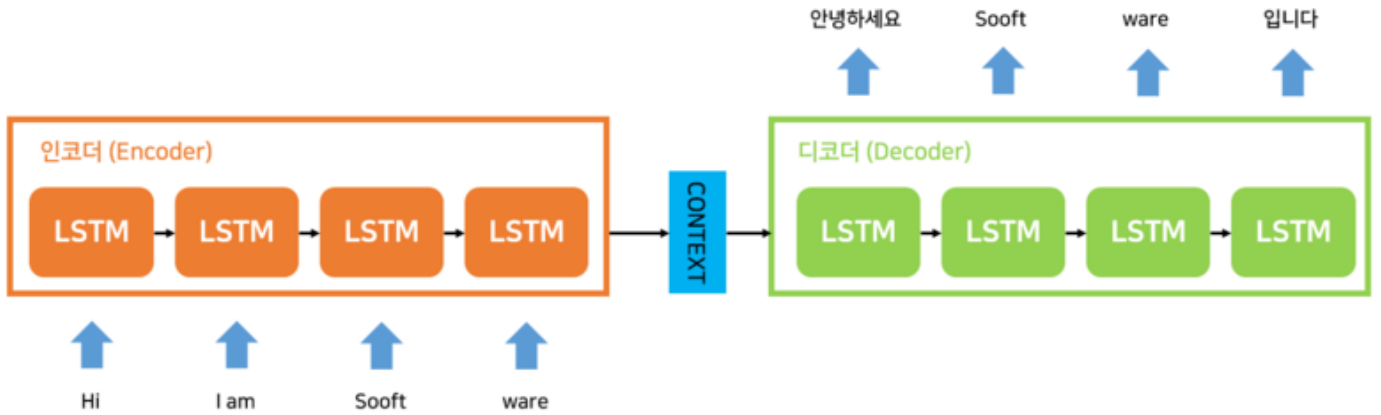
Sequence to Sequence Learning with Neural Networks (NIPS 2014)

박창협

Abstract

- 본 논문에서는 LSTM을 활용한 seq2seq 기계 번역 아키텍처를 제안함.
- 모델 구조에서 encoder/ decoder가 있는데 Lstm 을 이용.
- Lstm을 이용 했기 때문에 긴 문장에서도 성능이 높게 나옴.
- 본 논문에서 사용 된 데이터 셋은 WMT` 14 데이터 셋으로 영어를 불어로 번역하는 태스크에서 BLEU 스코어가 34.8
- 전통적인 통계적 기계 번역 모델(SMT)는 bleu 스코어가 33.3
- 추가적으로 입력문장에 포함되어 있는 단어들의 순서를 바꾸는 것이 더욱 성능을 비약적으로 향상 시켰다.



Introduction

- 이 때 (논문 나오기 전) 까지의 DNN(Deep Neural Networks)은 음성 인식이나, 사물 인식등에서 꾸준히 높은 성능을 보여 왔음.
- Dnn은 통계적 모델과 **유사하지만 더욱더 복잡한 함수(어려운 문제)를 학습(해결)하는 과정에서 우수**
- 그러나 기존의 DNN은 입력과 출력 차원이 일반적으로 고정 되어있기 때문에 분명한 한계가 존재
  - 특히, 기계번역 같은 **sequencial problem** 의 경우
- 이러한 문제를 해결 하기 위해서는 입,출력 차원이 가변적일 필요가 있는데, 본 논문이 나오기 이전에는 힘들었다.
- 본 논문에서는 **Lstm만으로 seq to seq 문제를 잘 해결할 수 있다**고 언급하고 있다.

