딥러닝팀

1팀

이수경
이승우
이은서
주혜인
홍현경

- Tokenization

Tokenization이란?

Text

"The cat sat on the mat."

↓

Tokens

"the", "cat", "sat", "on", "the", "mat", "."

문장/문서 단위의 자연어 데이터 집합

코퍼스(Corpus) 데이터 집합을 **토큰(token)**단위로 나누는 전처리 작업

ex) 구, 문장, 문단, 단어 등

- Tokenization

Out-of-Vocabulary

Train data tokens The, rabbit, cats, sea, eat, catch, octopus

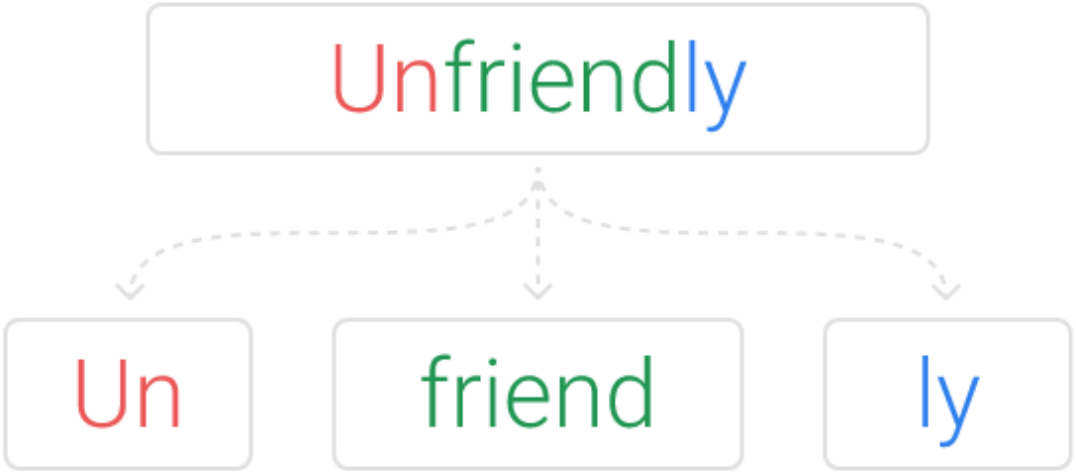
test data The cats catch the mouse ?

model

학습 데이터에 없는 단어는 처리하지 못하는 문제

- Subword Segmentation

Subword Segmentation이란?



"단어는 의미를 가진 더 작은 **서브워드**들의 조합으로 이루어진다"

➡ 모르는 단어를 쪼개서 의미를 추론

- Embedding

Embedding이란?

Vocabulary:
Man, woman, boy,
girl, prince,
princess, queen,
king, monarch

	1	2	3	4	5	6	7	8	9
man	1	0	0	0	0	0	0	0	0
woman	0	1	0	0	0	0	0	0	0
boy	0	0	1	0	0	0	0	0	0
girl	0	0	0	1	0	0	0	0	0
prince	0	0	0	0	1	0	0	0	0
princess	0	0	0	0	0	1	0	0	0
queen	0	0	0	0	0	0	1	0	0
king	0	0	0	0	0	0	0	1	0
monarch	0	0	0	0	0	0	0	0	1

사람이 쓰는 자연어를 컴퓨터가 이해할 수 있는 **벡터 형태**로 바꾸는 것

- Embedding

Embedding 모델

:자연어의 **통계적 패턴 정보**를 통해 벡터를 만드는 모델

유사한 벡터

ex) 호양이 = [0.9, -0.32, 0.27, 0.43]

'고양이'와의 상대적인 관계 학습

결합되어 응용될 수 있다

[Bag-of-words 가정] [언어 모델 Language Model] [분포 가설 Distribution Hypothesis]

