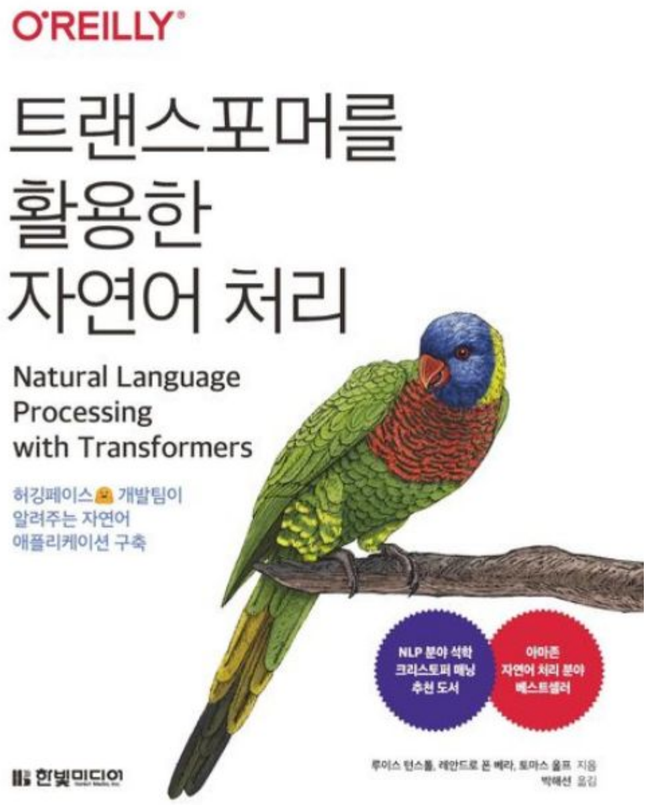
트랜스포머를 활용한 자연어처리

10장: 대규모 데이터셋 수집하기



소속
HUMANE랩

발표자
이다현

발표일시
2024년 08월 23일

24년 하계방학
스터디

목차

대규모 데이터 수집

- 대규모 말뭉치 구축의 어려움
  - 대규모 데이터셋은 대부분 고도의 자동화로 만들어짐
    - 이에 따른 편향, 저작권 위반, 위험한 콘텐츠 등의 문제가 있음
  - 말뭉치의 품질이 사전 훈련 모델의 품질에 영향을 미침
  - 말뭉치가 편향된 데이터라면 모델의 편향으로 이어짐
- 사용자 정의 코드 데이터셋 만들기
  - 깃허브 REST API 사용
  - 공개 데이터셋 이용 (예: 구글 빅쿼리)
  - 스트리밍 이용

- GPT와 GPT2가 생성한 문장 간의 비교

```
GPT completions:
1.
When they came back.
 " we need all we can get, " jason said once they had settled into the back of
the truck without anyone stopping them. " after getting out here, it 'll be up
to us what to find. for now
2.
When they came back.
 his gaze swept over her body. he 'd dressed her, too, in the borrowed clothes
that she 'd worn for the journey.
 " i thought it would be easier to just leave you there. " a woman like
3.
When they came back to the house and she was sitting there with the little boy.
 " don't be afraid, " he told her. she nodded slowly, her eyes wide. she was so
lost in whatever she discovered that tom knew her mistake

GPT-2 completions:
1.
When they came back we had a big dinner and the other guys went to see what
their opinion was on her. I did an hour and they were happy with it.
2.
When they came back to this island there had been another massacre, but he could
not help but feel pity for the helpless victim who had been left to die, and
that they had failed that day. And so was very, very grateful indeed.
3.
When they came back to our house after the morning, I asked if she was sure. She
said, "Nope." The two kids were gone that morning. I thought they were back to
being a good friend.

When Dost
```

토크나이저 구축하기

- 데이터셋에 맞는 최적의 토크나이저를 얻으려면 토크나이저를 직접 훈련해야 함
- 여기서는 파이썬 코드를 위한 토크나이저가 필요
  - 공백 기반 토크나이저를 사용하면 공백이 중요한 파이썬 코드의 특징을 파악하기 힘듦
  - 따라서 바이트 기반 토크나이저가 필요함
  - 일반 텍스트에서 훈련된 토크나이저는 코드의 "들여쓰기"를 인식하기 어려움
- 따라서 토크나이저를 재훈련 해야함

In [ ]:

```
from transformers import AutoTokenizer

python_code = r"""def say_hello():
    print("Hello, World!")
# Print it
say_hello()
"""
tokenizer = AutoTokenizer.from_pretrained("gpt2")
print(tokenizer(python_code).tokens())
```

