supervised Learning
  - ✓ Question Genration의 성능을 평가하는 surrogate measure로 QA 성능을 사용(EM, F1-score)

- 2 Step으로 진행 : Pre-training → Fine-tuning (End-to-End)



BERT vs. GPT



- Transformer Encoder Block
- Bi-directional Attention
- 문장 생성 불가능(?)

- Transformer Decoder Block
- Uni-directional Attention
- 문장 생성 가능

[BERT has a Mouth, and It Must Speak: BERT as a Markov Random Field Language Model (Alex Wang & Kyunghyun Cho)]

Paper Review

< Learning to Answer by Learning to Ask: Getting the Best of GPT-2 and BERT Worlds >

- Pre-training Step
  - ✓ GPT-2, BERT 각각의 모델에 대해서 QA Head 를 붙여, Question Answering Task 를 풀도록 학습시킨다. (SQuAD 1.1 사용)
  - ✓ 논문에는 정확히 언급되어 있진 않지만, 두 모델 모두 공개된 Pre-trained 모델을 가져와서 fine-tuning 하는 것으로 판단됨.
  - ✓ GPT-2의 경우에도 QA Task를 풀도록 하는 것이 Question 생성에 더 도움이 된다고 판단한 것으로 생각됨.

Paper Review

< Learning to Answer by Learning to Ask: Getting the Best of GPT-2 and BERT Worlds >

- Fine-tuning GPT-2 for Collaborative Generation
  - ✓ 앞서, QA Task 를 학습한 GPT-2 Pretrained 모델에 LM head 를 붙여서 Question Generation을 수행하도록 fine-tuning
  - ✓ 이 단계에서는 BERT와의 어떠한 상호작용도 없음, 논문에서는 이 모델을 LM_init 으로 표기
  - ✓ 또한, 정답의 위치를 표기해주기 위해 ">>" Answer tokens "<<" 와 같이 Highlight를 추가해주었음

최종현은 고양이 두 마리를 키우고 있다.
고양이의 이름은 >> **로미와 줄리** << 다.
로미와 줄리는 겁이 많은 편이다.
처음 보는 사람이 집에 가면 침대 밑에
숨어서 나오지 않는다. [QG_START]
최종현이 키우는 고양이들의 이름은?

최종현은 고양이 두 마리를 키우고 있다.
고양이의 이름은 >> **로미와 줄리** << 다.
로미와 줄리는 겁이 많은 편이다.
처음 보는 사람이 집에 가면 침대 밑에
숨어서 나오지 않는다. [QG_START]
최종현이 키우는 고양이들의 이름은?

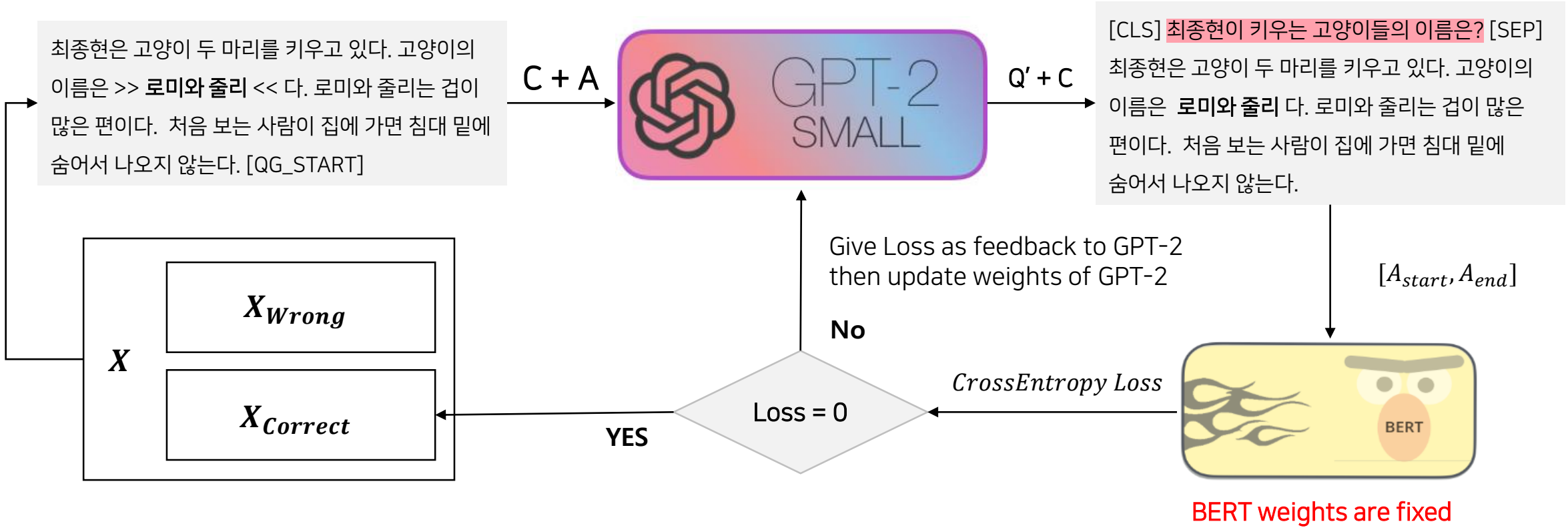
$$Q = \prod p(q_j \mid q_1, \ldots, q_{j-1}; Context, Answer)$$

