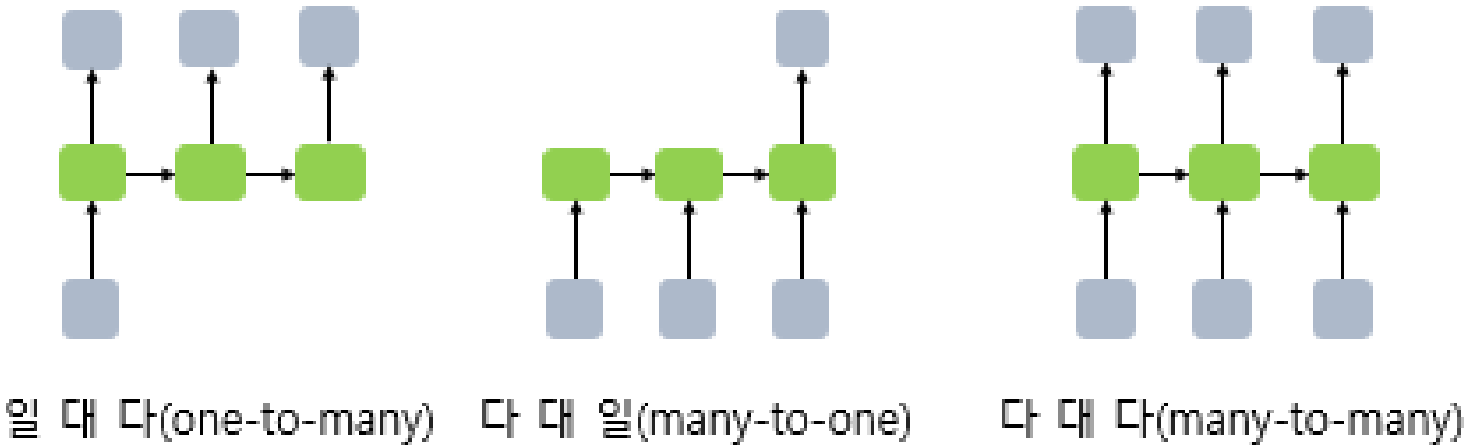
Attention

김균엽

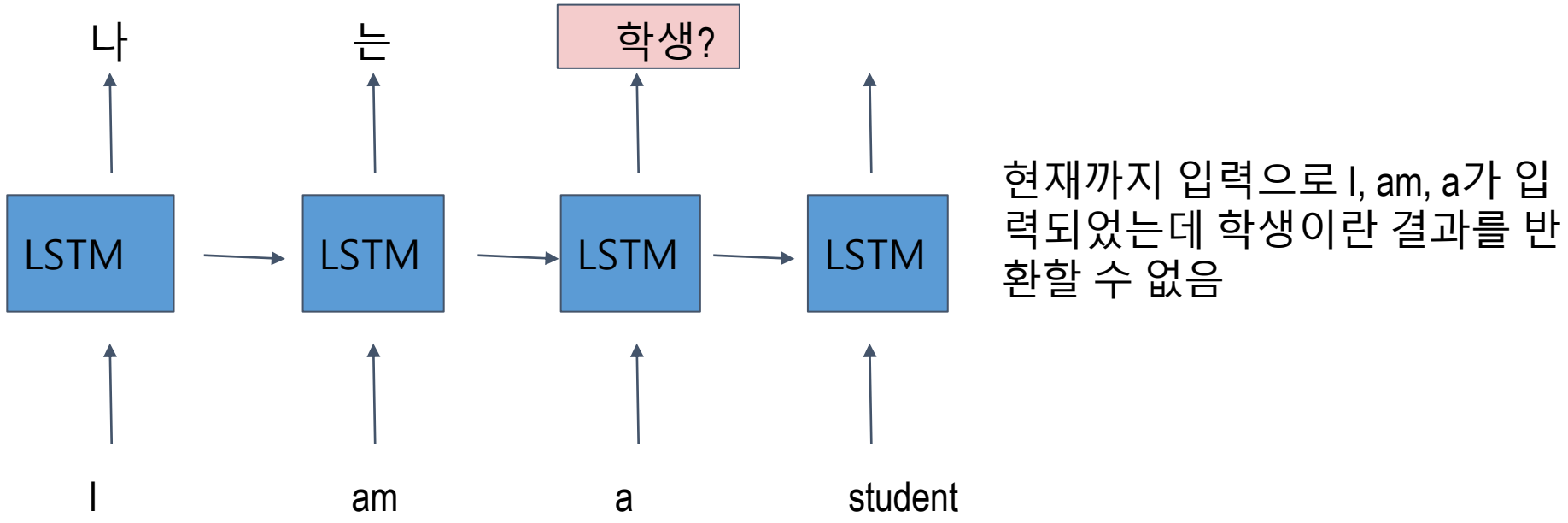
Seq2Seq

- **Translation in LSTM**
  - 번역을 LSTM과 같은 모델에서 진행하기 위해서는 many-to-many의 방법으로 in-out이 이루어져야함