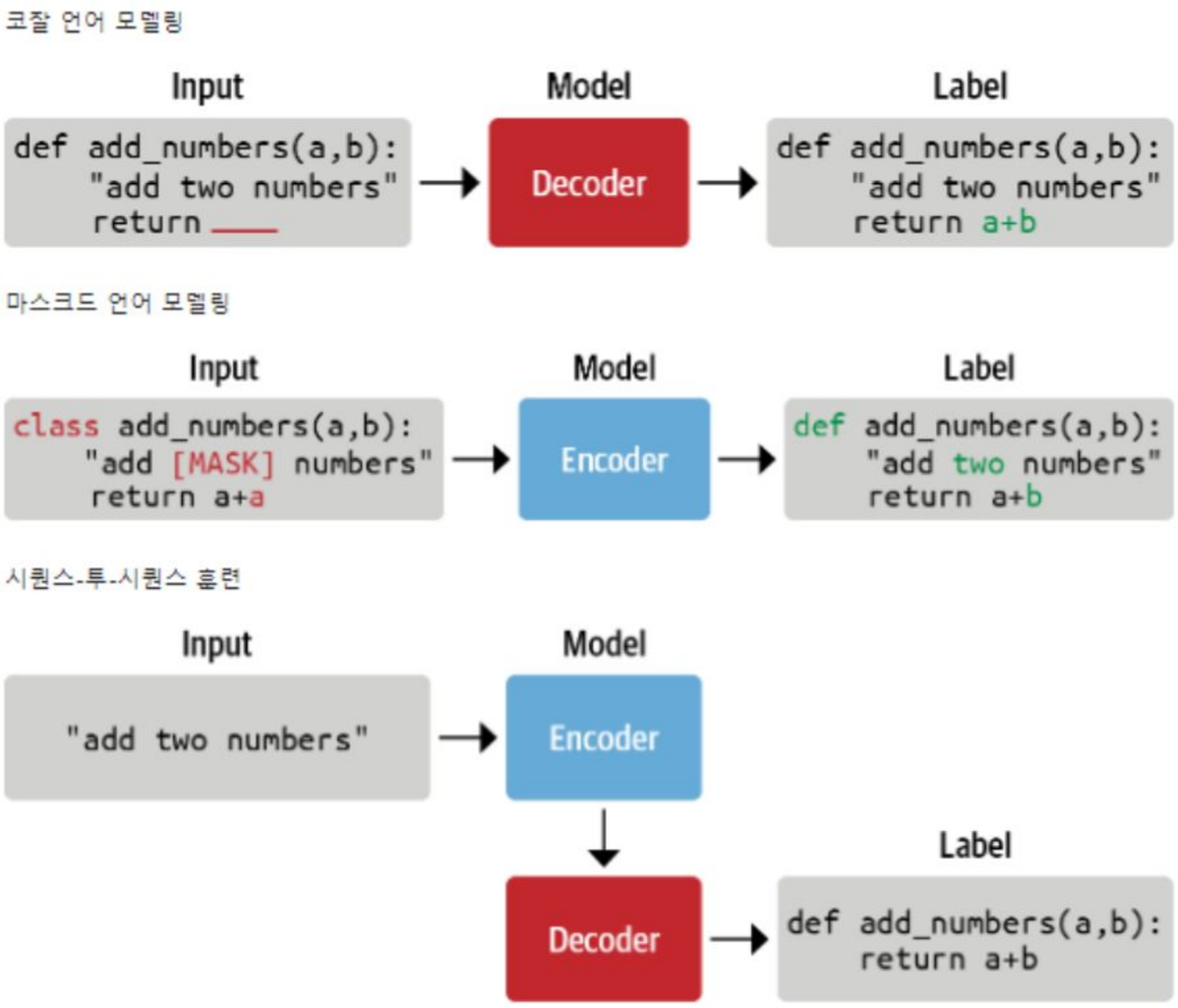
```
['def', 'ĠsayĠ', '_', 'hello', '():', 'Ċ', 'Ġ', 'Ġ', 'Ġ', 'Ġprint', '("', 
'Hello', ',', 'ĠWorld', '!"', ')', 'Ġ#', 'ĠPrint', 'Ġit', 'Ċ', 'Ċ', 'say', '_', 
'hello', '()', 'Ċ']
```

토크나이저 훈련하기

- 목표 어휘사전의 크기를 지정
- 토크나이저 모델을 훈련하기 위해 입력 문자열을 공급할 iterator 준비
- Train_new_frim_iterator() 매서드 호출

```
from tqdm.auto import tqdm

length = 100000
dataset_name = 'transformersbook/codeparrot-train'
dataset = load_dataset(dataset_name, split="train", streaming=True)
iter_dataset = iter(dataset)

def batch_iterator(batch_size=10):
    for _ in tqdm(range(0, length, batch_size)):
        yield [next(iter_dataset)['content'] for _ in range(batch_size)]

new_tokenizer = tokenizer.train_new_from_iterator(batch_iterator(),
                                                  vocab_size=12500,
                                                  initial_alphabet=base_vocab)
```

밑바닥부터 모델 훈련하기



- Casual LM: GPT 계열
  - 문장의 다음 단어를 예측하는 방식으로 훈련
- Masked LM: Bert 계열
  - 문장 중 일부 단어를 마스킹한 뒤 그 단어를 예측하는 방식으로 훈련
- Encoder-Docoder: T5
  - 입력 시퀀스를 받아서 대응하는 출력 시퀀스를 생성하는 방식