$$maximize\ the\ likelihood\ Q$$

Paper Review

< Learning to Answer by Learning to Ask: Getting the Best of GPT-2 and BERT Worlds >

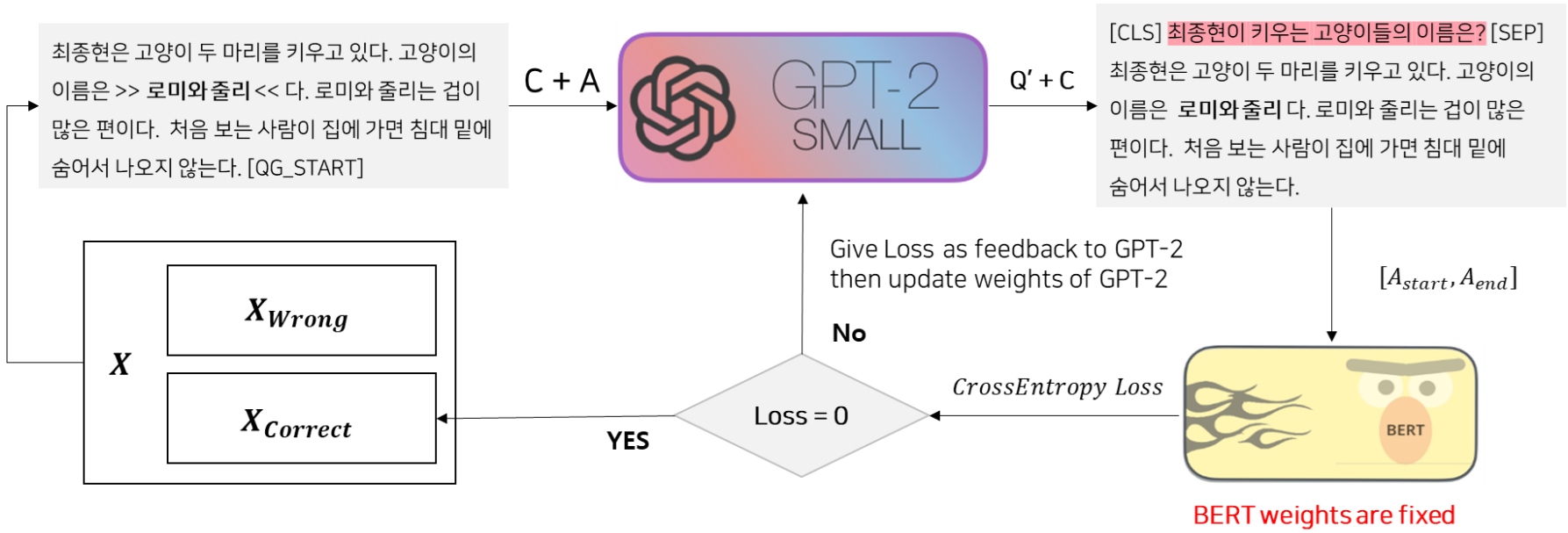
- Re-Fine-tuning GPT-2 with Collaborative Generation
  - ✓ 앞서, SQuAD Dataset 으로 fine-tuning 된 LM_init 모델의 Generation 성능을 높이기 위해 BERT 와의 상호작용 시도