Introduction

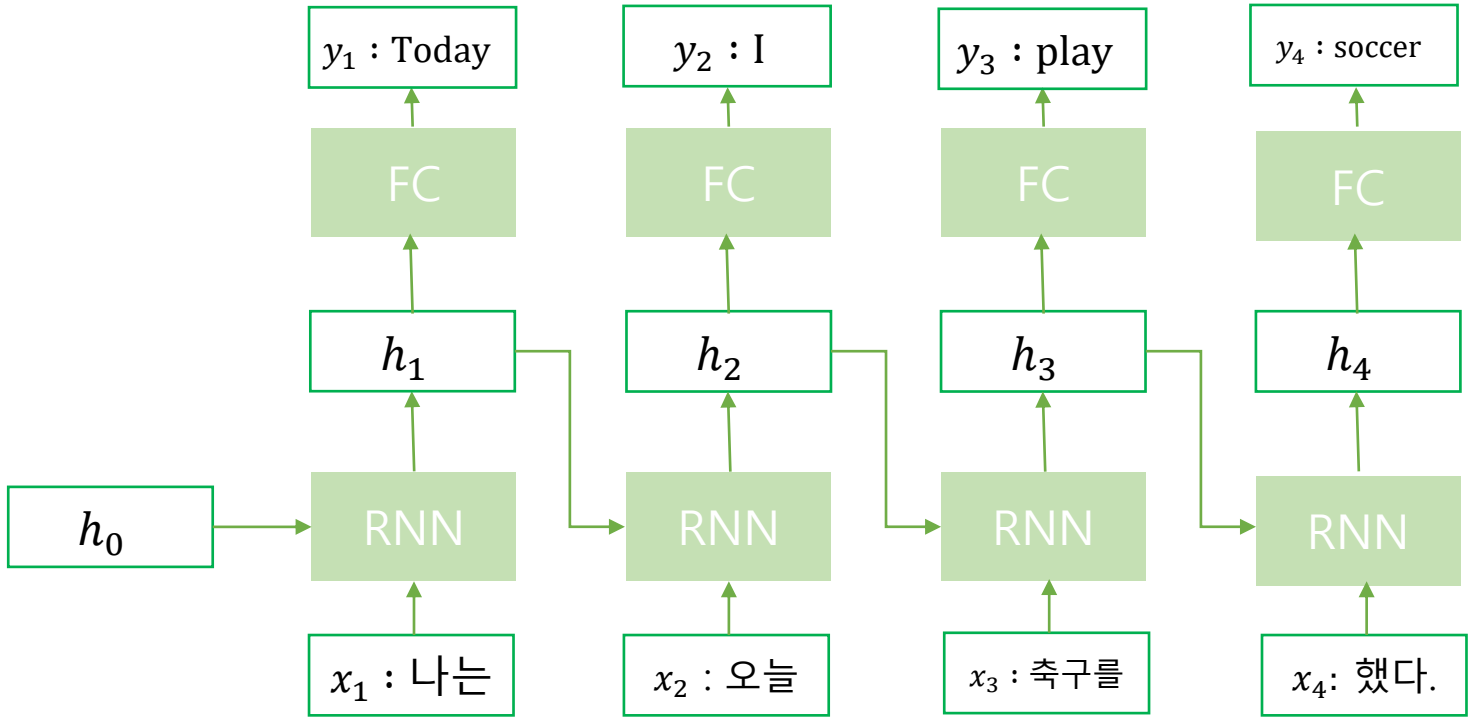
- 학습 과정
  1. 입력 데이터를 인코딩을 통해 고정된 차원의 context vector를 뽑는다.
  2. Context vector를 다시 디코더에 넣어 준다.
  3. 번역결과를 출력한다.
- 인코더와 디코더는 **lstm 4개를 중첩한 구조로 설계** 됨.
- 많은 논문에서 사용된 **wtm`14 데이터 셋에 대해서 34.81의 bleu score**를 얻음.
- 또한 **lstm과 smt를 같이 이용 했을 때,bleu score가 36.5** 까지 오름.
  - 논문 이전에 SOTA 성능이 37
- Lstm의 경우 긴 문장에서도 성능이 어느정도 보장 되기 때문에, **실제로 제안된 모델 아키텍쳐가 긴 Sequency 에서 잘 동작한다고 언급 되어 있습니다.**

SMT

- 전통적인 통계적 언어 모델은 **카운트 기반의 접근**을 사용함
  - P(지낸다|친구와 친하게) = $\frac{count(\textbf{친구와 친하게 지낸다})}{count(\textbf{친구와 친하게})}$
- 실제로 모든 문장에 대한 확률을 가지고 있으려면 **엄청난 양의 데이터가 필요.**
- 예를들어 “친구와 친하게”라는 시퀀스 자체가 학습 데이터에 존재하지 않는다면 ?
- 또한 긴문장을 처리하기가 매우 어렵다.
  - P(나는 공부를 마치고 집에서 밥을 먹었다) = P(나는) * P(공부를|나는) * P(마치고|나는 공부를) *P(집에서 | 나는 공부를 마치고) * P(밥을|나는 공부를 마치고 집에서) * P(먹었다 | 나는 공부를 마치고 집에서 밥을)
- 현실적인 해결책으로 n-GRAM 언어 모델 이 사용됨
  - 인접한 일부 단어만 고려하는 아이디어

RNN
기반의 번역 과정

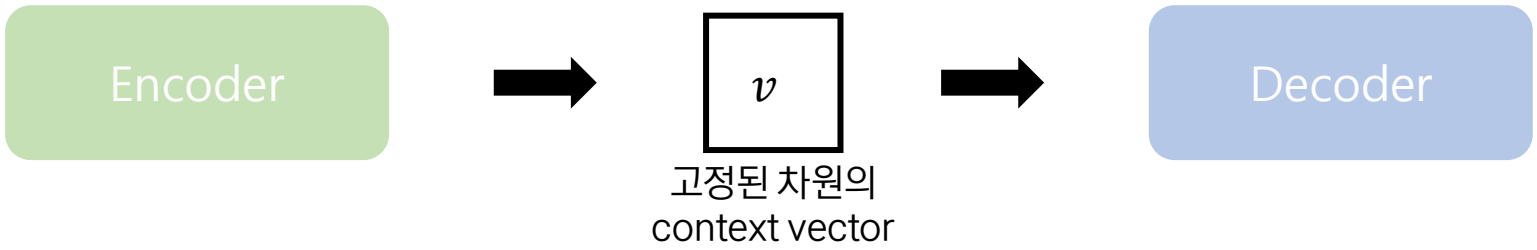
- 전통적인 초창기 RNN 기반의 언어 모델에서 번역이 이뤄지는 과정은 아래와 같다.
- RNN 기반의 기계번역은 입력과 출력의 크기가 같다고 가정한다.