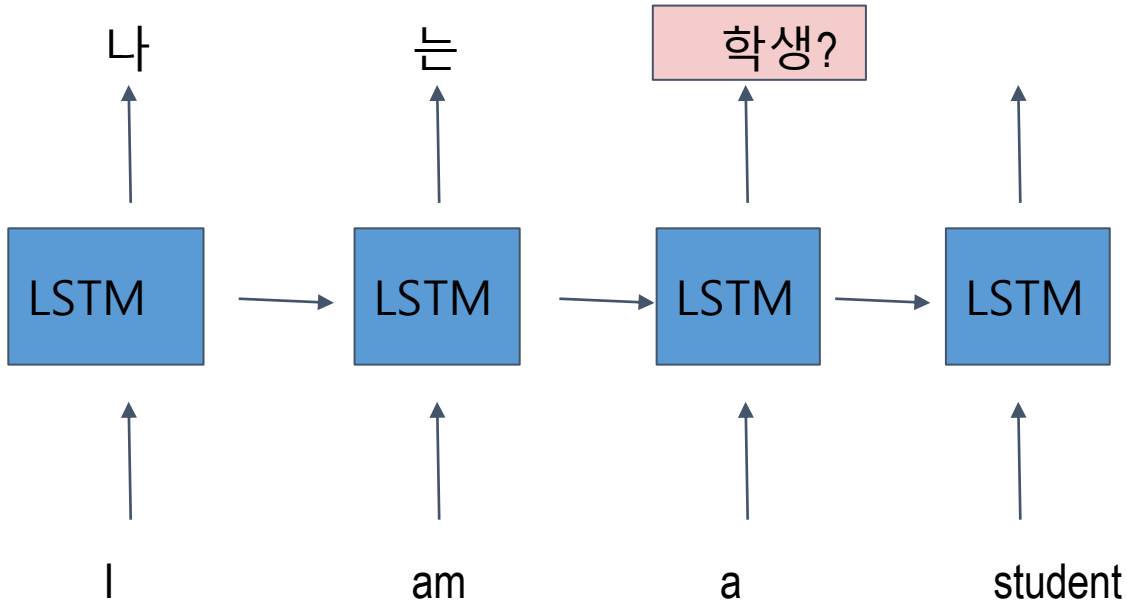
Seq2Seq

- **Translation in LSTM**
  - 하지만 입력의 길이가 매번 다르고 나라마다 어순이 다르기에 첫번째 단어만 보고 번역된 문장의 첫 단어를 예측하기란 쉽지 않음.
  - "나"라는 주어가 가장 앞에 나오는 언어가 있는 반면 주어가 뒤에 나오는 언어도 존재



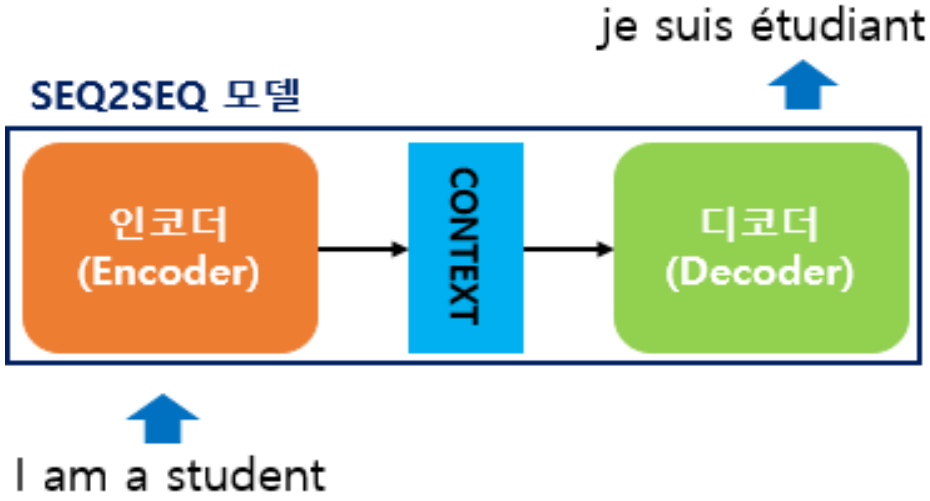
Seq2Seq

- **Translation in LSTM**
  - 다시말해, LSTM에서 결과를 도출할 때에는 현재 time의 이전 단어의 정보만을 가지고 결과를 도출
    → 모든 단어를 입력한 후 결과를 뽑는 방법 제안



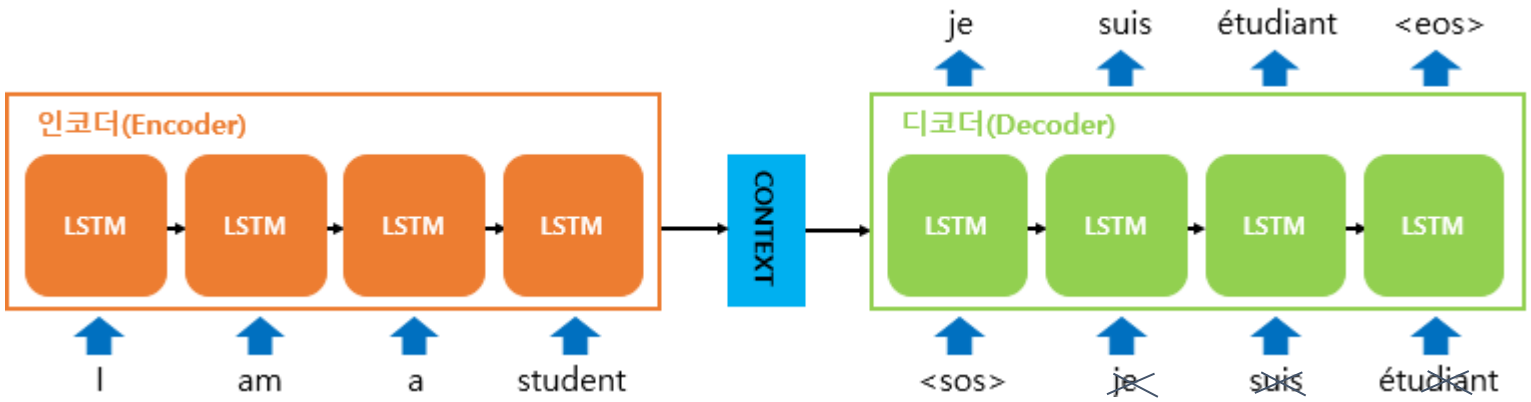
Seq2Seq

- **Seq2Seq Model**
  - 입력문장을 모두 입력받아 모든 token의 정보가 압축된 벡터인 context를 구성후 한 단어씩 디코딩해가는 방식
  - Encoder와 Decoder라는 두가지의 모듈을 가지고 Encoder에서는 입력된 단어를 압축, Decoder에서는 압축된 벡터를 푸는 형태를 가진다.