Paper Review

< Learning to Answer by Learning to Ask: Getting the Best of GPT-2 and BERT Worlds >

- Re-Fine-tuning GPT-2 with Collaborative Generation (End-to-End)
  - ✓ 파라미터 최적화 과정에서 Xwrong 의 개수가 작아지도록 학습하는 것이 목적
  - ✓ 모델이 업데이트 되는 과정에서 Xcorrect 에 대해서 Catastrophic forgetting 이 일어날 수 있음
  - ✓ 이를 방지하기 위해, Xcorrect 도 계속해서 샘플링하여 검증 과정을 거치고, 정답을 맞추지 못하면 Xwrong 으로 다시 편입



Qualitative Analysis

< Learning to Answer by Learning to Ask: Getting the Best of GPT-2 and BERT Worlds >

- 정성적인 평가 결과 꽤 높은 퀄리티의 Question이 생성되는 것을 확인할 수 있었음

Example-1:
"The Broncos defeated the Pittsburgh Steelers in the divisional round, 23–16, by scoring 11 points in the final three minutes of the game. They then beat the defending Super Bowl XLIX champion >>***New England Patriots***<< in the AFC Championship Game, 20–18, by intercepting a pass on New England's 2-point conversion attempt with 17 seconds left on the clock. Despite Manning's problems with interceptions during the season, he didn't throw any in their two playoff games. "

GPT-2 LM: Which team did the Broncos beat in the AFC Championship Game?
BERT Feedback: What team did the Broncos defeat in the AFC championship game?
GPT-2 Feedback: Who did the Broncos beat in the Super Bowl?
GT: Who won Super Bowl XLIX?

Example-2:
"The league eventually narrowed the bids to three sites: New Orleans' Mercedes-Benz Superdome, Miami's >>***Sun Life Stadium***<<, and the San Francisco Bay Area's Levi's Stadium. "

GPT-2 LM: What is the name of the stadium in Miami?
BERT Feedback: Which stadium did the league try to buy in Miami?
GPT2 Feedback: What stadium is in Miami?
GT: What venue in Miami was a candidate for the site of Super Bowl 50?

Experiments

< Learning to Answer by Learning to Ask: Getting the Best of GPT-2 and BERT Worlds >

- Quanitative Results
  - ✓ 유사한 방식의기존 연구인 QA-QG Dual 및 BERT의 feedbac을 받지 못한 LM_init 보다 제안된 방법이 더 높은 generation 성능을 보여줌
  - ✓ 기존 연구들과의 성능 비교를 조금 더 해야할 것 같다는 생각을 받았음
  - ✓ 앞서 소개한 BERT-HLSQG에 비하면 매우 떨어지는 성능 → Evalaution Metrics의 문제도 있음!
  - ✓ 최근 나온 QG 관련 논문들에서는 대부분 QA 성능을 통해 QG의 성능을 평가하고자 하는 방향

Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROGUE-L
QA-QG-Dual (Tang et al., 2017a)	-	-	-	5.03	-
LM-init (Radford et al., 2019)	24.85	17.85	11.06	6.85	33.56
Our Proposed Method	**31.46**	**19.50**	**12.41**	**7.84**	**34.51**

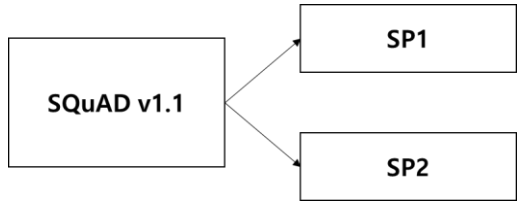
	Model	BLEU 1	BLEU 2	BLEU 3	BLEU 4	METEOR	ROUGE-L
SQuAD 73K	NQG-RC	43.09	25.96	17.50	12.28	16.62	39.75
	PLQG	43.47	28.23	20.40	15.32	19.29	43.91
	BERT-QG	34.17	15.52	8.36	4.47	14.78	37.60
	BERT-SQG	48.38	**33.15**	24.75	19.08	22.43	46.94
	BERT-HLSQG	**48.29**	33.12	**24.78**	**19.14**	**22.89**	**47.07**
SQuAD 81K	PLQG	44.51	29.07	21.06	15.82	19.67	44.24
	BERT-QG	34.18	15.51	8.57	4.97	14.57	37.65
	BERT-SQG	50.18	35.03	26.60	20.88	23.84	48.37
	BERT-HLSQG	**50.71**	**35.44**	**26.95**	**21.20**	**24.02**	**48.68**

