

# 14장 RNN을 이용한 인코더-디코더

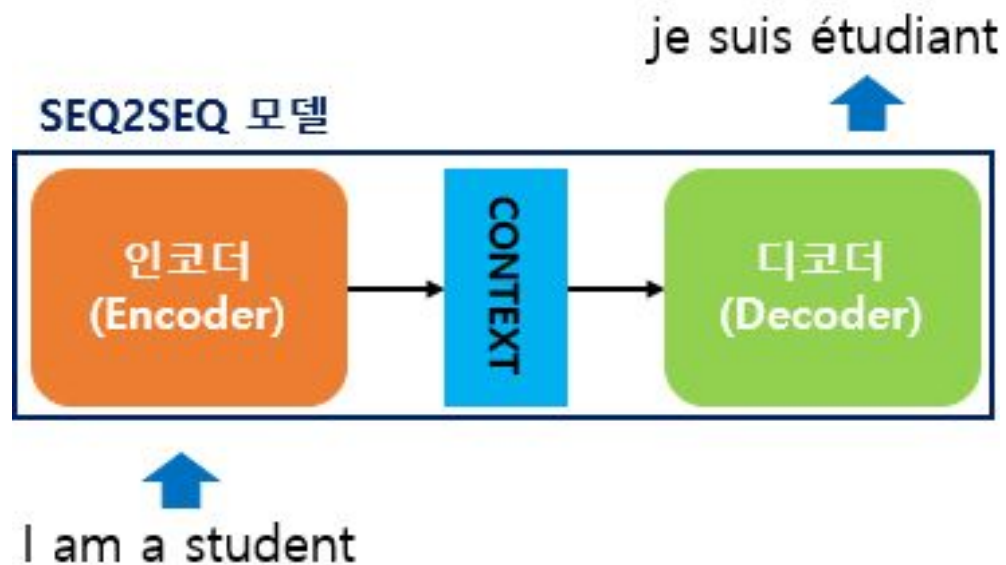
Encoder-Decoder using RNN

# 목차 Table of Contents

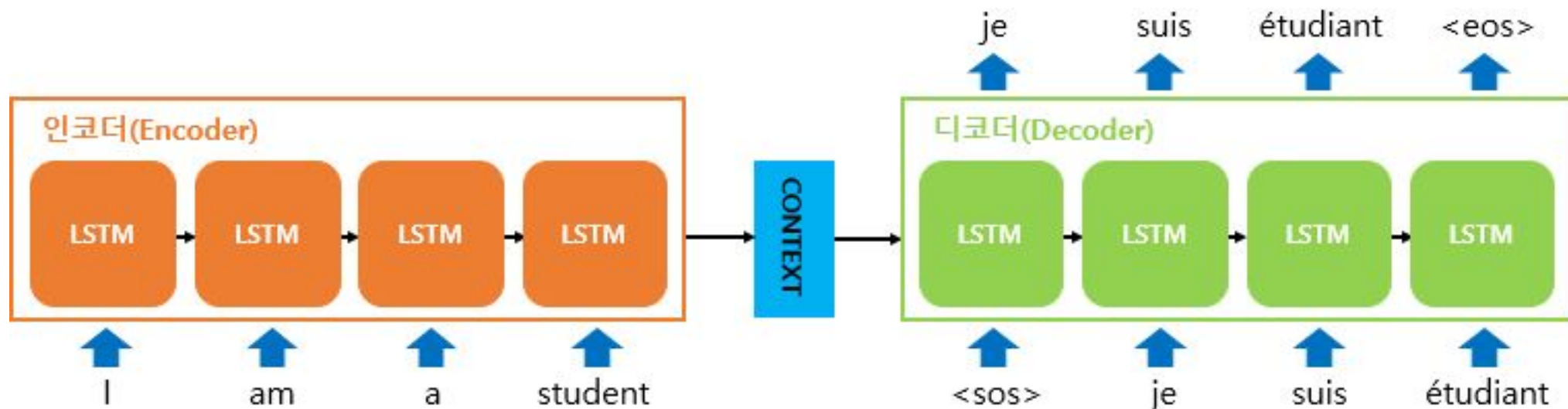
---

- 01 시퀀스-투-시퀀스(seq2seq)
- 02 Word-Level 번역기 만들기(Neural Machine Translation)
- 03 BLEU Score(Bilingual Evaluation Understudy Score)

