

기존 word embedding 방식 : 주변 단어로 학습 ~> 문맥 고려 x
⇒ 언어모델

언어모델

@ 모델

: 연구 대상 주제를 도면, 사진 사용 or 수식, 기호로 표현.
즉, 자연 법칙을 컴퓨터로 모사함으로써 시뮬레이션 가능

* 언어모델

· 학습 방향

- 주어진 단어들로부터 다음 등장 단어의 확률을 예측
⇒ 예측 good ~> 언어의 특성이 잘 반영된 모델

* Markov 확률 기반 언어 모델 : Markov Chain Model

· 학습방법

- 문장들에서 각 단어마다 다음 단어의 확률 학습 ~> 확률을 최대한 하도록 학습
- Markov 테이블 작성
- 같은 방식으로 문장들의 확률도 계산 가능

· 예시

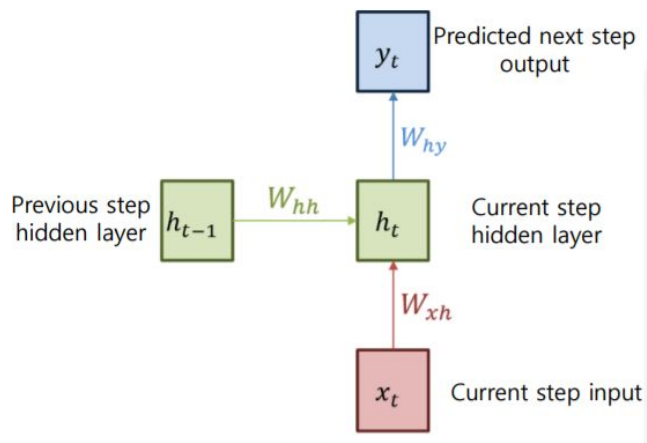
- p.42

* Recurrent Neural Network(RNN) 기반의 언어 모델

· 학습방법

- 현재 단계에서 다음 단계를 예측하는 과정에서, 이전 단계의 정보를 반영함.

기본적인 RNN의 구조



· 예시

- p.43

· 출력결과

- 앞선 단어들의 '문맥' 고려 ~> 마지막 출력
- Context vector 형성 가능
- Context vector + classification layer ⇒ 예측 모델

· 구조적 문제점

- 입력 sequence 길이가 매우 긴 경우 ~> 처음 token 정보 ↓
- 중요 x 토큰도 중요 토큰과 같은 영향력

* Attention 모델

· 배경

- 중요 feature은 더욱 중요하게 고려

· 학습단계

1. 기존 context vector + 각 step의 output
2. Feed Forward Fully Connected Layer에 입력 + softmax
3. 각 단계별 weight 출력 (중요도)
4. 각 step의 output * weight => 최종 context vector
5. 예측 결과 bad => Feed Forward Fully Connected Layer & weight 수정
6. 반복

· 장점

- 문맥에 따라 동적으로 할당되는 dynamic context vector 획득
- Attention 시각화를 통해 설명력 ↑
- p. 58

· 단점

- RNN을 순차적으로 연산 => 느림

* Self-attention 모델

· 배경

- 결국 필요한건 각 단어의 중요도(weight) 파악
- RNN 과정을 생략하고

· 학습과정

- p.65 ~ p.68

* Multi-head Self Attention 모델 (Bert 영역)

- Query, Key, Value로 구성된 attention layer을 동시에 여러 개 수행

* Transformer 모델

- multi-head attention으로 이루어진 encoder를 여러 층 쌓아서 encoding을 수행