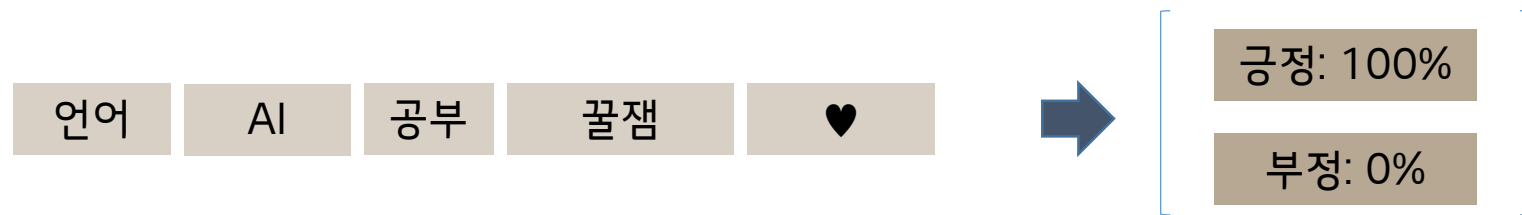


M7. Attention

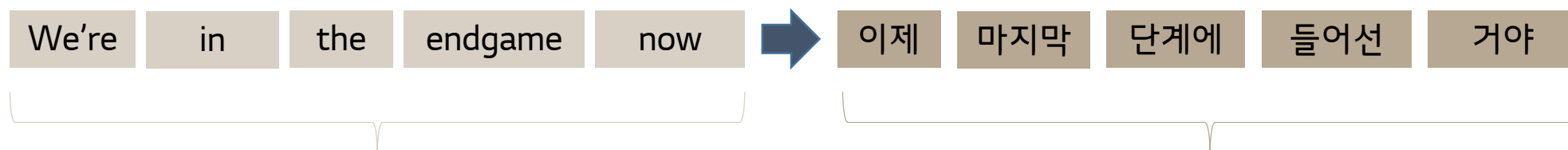
Until Now...



시퀀스 인풋

단일 아웃풋

Seq2Seq : Sequence to Sequence

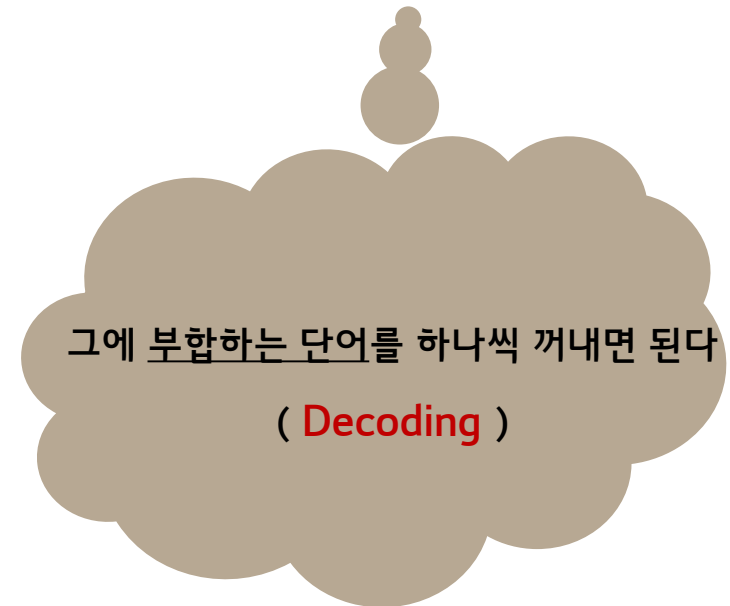
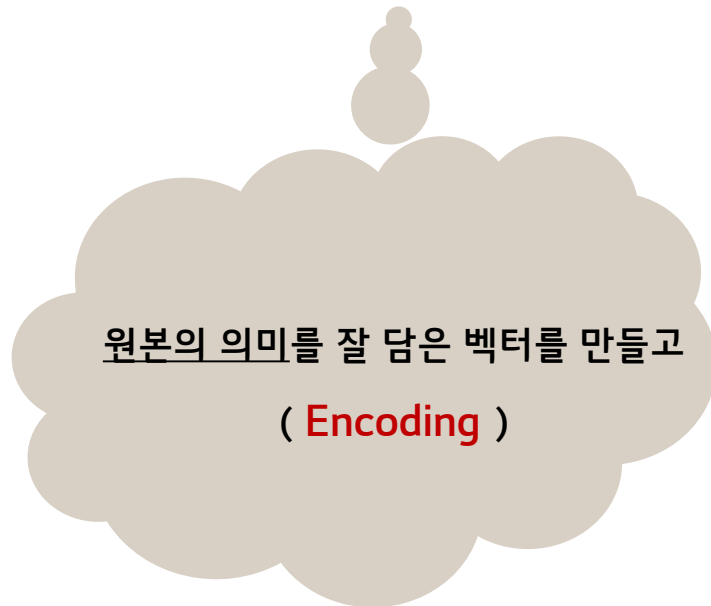
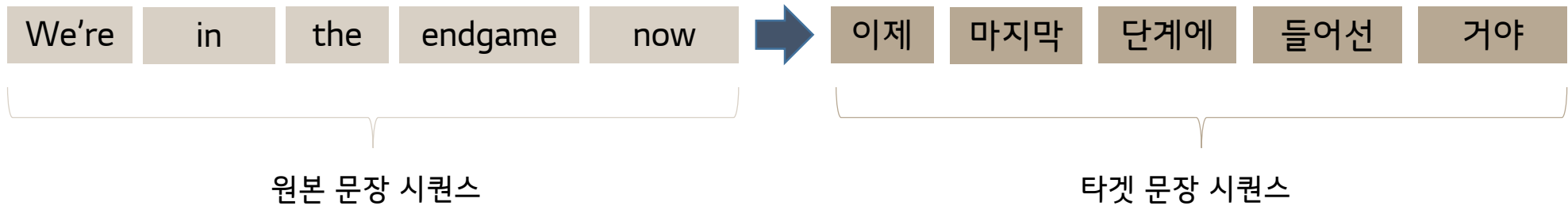