## 01 시퀀스-투-시퀀스 seq2seq

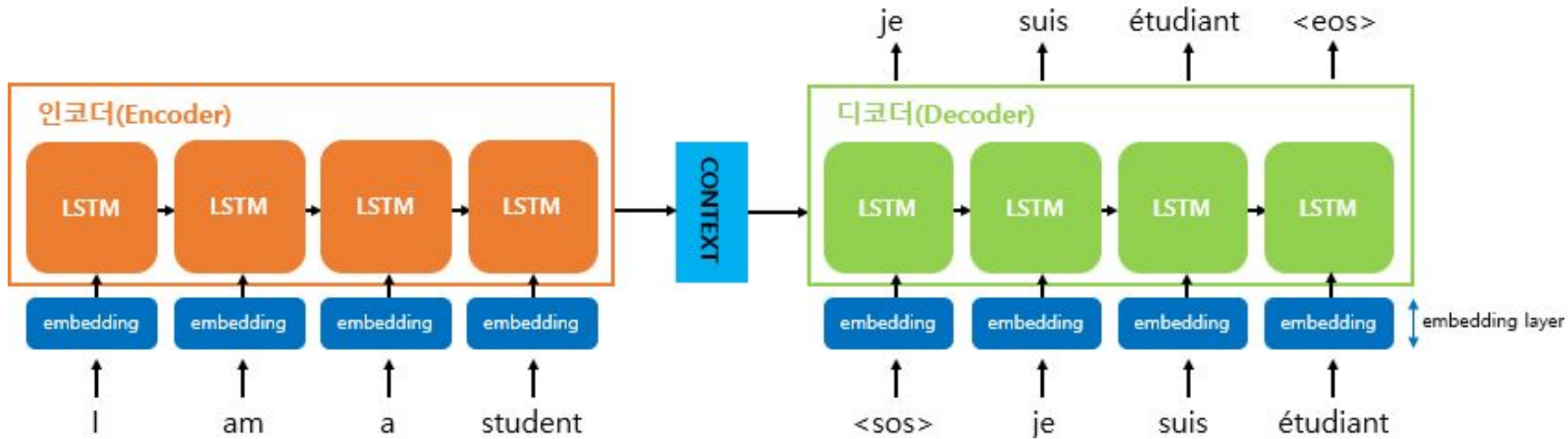


## 01 시퀀스-투-시퀀스 seq2seq



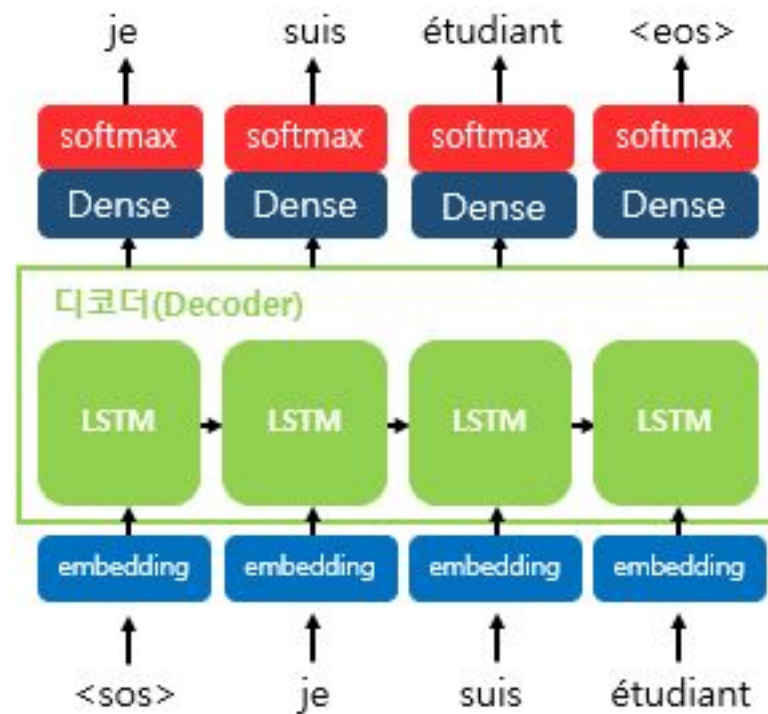
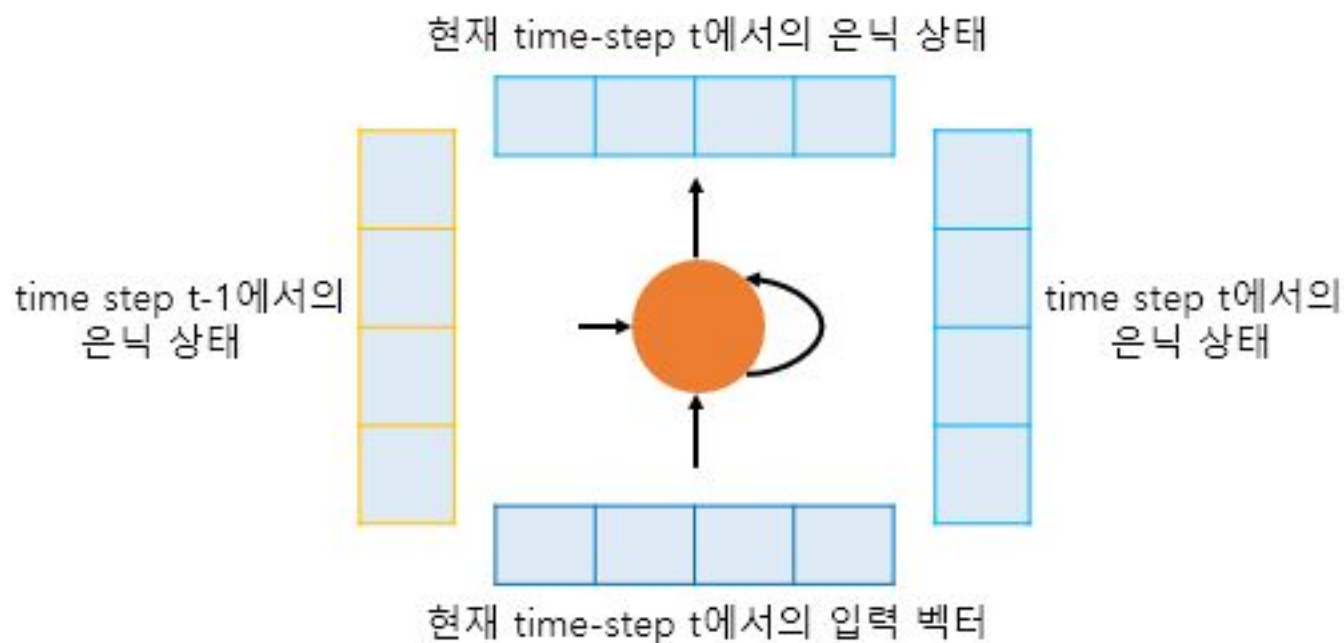
CONTEXT		0.15
		0.21
		-0.11
		0.91

## 01 시퀀스-투-시퀀스 seq2seq



I		0.157	am		0.78	a		0.75	student		0.88
		-0.25			0.29			-0.81			-0.17
		0.478			-0.96			0.96			0.29
		-0.78			0.52			0.12			0.48

