

# Neural Machine Translation by Jointly Learning to Align and Translate

Dzmitry Bahdanau  
Jacobs University Bremen, Germany


KyungHyun Cho Yoshua Bengio\*  
Universite de Montreal

ICLR 2015

KISTI-UST  
JUYEON YU

2021.05.21.FRI

# Neural Machine Translation by Jointly Learning to **Align** and Translate



Attention이라는 용어가 직접 등장하지않고

Align이라는 용어로 처음 소개

# Authors



## Kyunghyun Cho

[New York University](#)  
Verified email at nyu.edu - [Homepage](#)  
[Machine Learning](#) [Deep Learning](#)

TITLE	CITED BY	YEAR
<a href="#">Neural machine translation by jointly learning to align and translate</a> D Bahdanau, K Cho, Y Bengio ICLR 2015	17940	2014
<a href="#">Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation</a> K Cho, B van Merriënboer, C Gulcehre, F Bougares, H Schwenk, ... Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)	13944	2014
<a href="#">Show, Attend and Tell: Neural Image Caption Generation with Visual Attention</a> K Xu, J Ba, R Kiros, K Cho, A Courville, R Salakhutdinov, R Zemel, ... International Conference on Machine Learning	7019	2015
<a href="#">Empirical evaluation of gated recurrent neural networks on sequence modeling</a> J Chung, C Gulcehre, KH Cho, Y Bengio arXiv preprint arXiv:1412.3555	6723	2014
<a href="#">On the Properties of Neural Machine Translation: Encoder-Decoder Approaches</a> K Cho, B van Merriënboer, D Bahdanau, Y Bengio Eighth Workshop on Syntax, Semantics and Structure in Statistical ...	3790	2014
<a href="#">Attention-based models for speech recognition</a> J Chorowski, D Bahdanau, D Serdyuk, K Cho, Y Bengio The Twenty-ninth Annual Conference on Neural Information Processing Systems ...	1769	2015

# Authors



## Yoshua Bengio

Professor of computer science, [University of Montreal](#), Mila, IVADO, CIFAR  
umontreal.ca의 이메일 확인됨 - [홈페이지](#)

[Machine learning](#) [deep learning](#) [artificial intelligence](#)

TITLE	CITED BY	YEAR
<a href="#">Deep learning</a> Y LeCun, Y Bengio, G Hinton nature 521 (7553), 436-444	38705	2015
<a href="#">Gradient-based learning applied to document recognition</a> Y LeCun, L Bottou, Y Bengio, P Haffner Proceedings of the IEEE 86 (11), 2278-2324	36513	1998
<a href="#">Generative adversarial networks</a> IJ Goodfellow, J Pouget-Abadie, M Mirza, B Xu, D Warde-Farley, S Ozair, ... arXiv preprint arXiv:1406.2661	31023	2014
<a href="#">Deep learning</a> I Goodfellow, Y Bengio, A Courville, Y Bengio MIT press 1 (2)	26069	2016
<a href="#">Neural machine translation by jointly learning to align and translate</a> D Bahdanau, K Cho, Y Bengio arXiv preprint arXiv:1409.0473	17940	2014
<a href="#">Learning phrase representations using RNN encoder-decoder for statistical machine translation</a> K Cho, B Van Merriënboer, C Gulcehre, D Bahdanau, F Bougares, ... arXiv preprint arXiv:1406.1078	13888	2014

