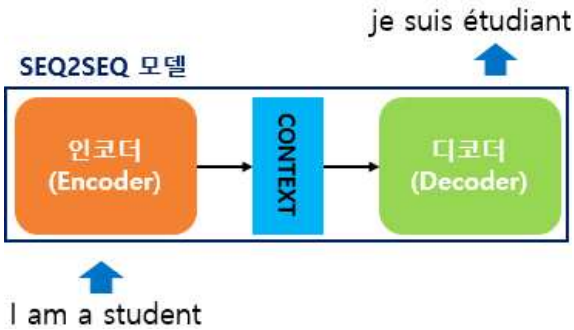
[15차시]

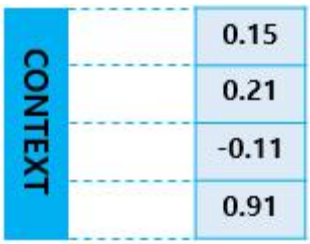
시퀀스-투-시퀀스(Sequence-to-Sequence, seq2seq)

참조; https://www.youtube.com/watch?v=0lgWzluKq1k

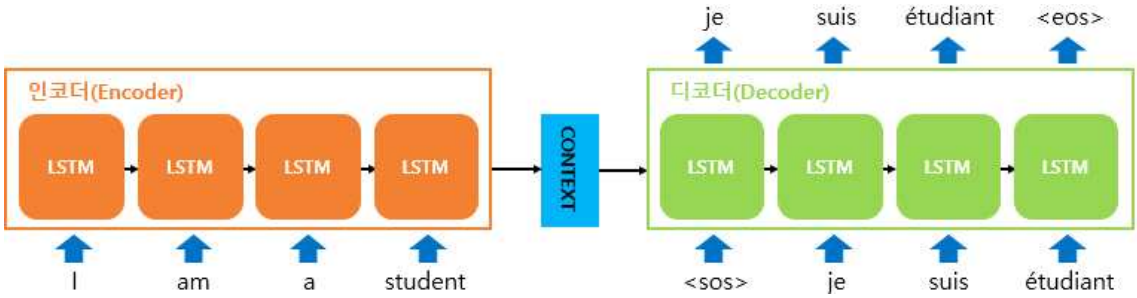
1. seq2seq 모델 내부의 모습



- seq2seq는 인코더와 디코더 아키텍쳐로 구성
- 인코더는 입력 문장의 모든 단어들을 순차적으로 입력받은 뒤에 마지막에 이 모든 단어 정보들을 압축해서 하나의 벡터로 만드는데, 이를 컨텍스트 벡터(context vector)라고 한다.
- 입력 문장의 정보가 하나의 컨텍스트 벡터로 모두 압축되면 인코더는 컨텍스트 벡터를 디코더로 전송한다.
- 디코더는 컨텍스트 벡터를 받아서 번역된 단어를 한 개씩 순차적으로 출력하게 된다.



위의 그림에서는 컨텍스트 벡터를 4의 사이즈로 표현하였지만, 실제 사용되는 seq2seq 모델에서는 보통 수백 이상의 차원을 갖고 있다.



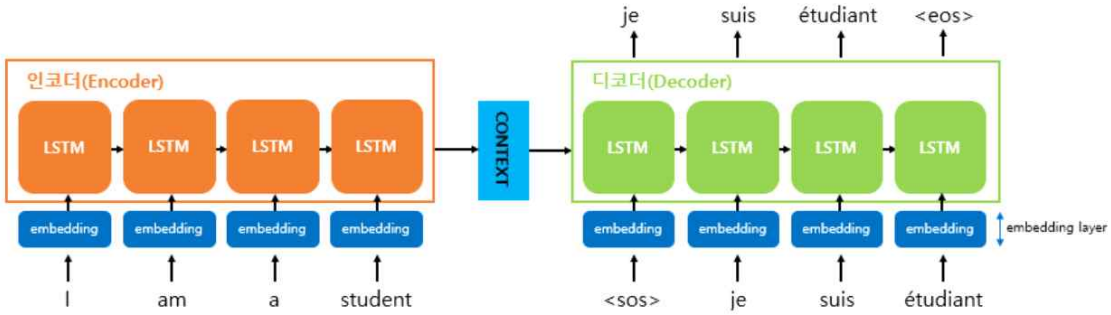
인코더 셀의 경우 입력 문장은 단어 토큰화를 통해서 단어 단위로 쪼개지고, 단어 토큰 각각은 RNN 셀의 각 시점의 입력이 된다. 인코더 RNN 셀은 모든 단어를 입력받은 뒤에 인코더 RNN 셀의 마지막 시점의 은닉 상태를 디코더 RNN 셀로 넘겨주는데 이를 컨텍스트 벡터라고 한다. 컨텍스트 벡터는 디코더 RNN 셀의 첫번째 은닉 상태로 사용된다.

