Generative Model

자네가 무언가 찾을 때
난 만들어 버린다네

Team KiYOUNG2, Chaeeun Kim, Haram Lee

Generative Model

Prompt

- **input**

```
model_inputs = [f"질문: {q} 지문: {c} </s>"
                for q, c in zip(examples["question"], examples["context"])]
```

- **결과**

EM : 44.167 / F1 : 51.146

```
question: {q}  context: {c}</s>
```

EM : 45.417 / F1 : 52.698

```
질문: {q}  지문: {c}</s>
```

EM : 45.417 / F1 : 52.502

```
질문: {q}\n
지문: {c}\n
</s>
```

EM : 44.583 / F1 : 51.328

```
문제: {q}  지문: {c}</s>
```

EM : 42.5 / F1 : 49.203

```
지문을 읽고 질문에 답하시오.\n질문: {q}
```

EM : 42.917 / F1 : 49.229

```
질문: {q}  내용: {c}</s>
```

EM : 45.833 / F1 : 52.236

```
질문: {q}\n{c}</s>
```

EM : 43.333 / F1 : 50.897

```
질문: {q}  제목: {t}  지문: {c}</s>
```

1. Summary

- **Train dataset**

Question	Context	**Summary**
Question	**Summary**	Context

- **Summary list**
  - **Abstractive** : 긴 요약 (동사형 종결 어미), kobart-base모델
  - **Bullet** : 짧은 요약 (명사형 종결 어미), kobart-base모델
  - Extractive : 핵심 문장 추출, 모델 brainbert모델

Augmentation

- **Example 1**

question : 처음으로 부실 경영인에 대한 보상 선고를 받은 회사는?
answer : 한보철강
context :
순천여자고등학교 졸업, 1973년 이화여자대학교를 졸업하고 1975년 제17회 사법시험에 합격하여 판사로 임용되었고 대법원 재판연구관, 수원지법 부장판사, 사법연수원 교수, 특허법원 부장판사 등을 거쳐 능력을 인정받았다. 2003년 최종영 대법원장의 지명으로 헌법재판소 재판관을 역임하였다.₩n₩n경제민주화위원회(위원장 장하성이 소액주주들을 대표해 한보철강 부실대출에 책임이 있는 이철수 전 제일은행장 등 임원 4명을 상대로 제기한 손해배상청구소송에서 서울지방법원 민사합의17부는 1998년 7월 24일에 "😄한보철강😄에 부실 대출하여 은행에 막대한 손해를 끼친 점이 인정된다"며 "원고가 배상을 청구한 400억원 전액을 은행에 배상하라"고 하면서 부실 경영인에 대한 최초의 배상 판결을 했다.₩n₩n2004년 10월 신행정수도의건설을위한특별조치법 위헌 확인 소송에서 9인의 재판관 중 유일하게 각하 견해를 내었다. 소수의견에서 전효숙 재판관은 다수견해의 문제점을 지적하면서 관습헌법 법리를 부정하였다. 전효숙 재판관은 서울대학교 근대법학교육 백주년 기념관에서 열린 강연에서, 국회가 고도의 정치적인 사안을 정치로 풀기보다는 헌법재판소에 무조건 맡겨서 해결하려는 자세는 헌법재판소에게 부담스럽다며 소회를 밝힌 바 있다.

model="abstractive"

max_len = 200

summary :
전효숙 재판관은 😄한보철강😄 부실대출에 책임이 있는 이철수 전 제일은행장 등 임원 4명을 상대로 제기한 손해배상청구소송에서 서울지방법원 민사합의17부는 1998년 7월 24일 "한보철강에 부실 대출하여 은행에 막대한 손해를 끼친 점이 인정된다"며 "원고가 배상을 청구한 400억원 전액을 은행에 배상하라"고 하면서 부실 경영인에 대한 최초의 배상 판결을 하였다.