# 1. Introduction

## Neural Network

### **이전 기계번역 방법들은 주로 Statistical Machine Translation**

- phrase based로서, 다양한 sub-component로 구성되어 있었고, 각 component는 각각 학습되고 구성

### **이후에는 하나의 큰 Neural Network를 이용한 translation 방법들이 제안**

- 대부분 encoder-decoder 형식
- Neural Network는 기존 system에 성능을 더하거나 결과를 re-rank하는 용도

### **이 논문에서는 오로지 Neural Network만을 사용**

### **Attention 기법을 사용해서 Neural Machine Translation의 성능을 향상시킨 내용**



# 1. Introduction

## Neural Machine Translation

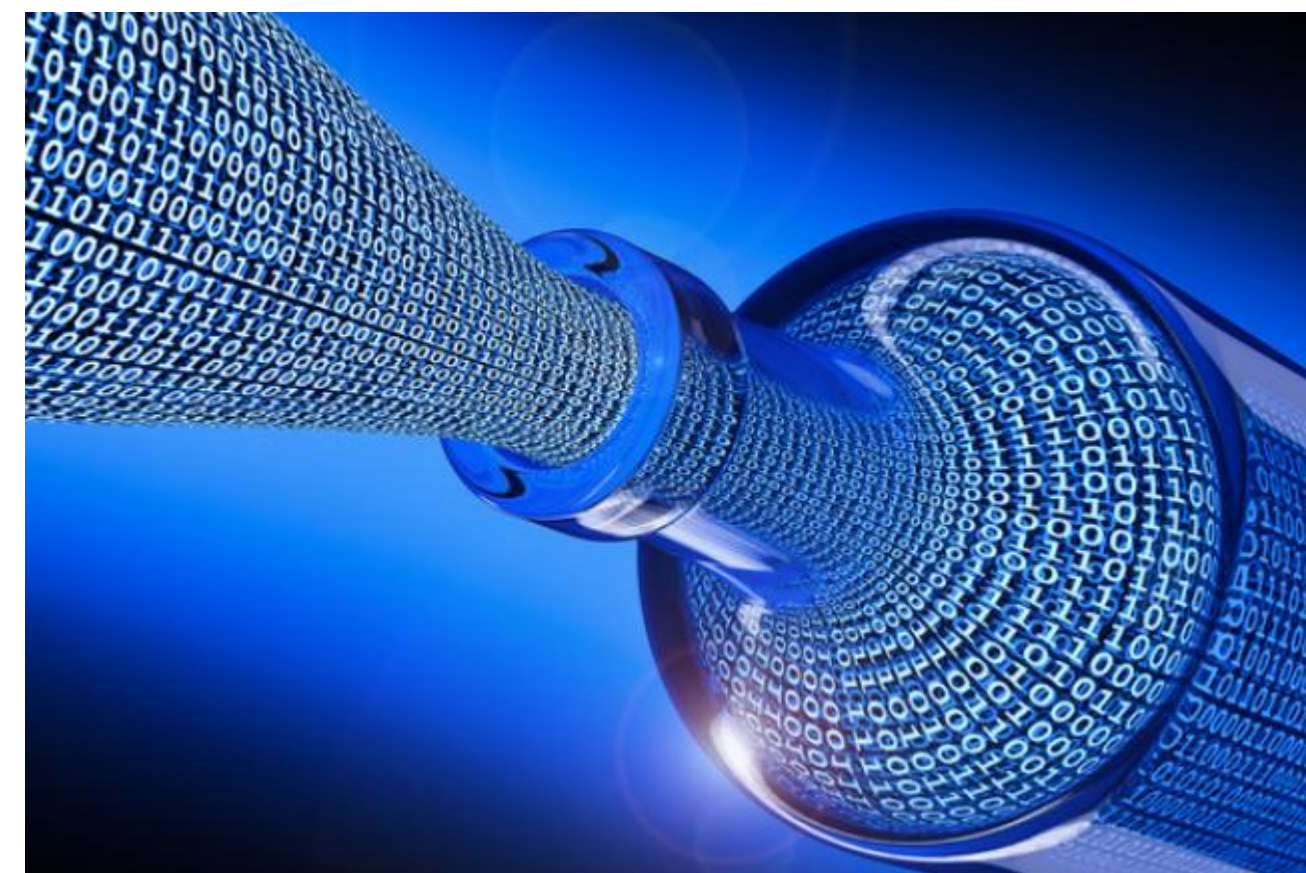
Encoder의 경우, 입력 문장을 고정 길이 벡터로 변환

Decoder는 해당 벡터를 이용해서 번역 결과를 생성

해당 모델의 입력으로 제공된 input sentence와 결과로 생성된 output sentence 사이의 probability를 최대화하는 방식으로 학습

이러한 encoder-decoder 구조는 Bottleneck 문제를 발생시킴

- encoder에서 전체 문장을 하나의 고정된 길이의 벡터로 생성할 때 발생
- 문장의 길이가 길어질수록 모델의 성능 저하
- 문장 전체의 정보를 짧은 고정 길이의 벡터로 모두 나타내기 어렵기 때문