시퀀스 인풋

시퀀스 아웃풋

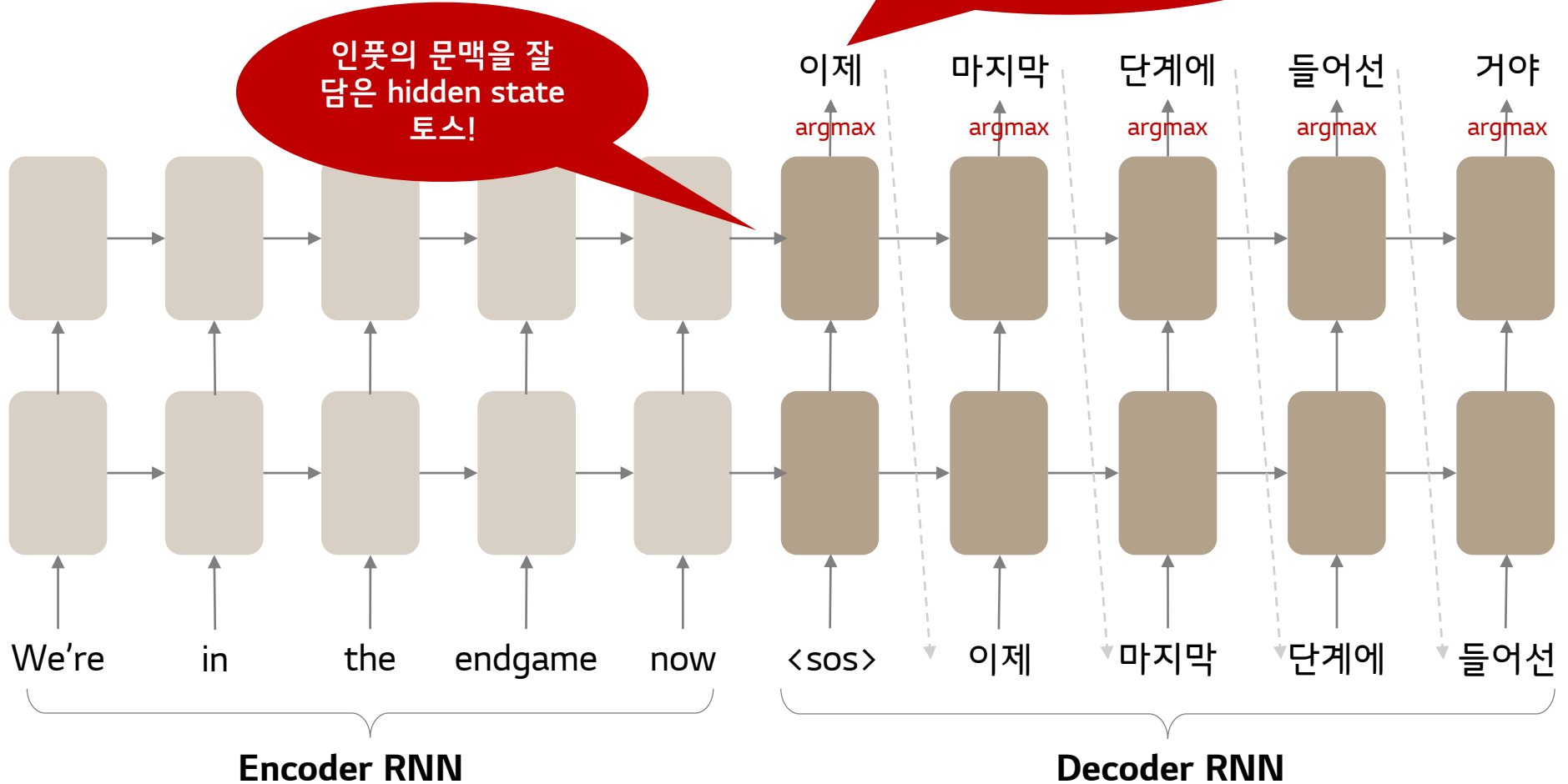
Seq2Seq : Sequence to Sequence

- 예시) 번역 과제를 어떻게 수행할 수 있을까?



RNN을 사용한 Seq2Seq 모델링

- 2개의 RNN을 인코더, 디코더로 사용하는 구조



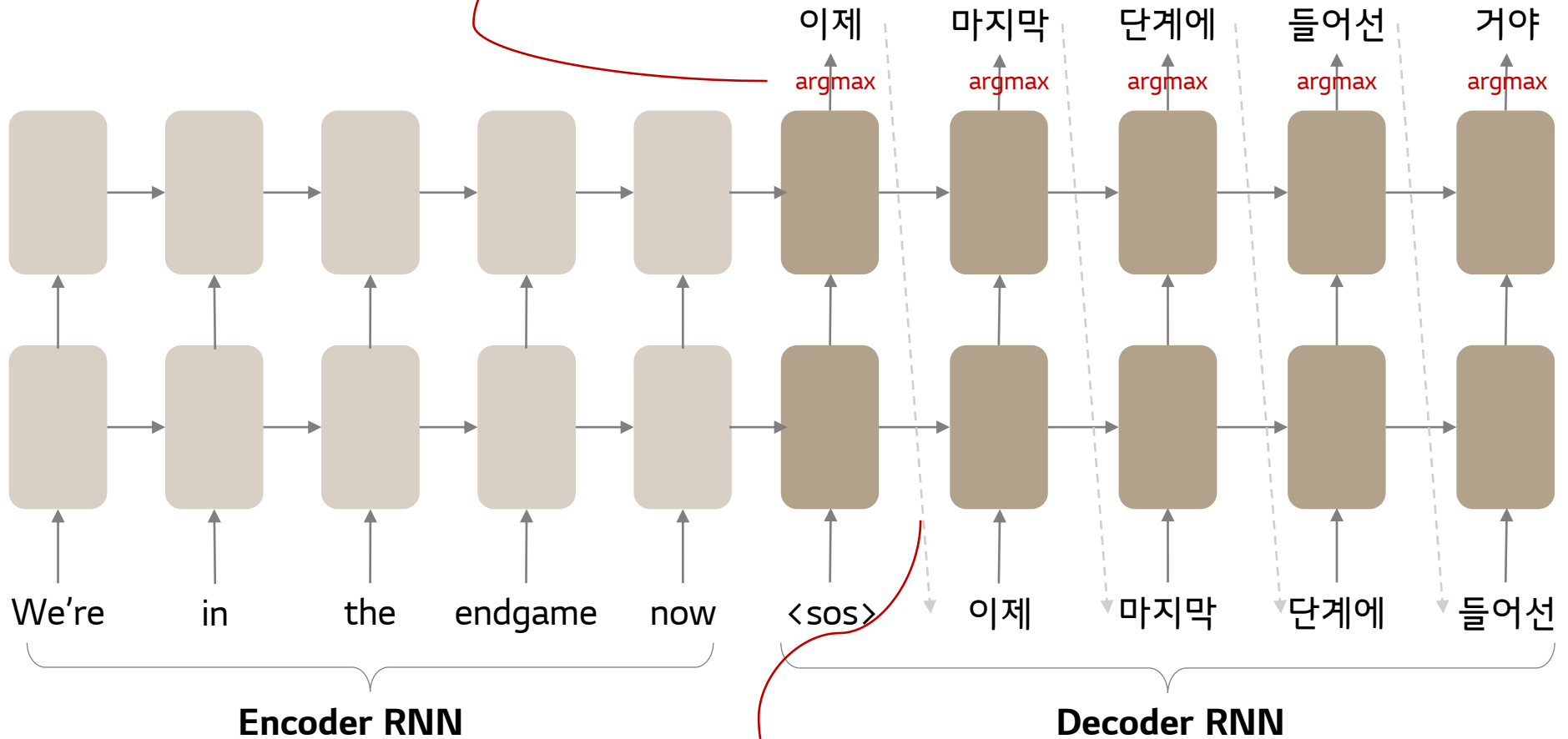
번역 대상이 되는 원본 문장을 인코딩 하여
Decoder RNN의 초기 hidden state로 전달