model="bullet"

summary :
😄한보철강😄 부실대출 배상 판결 9인 재판관 중 유일하게 각하 전효숙 "관습헌법 부정"

work with Daewoong Kim

Augmentation

- **Example 2**

question : 박지훈은 1라운드에서 몇 순위를 차지했는가?
answer : 전체 4순위
context :
대전유천초등학교에서 야구를 시작할 당시에는 내야수였다. 한밭중학교 3학년 때 연습 경기에서 오버핸드 투수로 등판해 대량 실점을 했고, 감독의 지시에 의해 나머지 두 경기에서 사이드암으로 던져 모두 승리를 따낸 직후부터 사이드암 투수로 전향했다.₩n대전고등학교 3학년 때 B로소 팀의 에이스로 부각됐지만 구속이 느리다는 이유로 프로 구단에서 큰 관심을 보이지 않아 2008년 신인 드래프트에서 지명되지 못해 단국대학교에 진학했다. 진학 후 신장이 커져 구속을 140km대로 끌어올리는 데 성공하며 1학년 때부터 대학 최고 사이드암으로 손꼽혔다. 이후 동기인 박지훈과 함께 마운드의 쌍두마차로 많은 경기를 꾸준히 책임졌고, 2학년 때 국가대표로도 뽑혀 프로 선수들과 함께 2009년 야구 월드컵에 참가했다.₩n그러나 3학년 이후 구속이 더 이상 오르지 않았고, 단조로운 구종으로 인해 2010년 세계 대학 야구 선수권 대회에서 쿠바 타자들을 상대로 난타당했다. 4학년 때 1.82의 평균자책점을 기록했음에도 불구하고 비슷한 성적을 올렸던 동기 박지훈이 1라운드 😄전체 4순위😄로 지명된 데 반해 2차 8라운드 전체 69순위로 간신히 프로에 입성했다.

model="abstractive"

max_len = 200

summary :
대전유천초등학교에서 야구를 시작할 당시에는 내야수였지만, 대전유천초등학교 3학년 때 연습 경기에서 오버핸드 투수로 등판해 대량 실점을 했고, 감독의 지시에 의해 사이드암으로 던져 모두 승리를 따낸 직후부터 사이드암 투수로 전향했다.

model="bullet"

summary :
대전유천초 3학년 때 오버핸드 투수로 등판해 대량 실점 2010년 세계 대학 야구 선수권 대회에서 쿠바 타자들 상대로 난타 당해

work with Daewoong Kim

Augmentation

- **결과**

기준 성능
- EM: 45.417
- F1: 52.698

model="abstractive"

앞 : abstractive + 원본 `[EM : 42.083/ F1 :48.59 ]`

뒤 : 원본 + abstractive `[EM : 42.5  F1 : 50.792]`

model="bullet"