## Main Idea

**이번 논문에서는 encoder-decoder 모델에서 새로운 구조를 추가해서 성능을 향상할 수 있는 방법을 제안**

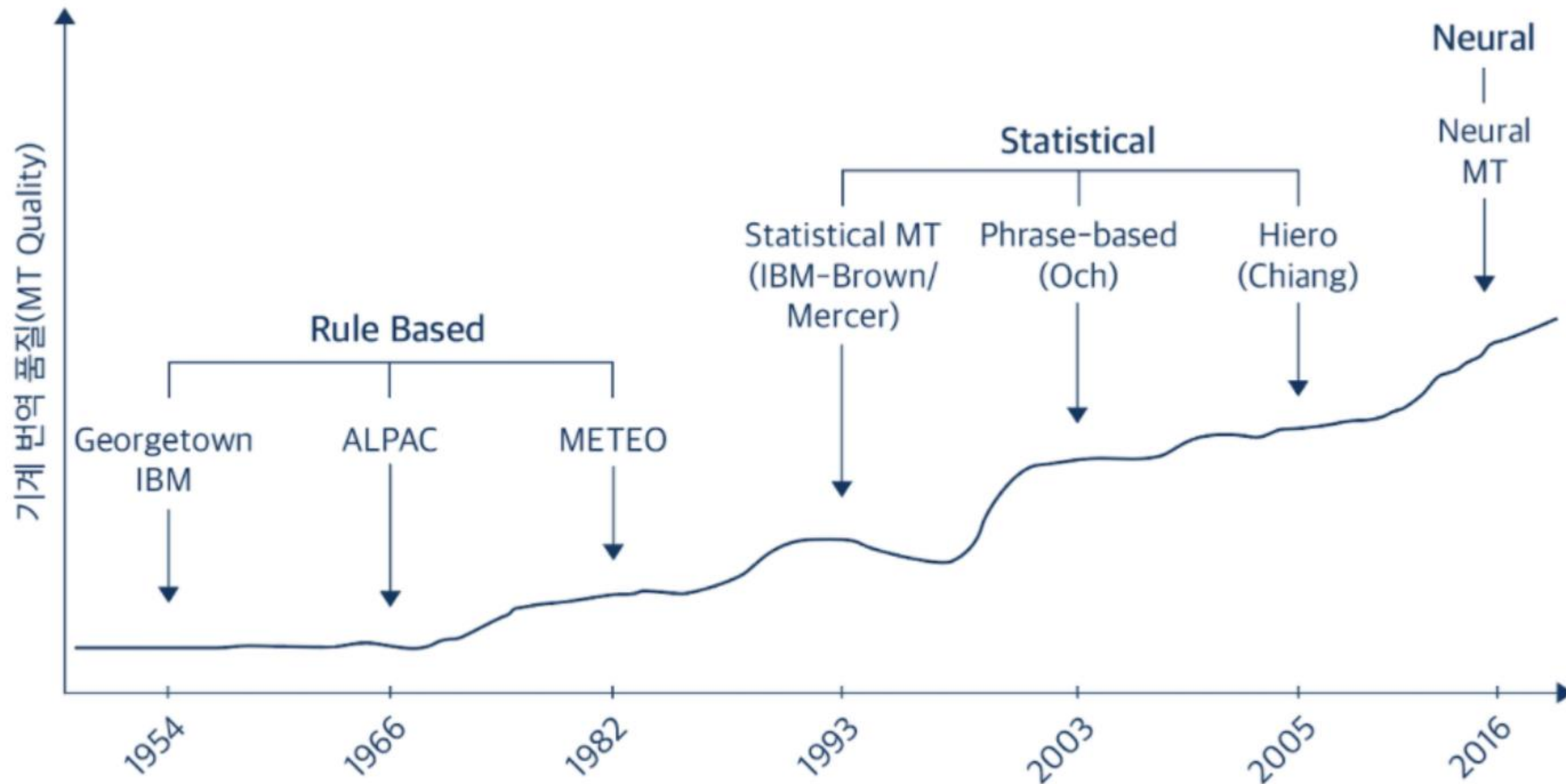
- decoder에서 하나의 결과를 만들어낼 때마다, 입력 문장을 순차적으로 탐색해서 현재 생성하려는 부분과 가장 관련있는 위치를 soft-search
- 최종적으로 encoder에서 생성한 context word 중 관련성이 크다고 판단되는 영역들과, decoder에서 이미 생성한 결과를 기반으로 다음 단어를 결과로 생성