Seq2Seq

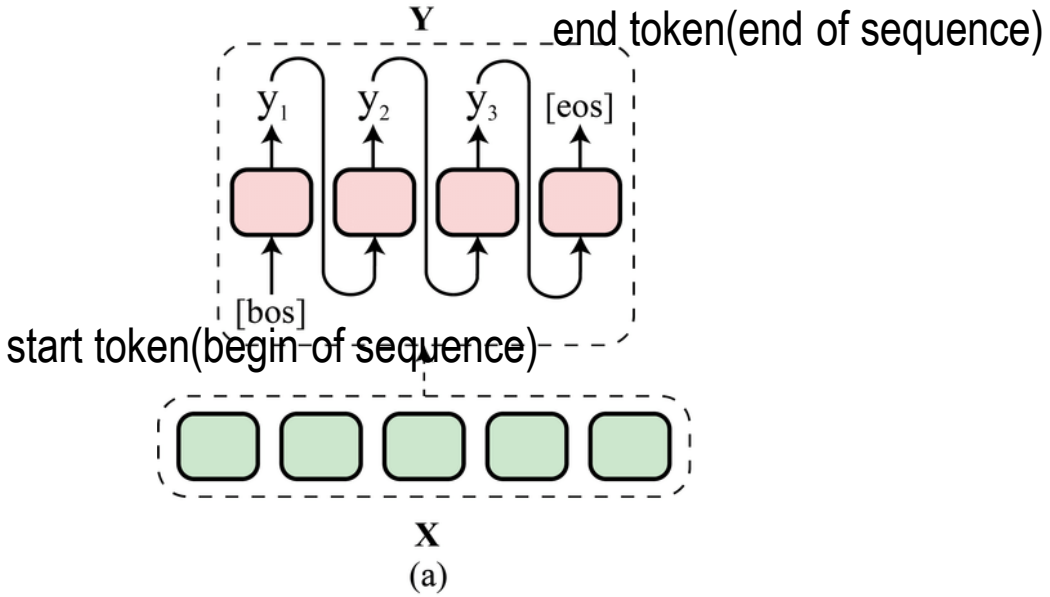
- **Seq2Seq Model**
  - Seq-2-Seq model의 경우 source문장을 encoder에서 처리하고 decoder에서는 target문장을 generation한다.
  - 문장을 generation하는 과정에서는 입력으로 정답을 사용할 수는 없기에 auto-regressive방법을 통해 generation한다.



정답을 모른다 가정하고 test하기 때문에
decoder의 입력 토큰을 설정할 수 없다.

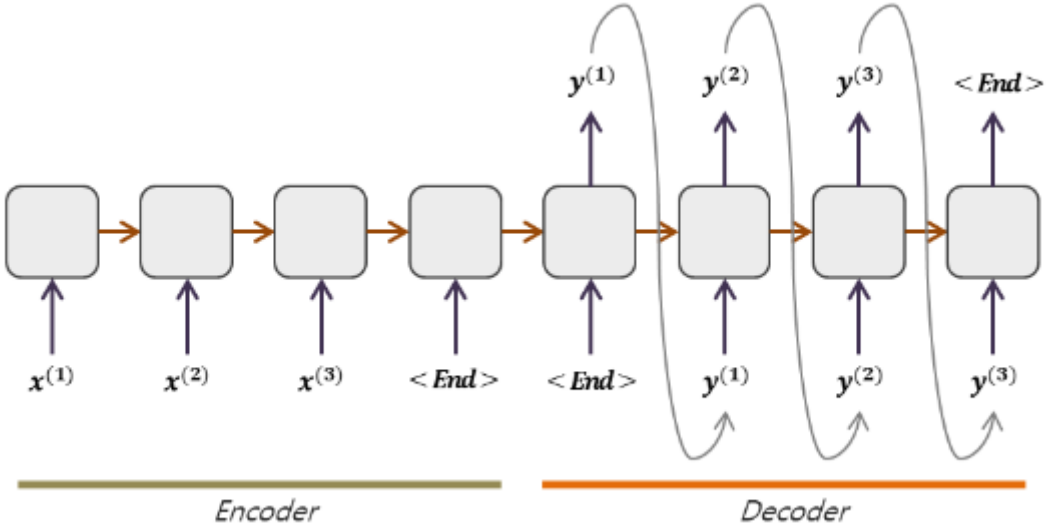
Seq2Seq

- **Auto regressive**
  - seq-2-seq의 decoder에서 정해진 입력토큰은 start token만을 사용한다.
  - 사전에 일부 token을 special token으로 지정한다.
  - 일반적으로 문장의 시작- start token, 문장의 끝- end token, padding token을 사용한다.₩
  - start token의 output을 target문장의 1st token이라 가정한다. 즉, next time에서는 current time의 output을 입력으로 사용한다.



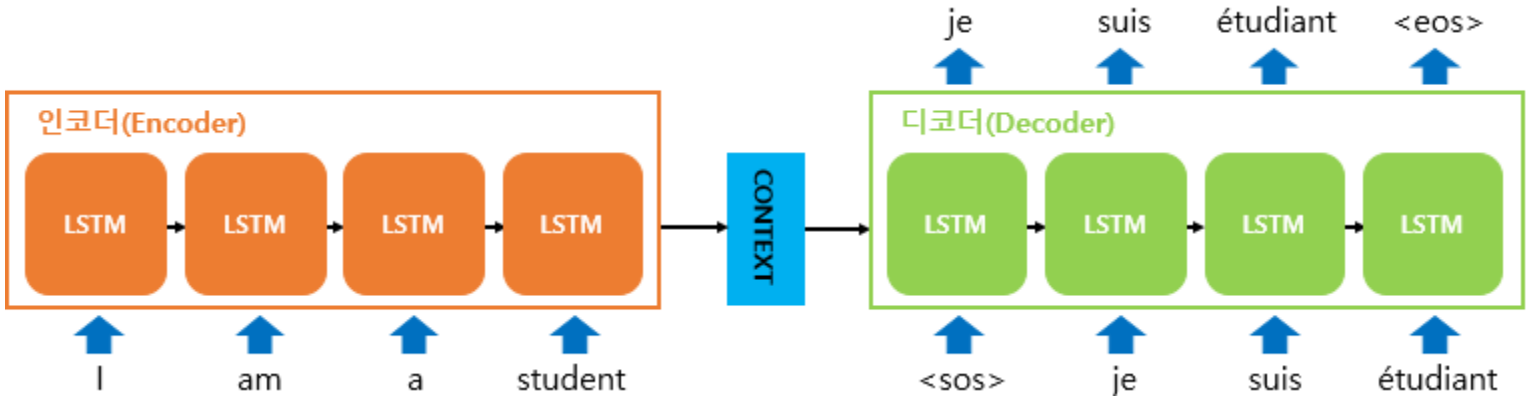
Seq2Seq

- **Auto regressive in seq-2seq model**
  - seq-2-seq의 decoder에서 정해진 입력토큰은 start token만을 사용한다.
  - 사전에 일부 token을 special token으로 지정한다.
  - 일반적으로 문장의 시작- start token, 문장의 끝- end token, padding token을 사용한다.₩
  - start token의 output을 target문장의 1st token이라 가정한다. 즉, next time에서는 current time의 output을 입력으로 사용한다.