유사한 등장빈도

고양이랑 유사한 빈도로 등장해서~

유사한 등장순서

고양이랑 유사한 등장순서를 가져서~

유사한 의미, 맥락

고양이랑 유사한 맥락에서 사용되어서~

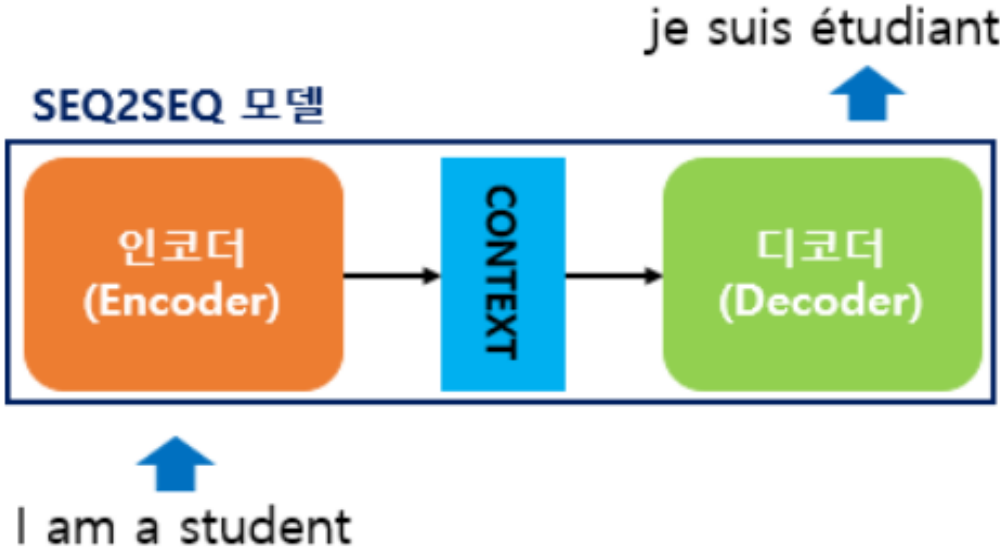
2 seq2seq

seq2seq 이란

Sequence-to-sequence

: 시퀀스 입력 – 시퀀스 출력

Ex) 기계 번역, 챗봇

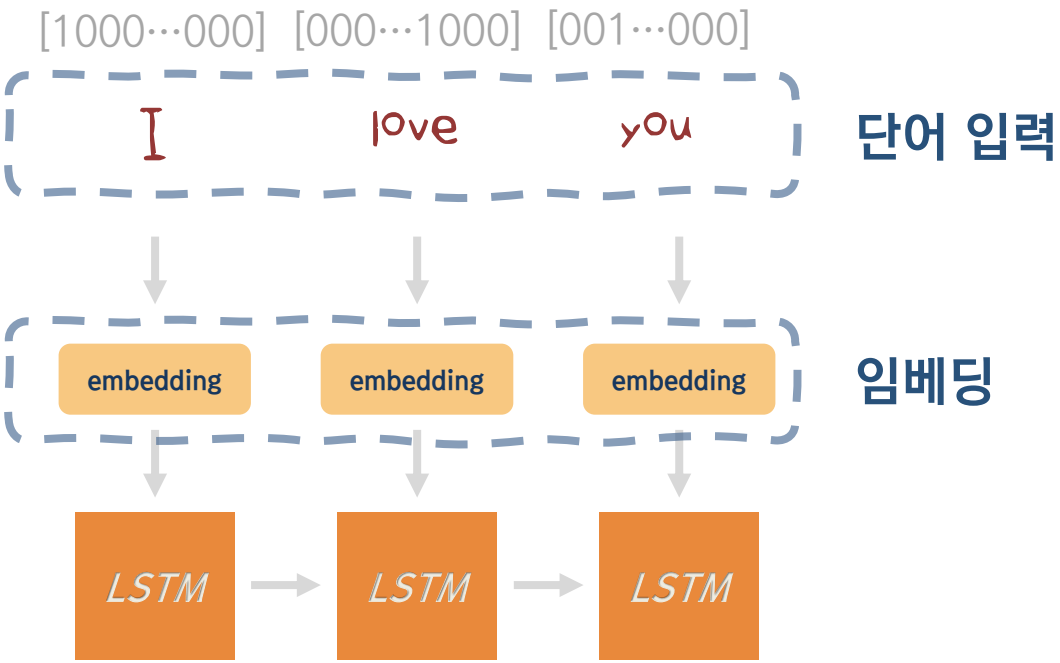