Table 5: Comparison between our model and the published methods using sentence level context

Experiments

< Learning to Answer by Learning to Ask: Getting the Best of GPT-2 and BERT Worlds >

- Semi-supervised Question Answering - ①
  - ✓ QG를 활용한 Semi-supervised 성능 평가를 위해 SQuAD v1.1 dataset 을 50% : 50% 로 split
  - ✓ SP1 을 이용해 먼저, 제안된 방법에 대한 성능 검증을 시도
  - ✓ GT data로 학습한 모델의 성능에 거의 가깝게 제안된 모델이 따라 잡는 것을 확인
  - ✓ 이를 통해, GPT-2가 생성한 질문이 정답을 찾는데 유의미한 질문을 생성했다고 판단 가능



Method	EM	F1
Supervised (Upper-bound)	79.60	87.30
LM-init (Radford et al., 2019)	67.51	77.15
Our Method (*GPT-2*)	70.61	79.73
Our Method (*BERT*)	**75.37**	**84.42**

Table 3: Question answering performance on SQuAD 1.1 (SP1), with exact measure (EM) and F1 metric. Our Method (*BERT*) denotes the proposed approach using GPT-2 for question generation as well as BERT as question answering. Our Method (*GPT-2*) denotes the approach employing GPT-2 for question generation as well as the modified GPT-2 QA module as discussed in the section dealing with the ablation study of the QA component.

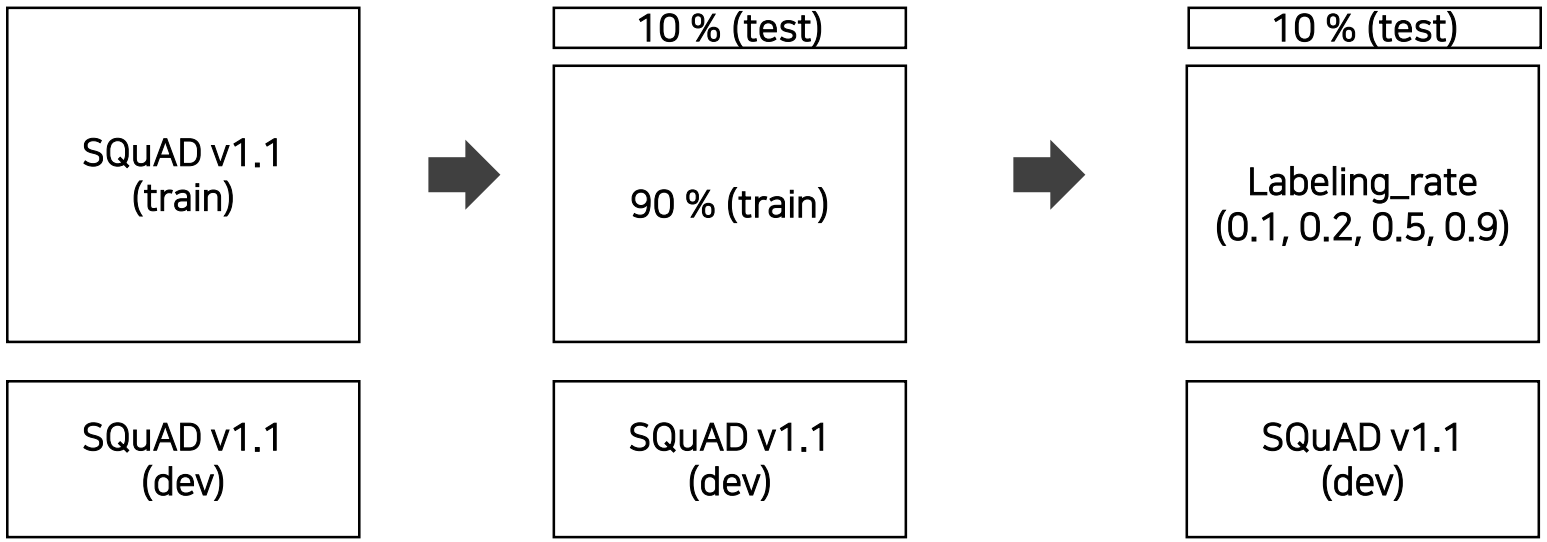
Method	EM	F1
Supervised (Upper-bound)	80.80	88.50
LM-init (Radford et al., 2019)	67.51	77.15
Our Method	**78.47**	**86.41**