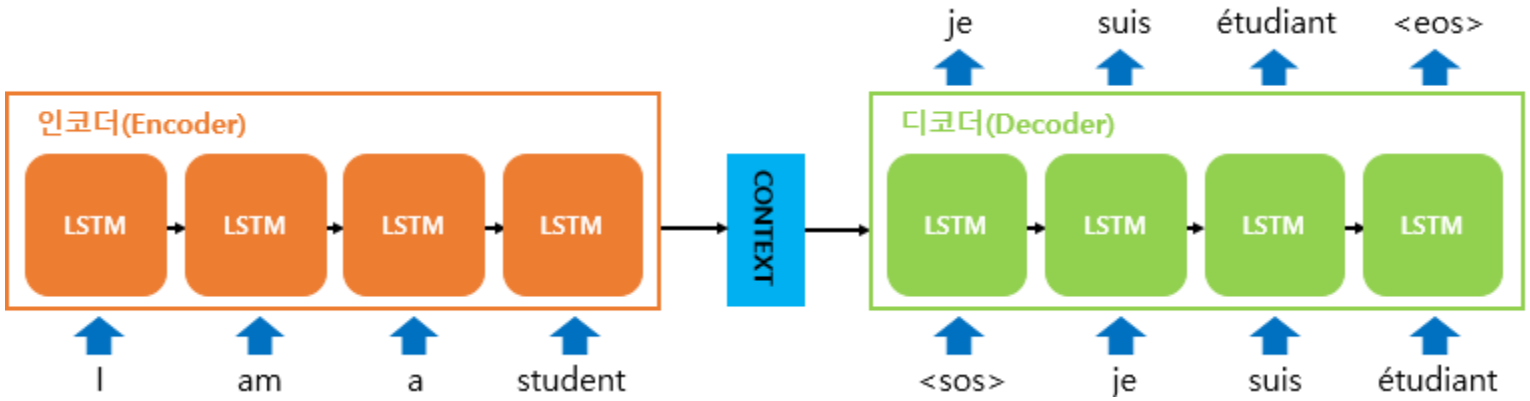
Seq2Seq

- **problem of Seq-2-seq model**
  - 기존의 Seq-2-Seq model은 LSTM을 Encoder와 Decoder로 사용하기에 LSTM과 RNN의 문제점을 그대로 계승한다.
  - LSTM의 vanishing gradient problem을 계승하기에 초기 단어의 정보가 소실된다.
  - 즉, encoder에서 source sentence를 context vector로 압축하는 과정에서 LSTM을 사용하기에 앞부분의 정보를 소실한다는 문제를 가진다.



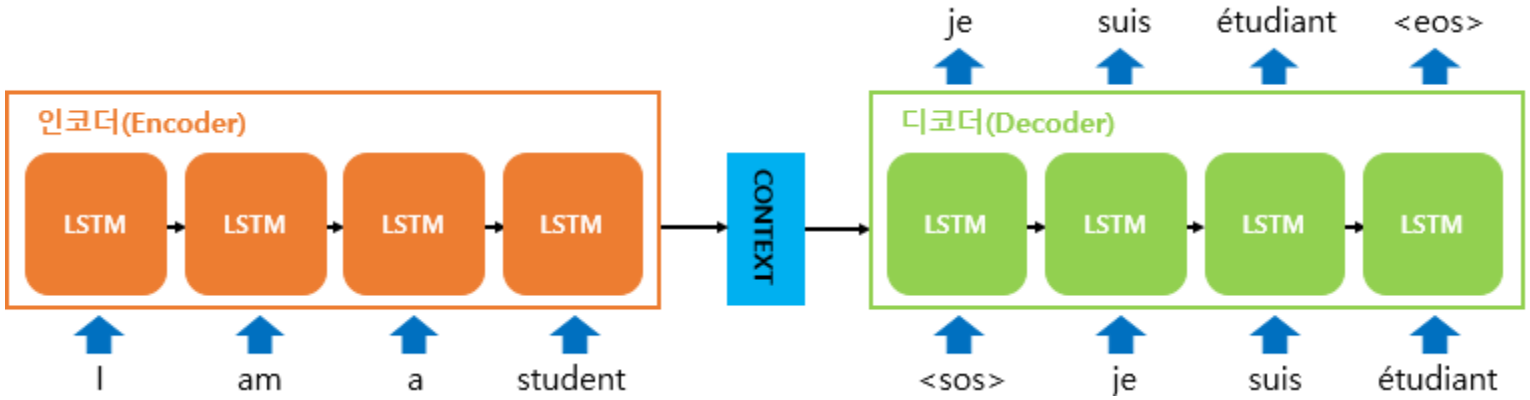
Seq2Seq

- **problem of Seq-2-seq model**
  - 또한 고정된 크기의 context vector에 모든 token에 대한 정보를 담으려하기에 정보의 손실이 일어난다.
  - 결과적으로, 문장의 길이가 길어질수록 낮은 번역품질을 보인다.