앞 : bullet + 원본 `[EM : 40.417 / F1 : 48.273 ]`

뒤 : 원본 + bullet `[EM : 45.833 / F1 : 52.933 ]` → 미세하지만 성능 오름👍

2. 육하원칙 / Question Generation / MRC

- **Train dataset**

  영재님께서 증강해주신 3,635개 데이터 추가

- **결과**

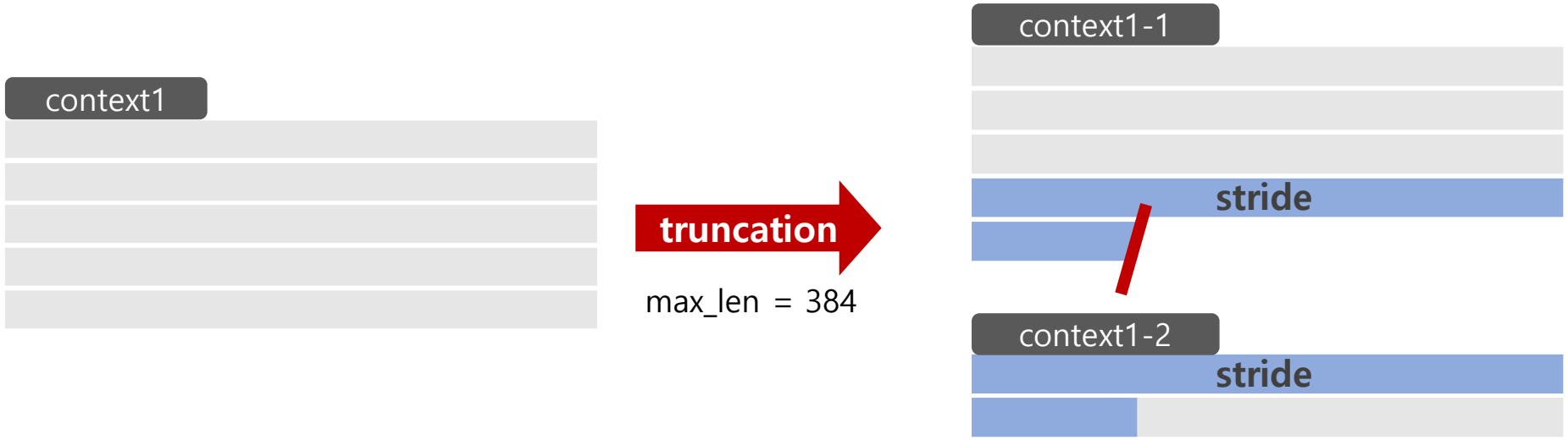
  [EM : 45 → 47 / F1 : 52 → 53] 👍

<추가 적용 대기>

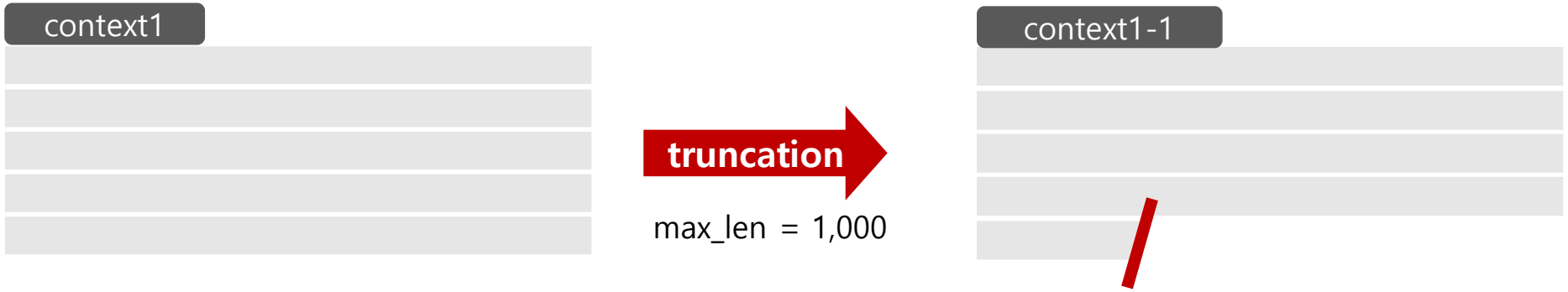
- question backtranslation
- context backtranslation
- punctuation
- underline embedding

- **Extractive Model과 비교**

Extractive Model



Generative Model



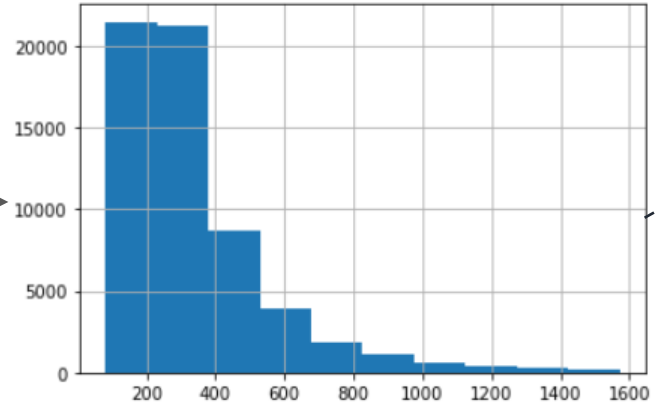
- **Generative Model Preprocessing**

context1-1

max_len = 1,000

정답 포함 → 변경사항 없음

정답 미포함 → answer= ""