- 디코더는 기본적으로 RNNLM(RNN Language Model)이다.
디코더는 초기 입력으로 문장의 시작을 의미하는 심볼 <sos>가 들어간다. <sos>가 입력되면, 다음에 등장할 확률이 높은 단어를 예측한다. 첫번째 시점(time step)의 디코더 RNN 셀은 다음에 등장할 단어로 je를 예측하고, 예측된 단어 je를 다음 시점의 RNN 셀의 입력으로 입력. 그리고 두번째 시점의 디코더 RNN 셀은 입력된 단어 je로부터 다시 다음에 올 단어인 suis를 예측하고, 또 다시 이것을 다음 시점의 RNN 셀의 입력으로 보낸다. 디코더는 이런 식으로 기본적으로 다음에 올 단어를 예측하고, 그 예측한 단어를 다음 시점의 RNN 셀의 입력으로 넣는 행위를 반복한다. 이 행위는 문장의 끝을 의미하는 심볼인 <eos>가 다음 단어로 예측될 때까지 반복한다.

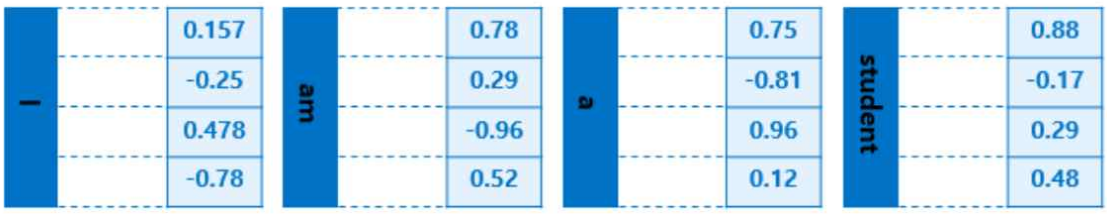
- seq2seq는 훈련 과정과 테스트 과정(또는 실제 번역기를 사람이 쓸 때)의 작동 방식이 조금 다르다. 훈련 과정에서는 디코더에게 인코더가 보낸 컨텍스트 벡터와 실제 정답인 상황인 <sos> je suis étudiant를 입력 받았을 때, je suis étudiant <eos>가 나와야 된다고 정답을 알려주면서 훈련.

반면 테스트 과정에서는 디코더는 오직 컨텍스트 벡터와 <sos>만을 입력으로 받은 후에 다음에 올 단어를 예측하고, 그 단어를 다음 시점의 RNN 셀의 입력으로 넣는 행위를 반복.

- 입, 출력에 쓰이는 단어 토큰들이 있는 부분을 좀 더 확대해보면



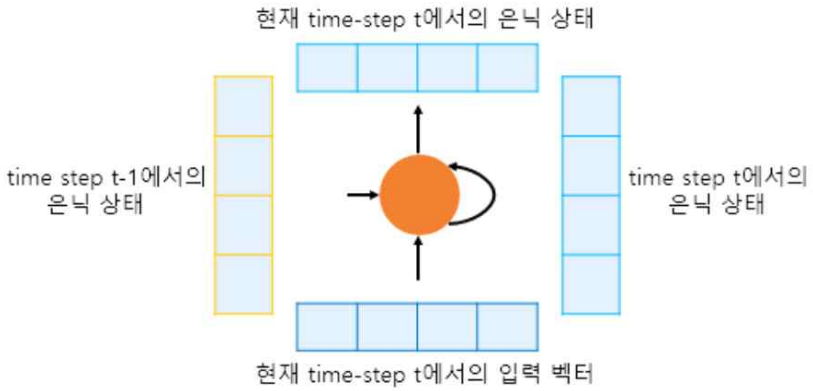
=> 자연어 처리에서 텍스트를 벡터로 바꾸는 방법으로 워드 임베딩이 사용.
즉, seq2seq에서 사용되는 모든 단어들은 워드 임베딩을 통해 임베딩 벡터로서 표현된 임베딩 벡터이다. 위 그림은 모든 단어에 대해서 임베딩 과정을 거치게 하는 단계인 임베딩층(embedding layer)의 모습을 보여준다.