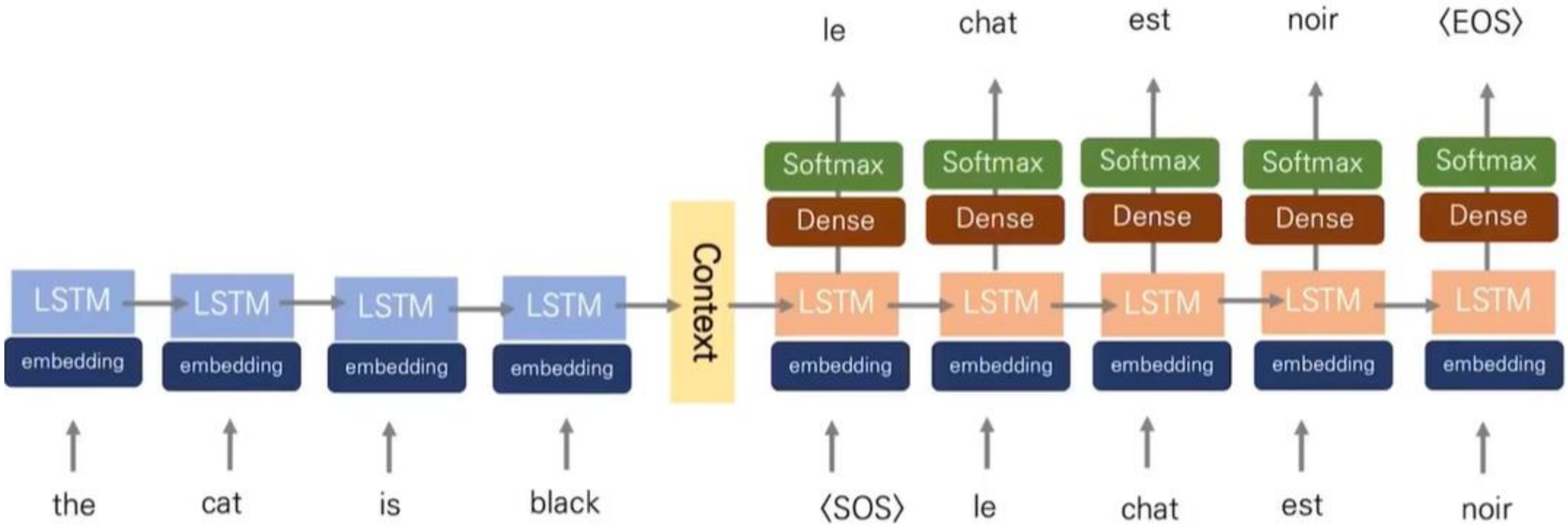
Sequence to Sequence Learning with Neural Networks (NIPS 2014)

RNN 기반의 seq2seq

- 전통적인 초창기 RNN 기반의 언어 모델은 다양한 한계점 존재
  - 이를 해결하기 위해 Encoder가 고정된 크기의 문맥 벡터(context vector)를 추출 하도록함.
  - 이후에 Context vector를 이용해서 Decoder가 번역 결과를 출력
  - 본 Seq2Seq 논문에서는 LSTM을 이용해 문맥 벡터를 추출하도록 하여 성능을 향상시킴.
    - LSTM은 긴 문장에도 유연하게 작동하기 때문.