인코더-디코더 2개의 RNN 아키텍처를 이어 붙인 모델

- seq2se의 구조

1) Encoder

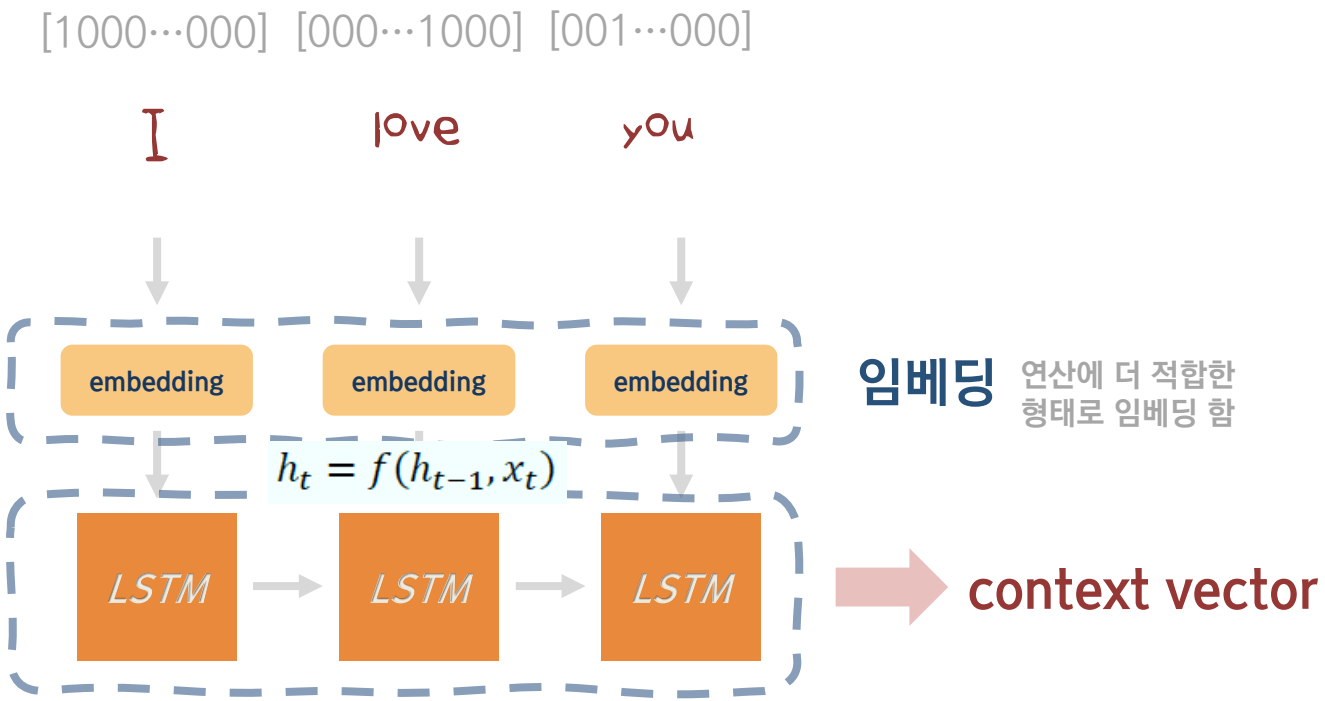
: 순차적으로 단어 입력



- seq2se의 구조

1) Encoder

: 순차적으로 단어 입력



- seq2seq의 구조

2) Context vector

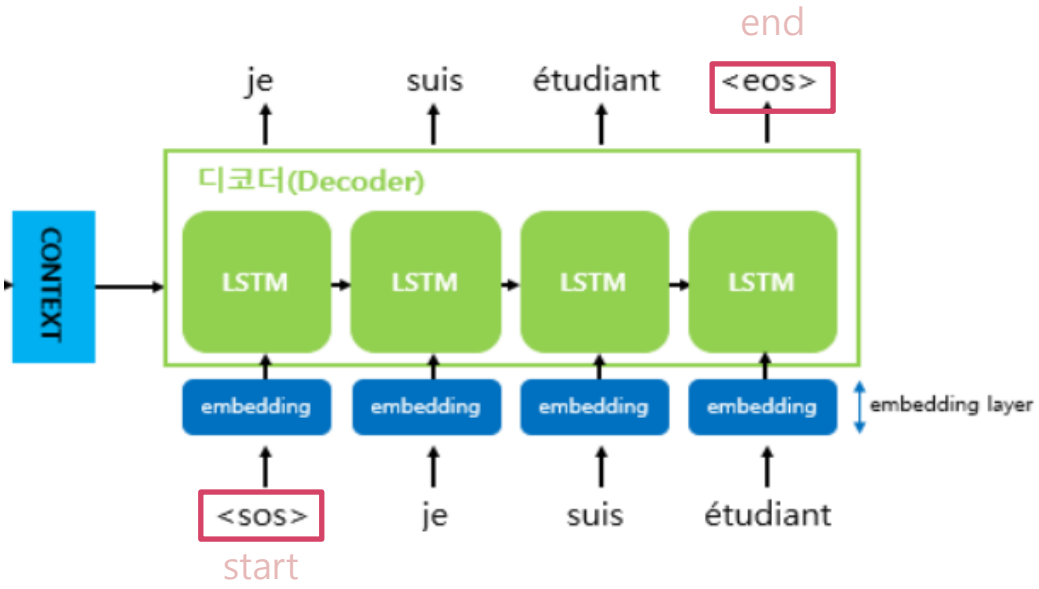