=> 예를 들어 I, am, a, student라는 단어들에 대한 임베딩 벡터는 위와 같은 모습을 가진다. 여기서는 사이즈를 4로 하였지만, 보통 실제 임베딩 벡터는 수백 개의 차원을 가질 수 있다.

RNN 셀)

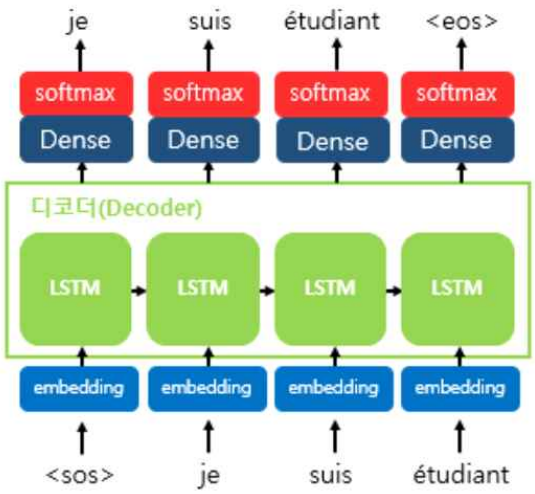
- 하나의 RNN 셀은 각각의 시점(time step)마다 두 개의 입력을 받는다.



=> 현재 시점(time step)을 t라고 할 때, RNN 셀은 t-1에서의 은닉 상태와 t에서의 입력 벡터를 입력으로 받고, t에서의 은닉 상태를 만든다. 이때 t에서의 은닉 상태는 바로 위에 또 다른 은닉층이나 출력층이 존재할 경우에는 위의 층으로 보내거나, 필요 없으면 값을 무시할 수 있다. 그리고 RNN 셀은 다음 시점에 해당하는 t+1의 RNN 셀의 입력으로 현재 t에서의 은닉 상태를 입력으로 보낸다.

- 현재 시점 t에서의 은닉 상태는 과거 시점의 동일한 RNN 셀에서의 모든 은닉 상태의 값들의 영향을 누적해서 받아온 값이라고 할 수 있다. 그렇기 때문에 컨텍스트 벡터는 인코더에서의 마지막 RNN 셀의 은닉 상태값을 말하는 것이며, 이는 입력 문장의 모든 단어 토큰들의 정보를 요약해서 담고 있다고 할 수 있다.

- 디코더는 인코더의 마지막 RNN 셀의 은닉 상태인 컨텍스트 벡터를 첫번째 은닉 상태의 값으로 사용한다. 디코더의 첫번째 RNN 셀은 이 첫번째 은닉 상태의 값과, 현재 t에서의 입력값인 <sos>로부터, 다음에 등장할 단어를 예측. 그리고 이 예측된 단어는 다음 시점인 t+1 RNN에서의 입력값이 되고, 이 t+1에서의 RNN 또한 이 입력값과 t에서의 은닉 상태로부터 t+1에서의 출력 벡터. 즉, 또 다시 다음에 등장할 단어를 예측하게 된다.
  - 디코더가 다음에 등장할 단어를 예측하는 부분을 확대해보면



=> seq2seq 모델은 선택될 수 있는 모든 단어들로부터 하나의 단어를 골라서 예측하는데 이를 예측하기 위해서 소프트맥스 함수를 사용한다. 디코더에서 각 시점(time step)의 RNN 셀에서 출력 벡터가 나오면, 해당 벡터는 소프트맥스 함수를 통해 출력 시퀀스의 각 단어별 확률값을 반환하고, 디코더는 출력 단어를 결정한다.