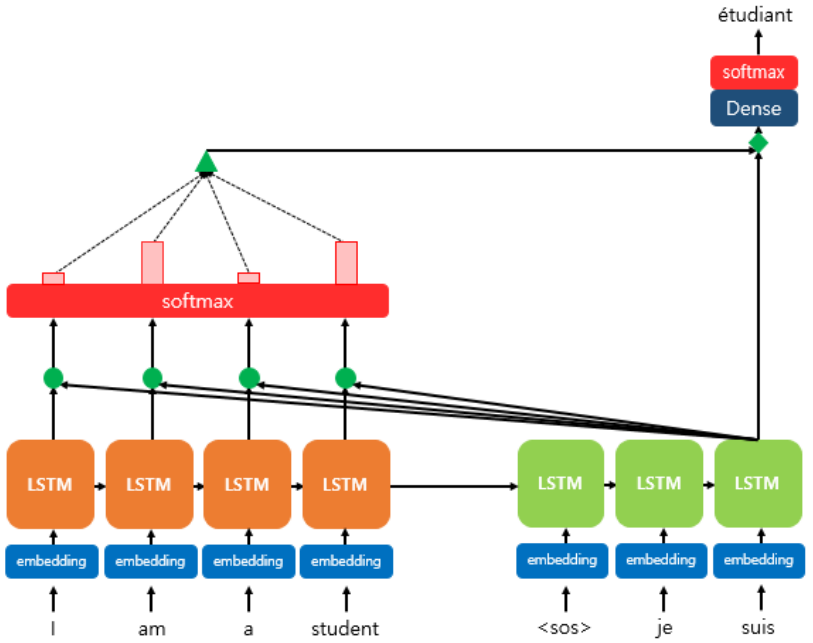
Seq2Seq

- **problem of Seq-2-seq model**
  - 번역되는 매 단어마다 해당 단어를 번역할때 더 중요한 입력단어가 존재
  - ex) je(나는)을 출력하는 과정에서는 I 단어의 정보가 중요
  - 하지만 이미 모든 단어정보가 균등하거나 뒤의 정보가 주요하게 구성된 Context에서 해당 단어만 집중하는 것은 불가능
  - 매 time마다 중요한 단어에 더 가중치를 두는 방법론 제안
    → attention



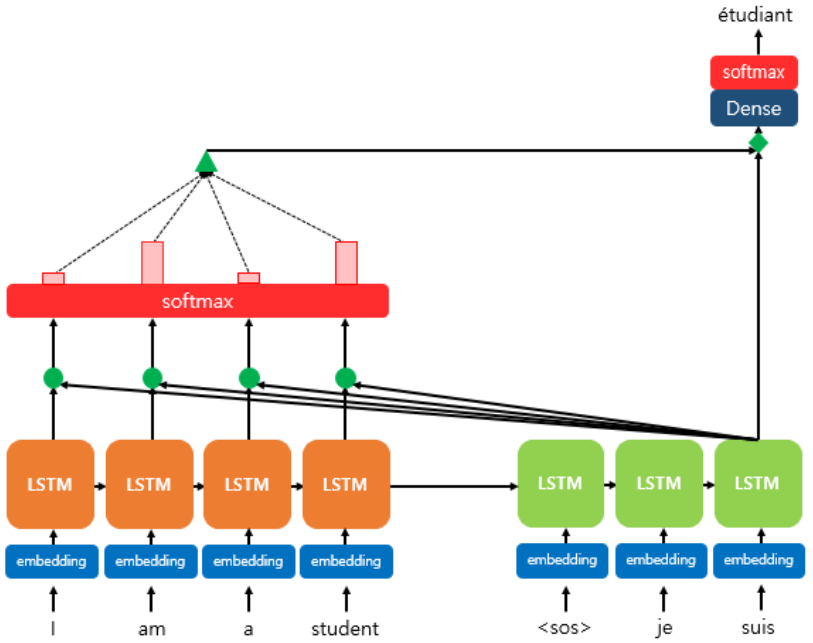
Attention

- **Attention in seq2seq**
  - 모델이 중요한 부분에 "집중"하게 하여 성능을 높이는 방법
  - 디코더에서 출력 단어를 예측하는 매 시점 (time step) 마다, 인코더에서 전체 입력 문장을 다시 한번 참고
  - Ex) les **pauvres** sont demunis -> the **poor** don't have money.
  - poor를 예측할 때 pauvres에 주목하도록



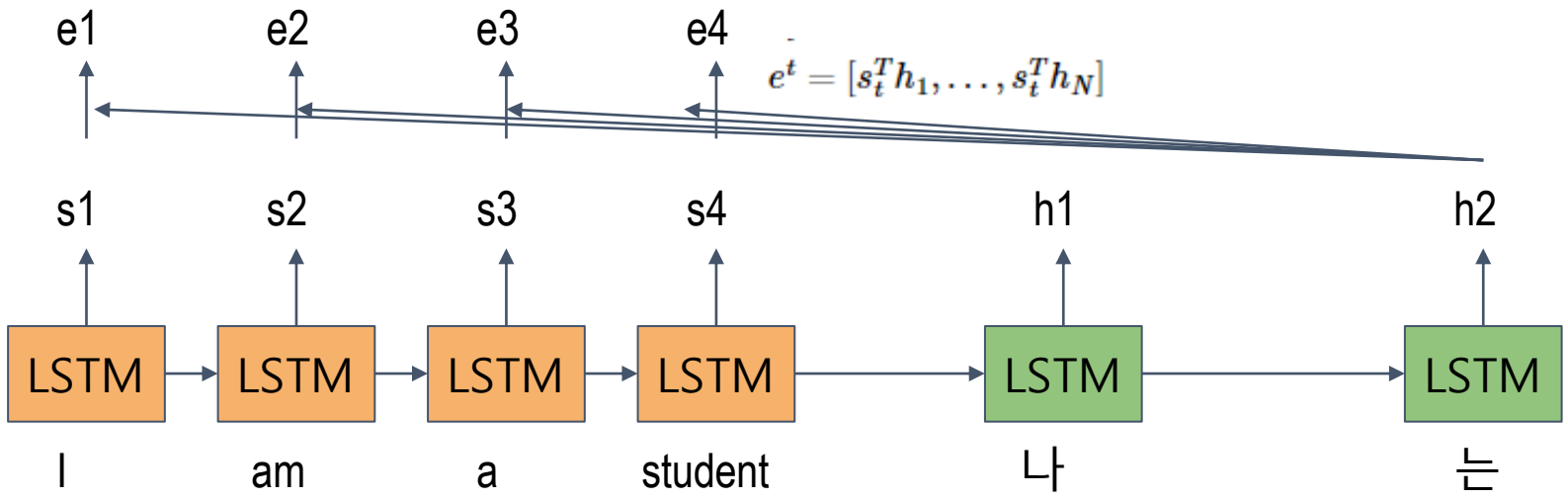
Attention

- **Attention in seq2seq**
  - Seq2seq에서는 현재 시점의 단어에 대한 정답(번역 결과)를 얻어내기 위해 encoder의 각 token중 어떤 token이 가장 연관있는지 계산하여 해당 정보를 사용하는 방법이다.
  - 아래의 그림을 기준으로는 suis 다음 단어를 얻어내기 위해 encoder의 각 단어의 hidden state와의 연산을 통해 각 token의 중요도를 찾아낸다.
  - 이후 weighted sum을 통해 각 token의 중요도에 비중을 둔 vector를 구성 후 최종 결과를 연산한다.