밑바닥부터 모델 훈련하기

- 모델 초기화

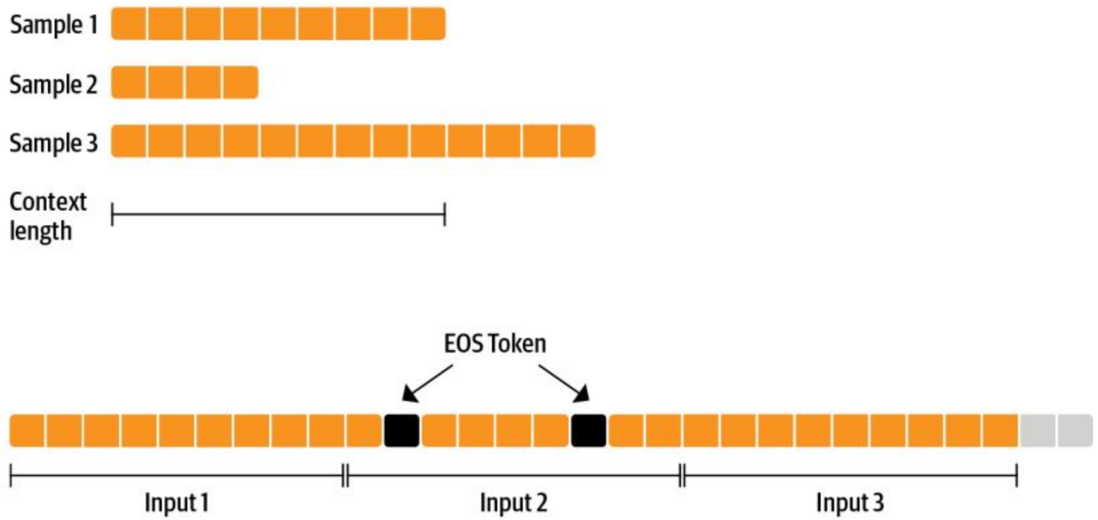
```
from transformers import AutoConfig, AutoModelForCausalLM, AutoTokenizer

tokenizer = AutoTokenizer.from_pretrained(org + "/" + model_ckpt)
config = AutoConfig.from_pretrained("gpt2-xl", vocab_size=len(tokenizer))
model = AutoModelForCausalLM.from_config(config)
```

```
print(f'GPT-2 (xl) 크기: {model_size(model)/1000**2:.1f}M parameters')
```

```
GPT-2 (xl) size: 1529.6M parameters
```

밑바닥부터 모델 훈련하기

- 데이터로더 구축
  - 문맥 크기를 꽉 채운 시퀀스를 모델에 제공하기 위해 여러 샘플을 토큰화 후 EOS 토큰으로 연결




```
examples, total_characters, total_tokens = 500, 0, 0
dataset = load_dataset('transformersbook/codeparrot-train', split='train',
                       streaming=True)

for _, example in tqdm(zip(range(examples), iter(dataset)), total=examples):
    total_characters += len(example['content'])
    total_tokens += len(tokenizer(example['content']).tokens())

characters_per_token = total_characters / total_tokens
```

밑바닥부터 모델 훈련하기

- 데이터로더 구축

```python
import torch
from torch.utils.data import IterableDataset

class ConstantLengthDataset(IterableDataset):

    def __init__(self, tokenizer, dataset, seq_length=1024,
                 num_of_sequences=1024, chars_per_token=3.6):
        self.tokenizer = tokenizer
        self.concat_token_id = tokenizer.eos_token_id
        self.dataset = dataset
        self.seq_length = seq_length
        self.input_characters = seq_length * chars_per_token * num_of_sequences
```

```python
def __iter__(self):
    iterator = iter(self.dataset)
    more_examples = True
    while more_examples:
        buffer, buffer_len = [], 0
        while True:
            if buffer_len >= self.input_characters:
                m=f"Buffer full: {buffer_len}>={self.input_characters:.0f}"
                print(m)
                break
            try:
                m=f"Fill buffer: {buffer_len}<{self.input_characters:.0f}"
                print(m)
                buffer.append(next(iterator)["content"])
                buffer_len += len(buffer[-1])
            except StopIteration:
                iterator = iter(self.dataset)

        all_token_ids = []
        tokenized_inputs = self.tokenizer(buffer, truncation=False)
        for tokenized_input in tokenized_inputs['input_ids']:
            all_token_ids.extend(tokenized_input + [self.concat_token_id])

        for i in range(0, len(all_token_ids), self.seq_length):
            input_ids = all_token_ids[i : i + self.seq_length]
            if len(input_ids) == self.seq_length:
                yield torch.tensor(input_ids)
```