encoder에 들어온 모든 정보들을 압축해서 담은 벡터

고정된 길이의 벡터

- seq2seq의 구조

3) Decoder

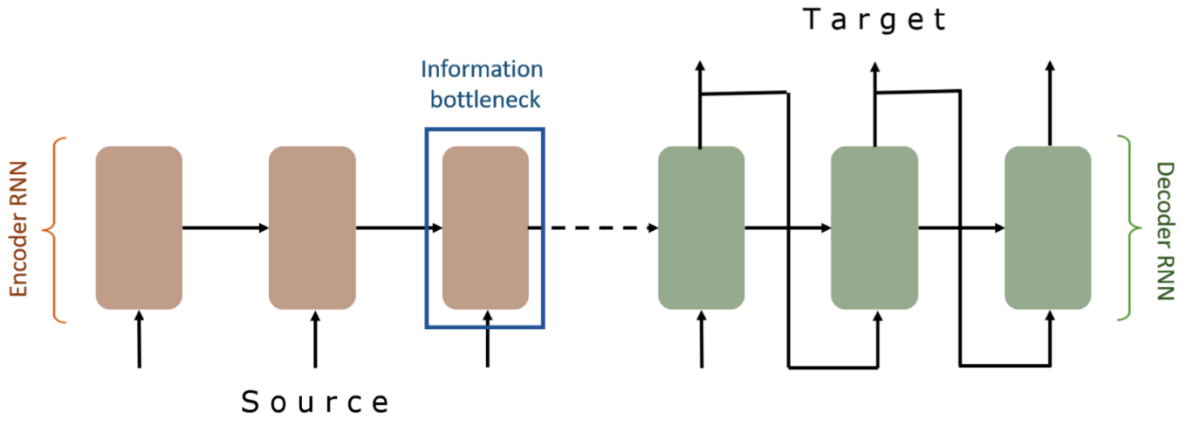
:번역된 단어를 순차적으로 출력



Context vector를 바탕으로 item by item으로 출력값 생성

- Attention Mechanism

Seq2Seq의 문제점



Bottleneck Problem (병목 현상)