**장점 중 하나는 입력 문장을 고정된 길이의 벡터로 표현하지 않아도 된다는 것**

- Decoder에서 연산을 진행하면서 encoder에서 생성한 context word를 계속해서 참조하기 때문에, 전체 문장의 정보를 하나의 벡터에 담으려고 하지 않아도 되고, 문장의 길이가 길어지더라도 성능을 유지

# 2. Background

## Machine Translation





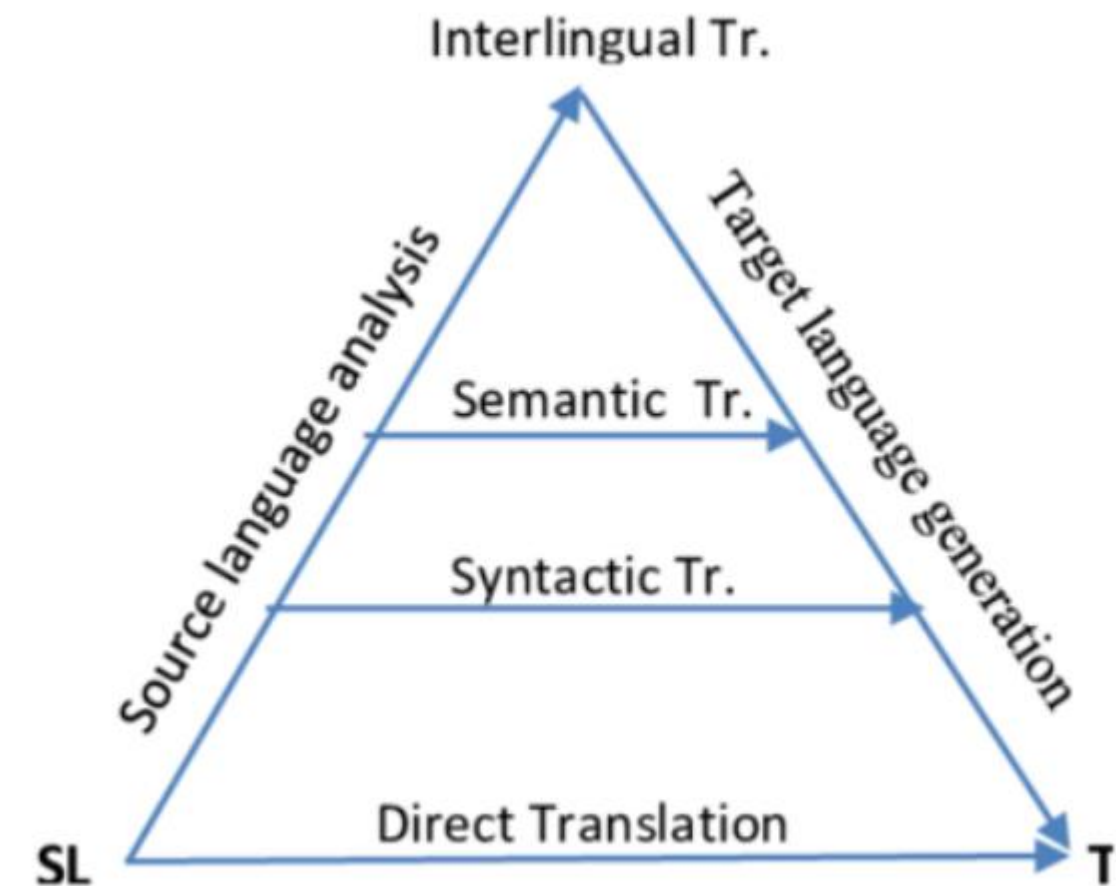
# 2. Background

## Rule-Based Machine Translation, RBMT

- 제2차 세계대전이 끝나고 냉전이 시작된 1950년대, 미국 국방성은 러시아어를 자동으로 번역하는 연구 시작
- 이때부터 1970년대에서 1980년대까지는 사람이 실제로 번역하는 프로세스를 본떠서 만든 기계 번역 고안
- 두 국가의 언어간 번역 사전을 사용하여 미리 정의된 mapping 관계에 의해 번역 수행
  - 형태소 분석(morphological analysis)
  - 구문 분석(syntactic analysis)
  - 의미 합성(semantic composition)