밑바닥부터 모델 훈련하기

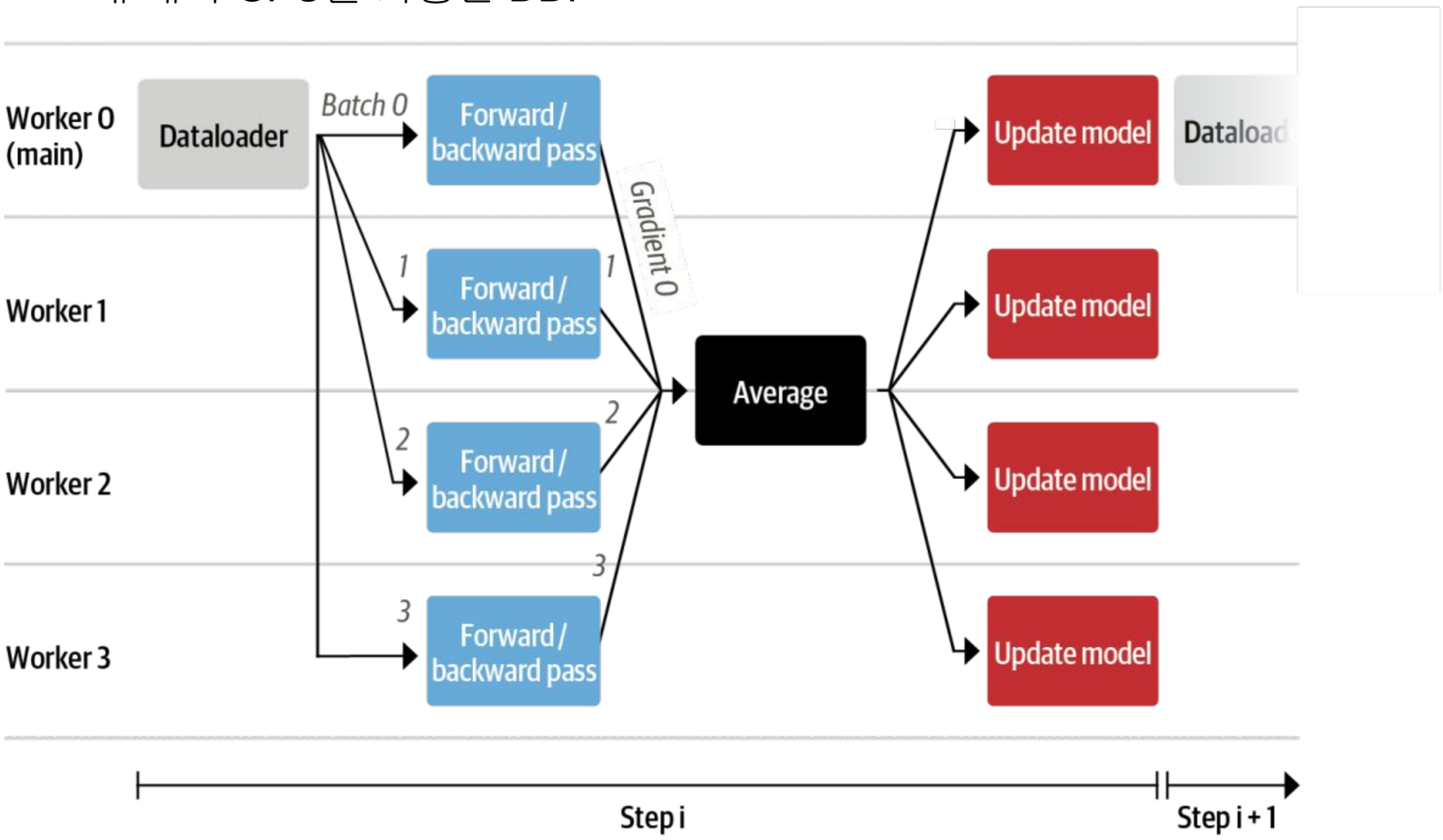
- 데이터로더 구축

```
shuffled_dataset = dataset.shuffle(buffer_size=100)
constant_length_dataset = ConstantLengthDataset(tokenizer, shuffled_dataset,
                                                num_of_sequences=10)
dataset_iterator = iter(constant_length_dataset)

lengths = [len(b) for _, b in zip(range(5), dataset_iterator)]
print(f"시퀀스 길이: {lengths}")
```

```
Fill buffer: 0<36864
Fill buffer: 3311<36864
Fill buffer: 9590<36864
Fill buffer: 22177<36864
Fill buffer: 25530<36864
Fill buffer: 31098<36864
Fill buffer: 32232<36864
Fill buffer: 33867<36864
Buffer full: 41172>=36864
Lengths of the sequences: [1024, 1024, 1024, 1024, 1024]
```

밑바닥부터 모델 훈련하기

- 훈련 루프 정의
  - 네 개의 GPU를 사용한 DDP



결과

- 사이킷런 모델도 만들 수 있을까?

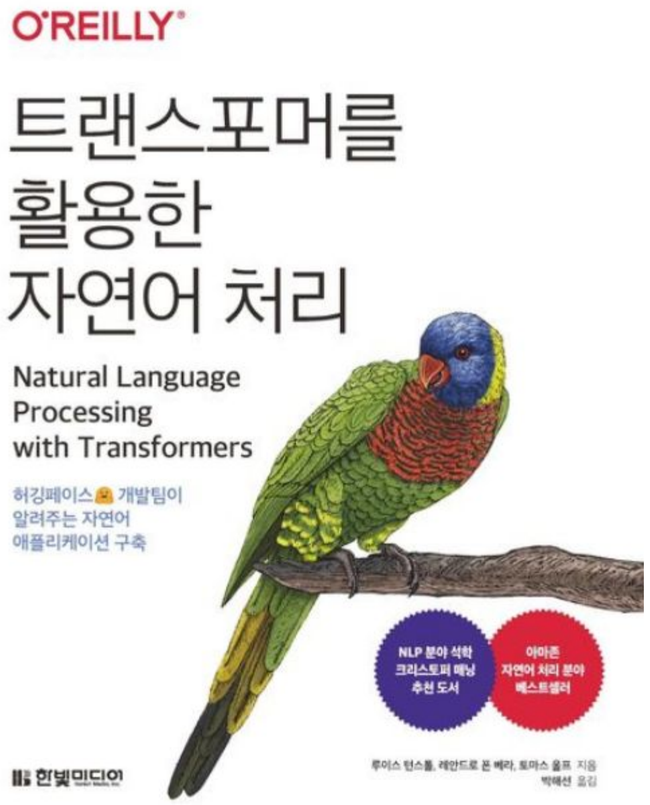
In [ ]:

```
prompt = '''X = np.random.randn(100, 100)
y = np.random.randint(0, 1, 100)

# fit random forest classifier with 20 estimators'''
complete_code(generation, prompt, max_length=96)
```

```
Setting `pad_token_id` to `eos_token_id`:0 for open-end generation.
reg = DummyRegressor()

forest = RandomForestClassifier(n_estimators=20)

forest.fit(X, y)
================================================================================

clf = ExtraTreesClassifier(n_estimators=100, max_features='sqrt')
clf.fit(X, y)
================================================================================

clf = RandomForestClassifier(n_estimators=20, n_jobs=n_jobs, random_state=1)
clf.fit(X, y)
================================================================================

clf = RandomForestClassifier(n_estimators=20)
clf.fit(X, y)
```