인코딩 된 히든에 기반하여 타겟 문장 생성

RNN을 사용한 Seq2Seq 모델링

-> 디코더 RNN은 이전 타임스텝에 디코딩된 단어를 인풋으로 받는다

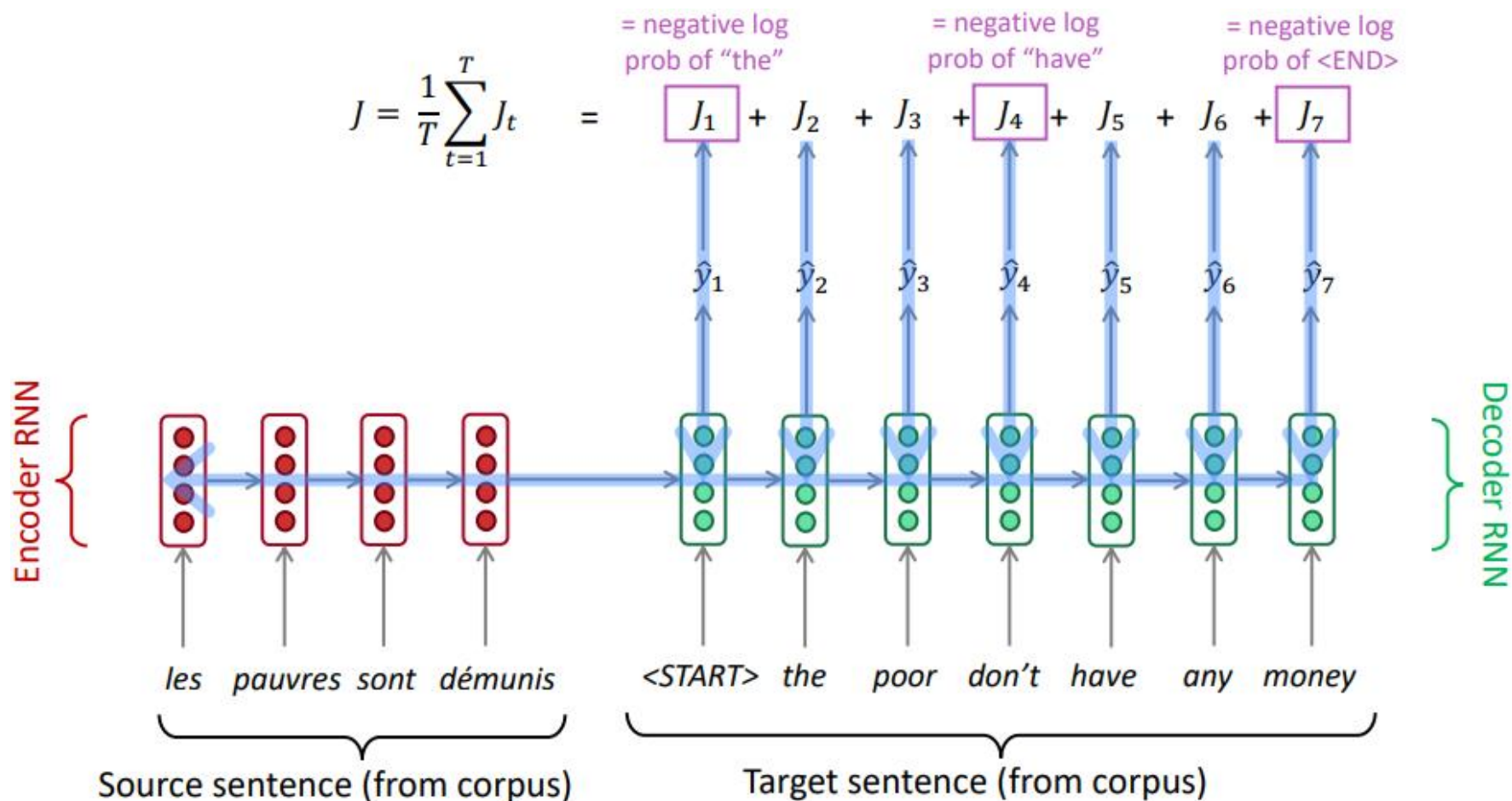
디코더 RNN의 아웃풋은
타겟 언어의 단어사전 개수 차원으로 매핑 됨



INPUT FEEDING
이전 단계에서 번역한 단어를 인풋으로 받아
현재 단계의 토큰을 생성

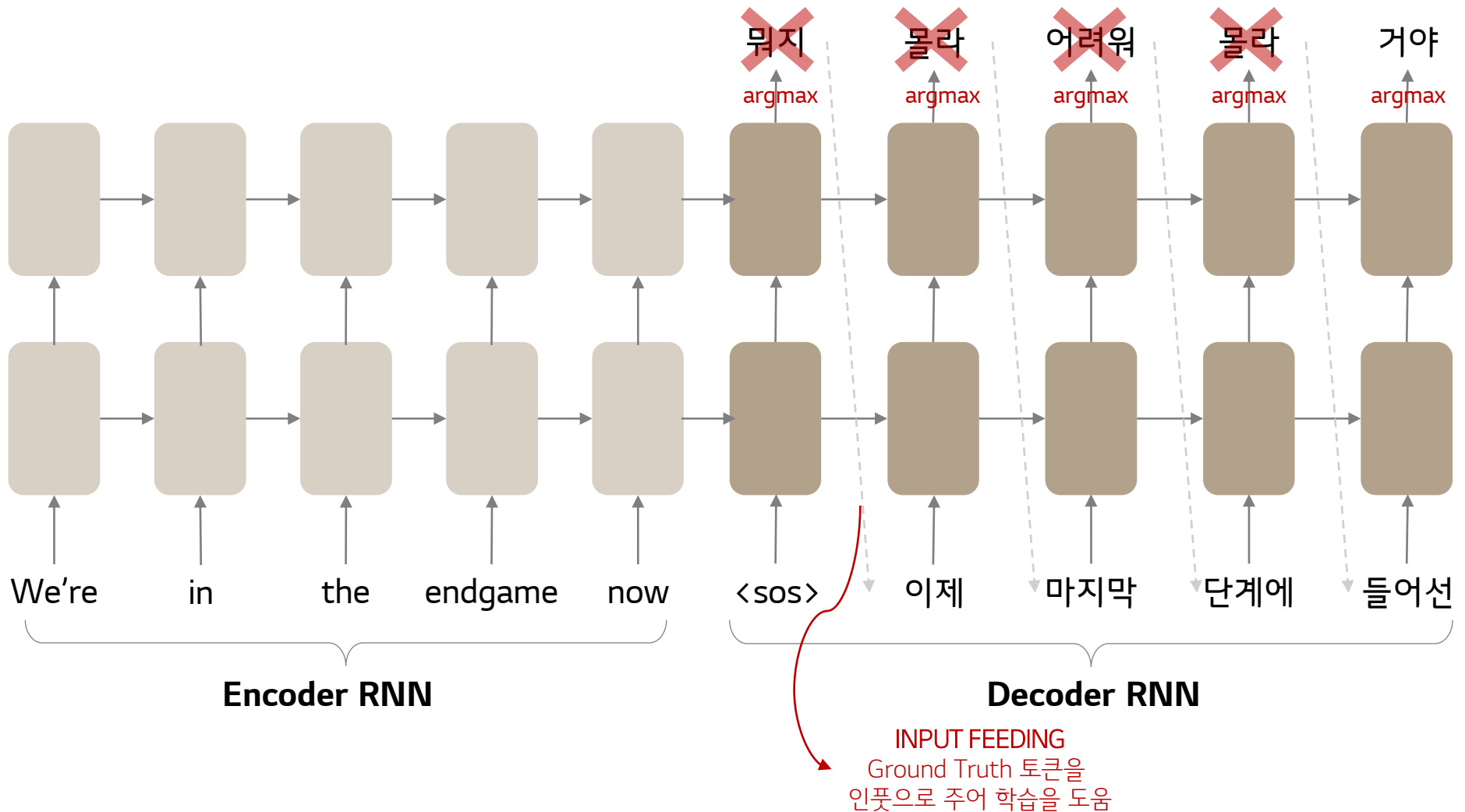
Seq2Seq RNN 학습 방법

: Seq2Seq는 하나의 시스템으로서, End-to-End back-propagation을 수행하며 학습



(참고) Teacher-forcing 학습

: 학습을 안정적으로 하기 위해 INPUT FEEDING에 실제 정답을 넣어주는 방법



Seq2Seq RNN 활용 예시

: 기계 번역 이외에 다양한 NLU 분야에서 활용할 수 있음

TASK	원본 (Source)	타겟 (Target)
기계 번역 (NMT)	원본 언어 텍스트	번역 언어 텍스트
요약 (summarization)	긴 원본 텍스트	짧은 요약문
대화 (dialog)	이전 발화 텍스트	현재 발화 텍스트
코드 생성 (code generation)	?	?

인공지능이 코딩을 한다??

The future of tech, with Kevin Scott and guests

Speakers Kevin Scott - [Microsoft](#); Peter Lee - [Microsoft](#); lila tretikov - [Microsoft](#); Luis Vargas - [Microsoft](#)

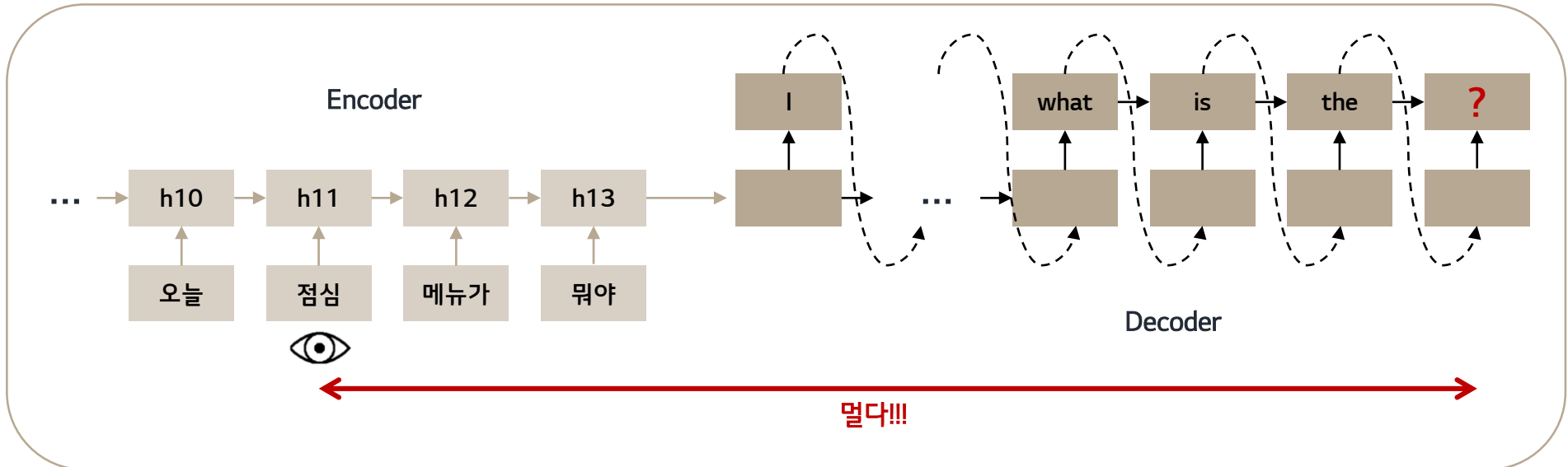
```
1  def is_palindrome(s):  
2      |      """Check whether a string is a palindrome"""  
3      |      return s == s[::-1]
```



인공지능이 작성 중...