Seq2seq**에서는 마지막 스텝의 hidden state**인 context vector를 **decoder**에 넘겨줌

단일 벡터가 **모든 source sentence의 정보**를 담고 있어야 함

마지막 스텝에서 **모든 정보를 인코**딩하여 정보가 쏠리는 현상 발생

- Attention Mechanism

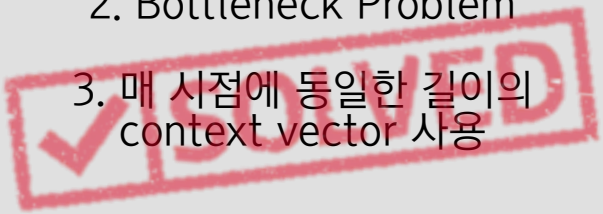
Attention의 등장

Seq2Seq

1. 장기 의존성 문제

2. Bottleneck Problem

3. 매 시점에 동일한 길이의 context vector 사용



하나의 고정된 길이의 벡터로 전체 맥락 나타내는 것은 불충분

입력 문장 길이 ↑ ~ **번역 품질** ↓

Attention

각 타임 스텝마다 생성된 context vector를 decoder에 넘겨줌

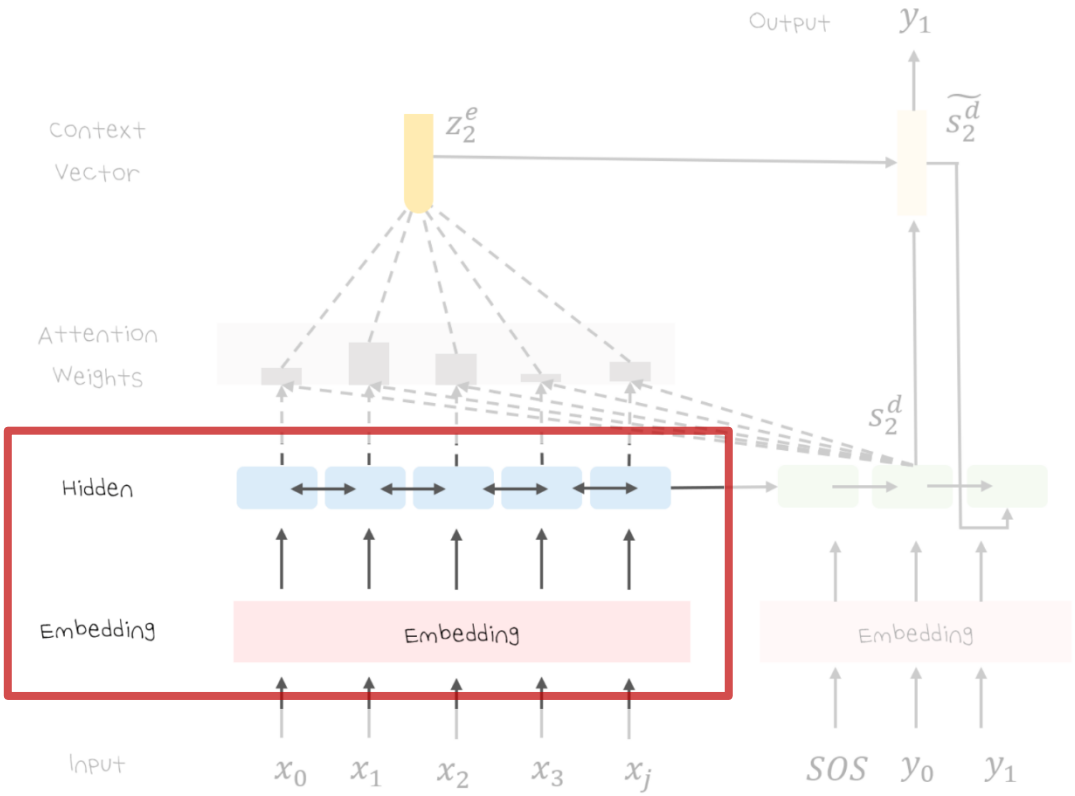
Decoding 단계에서

입력 문장 중 어떤 부분에 **집중**해야 하는지