트랜스포머를 활용한 자연어처리

11장: 향후 방향



소속
HUMANE랩

발표자
이다현

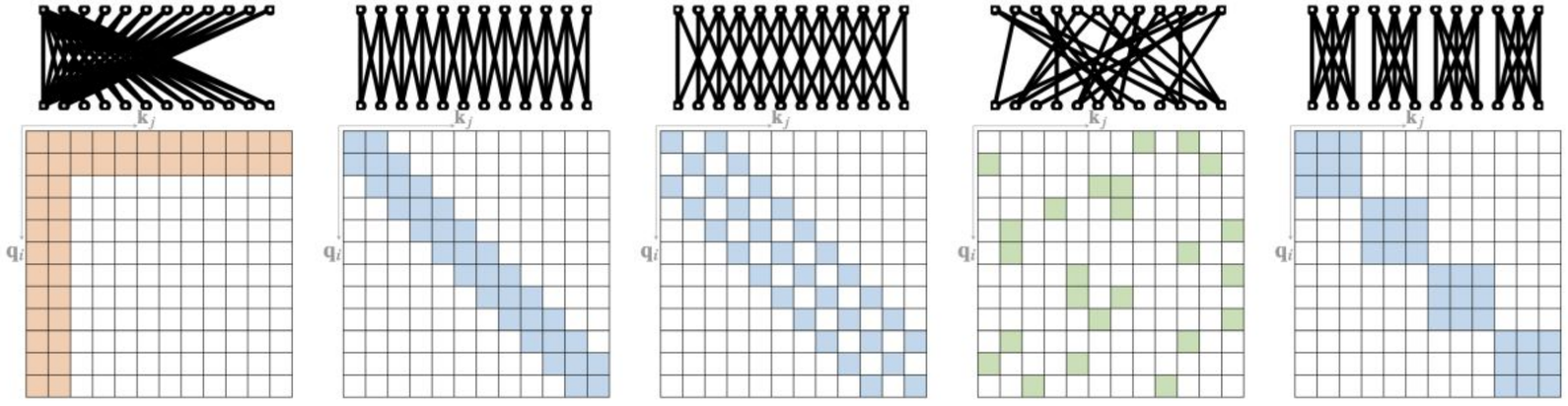
발표일시
2024년 08월 23일

24년 하계방학
스터디

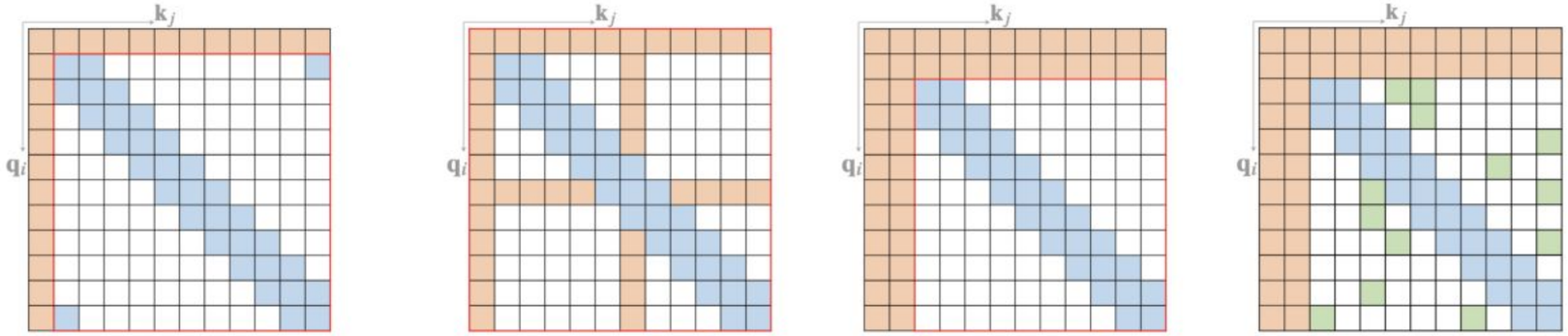
목차

어텐션 개선

- 계산을 줄이기 위한 희소 어텐션



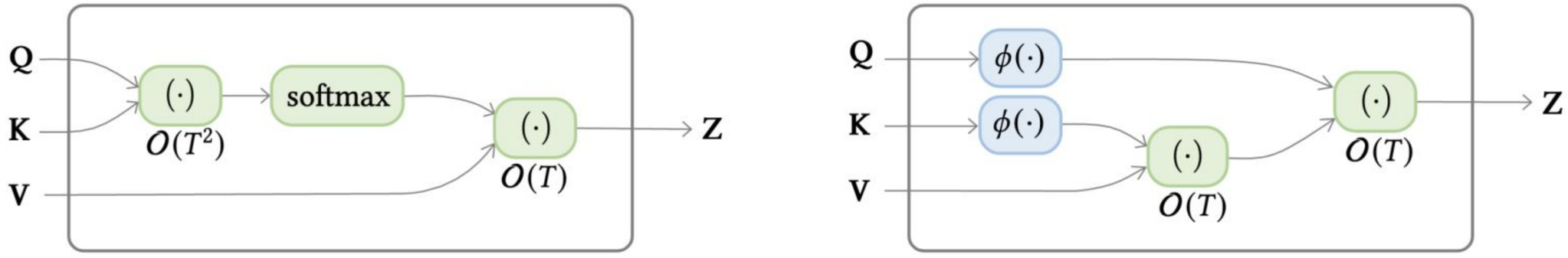
(a) global (b) band (c) dilated (d) random (e) block local