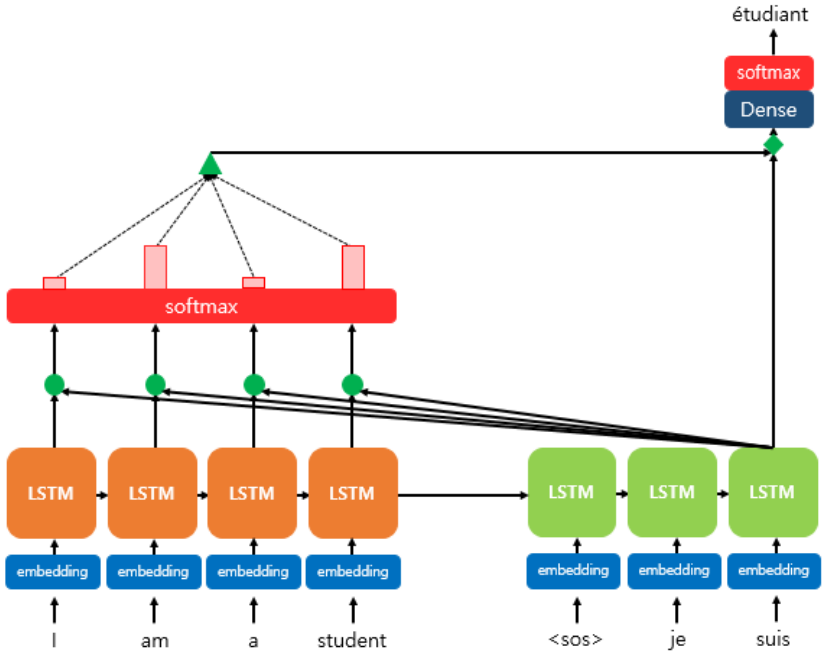
Attention

- **dot-product Attention in seq2seq**
  - attention score를 구하는 과정에서 dot-product 사용시 dot-product attention이라한다.
  - decoder의 현재 time의 hidden state를 encoder의 모든 token의 hidden state와 dot product를 하여 attention score를 구한다.



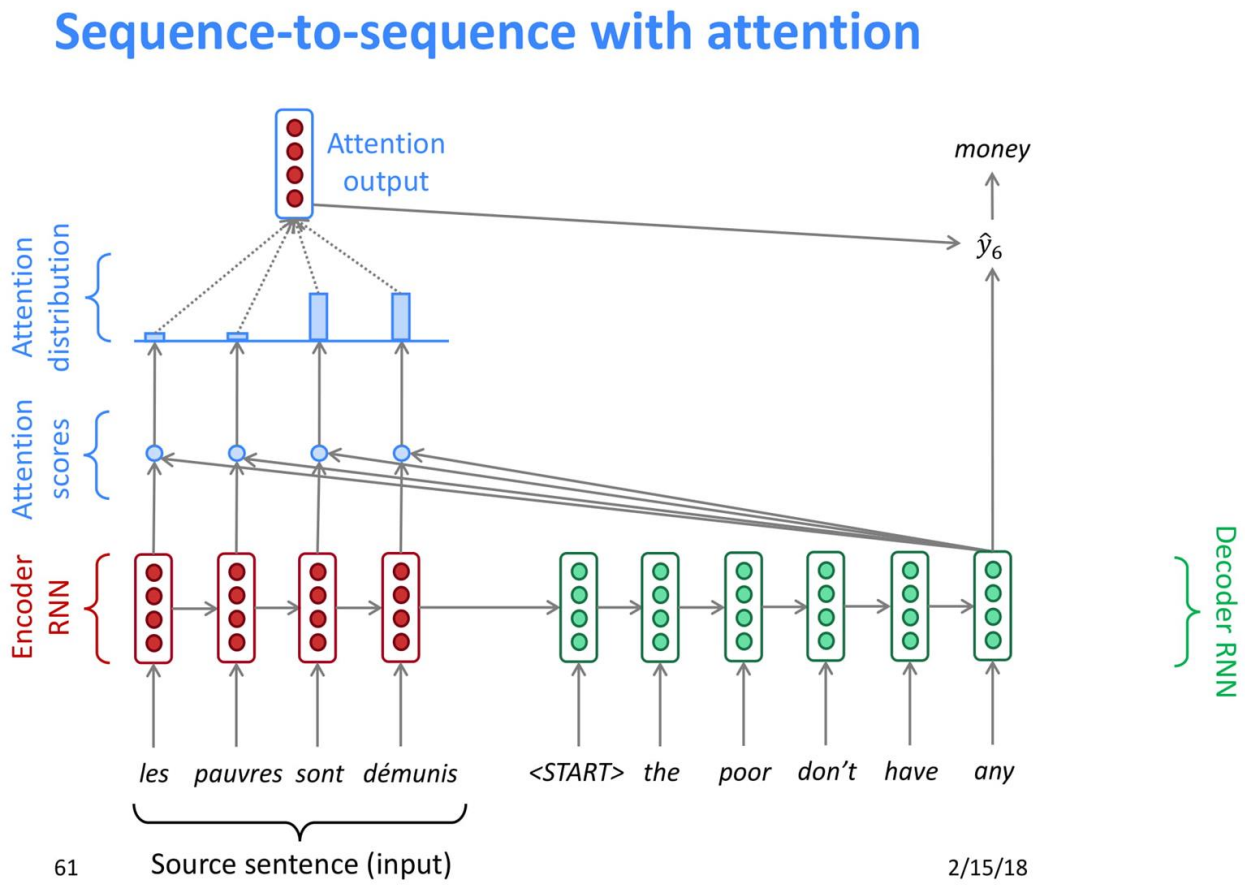
Attention

- **dot-product Attention in seq2seq**
  - 각 attention score에 softmax를 취하면 모든 score의 합이 1이 되게 변환된다.
  - 이를 통해 얻은 attention wieght와 각 encode의 hiddenstate를 곱한후 모든 encodr를 더해준다.
  - 이후 decode의 hidden state와 encode의 hidden state를 concat하여 결과를 도출할때 사용한다.