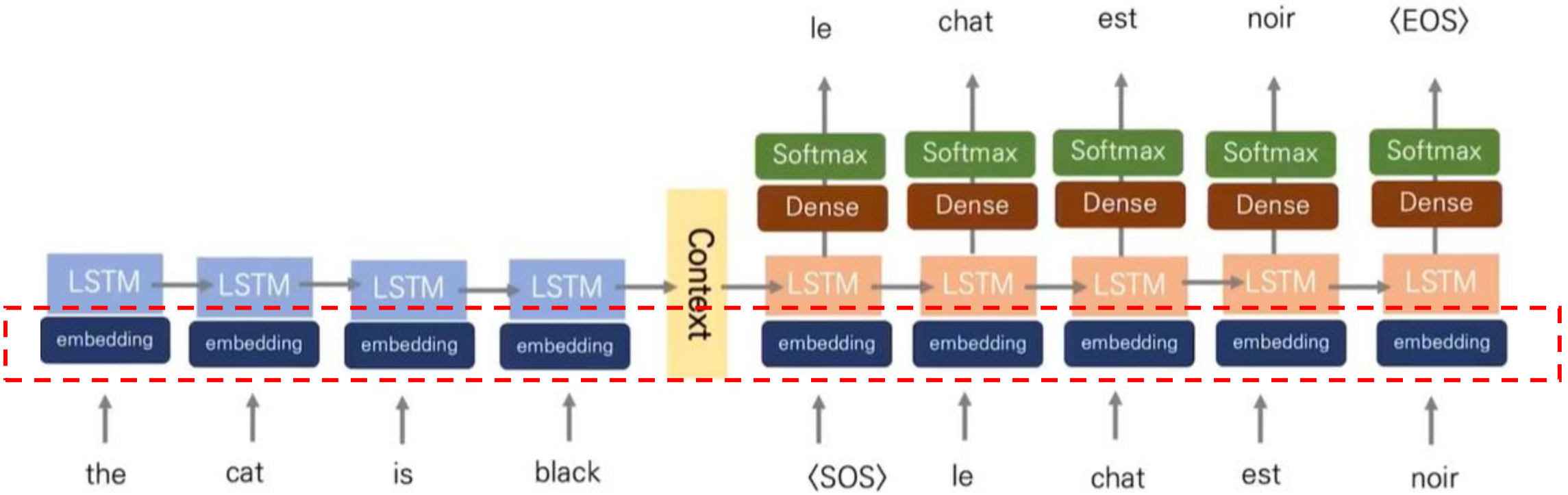


Model Architecture

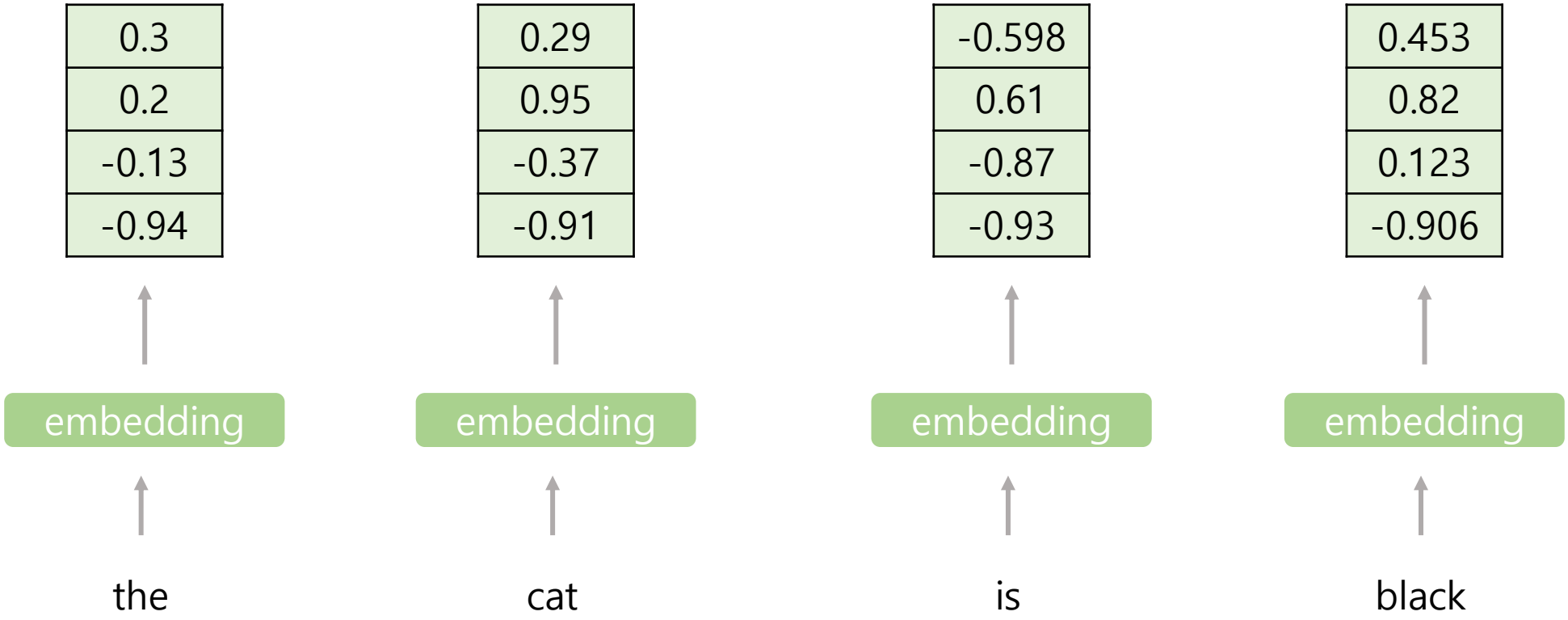


Model Architecture

1) Embedding

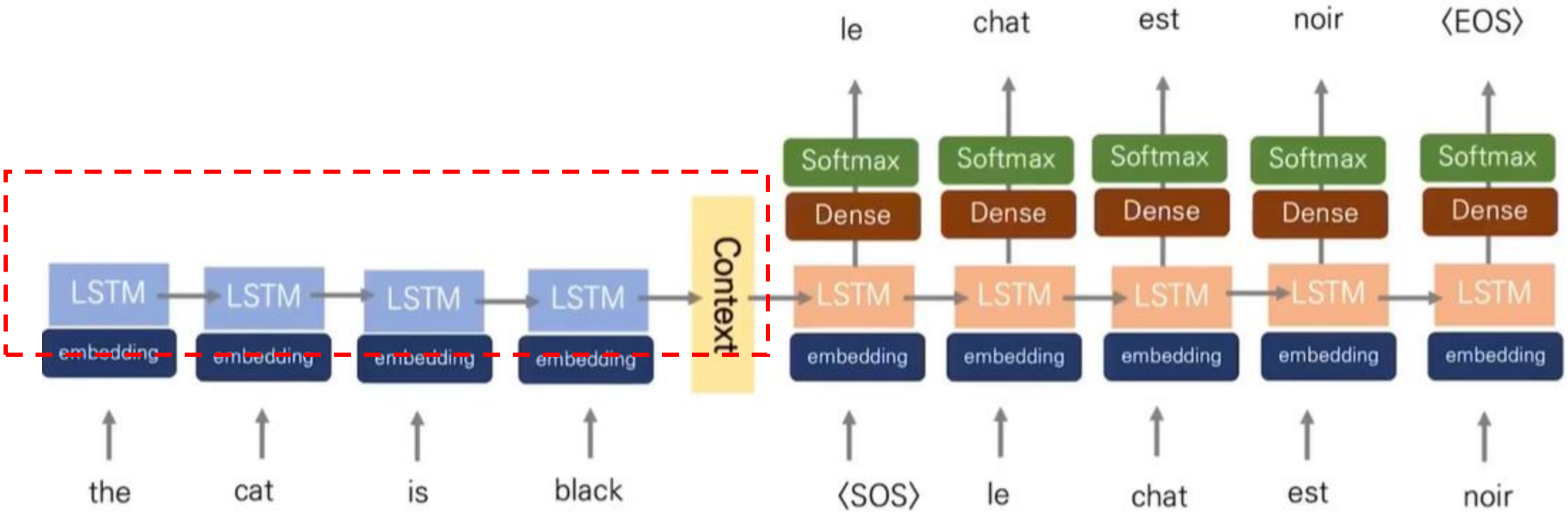


1) Embedding

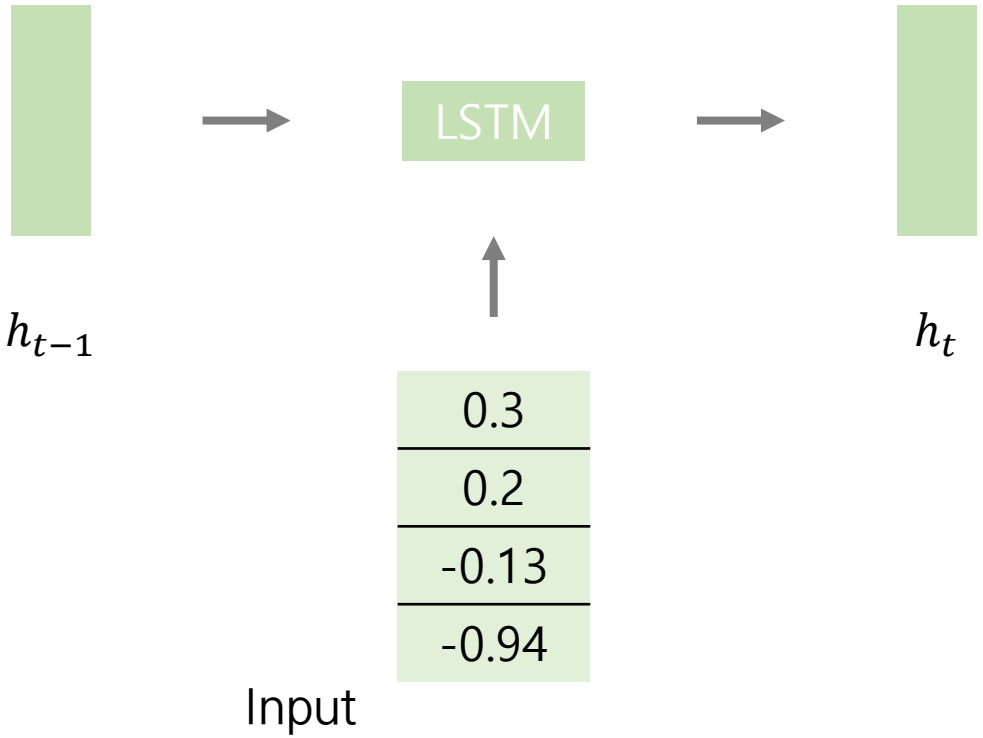


단어를 각각 임베딩 하여 encoder와 decoder의 input으로 활용

2) Encoder

