

## LM(Language Model)

### • 모델

- 정의 - 연구 대상 및 주제를 기호로 사용하여 표현한 것.
- 특징 - 자연의 방식을 컴퓨터가 모사함으로써 시뮬레이션 가능.



자연의 방식을 컴퓨터가 모사  $\Rightarrow$  언어모델

### • 언어 모델

- 입력된 단어를 기반으로 하여 다음의 단어를 예측하는 방식으로 학습.

#### \* 언어 모델의 예시

##### 1. Markov 모델

다음단어를 예측하기 위해 조건부 확률을 사용하는 언어 모델

단어 기반의 학습으로 변화.

##### 2. RNN 기반의 언어 모델

- RNN은 하든 노드가 방향을 가진 엣지의 형태로 연결된 순환 신경망.  $\rightarrow$  시계열 데이터 분석에 용이.
- 현재의 state가 다음의 state로 전달.

#### RNN의 활용

$\rightarrow$  RNN의 경우 시계열 데이터를 입력으로 받기 때문에, 문장의 의미 (context)를 갖는다.

문장의 긍정·부정 판단 / 문장의 다음단어를 예측.

### 3. Seq2Seq (Sequence to Sequence)

- Encoder + Decoder 구조

• Encoder  $\xrightarrow{\text{context vector}}$  Decoder

↳ RNN 에 의해 얻어진 데이터의 문맥, 벡터

ex) STT (Speech To Text)

기계번역

영상 자막 만들기.

\* RNN 기반의 Seq2Seq 의 문제점.

- sequence 가 길어지는 경우 처음 나온 token 정보만  
크기
- 고정된 context vector 로 인해 sequence 의 정보 향유  
이 어려움
- 불필요한 토큰 역시 학습에 영향을 미침.



Attention의 탄생.

### 4. Attention.

- Encoder Cell의 output 값을 사용!!

Attention 사용시 Decoder 이기 때 time step 마다 RNN의 output  
가시각으로 어떤 을 사용하여 Context Vector 사용

input에 집중하는지 Decoder 학습시 context vector를 조정하는 방안으로  
표현이 가능!! 학습.

- Encoder 역시 Decoder의 현재 state와 비교 연산을  
통하여 context vector 를 전달.

※ RNN 모델의 경우 연산에 너무 오랜시간을 필요로 함

=> RNN 모델을 없는 seq2seq 모델의 생성

↳ Transformer

## 5. Transformer.

\* Concept의 변환!

=> Encoder에서 input의 값을 잘표현 하도록 "학습"

• Encoder · Decoder 에서 RNN을 제거.

• self - Attention으로 아주 이전 모델.

↳ Concept!

=> 문장에서 단어들의 의미·중요도를 잘 표현하도록  
모델을 학습

• Multi-Head self Attention.

Query · Key · Value 로 구성된 Attention을  
여러개의 Head 에서 진행.

• Multi-Head Attention + FFNN 으로 구성된  
Encoder<sup>layer</sup>를 여러개 이어 붙이는 형식으로 만듦.

• 또한 Encoder 와 비슷한 형태의 Decoder layer를  
여러개 이어 Decoder를 만듦