Seq2Seq의 문제점?

- 인코더에 들어가는 문장의 길이가 길어지면, Seq2Seq의 성능이 저하된다.

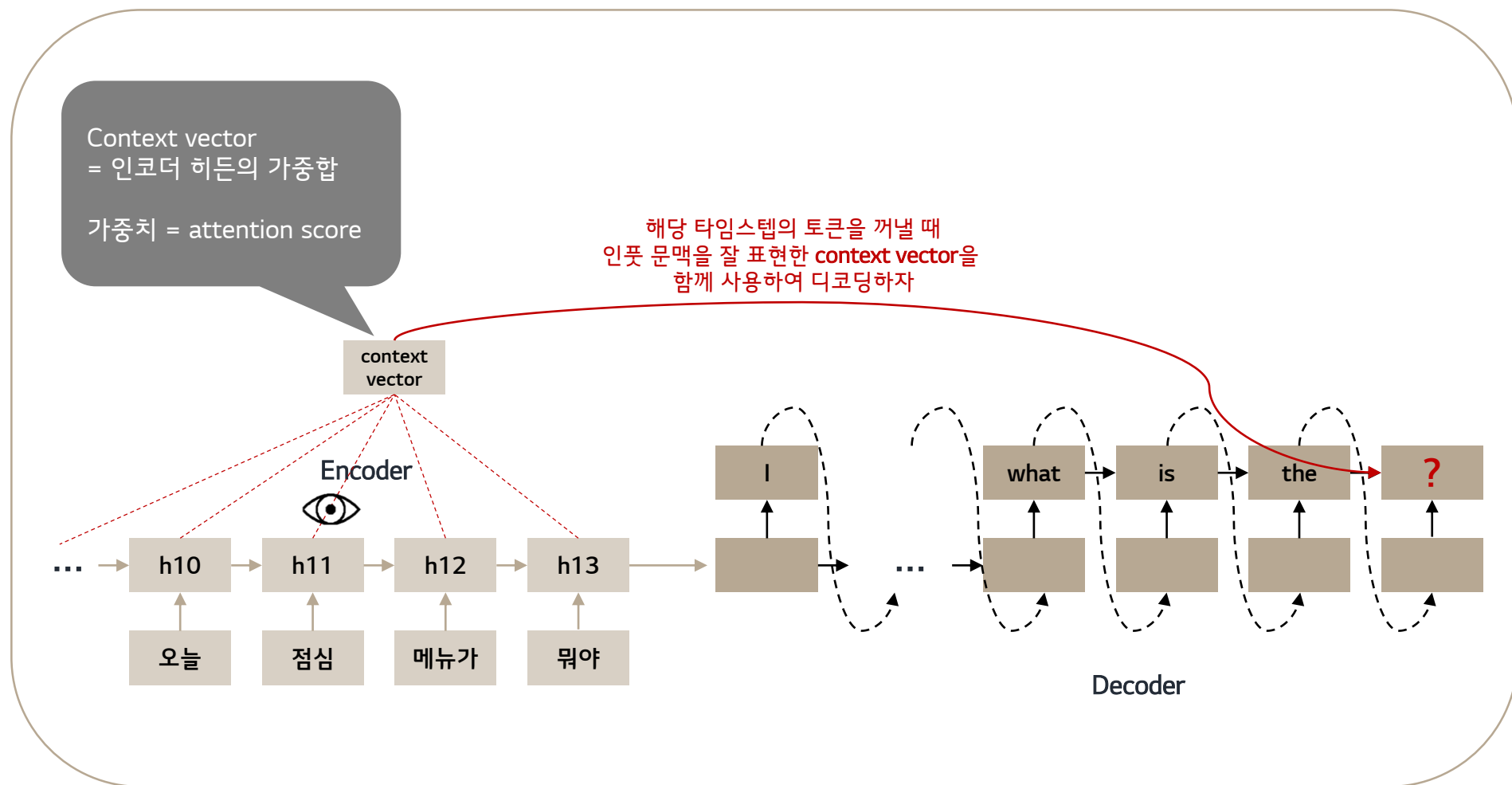


- 사람은 문장을 이해할 때, 문장 중에서 중요한 단어들을 좀 더 강조하여 이해함
- 예를 들어 문장의 앞 부분을 번역할 때는 원본 문장의 앞 부분을
- 문장의 뒷부분을 번역할 때는 원본 문장의 뒷부분을 좀 더 중요하게 봄

→ 이 모티브를 Seq2Seq에 녹여보자 !

Seq2Seq의 문제점?

- Motivation



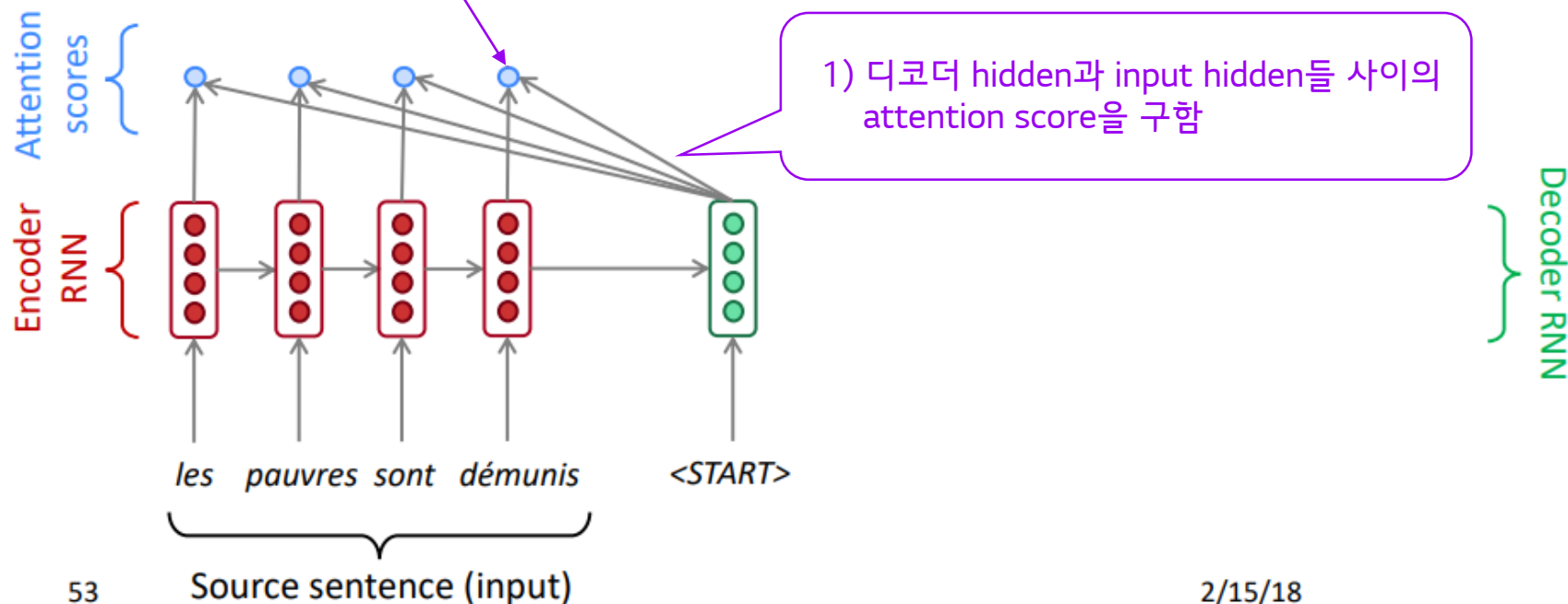
How?

