

# Transformer: Positional Encoding

Ki Hyun Kim

[nlp.with.deep.learning@gmail.com](mailto:nlp.with.deep.learning@gmail.com)

# Unlike RNN,

- Transformer는 위치 정보를 스스로 처리하지 않음 (ConvS2S도 마찬가지)
  - 입력 순서를 바꿔 넣으면 출력도 순서가 바뀐 채 같은 값이 나올 것
  - 마치 FC layer의 입력 feature 순서를 바꿔 학습해도 성능이 똑같은 것과 같음
- 따라서 위치(순서) 정보를 따로 인코딩해서 넣어줘야 함

# Positional Encoding

- 기존의 word embedding 값에 positional encoding 값을 더해줌

**Sentence Embedding Matrix**

Dimension ↑																
	0	1	2	3,2												
				1												
				0												
Position →																
																Word Embedding Vector

$$\begin{aligned} PE(pos = 3, dim\_idx = 2 = 2 \times i) \\ &= PE(pos = 3, i = 1) \\ &= \sin\left(\frac{pos}{10^{4 \times \frac{2 \times i}{d}}}\right) = \sin\left(\frac{3}{10^{4 \times \frac{2 \times 1}{d}}}\right) \end{aligned}$$

# vs Positional Embedding

- 사실 위치 정보도 integer 값이므로 embedding layer를 통해 임베딩 할 수 있음
- BERT와 같은 모델들은 positional encoding 대신에 positional embedding을 사용하기도 함

# Summary

- RNN과 달리, 순서(위치) 정보를 encoding해주는 작업이 필요
  - 학습이 아닌 단순 계산 후 encoding
- 학습에 의해 달라지는 값이 아니므로, 한번만 계산해 놓으면 됨