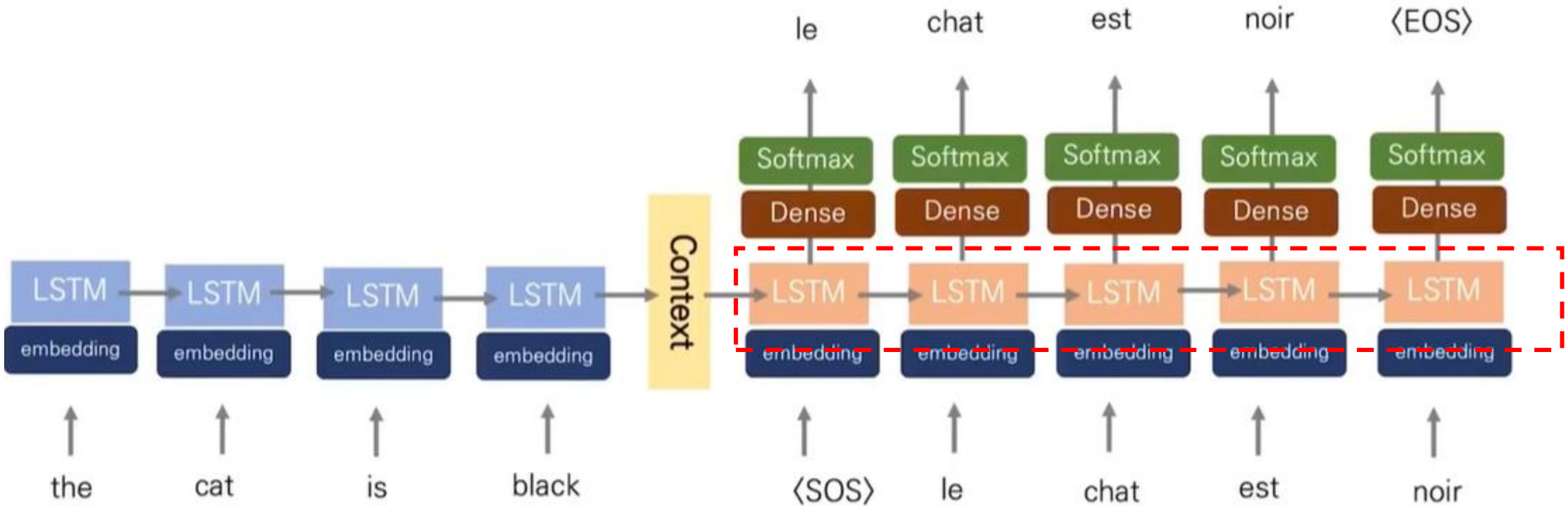

le
chat
est
noir
⟨EOS⟩
Softmax
Softmax
Softmax
Softmax
Softmax
Dense
Dense
Dense
Dense
Dense
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
embedding
embedding
embedding
embedding
embedding
embedding
embedding
embedding
embedding
embedding
Context
the
cat
is
black
⟨SOS⟩
le
chat
est
noir