# 01 시퀀스-투-시퀀스 seq2seq



# 01 시퀀스-투-시퀀스 seq2seq

- 문자 레벨 기계 번역기

**We are happy to be here** → **우리 가 여기 오--게 되--어서 기뻐--하 요**

병렬 corpus 데이터의 형태

	src	tar
	Watch me.	Regardez-moi !
59698	I'll take responsibility.	⚠ J'assumerai la responsabilité. ⚠n
49350	We're happy to be here.	⚠ Nous sommes heureux d'être ici. ⚠n
56905	Bears are very dangerous.	⚠ Les ours sont très dangereux. ⚠n
39277	How big is your house?	⚠ De quelle taille est ta maison ? ⚠n
55564	We have come a long way.	⚠ Nous avons fait un long voyage. ⚠n
1122	Don't move!	⚠ Ne bouge pas ! ⚠n
17884	He is intelligent.	⚠ Il est intelligent. ⚠n
58	No way!	⚠ Sans façons ! ⚠n
20241	The motor stopped.	⚠ Le moteur s'est arrêté. ⚠n
56880	Are you the group leader?	⚠ Es-tu la meneuse du groupe ? ⚠n

데이터 전처리

```
{ ' ': 1, '!': 2, '": 3, '$': 4, '%': 5, ... 종략 ... 'x': 73, 'y': 74, 'z': 75, 'é': 76, '': 77, '€': 78}
{'\t': 1, '\n': 2, ' ': 3, '!': 4, '": 5, ... 종략 ... 'ù': 98, 'œ': 99, 'C': 100, '\u2009': 101, '': 102, '': 103, '\u202f': 104}
```

target 문장의 정수 인코딩 : [[1, 3, 48, 53, 3, 4, 3, 2], [1, 3, 39, 53, 70, 55, 60, 57, 14, 3, 2], [1, 3, 28, 67, 73, 59, 57, 3, 4, 3, 2], [1, 3, 45, 53, 64, 73, 72, 3, 4, 3, 2], [1, 3, 45, 53, 64, 73, 72, 14, 3, 2]]

각 문자에 인덱스 부여 후 문장을 문자 단위로 정수 인코딩

target 문장 레이블의 정수 인코딩 : [[3, 48, 53, 3, 4, 3, 2], [3, 39, 53, 70, 55, 60, 57, 14, 3, 2], [3, 28, 67, 73, 59, 57, 3, 4, 3, 2], [3, 45, 53, 64, 73, 72, 3, 4, 3, 2], [3, 45, 53, 64, 73, 72, 14, 3, 2]]

교사 강요 학습을 위한 레이블 데이터

각 언어별 데이터  
길이를  
맞춰주기 위한 패딩  
작업

...의 한 인코딩

source 문장의 최대 길이 : 23  
target 문장의 최대 길이 : 76



# 01 시퀀스-투-시퀀스 seq2seq

## - 문자 레벨 기계 번역기

### 교사 강요를 통한 학습

```
encoder_inputs = Input(shape=(None, src_vocab_size))
encoder_lstm = LSTM(units=256, return_state=True)

# encoder_outputs은 여기서는 불필요
encoder_outputs, state_h, state_c = encoder_lstm(encoder_inputs)

# LSTM은 바닐라 RNN과는 달리 상태가 두 개. 은닉 상태와 셀 상태.
encoder_states = [state_h, state_c]

decoder_inputs = Input(shape=(None, tar_vocab_size))
decoder_lstm = LSTM(units=256, return_sequences=True, return_state=True)

# 디코더에게 인코더의 은닉 상태, 셀 상태를 전달.
decoder_outputs, _, _ = decoder_lstm(decoder_inputs, initial_state=encoder_states)

decoder_softmax_layer = Dense(tar_vocab_size, activation='softmax')
decoder_outputs = decoder_softmax_layer(decoder_outputs)

model = Model([encoder_inputs, decoder_inputs], decoder_outputs)
model.compile(optimizer="rmsprop", loss="categorical_crossentropy")

model.fit(x=[encoder_input, decoder_input], y=decoder_target, batch_size=64, epochs=40, validation_split=0.2)
```