: Decoder의 hidden state과 source 문장과의 attention score을 구함

Attention score을 구하는 방법은 다양함

1) Dot product : 디코더 hidden과 인코더 히든을 내적해 scalar 구하기

2) Attention weight 학습 : $\text{Score}(\mathbf{s}_{t-1}, \mathbf{h}_j) = \mathbf{v}^T \tanh(\mathbf{W}_a \mathbf{s}_{t-1} + \mathbf{U}_a \mathbf{h}_j)$

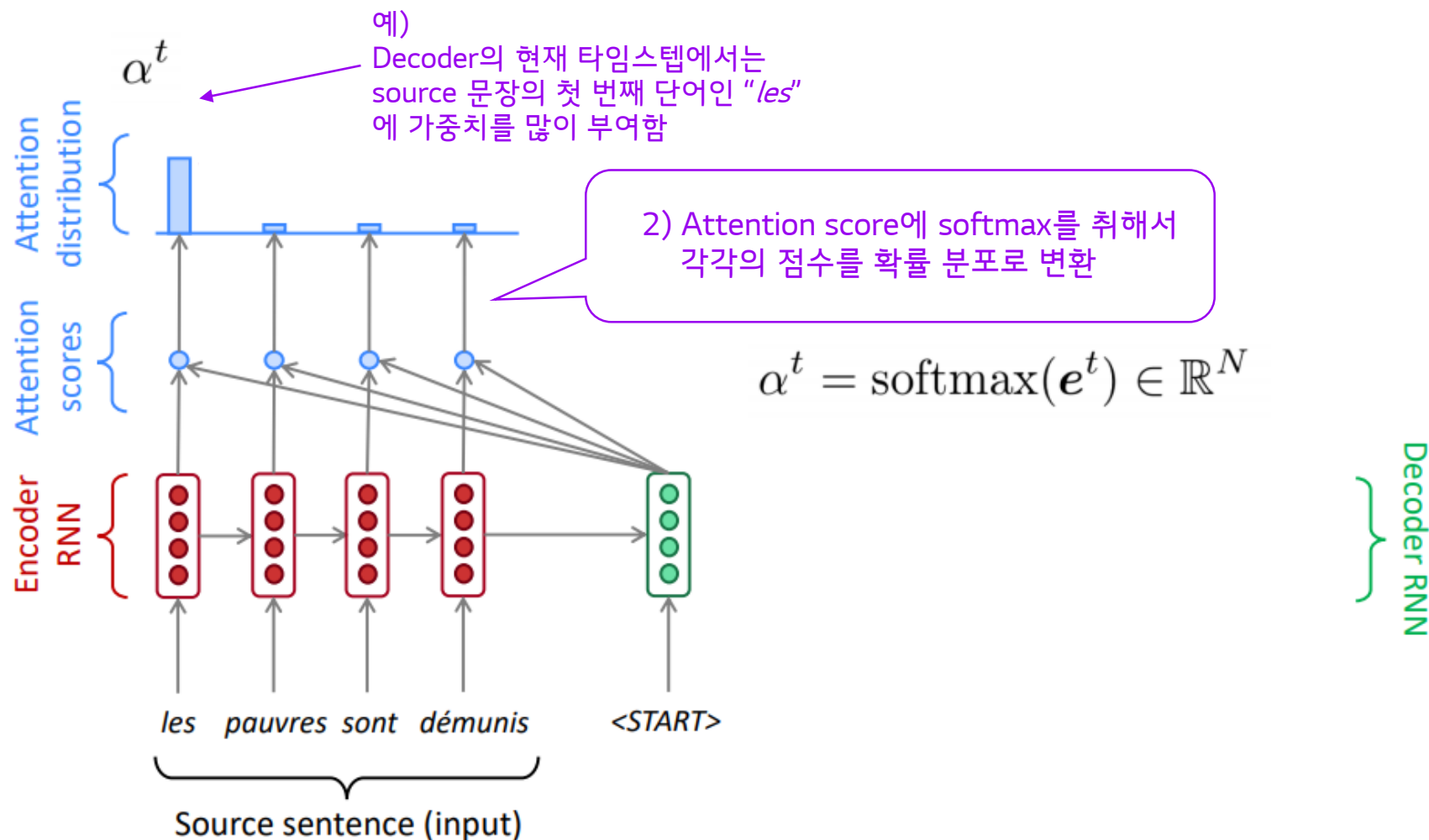


53

2/15/18

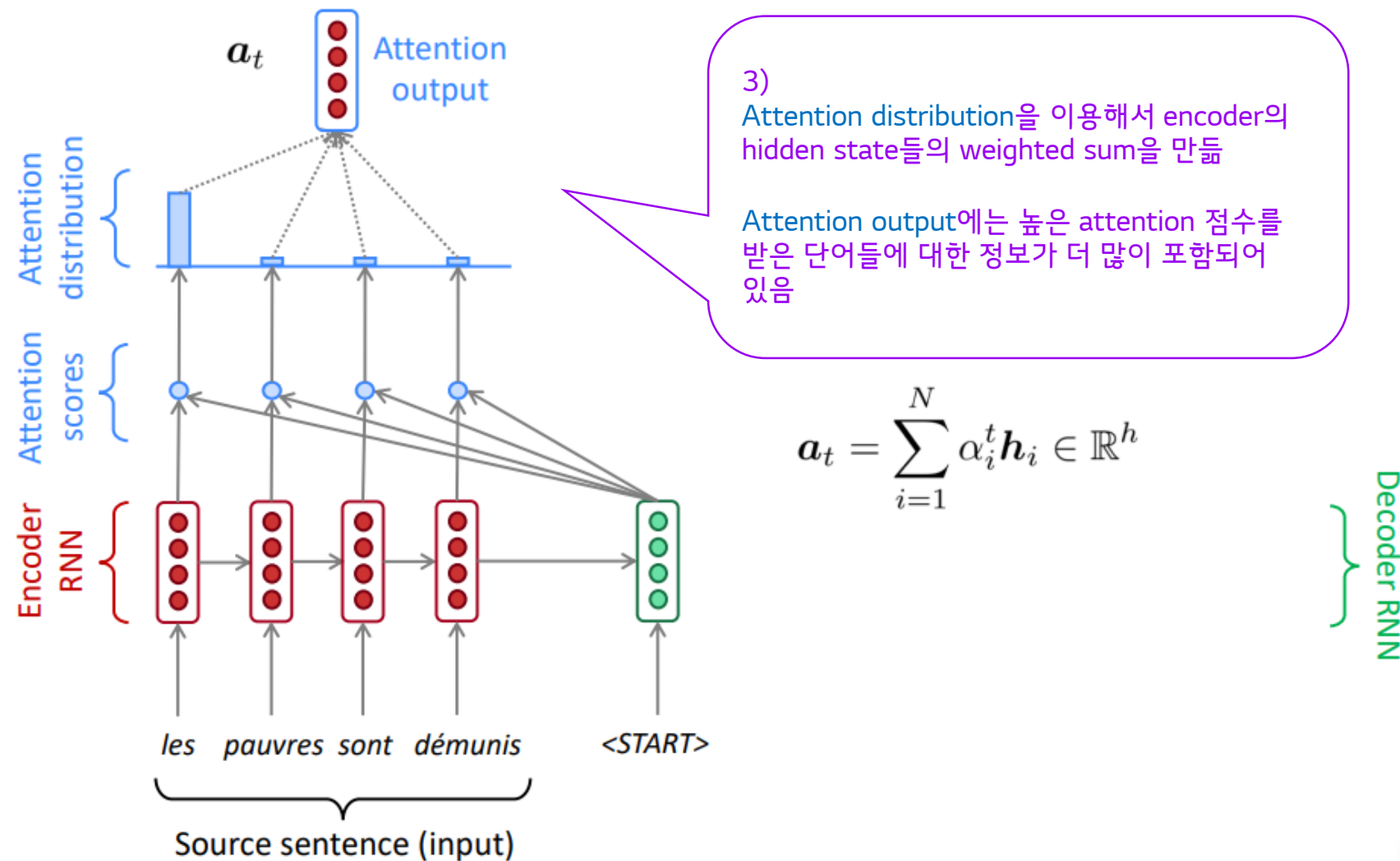
How?