2) Encoder

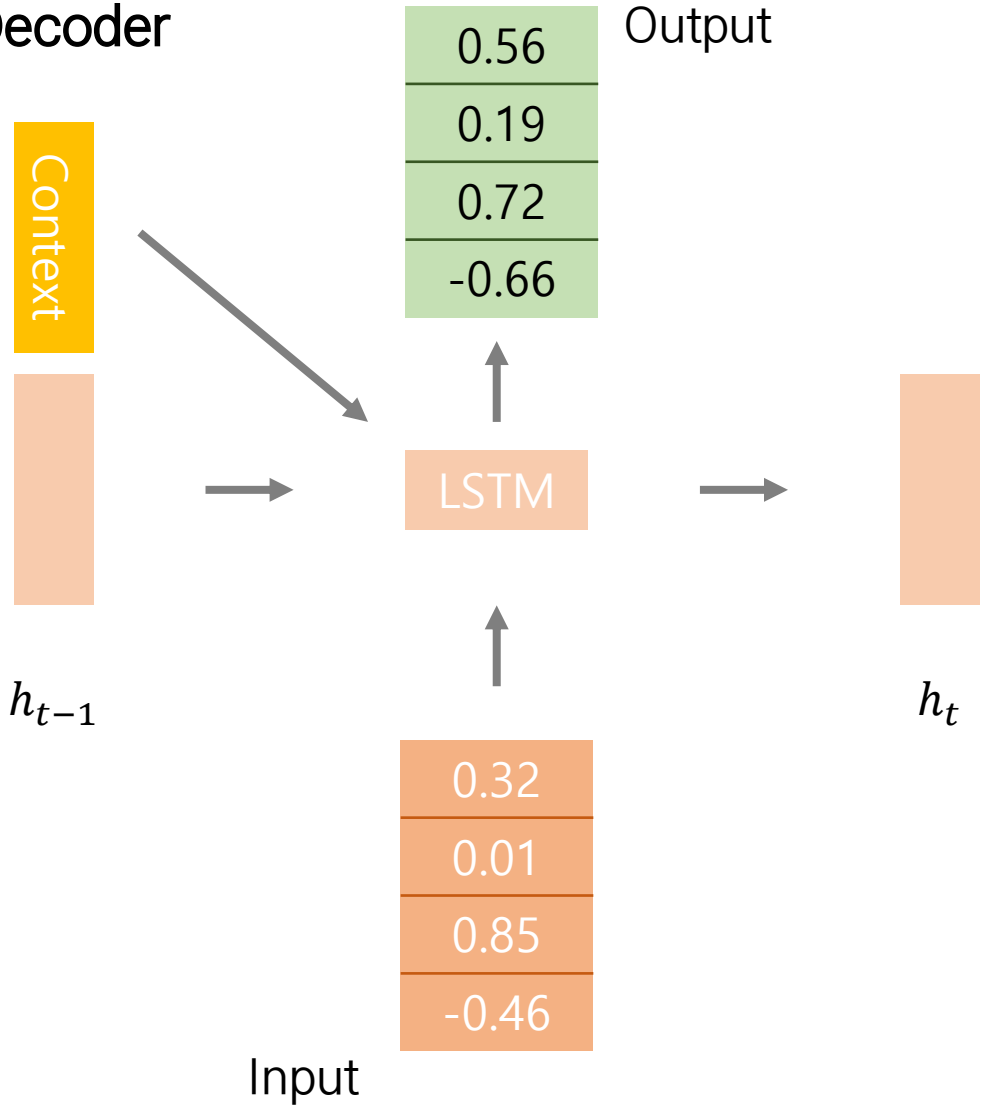


- ✓ 임베딩된 단어들 순차적으로 encoder의 입력으로 활용
- ✓ 임베딩 단어와 hidden state를 LSTM cell 로 연산해서 hidden state 업데이트
- ✓ 마지막 hidden state를 context vector로 지정