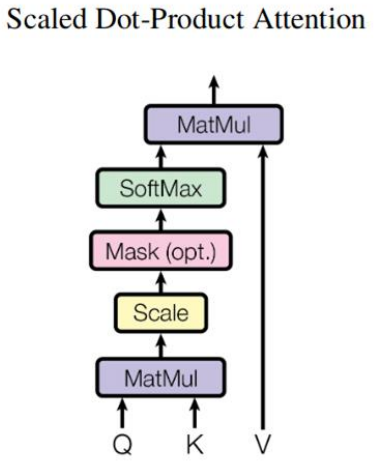
Attention

- **dot-product Attention in seq2seq**



Attention

- **Attention process**
  - attention은 크게 3가지의 input을 통해 연산을 진행한다.
  - Query(Q), Key(K), Value(V)
  - Query는 실험군, Key는 비교 대상, Value는 비교대상의 값이다.
  - Attention을 다시 말하면, 실험군(Query)와 비교대상(Key)을 비교하여 attention score를 구한 후 attention score에 기반하여 Value를 weighted sum하여 attention vector를 구하는 과정이다.



Attention

- **Attention**
  - dot product attention을 수식으로 나타내면 다음과 같다.
    - Q,K,V는 N개 token의 100 dim이라 가정(Nx100)
  - Q와 K를 dot product를 진행하여 attention score를 구한다.

$$Attention\ score(A) = Q * K^T$$

  - softmax를 통해 attention weight를 구한다.

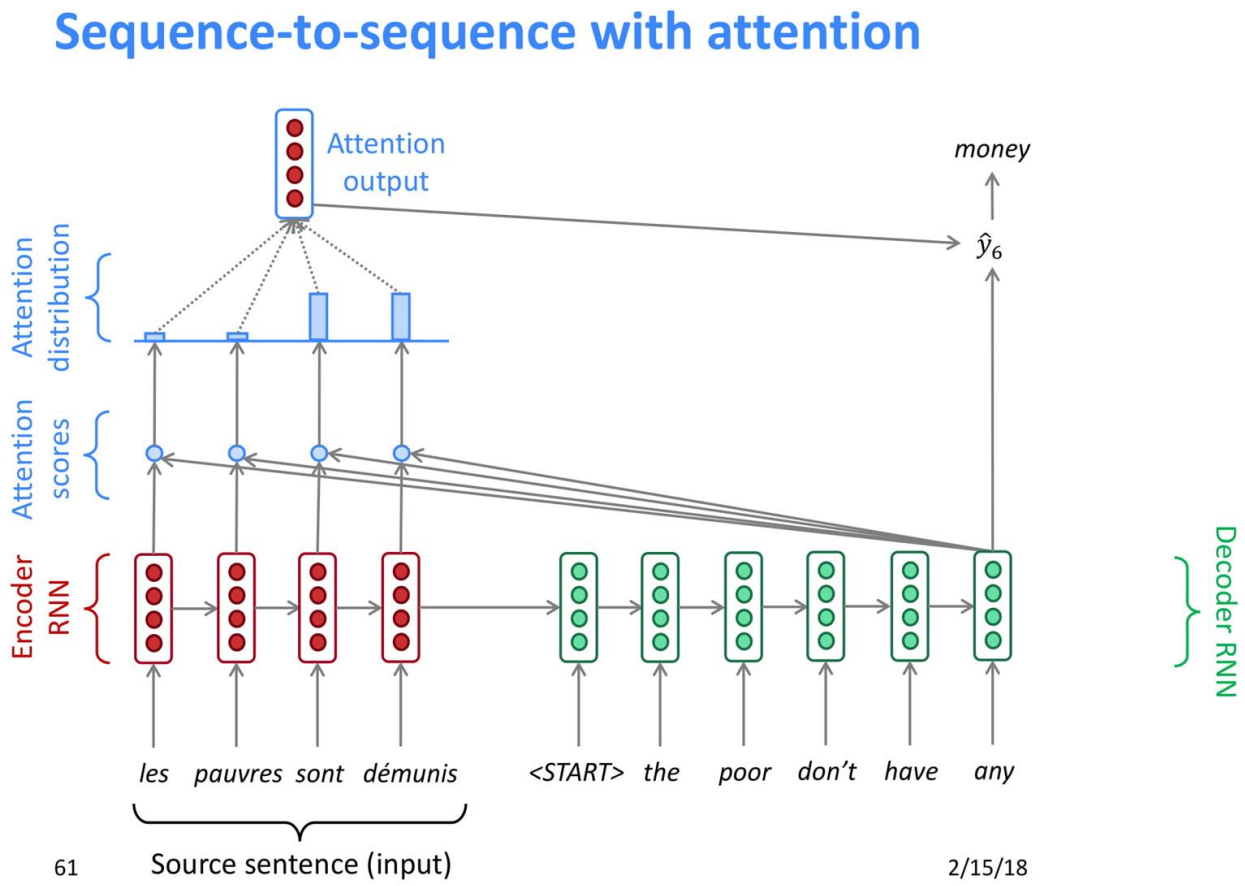
$$Attention\ weight(A') = softmax(A)$$

  - weighted sum을 통해 attention vector를 구한다.