### 한계점

- 언어 특성상 언어학적 지식을 정확하게 정의하기 어려움
- 완벽한 번역이 어려움



# 2. Background

## Statistical Machine Translation, SMT

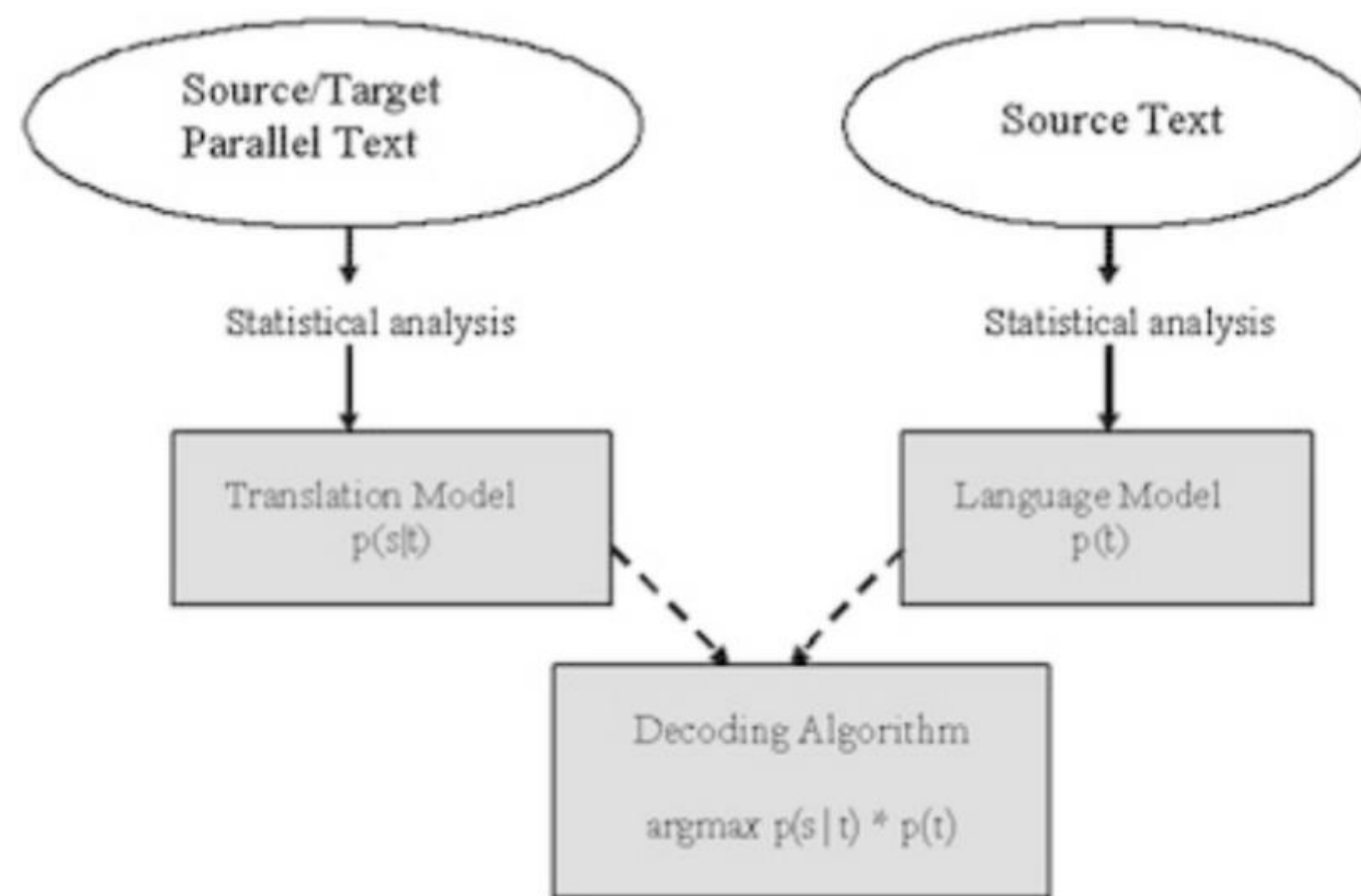
언어학자들이 설계한 규칙 기반 시스템보다 확률에 기반한 접근 방식의 결과가 더 좋음