3) Decoder

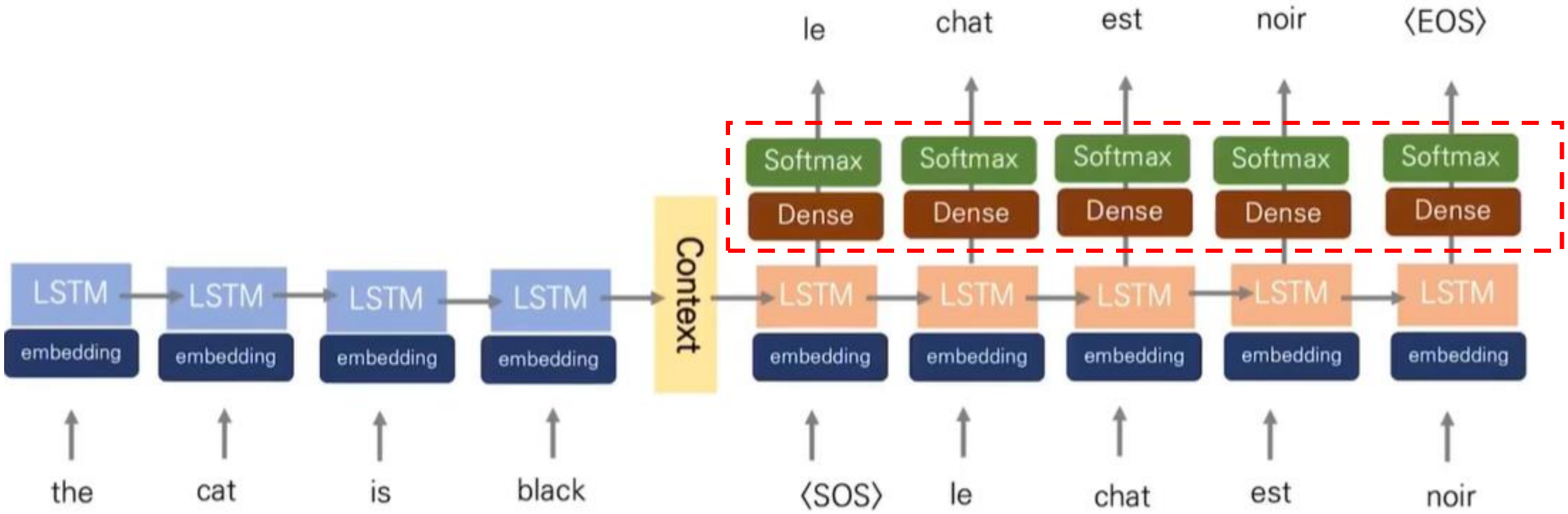

le
chat
est
noir
⟨EOS⟩
Softmax
Softmax
Softmax
Softmax
Softmax
Dense
Dense
Dense
Dense
Dense
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
LSTM
embedding
embedding
embedding
embedding
Context
embedding
embedding
embedding
embedding
embedding
the
cat
is
black
⟨SOS⟩
le
chat
est
noir

3) Decoder

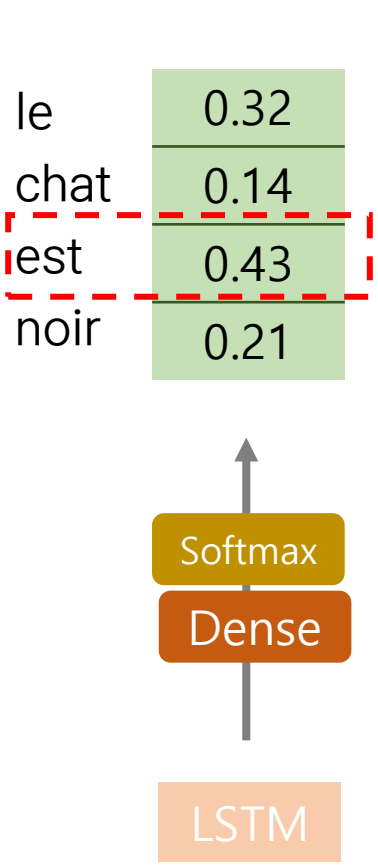


- ✓ 임베딩된 target 문장 단어들 순차적으로 decoder의 입력으로 활용
- ✓ 임베딩 단어와 hidden state 를 LSTM cell로 연산해서 hidden state를 업데이트하고 다음에 나올 확률이 높은 단어 예측
- ✓ Target sentence의 임베딩은 학습 시에만 필요

4) Softmax Layer



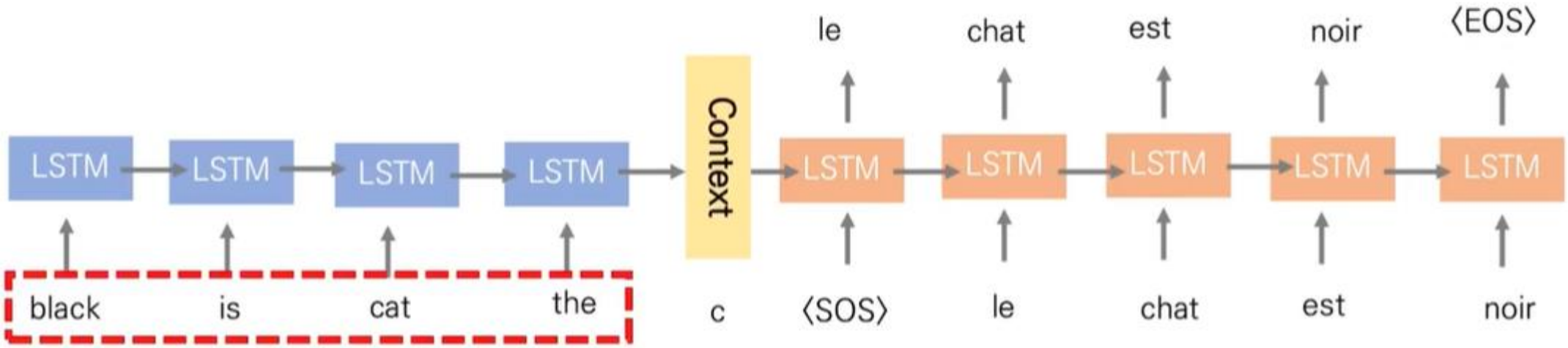
4) Softmax Layer



- ✓ Lstm의 output을 dense layer에 거치고, softmax 를 활용해 각 단어가 나올 확률값을 계산
- ✓ Softmax layer 를 거치며 각 토큰이 나올 확률값이 output 으로 나오게됨
- ✓ 이 예시에서는 Est가 나올 확률이 가장 높으므로, 다음 단어는 'est'로 선택

Experiment

1) Reversing the Source Sentences



- Source sentence의 **순서를 거꾸로 사용했을 때, BLEU score가 25.9 –〉 30.6 으로 향상**
- 이후 많은 논문에서도 source sentence 순서를 reverse 하여 학습하는 방식을 사용 하여 성능을 높임
- 성능 향상 이유 : source 의 첫 단어와 target 의 첫 단어와의 거리가 가까워지기 때문이라 추측.

