

What Do Position Embeddings Learn?

2025.05.26

민유안

Abstract / Introduction

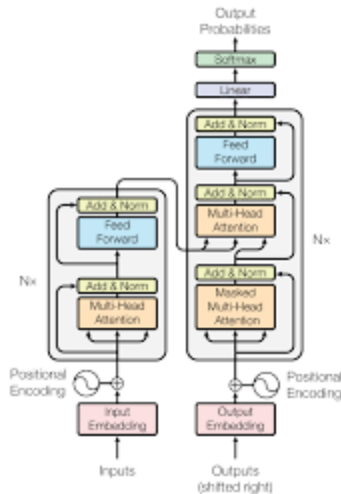
Background

기존 모델



단어의 순서를 모델링하기 위해 **순환 신경망 (RNN)** 활용
-> Time-dependency으로 인해 병렬화 불가

최신 모델





순서가 있는 시퀀스를 저장하는 대신,
Position Embedding을 통해 위치 정보를 활용

-> Time-dependency를 제거하여 병렬화 가능

Abstract / Introduction

- **Pre-trained Transformer language models**

- BERT (Devlin et al., 2018)  Transformer Encoder model
- RoBERTa (Liu et al., 2019)
- GPT-2 (Radford et al., 2019)  Transformer Decoder model

위의 3가지 Pre-trained Transformer language model과 기존 Transformer의 sinusoid 방식 Position Embedding을 대상으로 연구

- ✓ **About**

1 Position embedding이 위치의 의미를 학습하는가

2 다르게 학습된 Position embedding이 Transformer에 어떤 영향을 미치는가

Related Work

- Position Embedding 방식

1 Transformer: Pre-defined sinusoidal function 방식

2 Shaw et al. (2018)

: attention level에서의 Relative position representation을 제안

3 Dai et al. (2019)

: segment-level recurrence mechanism 적용, adaptive version of relative position embedding 사용

4 Wang et al. (2019)

: 임베딩 공간을 실수에서 복소수로 확장

새로운 learnable position embedding mapping 제안

Position Embedding Analysis

1 Sinusoid position embedding

$$PE_{(i,2j)} = \sin(i/10000^{2j/d_{model}}),$$
$$PE_{(i,2j+1)} = \cos(i/10000^{2j/d_{model}}),$$

pre-defined position embedding

Fixed, absolute

사전에 정의된 사인파 함수 바탕으로 위치 벡터 생성

2 Transformer Encoder, Decoder

Learnable, absolute

위치 임베딩 방식은 같지만, 사전 학습 방식과 사용 목적에 차이 존재

Do Embeddings Learn the Meaning of Positions?

- **Position Embedding 함수 정의**

Position space P 와 embedding space X 에 대하여, Position embedding function $f: P \rightarrow X$ mapping

- ✓ Sinusoid, Transformer Encoder model, Transformer Decoder model

- ✓ Position Embedding이 위치의 의미를 해석하는가?

- 1 학습된 임베딩 공간 X 가 단어의 절대 위치를 표현할 수 있는가?

- 2 위치 공간 P 와 임베딩 공간 X 가 동형인가?

Do Embeddings Learn the Meaning of Positions?

- 학습된 임베딩 공간 X 가 단어의 절대 위치를 표현할 수 있는가?
 - ✓ Embedding space X 를 Position space P 로 되돌렸을 때 위치 정보가 보존되어 있어야 한다

❖ Experiment

Embedding space X 에서 Position space P 로 되돌리는 **선형 회귀 모델 실험**

Embedding space의 X 를 Position space P 로 되돌리는 매핑 함수 g 정의,

Reversed mapping function $g: X \rightarrow P$ mapping 적용 후 **MAE 오차** 계산

Do Embeddings Learn the Meaning of Positions?

- 학습된 임베딩 공간 x 가 단어의 절대 위치를 표현할 수 있는가?

❖ Result

Type	PE	MAE
Learned	BERT	34.14
	RoBERTa	6.06
	GPT-2	1.03
Pre-Defined	sinusoid	0.0

- Sinusoid 방식의 pre-defined position embedding은 절대 위치를 정확하게 표현

- Transformer 디코더 모델 (GPT-2)에서는 약간의 오차 발생

- Transformer 인코더 모델 (BERT, RoBERTa)은 위치 관계를 정확하게 학습하지 못함

SVM, MLP와 같은 비선형 모델 실험시 overfitting이 자주 발생, 테스트 성능이 선형모델에 비해 낮았다

Do Embeddings Learn the Meaning of Positions?

- 위치 공간 P 와 임베딩 공간 X 가 동형인가?

✓ Position space P 와 Embedding space X 가 isomorphic하기 위해서는, 두 공간 사이의 거리 연산에 대한 일대일 대응이 존재해야 한다

❖ Experiment

거리 매핑 함수를 정의, $h(x_i, x_j) = |i-j|$

이때, 위치 간 거리를 선형회귀로 예측하는 데에 어려움이 있어 두 위치의 순서를 예측하는 binary classification problem으로 실험

$h(x_i, x_j) = 0 \text{ or } 1$

Do Embeddings Learn the Meaning of Positions?