### 모델 테스트

```
encoder_model = Model(inputs=encoder_inputs, outputs=encoder_states)

# 이전 시점의 상태들을 저장하는 텐서
decoder_state_input_h = Input(shape=(256,))
decoder_state_input_c = Input(shape=(256,))
decoder_states_inputs = [decoder_state_input_h, decoder_state_input_c]

# 문장의 다음 단어를 예측하기 위해서 초기 상태(initial_state)를 이전 시점의 상태로 사용.
# 뒤의 함수 decode_sequence()에 동작을 구현 예정
decoder_outputs, state_h, state_c = decoder_lstm(decoder_inputs, initial_state=decoder_states_inputs)

# 훈련 과정에서와 달리 LSTM의 리턴하는 은닉 상태와 셀 상태를 버리지 않음.
decoder_states = [state_h, state_c]
decoder_outputs = decoder_softmax_layer(decoder_outputs)
decoder_model = Model(inputs=[decoder_inputs] + decoder_states_inputs, outputs=[decoder_outputs] + decoder_states)

index_to_src = dict((i, char) for char, i in src_to_index.items())
index_to_tar = dict((i, char) for char, i in tar_to_index.items())
```

### 결과물

```
-----
입력 문장: Hi.
정답 문장: Salut !
번역 문장: Salut.
-----
입력 문장: I see.
정답 문장: Aha.
번역 문장: Je change.
-----
입력 문장: Hug me.
정답 문장: Serrez-moi dans vos bras !
번역 문장: Serre-moi dans vos patents !
-----
입력 문장: Help me.
정답 문장: Aidez-moi.
번역 문장: Aidez-moi.
-----
입력 문장: I beg you.
정답 문장: Je vous en prie.
번역 문장: Je vous en prie.
```



## 02 Word-Level 번역기 만들기 Neural Machine Translation

병렬 corpus 데이터의 형태

	src	tar
	Watch me.	Regardez-moi !
59698	I'll take responsibility.	₩t J'assumerai la responsabilité. ₩n
49350	We're happy to be here.	₩t Nous sommes heureux d'être ici. ₩n
56905	Bears are very dangerous.	₩t Les ours sont très dangereux. ₩n
39277	How big is your house?	₩t De quelle taille est ta maison ? ₩n
55564	We have come a long way.	₩t Nous avons fait un long voyage. ₩n
1122	Don't move!	₩t Ne bouge pas ! ₩n
17884	He is intelligent.	₩t Il est intelligent. ₩n
58	No way!	₩t Sans façons ! ₩n
20241	The motor stopped.	₩t Le moteur s'est arrêté. ₩n
56880	Are you the group leader?	₩t Es-tu la meneuse du groupe ? ₩n

데이터 전처리

```
인코더의 입력 : [['go', '.'], ['go', '.'], ['go', '.'], ['hi', '.'], ['hi', '.']]
디코더의 입력 : [['<sos>', 'va', '!'], ['<sos>', 'marche', '.'], ['<sos>', 'bouge', '!'], ['<sos>', 'salut', '!'], ['<sos>', 'salut', '.']]
디코더의 레이블 : [['va', '!', '<eos>'], ['marche', '.', '<eos>'], ['bouge', '!', '<eos>'], ['salut', '!', '<eos>'], ['salut', '.', '<eos>']]
```

```
array([ 2, 18, 5, 16, 173, 1, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0], dtype=int32)
```

각 단어에 인덱스 부여 후 문장을 단어 단위로 정수 인코딩

```
array([ 18, 5, 16, 173, 1, 3, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0], dtype=int32)
```

교사 강요 학습을 위한 레이블 데이터

각 언어별 데이터  
길이를  
맞춰주기 위한 패딩  
작업

```
인코더의 입력의 크기(shape) : (33000, 8)
디코더의 입력의 크기(shape) : (33000, 16)
디코더의 레이블의 크기(shape) : (33000, 16)
```