: Decoder의 hidden state와 source 문장과의 dot product를 통해서 중요도를 측정



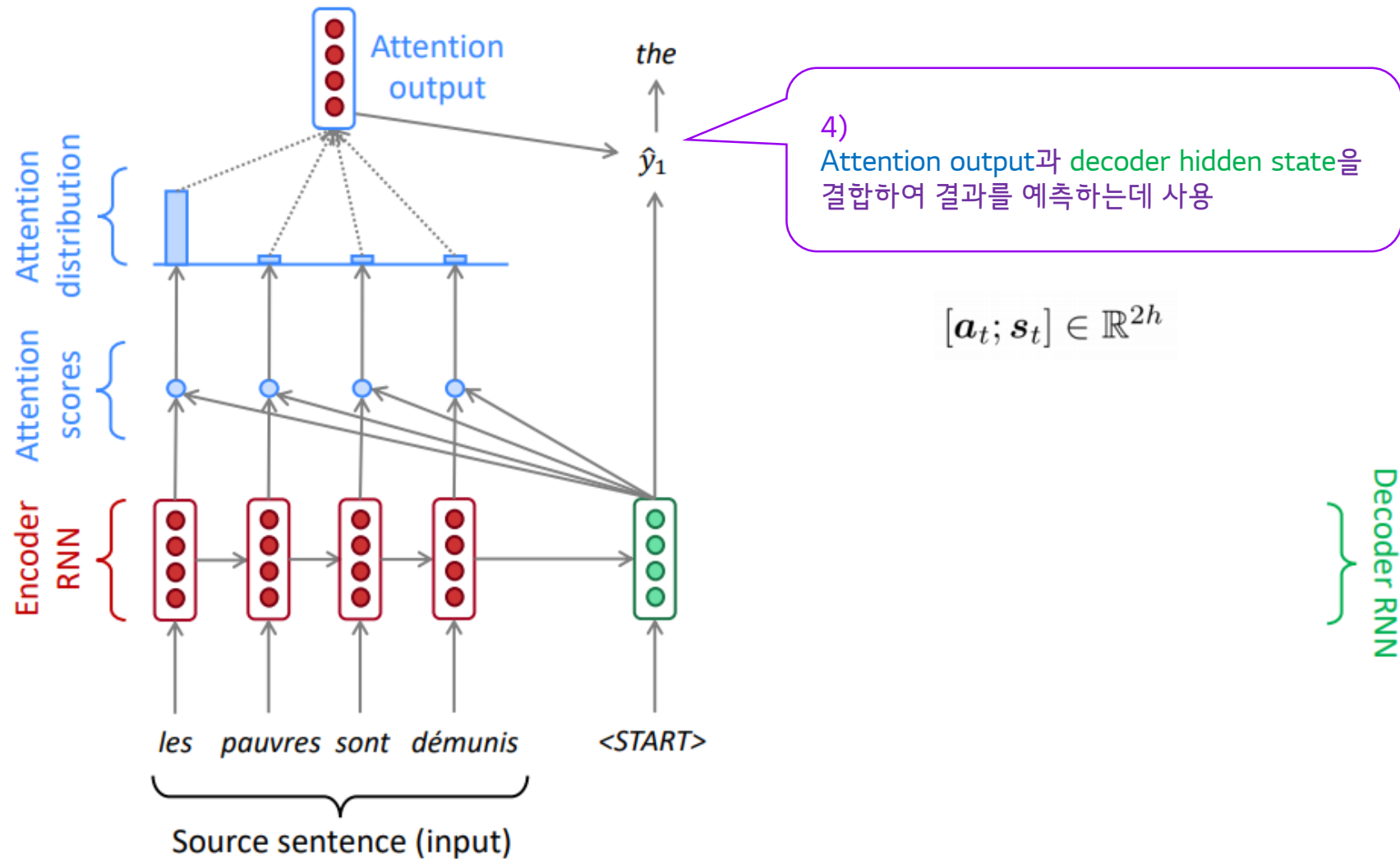
How?

: 이후 attention 가중치를 이용해 인코더 hidden state들의 가중 합 벡터를 만든다



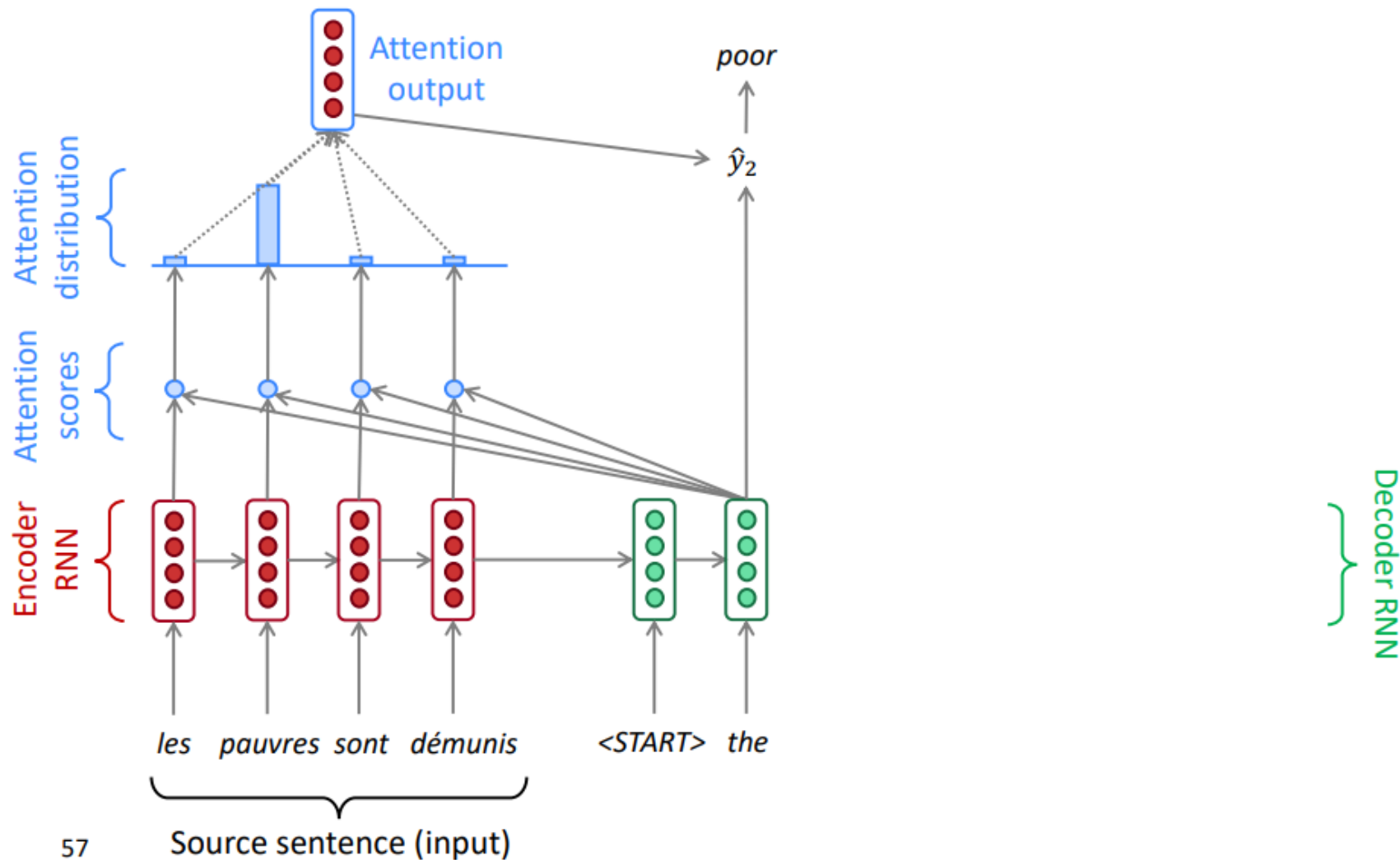
How?