타임 스텝 별로 알 수 있음

- Structure of Attention

Encoder



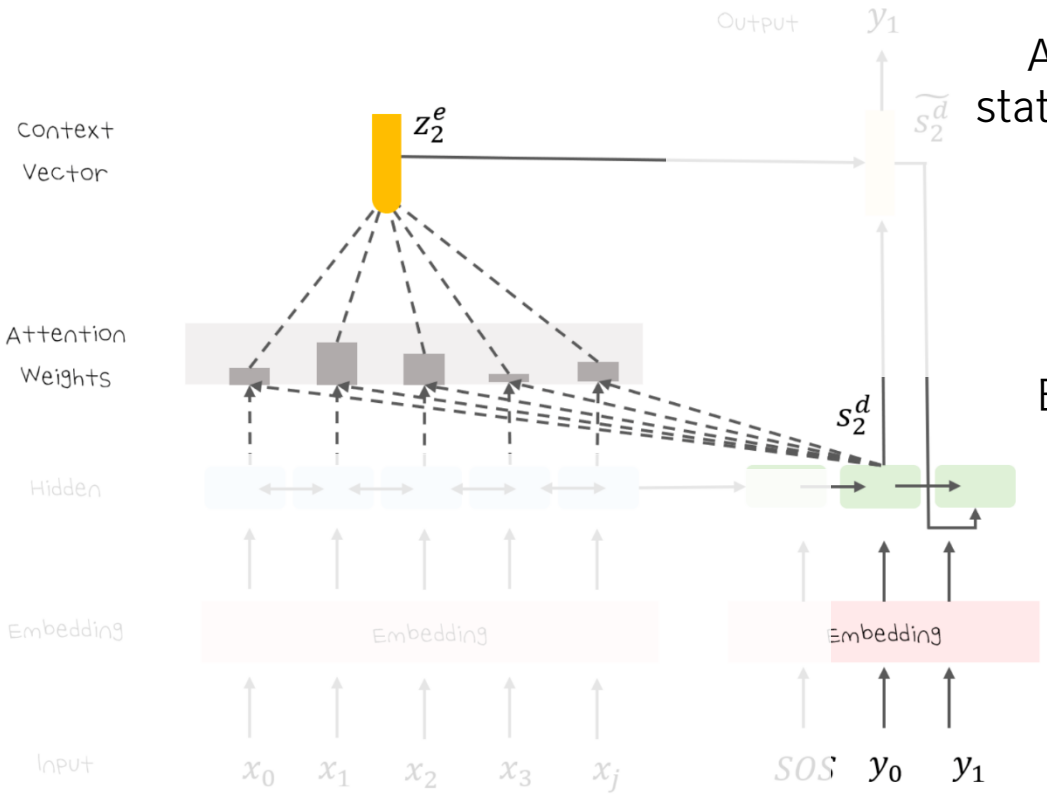
여기까지 인코더

Embedding layer 거친 값들이
RNN(LSTM, GRU) 모델로 들어가서
Hidden state 생성

Source data가
embedding layer로 들어감

- Structure of Attention

Decoder



이때 이번 시점의 decoder 출력 값과 인코더 hidden state들이 유사한 만큼 Attention! 되는 것임

Attention score와 Encoder hidden state들을 곱하여 Attention context 생성!

Decoder의 이전 시점 hidden state와 Encoder에서 구했던 hidden state들을 비교하여 attention score 생성!

Encoder의 입력이 끝나면, Decoder의 입력 시작!