## 02 Word-Level 번역기 만들기 Neural Machine Translation

## 교사 강요를 통한 학습

```
embedding_dim = 64
hidden_units = 64

# 인코더
encoder_inputs = Input(shape=(None,))
enc_emb = Embedding(src_vocab_size, embedding_dim)(encoder_inputs) # 임베딩 층
enc_masking = Masking(mask_value=0.0)(enc_emb) # 패딩 0은 연산에서 제외
encoder_lstm = LSTM(hidden_units, return_state=True) # 상태값 리턴을 위해 return_state는 True
encoder_outputs, state_h, state_c = encoder_lstm(enc_masking) # 은닉 상태와 셀 상태를 리턴
encoder_states = [state_h, state_c] # 인코더의 은닉 상태와 셀 상태를 저장

# 디코더
decoder_inputs = Input(shape=(None,))
dec_emb_layer = Embedding(tar_vocab_size, hidden_units) # 임베딩 층
dec_emb = dec_emb_layer(decoder_inputs) # 패딩 0은 연산에서 제외
dec_masking = Masking(mask_value=0.0)(dec_emb)

# 상태값 리턴을 위해 return_state는 True, 모든 시점에 대해서 단어를 예측하기 위해 return_sequences는 True
decoder_lstm = LSTM(hidden_units, return_sequences=True, return_state=True)

# 인코더의 은닉 상태를 초기 은닉 상태(initial_state)로 사용
decoder_outputs, _, _ = decoder_lstm(dec_masking,
                                     initial_state=encoder_states)

# 모든 시점의 결과에 대해서 소프트맥스 함수를 사용한 출력층을 통해 단어 예측
decoder_dense = Dense(tar_vocab_size, activation='softmax')
decoder_outputs = decoder_dense(decoder_outputs)

# 모델의 입력과 출력을 정의.
model = Model([encoder_inputs, decoder_inputs], decoder_outputs)

model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['acc'])

model.fit(x=[encoder_input_train, decoder_input_train], y=decoder_target_train, \
        validation_data=[encoder_input_test, decoder_input_test], decoder_target_test), \
        batch_size=128, epochs=50)
```

## 모델 테스트

```
# 인코더
encoder_model = Model(encoder_inputs, encoder_states)

# 디코더 설계 시작
# 이전 시점의 상태를 보관할 텐서
decoder_state_input_h = Input(shape=(hidden_units,))
decoder_state_input_c = Input(shape=(hidden_units,))
decoder_states_inputs = [decoder_state_input_h, decoder_state_input_c]

# 훈련 때 사용했던 임베딩 층을 재사용
dec_emb2 = dec_emb_layer(decoder_inputs)

# 다음 단어 예측을 위해 이전 시점의 상태를 현 시점의 초기 상태로 사용
decoder_outputs2, state_h2, state_c2 = decoder_lstm(dec_emb2, initial_state=decoder_states_inputs)
decoder_states2 = [state_h2, state_c2]

# 모든 시점에 대해서 단어 예측
decoder_outputs2 = decoder_dense(decoder_outputs2)

# 수정된 디코더
decoder_model = Model(
    [decoder_inputs] + decoder_states_inputs,
    [decoder_outputs2] + decoder_states2)

for seq_index in range(10000):
    # 인코딩
    input_seq = encoder_model.predict(input_sequences[seq_index])
    decoded_sentence, _, _ = decoder_model.predict([input_seq] + decoder_states)

    print("입력문장 :", input_sequences[seq_index])
    print("정답문장 :", encoder_outputs[seq_index])
    print("번역문장 :", decoded_sentence)

    # 입력문장 : when does it end ?
    # 정답문장 : quand est ce que ca finit ?
    # 번역문장 : quand est ce que ca marche ?
    print("-----")

    print("입력문장 :", input_sequences[seq_index])
    print("정답문장 :", encoder_outputs[seq_index])
    print("번역문장 :", decoded_sentence)

    # 입력문장 : it s sand .
    # 정답문장 : c est du sable .
    # 번역문장 : c est de l eau .
```

## 결과물

입력문장 : when does it end ?  
정답문장 : quand est ce que ca finit ?  
번역문장 : quand est ce que ca marche ?