$$Attention\ vector = A' * V$$

Attention

- **dot-product Attention in seq2seq**

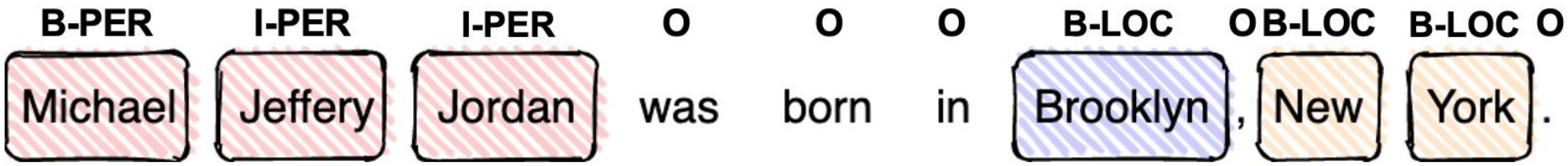


NER with Seq2seq

김균엽

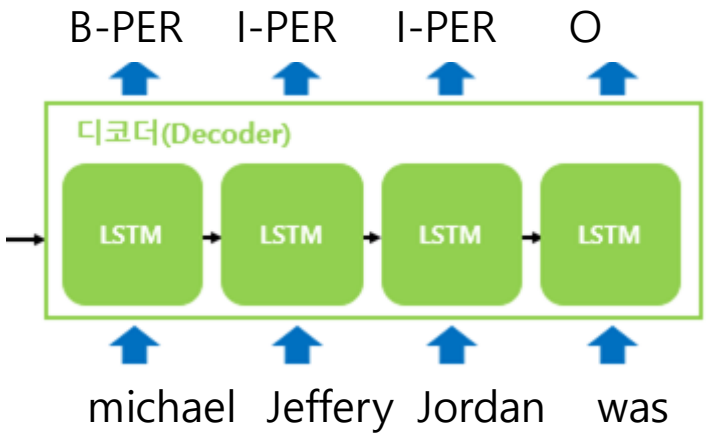
NER

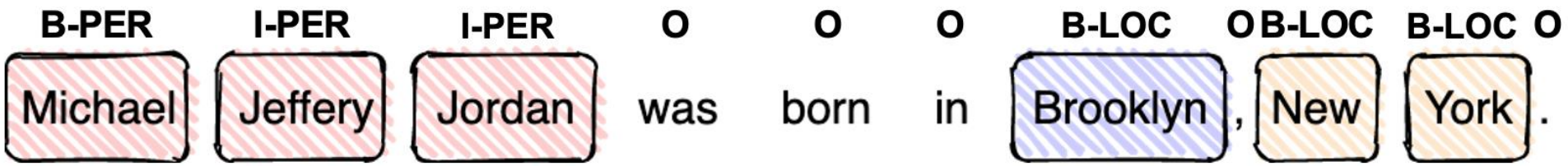
- **Name Entity Recognization**
  - 개체명 인식 task로 각 단어들이 어떠한 개체를 가르키는지(기관, 시간 등)을 나타낸다.
  - BIO tag를 일반적으로 label로 사용한다. 이때 B는 개체명의 시작 token I는 개체명의 내부 token O는 개체명이 아닌 토큰을 이용한다.



NER

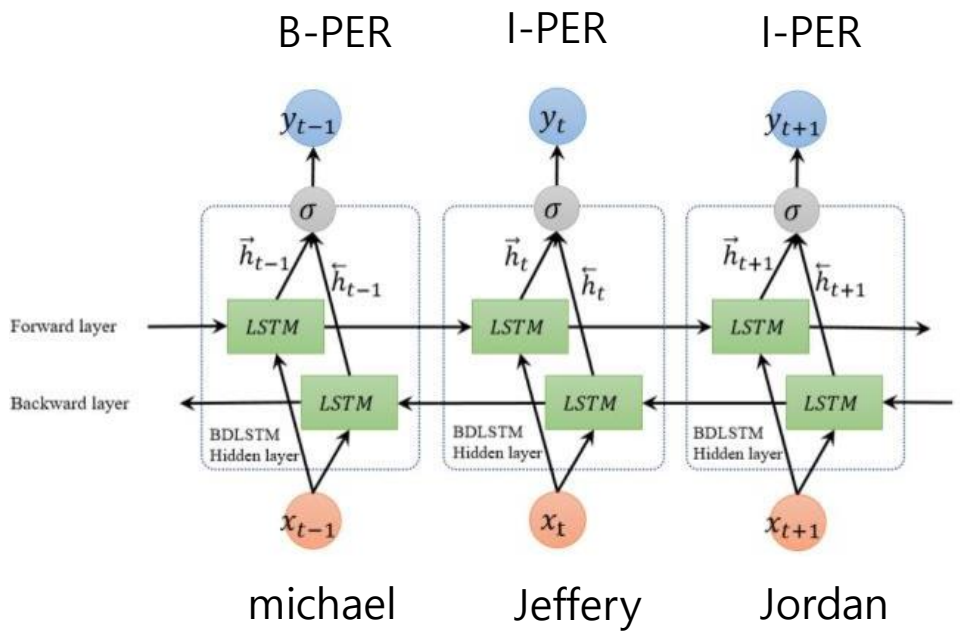
- **Name Entity Recognization**
  - RNN model을 이용하여 NER을 진행한다.

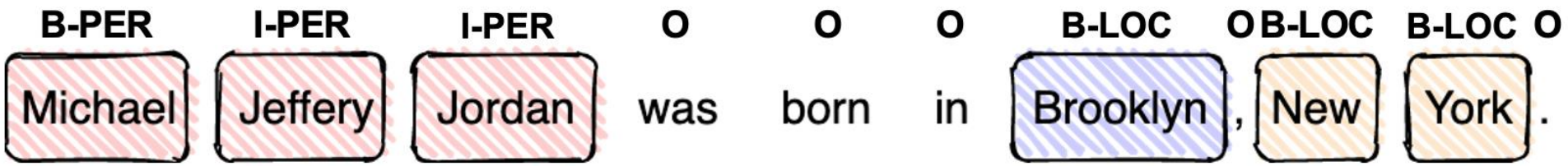




NER

- **Name Entity Recognization**
  - bi directional LSTM model을 이용하여 NER을 진행한다.





NER

- **Name Entity Recognization**
  - Seq2Seq model을 이용하여 NER을 진행한다.