<length of tokenized contexts>

메모리가 터지지 않으면서
대부분의 내용을 포함할 수 있는 길이

- **Generative Model Preprocessing**

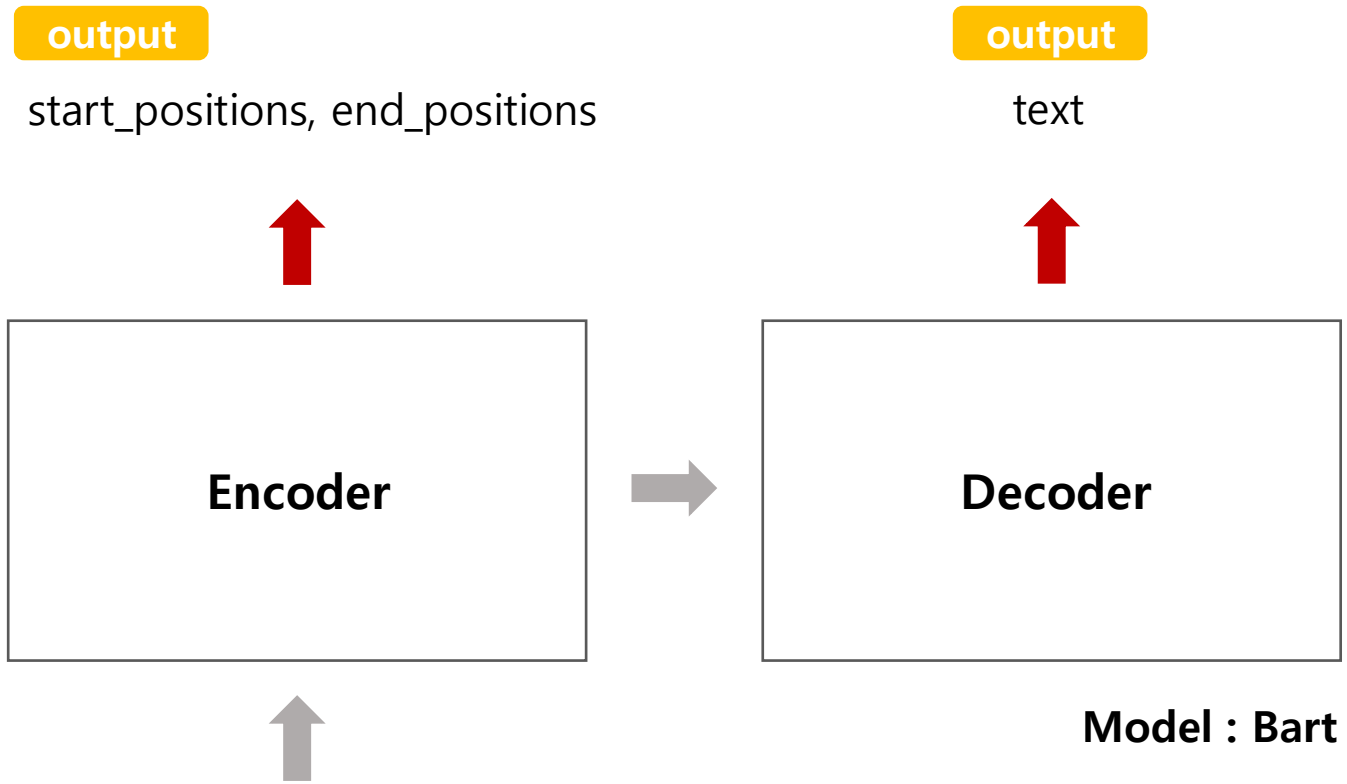
return_offsets_mapping과
return_overflowing_tokens 필요 없음

Answer Tokenizing

```
labels = [f"{answer['text'][0]} </s>" for answer in examples["answers"]]
with tokenizer.as_target_tokenizer():
    labels = tokenizer(
        labels,
        max_length=data_args.max_label_length,
        padding=data_args.pad_to_max_length,
        truncation=True,
    )["input_ids"]
return labels
```