Table 4: Question answering performance on SQuAD 1.1 (all), with exact measure (EM) and F1. *Our Method* denotes to the BERT QA model trained on entire training set of SQuAD, but half the training data is fully supervised (SP2), but the other half (SP1) is generated by our proposed method, as discussed in the section dealing with quantitative evaluation using surrogate measure.

Experiments

< Learning to Answer by Learning to Ask: Getting the Best of GPT-2 and BERT Worlds >

- Semi-supervised Question Answering - ②
  - ✓ QG를 활용한 Semi-supervised 성능 평가를 위해 SQuAD v1.1 dataset 을 split
  - ✓ Labeling_rate에 해당하는 만큼의 데이터는 GT-Question을 이용, 나머지 데이터는 Generated-Question을 이용하여 QA 성능 평가
  - ✓ Yang et al. ,2017 실험 결과와 비교



Experiments

< Learning to Answer by Learning to Ask: Getting the Best of GPT-2 and BERT Worlds >

- Semi-supervised Question Answering - ②
  - ✓ GT Question 사용 비율을 줄일 수록 성능이 떨어짐
  - ✓ 하지만, 50%까지는 성능 하락폭이 크지 않고
  - ✓ GT:GEN = 10:90 의 경우도 어느 정도 성능이 나올 수 있음을 확인
  - ✓ 기존 연구와 비교했을 때, 상당한 폭으로 성능 향상이 이루어짐

Labeling rate	Method	Dev F1	Test F1	Test EM
0.1	Gen + GAN (Ganin and Lempitsky, 2015)	0.4897	0.4373	0.2885
0.1	Gen + dual (He et al., 2016)	0.5036	0.4555	0.3005
0.1	Gen + domain (Yang et al., 2017)	0.5234	0.4703	0.3145
0.1	Gen + domain + adv (Yang et al., 2017)	0.5313	0.4802	0.3218
0.1	Our Proposed Method	**0.6931**	**0.6391**	**0.4741**
0.2	Gen + GAN (Ganin and Lempitsky, 2015)	0.5525	0.5037	0.3470
0.2	Gen + dual (He et al., 2016)	0.5720	0.5192	0.3612
0.2	Gen + domain (Yang et al., 2017)	0.5749	0.5216	0.3658
0.2	Gen + domain + adv (Yang et al., 2017)	0.5867	0.5394	0.3781
0.2	Our Proposed Method	**07614**	**0.7053**	**0.5476**
0.5	Gen + GAN (Ganin and Lempitsky, 2015)	0.6110	0.5590	0.4044
0.5	Gen + dual (He et al., 2016)	0.6368	0.5746	0.4163
0.5	Gen + domain (Yang et al., 2017)	0.6378	0.5826	0.4261
0.5	Gen + domain + adv (Yang et al., 2017)	0.6375	0.5831	0.4267
0.5	Our Proposed Method	**0.8185**	**0.7564**	**0.6056**
0.9	Gen + GAN (Ganin and Lempitsky, 2015)	0.6396	0.5874	0.4317
0.9	Gen + dual (He et al., 2016)	0.6511	0.5892	0.4340
0.9	Gen + domain (Yang et al., 2017)	0.6611	0.6102	0.4573
0.9	Gen + domain + adv (Yang et al., 2017)	0.6585	0.6043	0.4497
0.9	Our Proposed Method	**0.8409**	**0.7755**	**0.6282**

Table 2: Performance with various labeling rates and methods with unlabeled dataset comprising 50k samples. "Dev" denotes the development set, and "Test" denotes the test set, with exact measure (EM) and F1 metric. Results from all approaches apart from the proposed one are taken from (Yang et al., 2017).

Thank You!