입력문장 : it s sand .  
정답문장 : c est du sable .  
번역문장 : c est de l eau .

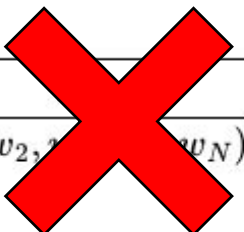
입력문장 : i didn t go .  
 정답문장 : je n y suis pas allee .  
 번역문장 : je ne suis pas encore .

입력문장 : it was a mistake .  
정답문장 : ce fut une erreur .  
번역문장 : il s agit d une blague .

입력문장 : it boggles my mind .  
정답문장 : ca me laisse perplexe .  
번역문장 : ca m en femme .

## 03 BLEU Score Bilingual Evaluation Understudy Score

Perplexity(PPL)

$$PPL(W) = P(w_1, w_2, w_3, \dots, w_N)^{-\frac{1}{N}} = \sqrt[N]{\frac{1}{P(w_1, w_2, \dots, w_N)}} = \sqrt[N]{\frac{1}{\prod_{i=1}^N P(w_i | w_1, w_2, \dots, w_{i-1})}} = \sqrt[N]{\frac{1}{\prod_{i=1}^N P(w_i | w_{i-1})}}$$


Unigram Precision

$$\text{Unigram Precision} = \frac{\text{Ref들 중에서 존재하는 Ca의 단어의 수}}{\text{Ca의 총 단어 수}} = \frac{\text{the number of Ca words(unigrams) which occur in any Ref}}{\text{the total number of words in the Ca}}$$

### Example 1

Candidate: 기계 번역된 문장

Reference: 사람이 직접 번역한 문장

- Candidate1 : It is a guide to action which ensures that the military always obeys the commands of the party.
- Candidate2 : It is to insure the troops forever hearing the activity guidebook that party direct.
- Reference1 : It is a guide to action that ensures that the military will forever heed Party commands.
- Reference2 : It is the guiding principle which guarantees the military forces always being under the command of the Party.
- Reference3 : It is the practical guide for the army always to heed the directions of the party.

## 03 BLEU Score Bilingual Evaluation Understudy Score

### - Unigram Precision

$$\text{Unigram Precision} = \frac{\text{Ref들 중에서 존재하는 Ca의 단어의 수}}{\text{Ca의 총 단어 수}} = \frac{\text{the number of Ca words(unigrams) which occur in any Ref}}{\text{the total number of words in the Ca}}$$

#### Example 2

Candidate: 기계 번역된 문장

Reference: 사람이 직접 번역한 문장

- Candidate : the the the the the the the
- Reference1 : the cat is on the mat
- Reference2 : there is a cat on the mat

$$\rightarrow \text{MUP} = \frac{2}{7}$$

### - Modified Unigram Precision

$$\text{Modified Unigram Precision} = \frac{\text{Ca의 각 유니그램에 대해 } Count_{clip} \text{을 수행한 값의 총 합}}{\text{Ca의 총 유니그램 수}} = \frac{\sum_{unigram \in Candidate} Count_{clip}(unigram)}{\sum_{unigram \in Candidate} Count(unigram)}$$

$$Count_{clip} = \min(Count, Max\_Ref\_Count) \rightarrow \text{중복 집계 제거!}$$

## 03 BLEU Score Bilingual Evaluation Understudy Score

### - Modified Unigram Precision

$$\text{Modified Unigram Precision} = \frac{\text{Ca의 각 유니그램에 대해 } Count_{clip} \text{을 수행한 값의 총합}}{\text{Ca의 총 유니그램 수}} = \frac{\sum_{unigram \in Candidate} Count_{clip}(unigram)}{\sum_{unigram \in Candidate} Count(unigram)}$$

$$p_1 = \frac{\sum_{unigram \in Candidate} Count_{clip}(unigram)}{\sum_{unigram \in Candidate} Count(unigram)} \quad \leftarrow \text{unigram( = 1-gram)}$$