- 위치 공간 P와 임베딩 공간 X가 동형인가?

❖ Result

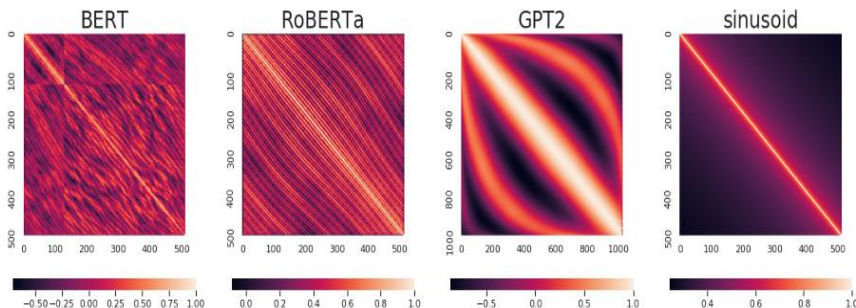
Type	PE	Error Rate
Learned	BERT	19.72%
	RoBERTa	7.23%
	GPT-2	1.56%
Pre-Defined	sinusoid	5.08%

- Transformer 인코더 모델 (BERT, RoBERTa)은 위치 간의 관계 정보를 덜 학습하였다
- Sinusoid 방식은 Transformer 인코더 모델보다는 위치 간의 관계 정보를 잘 학습하였으나, GPT-2 모델에 비해 오차율이 높았다
- GPT-2 모델의 상대 위치 관계를 가장 잘 학습하였다

What do Transformer Encoders Capture about Positions?

- **Position-Wise Cosine Similarity**

위치 임베딩 간의 cos-similarity를 계산하여 시각화



- **GPT-2 / sinusoid 방식**

위치 순서에 따른 주기성 패턴 관찰

- **BERT**

임베딩 벡터가 인접한 위치끼리는 유사하지만 장기적으로는 설명 가능한 패턴이 나타나지 않는다

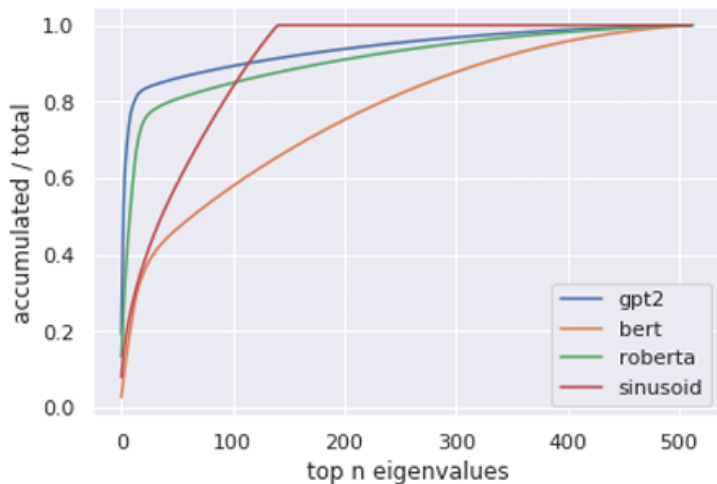
- **RoBERTa**

BERT와 유사하지만, 인접 위치 사이에서 비주기적인 패턴

What do Transformer Encoders Capture about Positions?

▪ Informativeness of Position Embeddings

학습된 위치 임베딩을 대상으로 SVD 수행하여
고유벡터 분석



✓ 상위 n 개의 고유값 누적합이 전체 고유값에서 차지하는 비율을 시각화

- 작은 n 으로도 전체 정보의 대부분을 설명할 수 있어야 이상적인 위치 임베딩
- BERT는 전체 정보의 대부분을 설명하기 위해 매우 큰 n 을 필요로 한다
- RoBERTa도 GPT-2와 Sinusoid 방식에 비해 더 많은 n 을 필요로 한다

What Makes the Differences?

- Transformer Encoder model vs Transformer Decoder model

✓ Pre-Training Objectives

1 Transformer Encoder model

$$\mathcal{L}(\mathcal{U}) = \sum_i \log P(u_i | u_1, \dots, u_{i-1}, u_{i+1}, \dots, u_l; \theta)$$

특정 단어 u_i 를 그 주변 문맥으로 예측하는
확률을 극대화

2 Transformer Decoder model

$$\mathcal{L}(\mathcal{U}) = \sum_i \log P(u_i | u_1, \dots, u_{i-1}; \theta)$$

이전 단어들로 다음 단어 u_i 를 예측

Transformer의 인코더는 정확한 절대 위치 정보를 필요로 하지 않는다

지역적인 위치 정보만을 학습하여 인접한 위치 정보에 집중하도록 한다

What Makes the Differences?