Ensemble
Extractive and Generative Models

Model Architecture

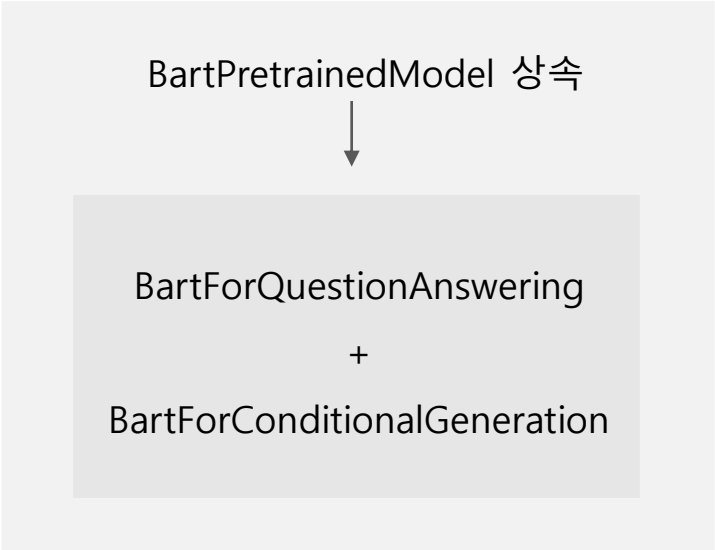


input

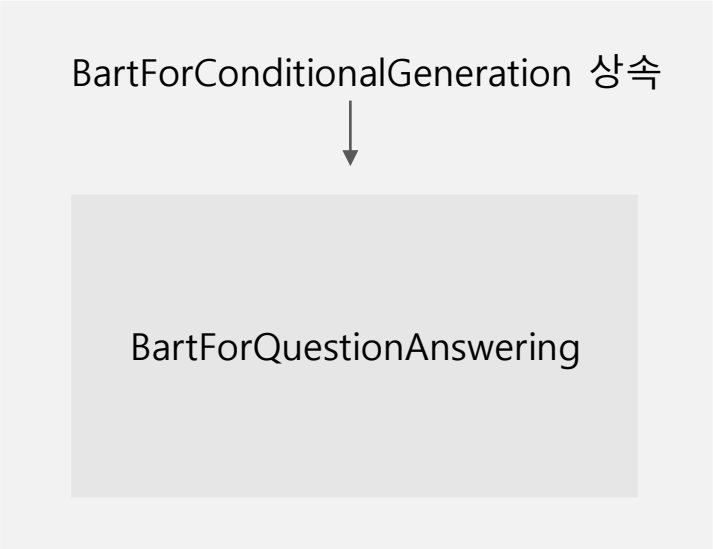
- question, context
- answer (start_positions, end_positions, text)

Model Architecture

```
class BartForExtractionGenerationEnsemble(BartPretrainedModel)
```




```
class BartForExtractionGenerationEnsemble(GC)
```



Extractive Model Loss

```python
sequence_output = outputs.encoder_last_hidden_state

logits = self.qa_outputs(sequence_output)
start_logits, end_logits = logits.split(1, dim=-1)
start_logits = start_logits.squeeze(-1).contiguous()
end_logits = end_logits.squeeze(-1).contiguous()
```

```python
total_loss = None
if start_positions is not None and end_positions is not None:
    if len(start_positions.size()) > 1:
        start_positions = start_positions.squeeze(-1)
    if len(end_positions.size()) > 1:
        end_positions = end_positions.squeeze(-1)
    ignored_index = start_logits.size(1)
    start_positions = start_positions.clamp(0, ignored_index)
    end_positions = end_positions.clamp(0, ignored_index)

    loss_fct = CrossEntropyLoss(ignore_index=ignored_index)
    start_loss = loss_fct(start_logits, start_positions)
    end_loss = loss_fct(end_logits, end_positions)
    total_loss = (start_loss + end_loss) / 2
```

Generative Model Loss

```python
lm_logits = self.lm_head(
    outputs.last_hidden_state) + self.final_logits_bias
```

```python
masked_lm_loss = None
if labels is not None:
    loss_fct = CrossEntropyLoss()
    masked_lm_loss = loss_fct(
        lm_logits.view(-1, self.config.vocab_size), labels.view(-1))
```

Loss

```python
if not return_dict:
    ext_output = (
        start_logits,
        end_logits,
    ) + outputs[1:]
    gen_output = (lm_logits,) + outputs[1:]
    return ((total_loss + masked_lm_loss,) + ext_output + gen_output)
```

Preprocessing

```
output = tokenizer(
    [f"질문: {q} 지문: {c} </s>" for q, c in zip(examples["question"], examples["context"])],
    max_length=data_args.max_seq_length,
    padding=data_args.pad_to_max_length,
    truncation=True,
)
```

Generative

cls_token sep_token 직접 붙여줍니다!

Generative + Extractive

```
output = tokenizer(
    [f"<s> 질문: {q} 지문: </s>" for q in examples["question"]],
    [f"{c} </s>" for c in examples["context"]],
    max_length=data_args.max_seq_length,
    padding=data_args.pad_to_max_length,
    return_offsets_mapping=True,
    truncation=True,
)
```

return_offsets_mapping=True
➔ start_positions, end_positions mapping을 위해서!

Bert (or Roberta)

```
tokenizer.decode(tokenized_examples.input_ids[0][:18])
```

```
'[CLS] 대통령을 포함한 미국의 행정부 견제권을 갖는 국가 기관은? [SEP]'
```

```
offset_mapping[0][:18]
```

```
[(0, 0),
 (0, 3),
 (3, 4),
 (5, 7),
 (7, 8),
 (9, 11),
 (11, 12),
 (13, 16),
 (17, 19),
 (19, 20),
 (20, 21),
 (22, 23),
 (23, 24),
 (25, 27),
 (28, 30),
 (30, 31),
 (31, 32),
 (0, 0)]
```

Bart

```
tokenizer.decode(tokenized_examples.input_ids[0][:13])
```

```
'<s>대통령을 포함한 미국의 행정부 견제권을 갖는 국가 기관은?</s>'
```

```
offset_mapping[0][:14]
```

```
[(0, 3),
 (3, 6),
 (6, 7),
 (7, 11),
 (11, 15),
 (15, 19),
 (19, 22),
 (22, 24),
 (24, 27),
 (27, 30),
 (30, 34),
 (34, 35),
 (35, 39),
 (0, 2)]
```

➔ start_positions, end_positions 을 찾을 때 문제 발생!