#### Example 1

- Candidate1 : It is a guide to action which ensures that the military always obeys the commands of the party.
- Candidate2 : It is to insure the troops forever hearing the activity guidebook that party direct.
- **Candidate3 : the that military a is It guide ensures which to commands the of action obeys always party the.**
- Reference1 : It is a guide to action that ensures that the military will forever heed Party commands.
- Reference2 : It is the guiding principle which guarantees the military forces always being under the command of the Party.
- Reference3 : It is the practical guide for the army always to heed the directions of the party.

### - N-gram Precision

$$p_n = \frac{\sum_{n\text{-gram} \in Candidate} Count_{clip}(n\text{-gram})}{\sum_{n\text{-gram} \in Candidate} Count(n\text{-gram})}$$

#### Example 2

- Candidate1 : the the the the the the the
- Candidate2 : the cat the cat on the mat
- Reference1 : the cat is on the mat
- Reference2 : there is a cat on the mat

바이그램	the cat	cat the	cat on	on the	the mat	SUM
<i>Count</i>	2	1	1	1	1	6
<i>Count<sub>clip</sub></i>	1	0	1	1	1	4



## 03 BLEU Score Bilingual Evaluation Understudy Score

### - N-gram Precision

$$p_n = \frac{\sum_{n\text{-gram} \in \text{Candidate}} \text{Count}_{\text{clip}}(n\text{-gram})}{\sum_{n\text{-gram} \in \text{Candidate}} \text{Count}(n\text{-gram})}$$

$p_n$  : 각 gram의 보정된 정밀도입니다.

$N$  : n-gram에서  $n$ 의 최대 숫자입니다. 보통은 4의 값을 가집니다.  $N$ 이 4라는 것은  $p_1, p_2, p_3, p_4$ 를 사용한다는 것을 의미합니다.

$w_n$  : 각 gram의 보정된 정밀도에 서로 다른 가중치를 줄 수 있습니다. 이 가중치의 합은 1로 합니다. 예를 들어  $N$ 이 4라고 하였을 때,

$p_1, p_2, p_3, p_4$ 에 대해서 동일한 가중치를 주고자한다면 모두 0.25를 적용할 수 있습니다.

### - BLEU

$$BLEU = \exp\left(\sum_{n=1}^N w_n \log p_n\right)$$

### - Brevity Penalty

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}$$

#### Example 3

##### Example 1

Candidate4 : it is

- Candidate 1: I always invariably perpetually do.
- Candidate 2: I always do.
- Reference 1: I always do.
- Reference 2: I invariably do.
- Reference 3: I perpetually do.


$$BLEU = BP \times \exp\left(\sum_{n=1}^N w_n \log p_n\right)$$

## 03 BLEU Score Bilingual Evaluation Understudy Score

```
import nltk.translate.bleu_score as bleu
```

[Copy](#)

```
candidate = 'It is a guide to action which ensures that the military always obeys the commands of the party'
references = [
    'It is a guide to action that ensures that the military will forever heed Party commands',
    'It is the guiding principle which guarantees the military forces always being under the command of the Party',
    'It is the practical guide for the army always to heed the directions of the party'
]

print('실습 코드의 BLEU :', bleu_score(candidate.split(), list(map(lambda ref: ref.split(), references))))
print('패키지 NLTK의 BLEU :', bleu.sentence_bleu(list(map(lambda ref: ref.split(), references)), candidate.split()))
```

실습 코드의 BLEU : 0.5045666840058485

패키지 NLTK의 BLEU : 0.5045666840058485



감사합니다  
**Q&A**