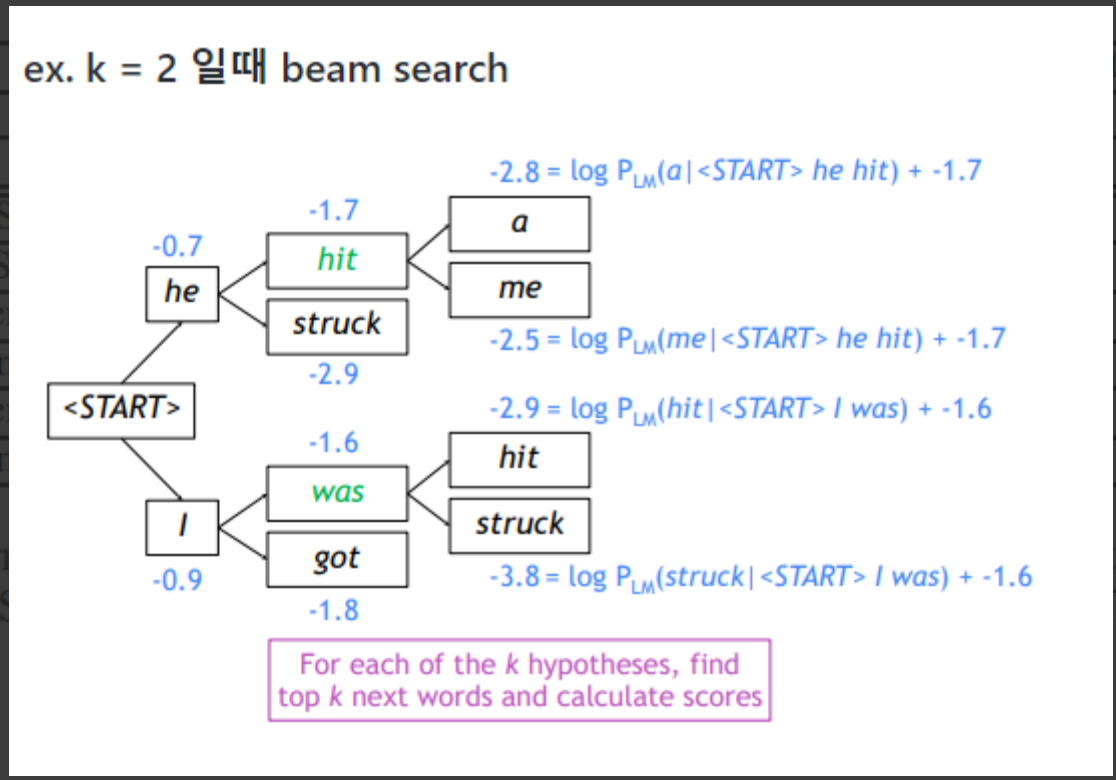
Experiment

2) Traning details

- LSTM에 들어가는 **파라미터 들은 -0.08 과 0.08 사이의 값으로 uniform 분포를** 따르도록 초기화
- Stochastic gradient descent (SGD) 사용.
- Batch size : 128 -〉 학습 과정에서 사용하는 문장의 개수가 128개씩 입력됨.
  - 한번에 들어가는 문장의 길이를 최대한 맞춰서 효율적으로 학습 진행
  - 문장의 길이가 다르면 padding 처리해야되기 때문.

Experiment

3) Results



✓ 순수하게 lstm만 앙상블해도 높은 성능을 보이는 것을 확인 할 수 있음.

Experiment

3) Results

Method	test BLEU score (ntst14)
Bahdanau et al. [2]	28.45
Baseline System [29]	33.30
Single forward LSTM, beam size 12	26.17
Single reversed LSTM, beam size 12	30.59
Ensemble of 5 reversed LSTMs, beam size 1	33.00
Ensemble of 2 reversed LSTMs, beam size 12	33.27
Ensemble of 5 reversed LSTMs, beam size 2	34.50
Ensemble of 5 reversed LSTMs, beam size 12	**34.81**

Table 1: The performance of the LSTM on WMT'14 English to French test set (ntst14). Note that an ensemble of 5 LSTMs with a beam of size 2 is cheaper than of a single LSTM with a beam of size 12.

✓ 순수하게 lstm만 앙상블해도 높은 성능을 보이는 것을 확인 할 수 있음.

Conclusion

- **Lstm을 깊게 쌓아서 기존에 존재했던 통계적 기계 번역 방법 (SMT) 보다 더 좋은 성능**을 낼 수 있다는 것을 실험을 통해 잘 보여줌.
- **입력 단어의 순서를 바꾸는 것이 기존 순서를 그대로 유지하는 것보다 더 성능을 개선** 할 수 있음.
- 논문 내용을 요약하면,
  - ✓ Input data를 lstm으로 이뤄진 encoder에 넣어 conext vector 생성
  - ✓ Context vector가 마찬가지로 lstm으로 구성된 decoder를 거쳐서 번역 결과를 출력함.
  - ✓ 이때 input data의 입력 순서를 바꿔주면 더 성능이 개선됨.

감사합니다.