Preprocessing

※ start_positions, end_positions 을 찾을 때 문제 발생하는 경우 예시

```
if not (
    offsets[token_start_index][0] <= start_char and
    offsets[token_end_index][1] >= end_char
):
    tokenized_examples["start_positions"].append(cls_index)
    tokenized_examples["end_positions"].append(cls_index)
else:
    while (
        token_start_index < len(offsets) and
        offsets[token_start_index][0] <= start_char
    ):
        token_start_index += 1
    tokenized_examples["start_positions"].append(
        token_start_index - 1)

    while offsets[token_end_index][1] >= end_char:
        token_end_index -= 1
    tokenized_examples["end_positions"].append(
        token_end_index + 1)
```

'title': '삼성 애니콜',
'context': '애플의 아이폰을 도입한 이후로 KT와 삼성간의 불화가 일기 시작하였다. … 후략'
'question': "KT가 '쇼옴니아' 명칭을 사용하지 못하는 계기가 된 핸드폰은 어느 회사 제품인가?",
'answers': {'answer_start': [0], 'text': ['애플']}

만약 answer의 위치가 context 맨 앞이라면,
end_position 을 찾을 때 while 문을 빠져나오지 못한다.

Preprocessing

```python
for i, offsets in enumerate(offset_mapping):
    input_ids = tokenized_examples["input_ids"][i]
    cls_index = input_ids.index(tokenizer.cls_token_id)  # cls index
    sep_index = input_ids.index(tokenizer.sep_token_id)  # sep index

    offsets[cls_index] = (0, 0)  # 필수!
    offsets[sep_index] = (0, 0)  # 필수!

    sequence_ids = tokenized_examples.sequence_ids(i)
    answers = examples[ANSWER_COLUMN_NAME][i]
    context_index = 0 if pad_on_right else 1

    if len(answers["answer_start"]) == 0:
        tokenized_examples["start_positions"].append(cls_index)
        tokenized_examples["end_positions"].append(cls_index)
    else:
        start_char = answers["answer_start"][0]
        end_char = start_char + len(answers["text"][0])

        token_start_index = 0
        while sequence_ids[token_start_index] == context_index:
            token_start_index += 1

        token_end_index = len(input_ids) - 1
        while sequence_ids[token_end_index] == context_index or input_ids[token_end_index] == tokenizer.pad_token_id:
            token_end_index -= 1
```

Preprocessing

tokens

[CLS] 대통령 ##을 포함 ##한 미국 ##의 행정부 견제 ##권 ##을 갖 ##는 국가 기관 ##은 ? [SEP] 미국 상의 ##원 또는 미국 상원 (United State ##s Se ##n ##ate) 은 양 ##원 ##제 ##인 미국 의회 ##의 상원 ##이다 . ... [PAD] [PAD]

sequence_ids

[None, 0, 0, 0, ..., 0, None, 1, 1, 1, ..., 1, None, None, ..., None]

---

tokens

<s> 대통령 ##을 포함 ##한 미국 ##의 행정부 견제 ##권 ##을 갖 ##는 국가 기관 ##은 ? </s> 미국 상의 ##원 또는 미국 상원 (United State ##s Se ##n ##ate) 은 양 ##원 ##제 ##인 미국 의회 ##의 상원 ##이다 . ... <pad> <pad>

sequence_ids

[0, 0, 0, 0, ..., 0, 0, 1, 1, 1, ..., 1, None, None, ..., None]

or

[0, 0, 0, 0, ..., 0, 0, None, None, None, ..., None, None, None, ..., None]

Preprocessing

```python
for i, offsets in enumerate(offset_mapping):
    input_ids = tokenized_examples["input_ids"][i]
    cls_index = input_ids.index(tokenizer.cls_token_id)  # cls index
    sep_index = input_ids.index(tokenizer.sep_token_id)  # sep index

    offsets[cls_index] = (0, 0)  # 필수!
    offsets[sep_index] = (0, 0)  # 필수!

    sequence_ids = tokenized_examples.sequence_ids(i)
    answers = examples[ANSWER_COLUMN_NAME][i]
    context_index = 0 if pad_on_right else 1

    if len(answers["answer_start"]) == 0:
        tokenized_examples["start_positions"].append(cls_index)
        tokenized_examples["end_positions"].append(cls_index)
    else:
        start_char = answers["answer_start"][0]
        end_char = start_char + len(answers["text"][0])

        token_start_index = 0
        while sequence_ids[token_start_index] == context_index:
            token_start_index += 1

        token_end_index = len(input_ids) - 1
        while sequence_ids[token_end_index] == context_index or input_ids[token_end_index] == tokenizer.pad_token_id:
            token_end_index -= 1
```