(a) Star-Transformer (b) Longformer (c) ETC (d) BigBird

어텐션 개선

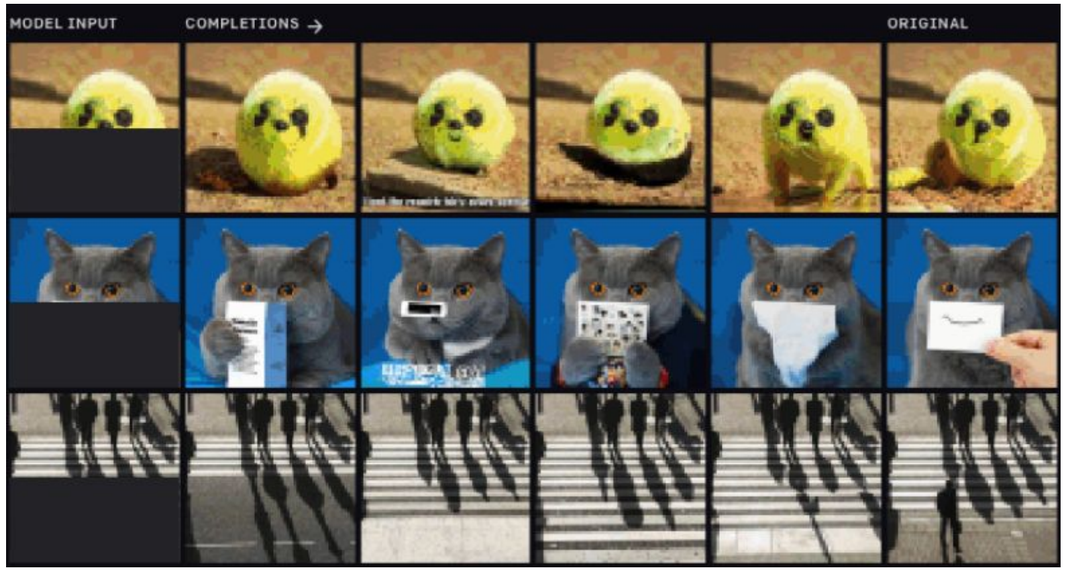
- 계산을 줄이기 위한 선형화된 셀프 어텐션



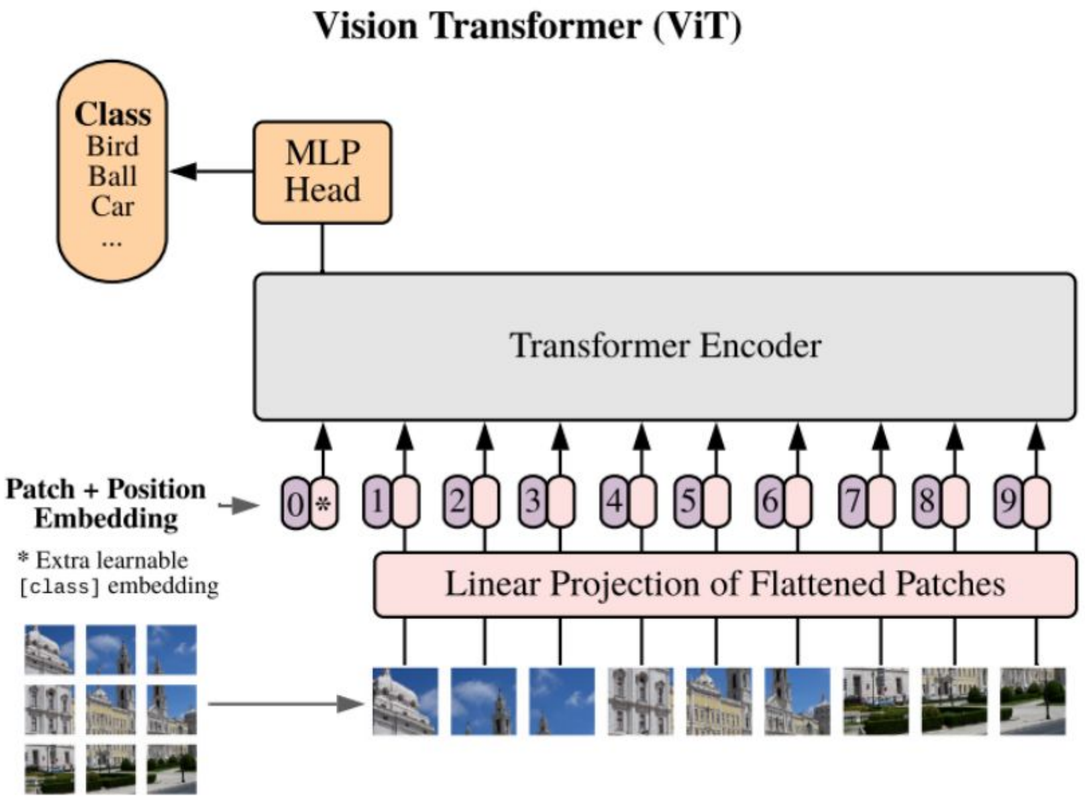
(a) standard self-attention

(b) linearized self-attention

텍스트를 넘어서 - 비전



- iGPT
  - [Chen et al. 2020] Generative Pretraining from Pixels 2020, ICML



- ViT
  - Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." arXiv preprint arXiv:2010.11929 (2020).