: 인코더의 가중 합 된 정보와 디코더의 히든을 이용해 이번 단계의 결과를 예측하는 데 사용한다



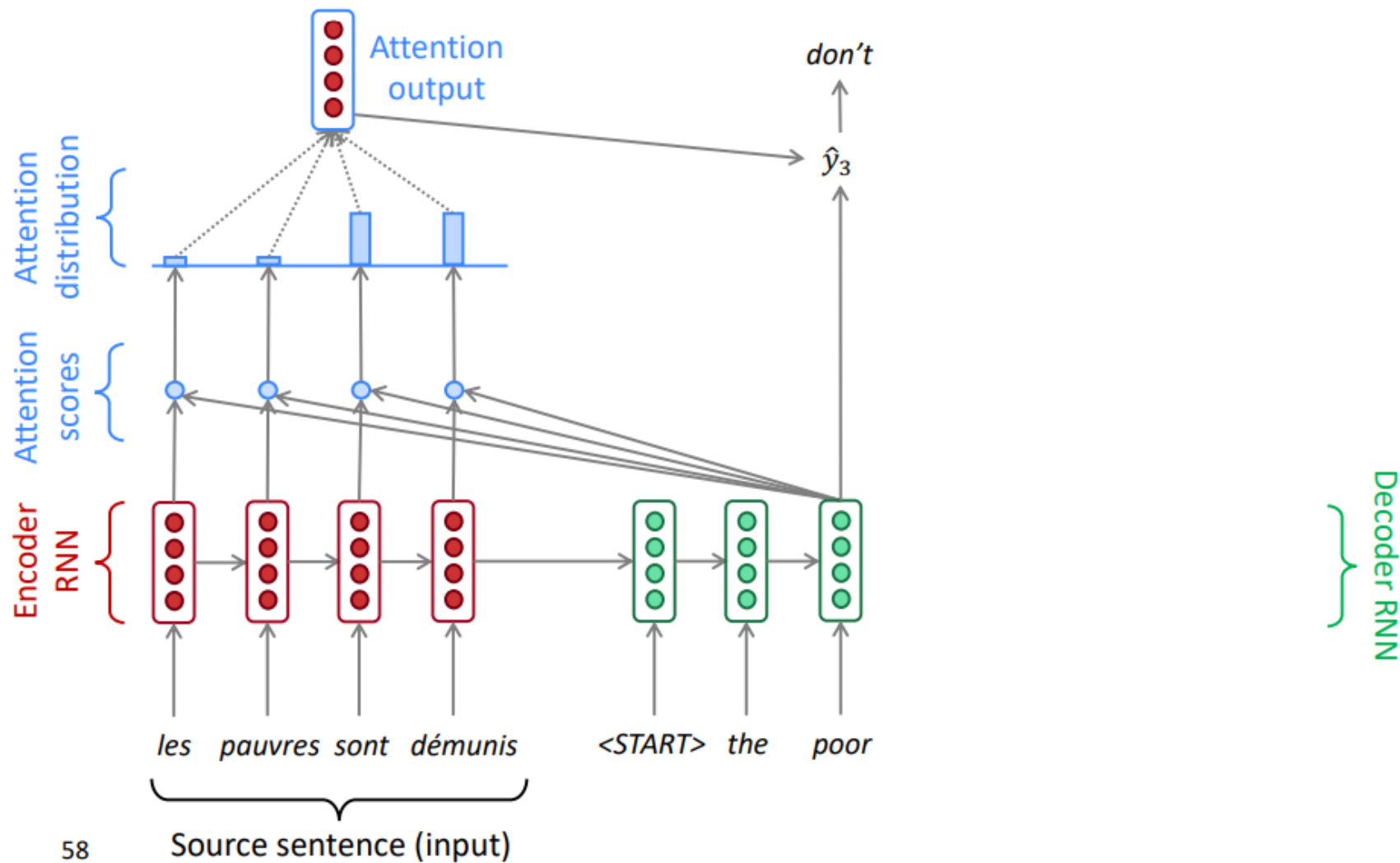
How?

: 디코더 타임스텝을 진행함에 따라 이러한 방식으로 계속 진행해 나감



How?

: 디코더 타임스텝을 진행함에 따라 이러한 방식으로 계속 진행해 나감



나만의 표표ㄱ 만들기

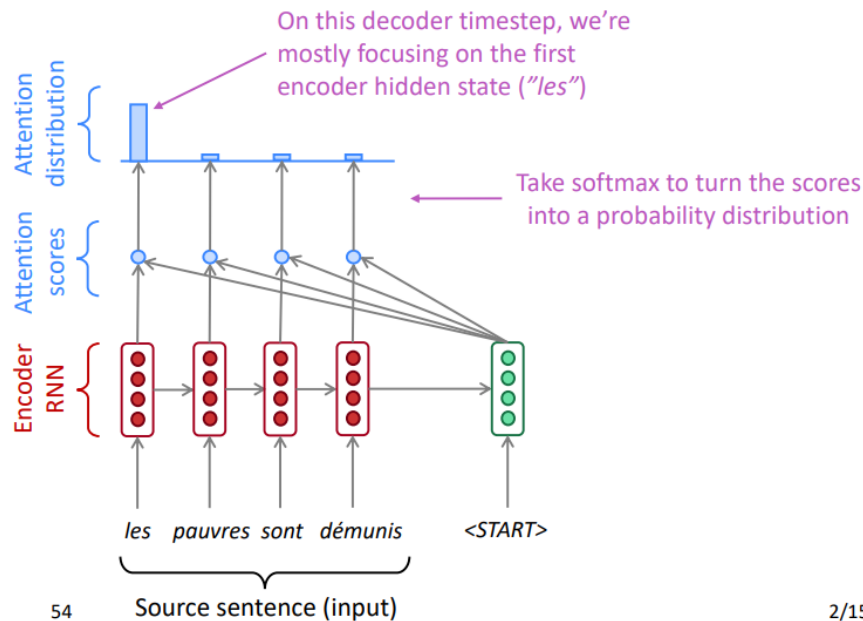
실습_5_Attention.ipynb



나만의 표표기 만들기

- 학습 목표 :

- GRU를 사용한 Seq2Seq 구조를 이해한다.
- Seq2Seq 디코더에 attention 구조를 추가한 모델을 구현한다.
- TensorFlow FunctionalAPI를 사용해 원하는 구조를 구현하고 학습한다.
- 학습된 모델을 이용해 한국어를 인풋으로 받아 영어를 리턴하는 추론 함수를 구현할 수 있다.



54

2/15/18

Decoder RNN

바다나우 어텐션 구현!

$$\text{Score}(\mathbf{s}_{t-1}, \mathbf{h}_j) = \mathbf{v}^T \tanh(\mathbf{W}_a \mathbf{s}_{t-1} + \mathbf{U}_a \mathbf{h}_j)$$

나만의 표표 만들기

- 데이터 : 대화체 데이터 100,000건

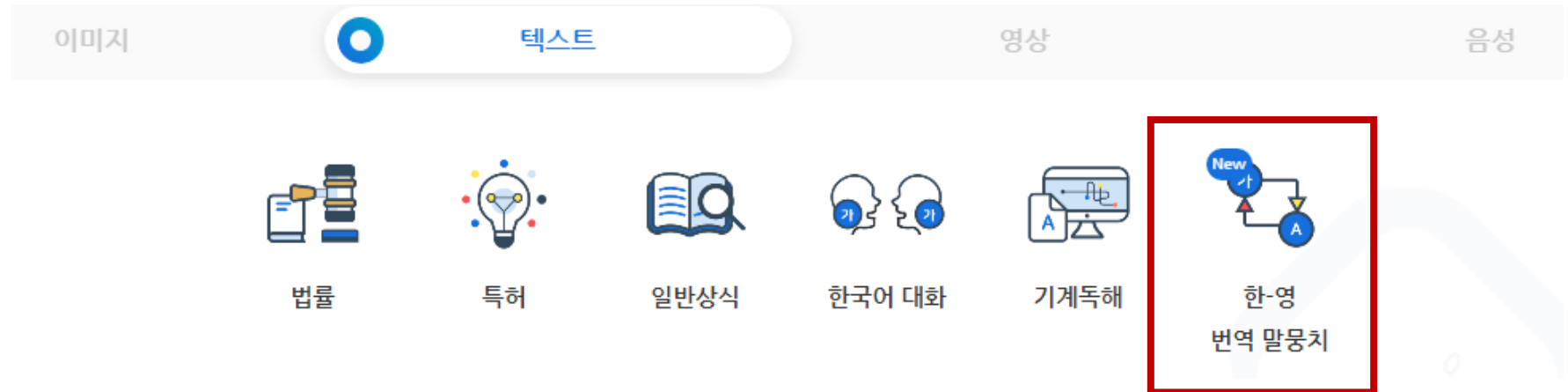
원문	번역문
이번 신제품 출시에 대한 시장의 반응은 어떤가요?	How is the market's reaction to the newly rele...
판매량이 지난번 제품보다 빠르게 늘고 있습니다.	The sales increase is faster than the previous...
그렇다면 공장에 연락해서 주문량을 더 늘려야겠네요.	Then, we'll have to call the manufacturer and ...
네, 제가 연락해서 주문량을 2배로 늘리겠습니다.	Sure, I'll make a call and double the volume o...
지난 회의 마지막에 논의했던 안건을 다시 볼까요?	Shall we take a look at the issues we discusse...