Seq2SeqTrainer (Transformers)

```python
class Seq2SeqTrainer(Trainer):
    def evaluate(
        self,
        eval_dataset: Optional[Dataset] = None,
        ignore_keys: Optional[List[str]] = None,
        metric_key_prefix: str = "eval",
        max_length: Optional[int] = None,
        num_beams: Optional[int] = None,
    ) -> Dict[str, float]:
        self._max_length = max_length if max_length is not None else self.args.generation_max_length
        self._num_beams = num_beams if num_beams is not None else self.args.generation_num_beams
        return super().evaluate(eval_dataset, ignore_keys=ignore_keys, metric_key_prefix=metric_key_prefix)

    def predict(
        self,
        test_dataset: Dataset,
        ignore_keys: Optional[List[str]] = None,
        metric_key_prefix: str = "eval",
        max_length: Optional[int] = None,
        num_beams: Optional[int] = None,
    ) -> PredictionOutput:
        self._max_length = max_length if max_length is not None else self.args.generation_max_length
        self._num_beams = num_beams if num_beams is not None else self.args.generation_num_beams
        return super().predict(test_dataset, ignore_keys=ignore_keys, metric_key_prefix=metric_key_prefix)
```

➔ Seq2SeqTrainer 는 Trainer 에서 일부 변수 셋팅이 추가되었다!

QuestionAnsweringSeq2SeqTrainer (Ours)

```python
class QuestionAnsweringSeq2SeqTrainer(Seq2SeqBaseTrainer):
    def evaluate(
        self,
        eval_dataset: Optional[datasets.Dataset] = None,
        eval_examples: Optional[datasets.Dataset] = None,
        ignore_keys: Optional[List[str]] = None,
        metric_key_prefix: str = "eval",
        mode: str = "evaluate",
        max_length: Optional[int] = None,
        num_beams: Optional[int] = None,
    ) -> Dict[str, float]:
        self._max_length = max_length if max_length is not None else self.args.generation_max_length
        self._num_beams = num_beams if num_beams is not None else self.args.generation_num_beams
        ...
    def predict(
        self,
        test_dataset: datasets.Dataset,
        test_examples: datasets.Dataset,
        ignore_keys: Optional[List[str]] = None,
        metric_key_prefix: str = "test",
        mode: str = "test",
        max_length: Optional[int] = None,
        num_beams: Optional[int] = None,
    ) -> PredictionOutput:
        self._max_length = max_length if max_length is not None else self.args.generation_max_length
        self._num_beams = num_beams if num_beams is not None else self.args.generation_num_beams
        ...
```

➔ 따라서, QuestionAnswering 을 위해 evaluate, predict를 오버라이드 할 때도 이 부분을 추가해준다.

Trainer

```python
class QuestionAnsweringEnsembleTrainer(QuestionAnsweringSeq2SeqTrainer):
    def __init__(
        self,
        *args,
        eval_examples: datasets.Dataset = None,
        post_process_function: Callable = None,
        **kwargs
    ):
        super().__init__(*args, eval_examples=eval_examples,
                         post_process_function=post_process_function, **kwargs)
        self.label_names = ["start_positions", "end_positions", "labels"]
```

➔ Trainer의 self.label_names 를 우리가 원하는 형식에 맞게 세 개로 수정!

```python
class Trainer:
    def __init__(
        self,
        model: Union[PreTrainedModel, nn.Module] = None,
        args: TrainingArguments = None,
        data_collator: Optional[DataCollator] = None,
        train_dataset: Optional[Dataset] = None,
        eval_dataset: Optional[Dataset] = None,
        tokenizer: Optional[PreTrainedTokenizerBase] = None,
        model_init: Callable[[], PreTrainedModel] = None,
        compute_metrics: Optional[Callable[[EvalPrediction], Dict]] = None,
        callbacks: Optional[List[TrainerCallback]] = None,
        optimizers: Tuple[torch.optim.Optimizer, torch.optim.lr_scheduler.LambdaLR] = (None, None),
    ):
        ...
        default_label_names = (
            ["start_positions", "end_positions"]
            if type(self.model).__name__ in MODEL_FOR_QUESTION_ANSWERING_MAPPING_NAMES.values()
            else ["labels"]
        )
        self.label_names = default_label_names if self.args.label_names is None else self.args.label_names
        ...
```

QuestionAnsweringEnsembleTrainer (Ours)

× Add a condition for checking labels (#14211)

Browse files

master (#14211)

hrxorxm committed 8 days ago Verified

1 parent b338596 commit e823d8198a2689255a24815ae69e78dfc5f9719e

Showing 1 changed file with 6 additions and 3 deletions.