텍스트를 넘어서 - 테이블

Table

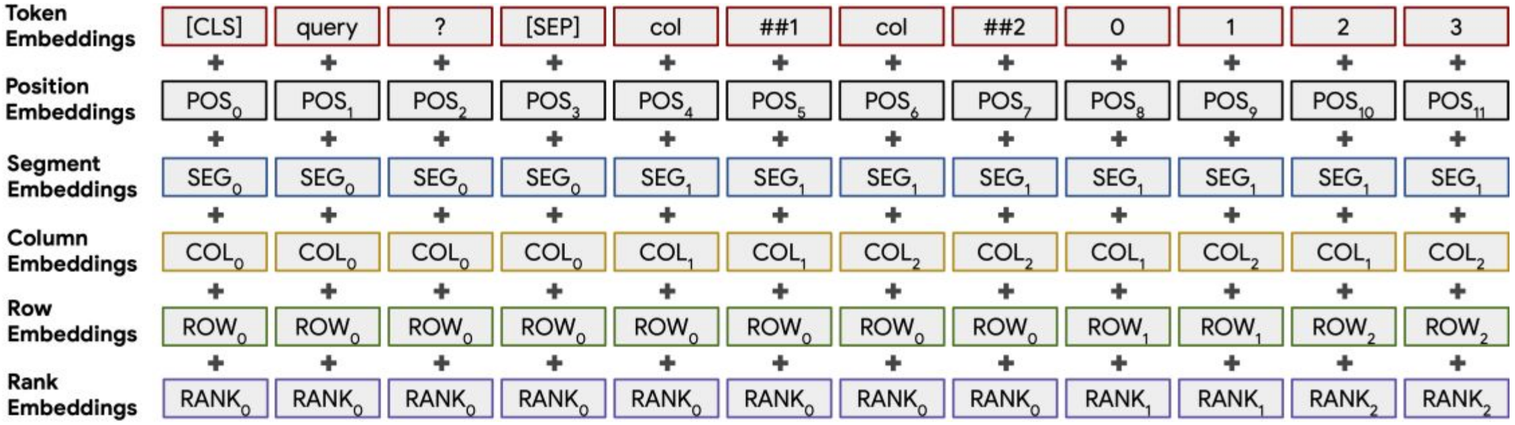
Rank	Name	No. of reigns	Combined days
1	Lou Thesz	3	3,749
2	Ric Flair	8	3,103
3	Harley Race	7	1,799
4	Dory Funk Jr.	1	1,563
5	Dan Severn	2	1,559
6	Gene Kiniski	1	1,131

Example questions

#	Question	Answer	Example Type
1	*Which wrestler had the most number of reigns?*	Ric Flair	Cell selection
2	*Average time as champion for top 2 wrestlers?*	AVG(3749,3103)=3426	Scalar answer
3	*How many world champions are there with only one reign?*	COUNT(Dory Funk Jr., Gene Kiniski)=2	Ambiguous answer
4	*What is the number of reigns for* Harley Race?	7	Ambiguous answer
5	*Which of the following wrestlers were ranked in the bottom 3?*	{Dory Funk Jr., Dan Severn, Gene Kiniski}	Cell selection
	Out of these, who had more than one reign?	Dan Severn	Cell selection