- ☒ (모두 선택)
- ☒ 비즈니스
- ☒ 스포츠
- ☒ 여행/쇼핑
- ☒ 의학
- ☒ 일상대화

나만의 데이터 만들기

- (참고) AI HUB 한국어 데이터셋

<http://www.aihub.or.kr/>

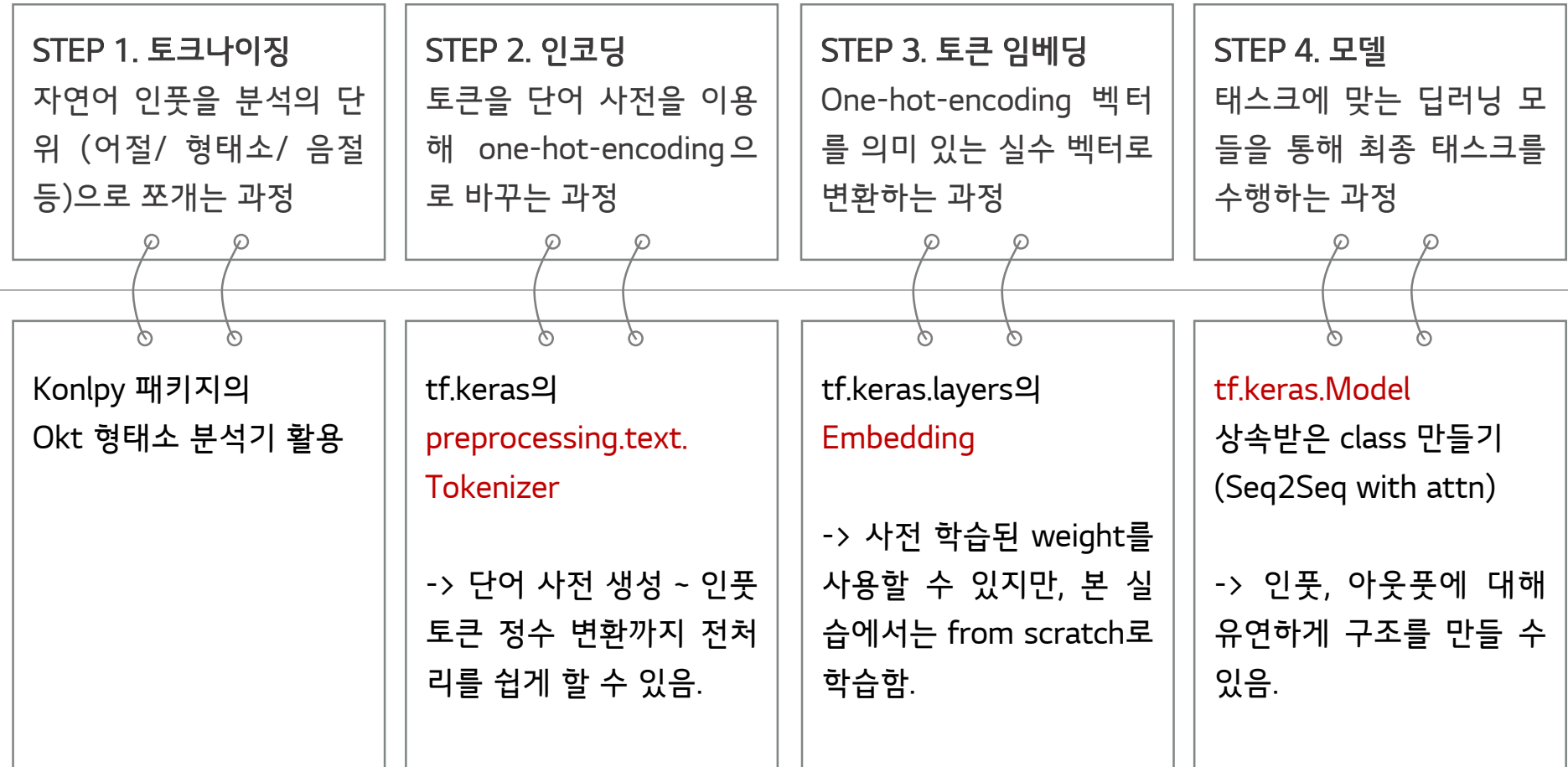


NIA에서 제공중인 데이터셋 중
기계번역 - 대화체 데이터 사용

분야	설명	수량
뉴스	뉴스 텍스트	80만 문장
정부 웹사이트/저널	정부/지자체 홈페이지, 간행물	10만 문장
법률	행정 규칙, 자치 법규	10만 문장
한국문화	한국 역사, 문화 콘텐츠	10만 문장
구어체	자연스러운 구어체 문장	40만 문장
대화체	상황/시나리오 기반 대화 세트	10만 문장
합계		160만 문장

TensorFlow2.0 Keras로 모델 만들기

- Step by step



TensorFlow2.0 Keras로 모델 만들기

- TF2.0 Keras 사용하기 – basic
 - ✓ 간단한 모델은 Sequential로 layer를 쌓고, compile한 후 fit 메서드로 학습할 수 있음.

```
1 model = tf.keras.Sequential([
2     # 64개의 유닛을 가진 완전 연결 층을 모델에 추가합니다:
3     layers.Dense(64, activation='relu', input_shape=(32,)),
4     # 또 하나를 추가합니다:
5     layers.Dense(64, activation='relu'),
6     # 10개의 출력 유닛을 가진 소프트맥스 층을 추가합니다:
7     layers.Dense(10, activation='softmax')])
8
9 model.compile(optimizer=tf.keras.optimizers.Adam(0.001),
10               loss='categorical_crossentropy',
11               metrics=['accuracy'])

```

예제 `Dataset` 객체를 만듭니다:

```
dataset = tf.data.Dataset.from_tensor_slices((data, labels))
dataset = dataset.batch(32)
```

Dataset에서 `fit` 메서드를 호출할 때 `steps_per_epoch` 설정을 잊지 마세요.

```
model.fit(dataset, epochs=10, steps_per_epoch=30)
```


TensorFlow2.0 Keras로 모델 만들기

- TF2.0 Keras 사용하기 – 커스텀 모델 만들기
 - ✓ TensorFlow2.x에서는 Model을 subclass하여 원하는 모델을 만들 수 있음.

subclassed model 코드

```
1 class MLP(keras.Model):
2
3     def __init__(self, **kwargs):
4         super(MLP, self).__init__(**kwargs)
5         self.dense_1 = layers.Dense(64, activation='relu')
6         self.dense_2 = layers.Dense(10)
7
8     def call(self, inputs):
9         x = self.dense_1(inputs)
10        return self.dense_2(x)
11
12 # 모델 인스턴스화 하기
13 mlp = MLP()
14 # 모델 state를 만들어야 함. 모델은 한 번 호출되기 전까지 state를 가지고 있지 않음
15 # The model doesn't have a state until it's called at least once.
16 _ = mlp(tf.zeros((1, 32)))
```

모델에 사용되는 layer 정의

Call method를 정의하여
forward pass가 어떻게 이루어지는
지 정의함

TensorFlow2.0 Keras로 모델 만들기

- TF2.0 Keras 사용하기 – 커스텀 모델 만들기

- ✓ 커스텀 모델은 loss를 직접 디자인하고 모델 구조를 유연하게 짜는 데에 유용함!

Loss function 정의

손실 함수와 그래디언트 함수 정의하기

훈련과 평가단계에서 모델의 **손실(loss)**을 계산해야 합니다. 손실은 모델의 예측이 원하는 레이블과 얼마나 일치하는지, 또한 모델이 잘 작동하는지에 대한 척도로 사용됩니다. 이 값을 최소화하고, 최적화 해야합니다.

모델의 손실은 `tf.keras.losses.categorical_crossentropy` 함수를 사용해 계산할 것입니다. 이 함수는 모델의 클래스(레이블)과 예측된 값(로짓)을 입력받아 샘플의 평균 손실을 반환합니다.

```
loss_object = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True)
```

```
def loss(model, x, y):
    y_ = model(x)

    return loss_object(y_true=y, y_pred=y_)

l = loss(model, features, labels)
print("손실 테스트: {}".format(l))
```

손실 테스트: 1.6921892166137695

모델을 최적화하기 위해 사용되는 **그래디언트(gradient)**를 계산하기 위해 `tf.GradientTape` 컨텍스트를 사용합니다.

```
def grad(model, inputs, targets):
    with tf.GradientTape() as tape:
        loss_value = loss(model, inputs, targets)
    return loss_value, tape.gradient(loss_value, model.trainable_variables)
```

tf.GradientTape

colab

[44] translate("오늘따라 피자 먹고 싶다")

Input: <start> 오늘 따라 피자 먹고 싶다 <end>
 Predicted translation: i want to have a pizza today <end>