Split Unified

9 src/transformers/trainer_seq2seq.py

```diff
@@ -196,9 +196,12 @@ def prediction_step(
196 196         if self.args.prediction_loss_only:
197 197             return (loss, None, None)
198 198
199     -       labels = inputs["labels"]
200     -       if labels.shape[-1] < gen_kwargs["max_length"]:
201     -           labels = self._pad_tensors_to_max_len(labels, gen_kwargs["max_length"])
    199 +       if has_labels:
    200 +           labels = inputs["labels"]
    201 +           if labels.shape[-1] < gen_kwargs["max_length"]:
    202 +               labels = self._pad_tensors_to_max_len(labels, gen_kwargs["max_length"])
    203 +       else:
    204 +           labels = None
202 205
203 206         return (loss, generated_tokens, labels)
204 207
```

➔ labels(정답)을 주지 않고도 generation을 해야 하는데, 원래 Seq2Seq prediction_step 코드에서도 이런 점이 반영이 안되어 있어서 수정 후 PR!

Cross Attention

논문 : Attention-guided Generative Models for Extractive Question Answering
링크 : https://arxiv.org/abs/2110.06393

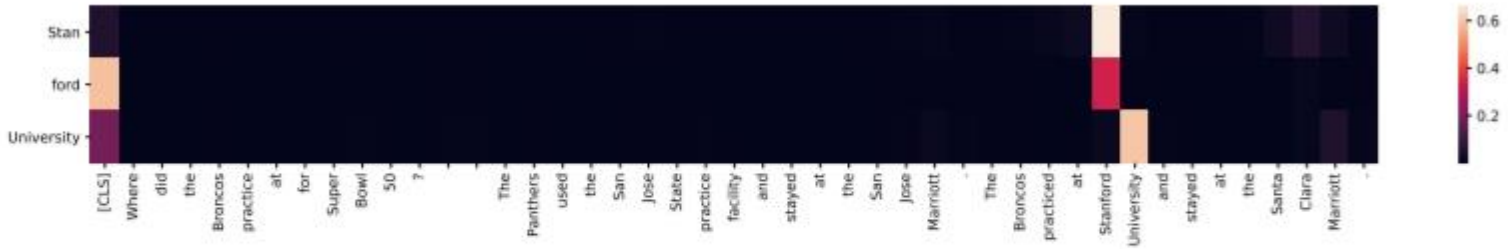


Figure 1: Average cross-attention between generated tokens and input context tokens on an example from SQuAD-v1.1 dataset. Each row and column corresponds to a decoder generated token and an input token, respectively. The generated token sequence ['Stan', 'ford', 'University'] attends to the input tokens ['Stanford', 'University'].

➔ generation 된 token 들과, context 내에 있는 answer span tokens 의 cross-attention 값을 확인해보니 높다!

Cross Attention

논문 : Attention-guided Generative Models for Extractive Question Answering
링크 : https://arxiv.org/abs/2110.06393

$$\Pr[start = i] = CrossAttn(\hat{y}_1, x_i),$$
$$\Pr[end = i] = CrossAttn(\hat{y}_t, x_i).$$

➔ start position 이 i 일 확률과 end position이 i 일 확률을 Cross Attention을 통해 구한 후, 실제 정답 위치와의 차이를 이용한 loss 를 추가하자!

Generative loss (기존)

$$\ell_{gen}(q, C, a) = -\sum_{i=1}^{t} \log \Pr\left(y_i | y_{1\ldots,i-1}; q, C\right)$$

Extractive loss (추가!)

$$\ell_{span}(q, C, start, end) = CE(start, CrossAttn(y_1, C)) + CE(end, CrossAttn(y_t, C)),$$

Final joint training loss

$$\ell_{joint}(q, C, a) = (1 - \lambda)\ell_{gen} + \lambda\ell_{span},$$