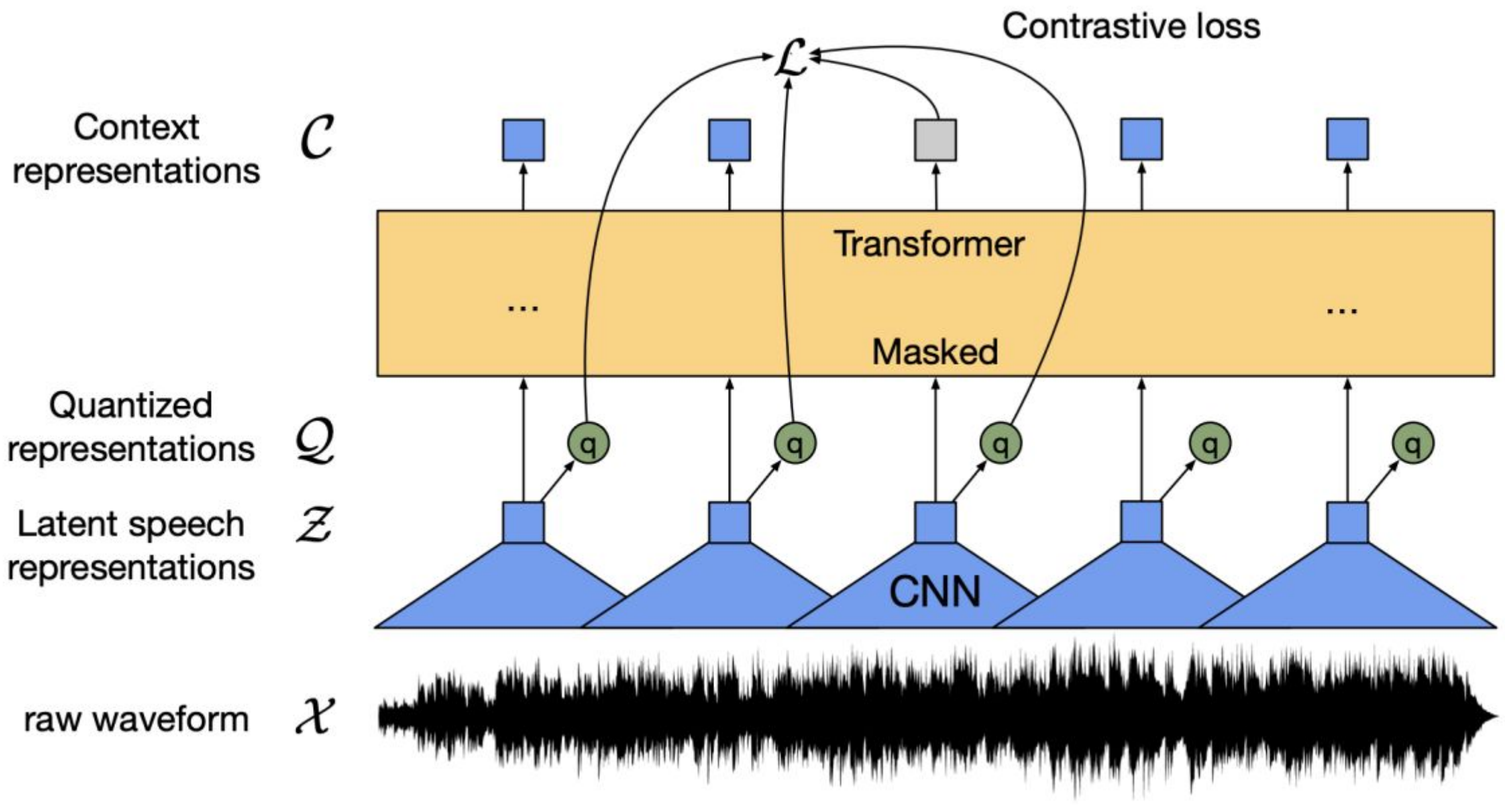
Table

col1	col2
0	1
2	3



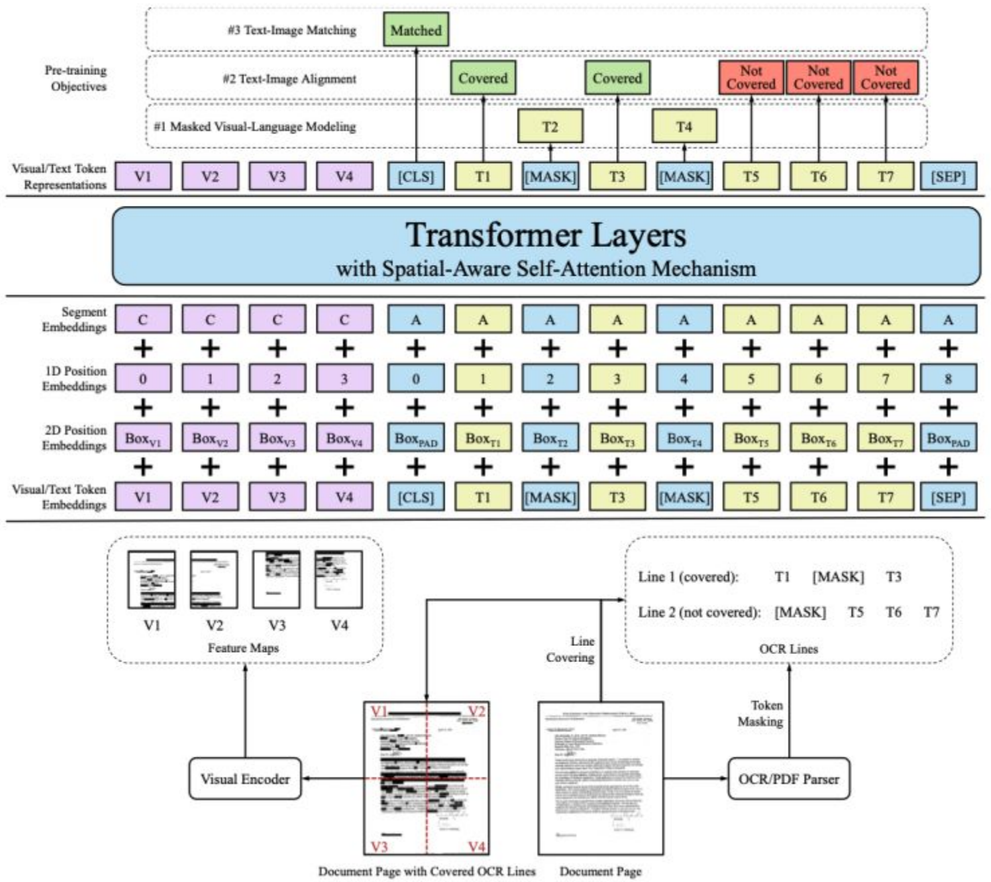
멀티모달 트랜스포머

- Speech-to-Text
  - wav2vec 2.0 아키텍처



멀티모달 트랜스포머

- LayoutLMv2모델 아키텍처와 사전 훈련 전략



멀티모달 트랜스포머

- CLIP