- BERT vs RoBERTa

- ✓ Main improvements from BERT to RoBERTa

- 1 학습 도중 배치 사이즈를 점진적으로 증가 (256 > 8192)

- 2 토큰 마스킹 방식을 동적으로 변화

- 3 추가적인 다음 문장 예측 loss 항목 제거

RoBERTa의 cos-similarity 시각화 결과는 BERT와 유사하지만 128 위치에서의 단절 x

-> 큰 배치 사이즈로 사전학습을 수행할 경우, Transformer 인코더에서의 위치 임베딩이 더 명확해진다

Performance Effect

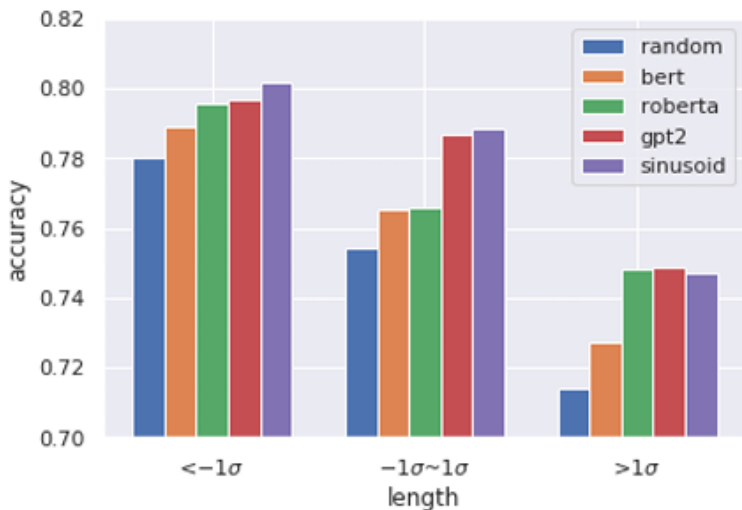
Text classification

✓ Experiment Setup

Dataset: SST2, TREC, SUBJ, CR, MR, MPQA

Model: 1-layer Transformer encoder

Position Embedding: Random, BERT, RoBERTa, GPT-2, Sinusoid



실험 결과

- BERT, RoBERTa의 성능이 낮았다
- Transformer는 모든 입력 토큰의 정보를 활용하기에, 절대 위치 정보가 상대 위치 정보보다 중요하게 작용
- 짧은 문장 기반 Task에서는 위치 임베딩간의 차이가 거의 없다

Performance Effect

▪ Language Modeling

Encoder: masked language model

Decoder: autoregressive model

Masked language model은 지역적인 위치 정보를 학습

Skip Position Attack Experiment

$$z_i = WE(x_i) + PE(i)$$

$$z_i = WE(x_i) + PE(i * k).$$

기존의 방식

입력 임베딩을 단어 임베딩과 위치 임베딩의 합으로 표현

Skip-Position Attack

위치 인덱스에 상수를 곱해 건너뛴 위치 정보를 사용

단기적인 지역 정보만 학습하였다면, 근처 위치를 건너뛰어 위치 정보가 손실

절대 위치 정보를 학습하였다면, 건너뛰어도 순서가 유지되므로 영향이 상대적으로 적다

Performance Effect

▪ Language Modeling

✓ Experiment Setup

Dataset: Wikitext-2, Wikitext-103

Model: BERT-Base (masked), GPT-2 Base (autoregressive)

Position Embedding: Random, BERT, RoBERTa, GPT-2, Sinusoid

LM	PE	Wikitext-2		Wikitext-103	
		Perplexity	Δ	Perplexity	Δ
MLM	BERT	147.93	-	12.45	-
	+skip position	198.61	(+50.68)	323.12	(+310.67)
	RoBERTa	157.98	-	12.61	-
	+skip position	199.13	(+41.14)	14.44	(+1.83)
Autoregressive	GPT-2	172.97	-	25.83	-
	+skip position	171.20	(-1.77)	25.74	(-0.09)

실험 결과

- 평균적으로는 BERT와 RoBERTa가 GPT-2보다 퍼플렉시티가 낮다 (양방향 문맥 파악)
- Skip-position attack | masked language model의 성능을 크게 약화시킨다

Performance Effect

Language Modeling

✓ Experiment Setup

Dataset: Multi30k EN-DE (WMT16)

Model: 6-layer Transformer, FFN=1024, hidden=512, head=4

인코더와 디코더의 위치 임베딩을 각각 설정

PE		BLEU	
Encoder	Decoder	Full Set	Length > 2 σ
Random	Random	32.19	18.04
BERT	-	35.54	22.98
GPT-2	-	34.36	22.05
-	BERT	32.08	17.77
-	GPT-2	32.81	18.05
BERT	GPT-2	34.11	21.29
GPT-2	BERT	32.80	21.96
BERT	BERT	35.94	23.60
RoBERTa	RoBERTa	35.47	24.50
GPT-2	GPT-2	35.80	25.12

실험 결과

인코더: BERT와 GPT-2 모두 성능 향상, BERT의 효과가 더 크다

디코더: GPT-2 효과가 더 크다, BERT는 성능을 더 저해할 수도 있다

혼합: 성능이 떨어질 수 있다

긴 문장에 대해서는 GPT-2 위치 임베딩이 가장 뛰어나다