단어나 구문이 가진 여러 가지 의미를 저장해 놓는 시스템

문장을 입력하면 이를 단어나 구문 단위로 쪼갬 뒤 통계적으로 가장 본래 의미에 가깝다고 판단되는 번역물을 내놓음

### 한계점

- SMT는 여러 모듈로 구성되어 시스템 복잡도 높음(NMT는 단 한개의 언어모델 생성)
- 풍부한 parallel data(언어1-언어2 대응 번역 데이터) 필요
- 입/출력 시퀀스 간에 대응관계인 단어가 없을 수 있음(일대다, 다대일)
- 국가 언어 간 signifi의 차이



# 2. Background

## Neural Machine Translation, NMT

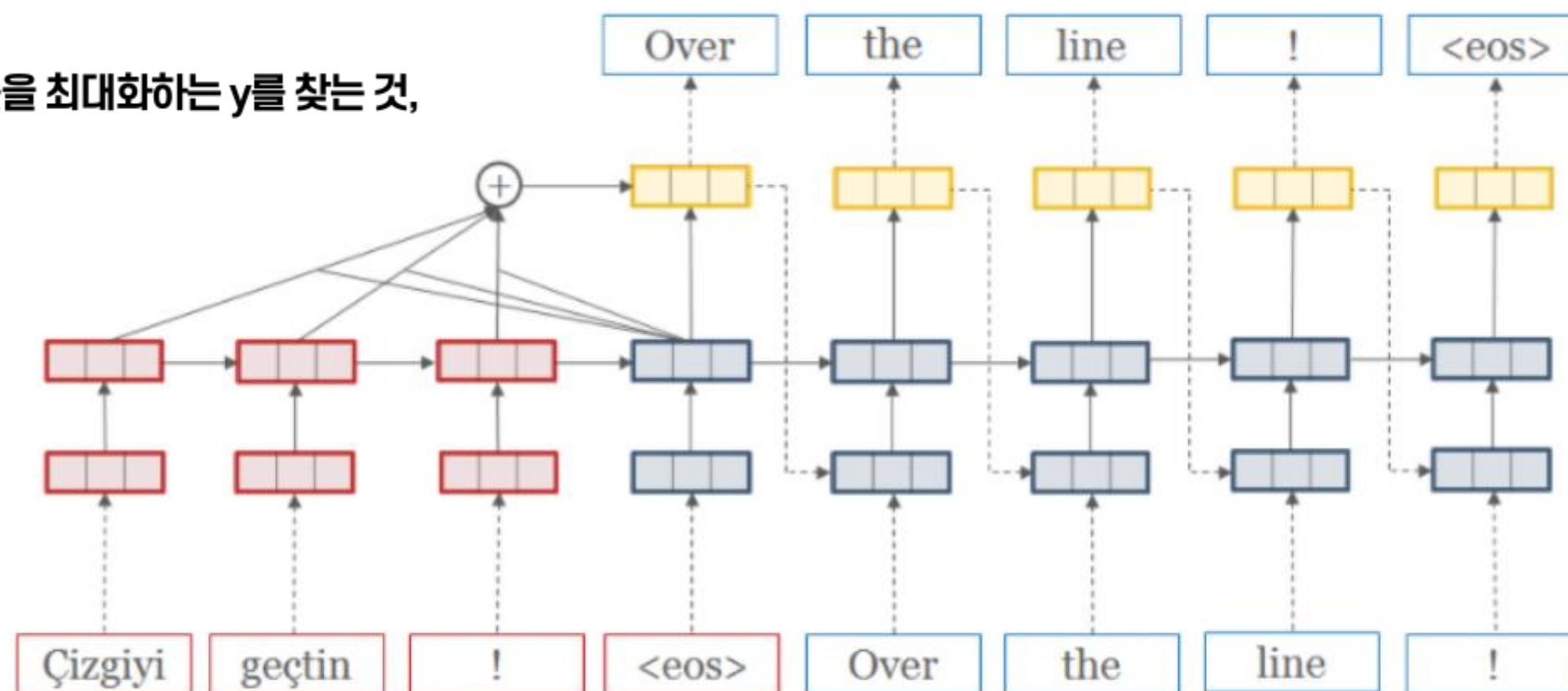
기계번역을 단일 신경망으로 수행하는 방법

해당 뉴럴네트워크 구조를 seq2seq라고 부르며, 이것은 두 개의 RNN을 포함

입력 문장을 문장 벡터 형태로 변환, 이를 기반으로 출력 언어의 문장을 생성하는 방식으로 번역을 수행

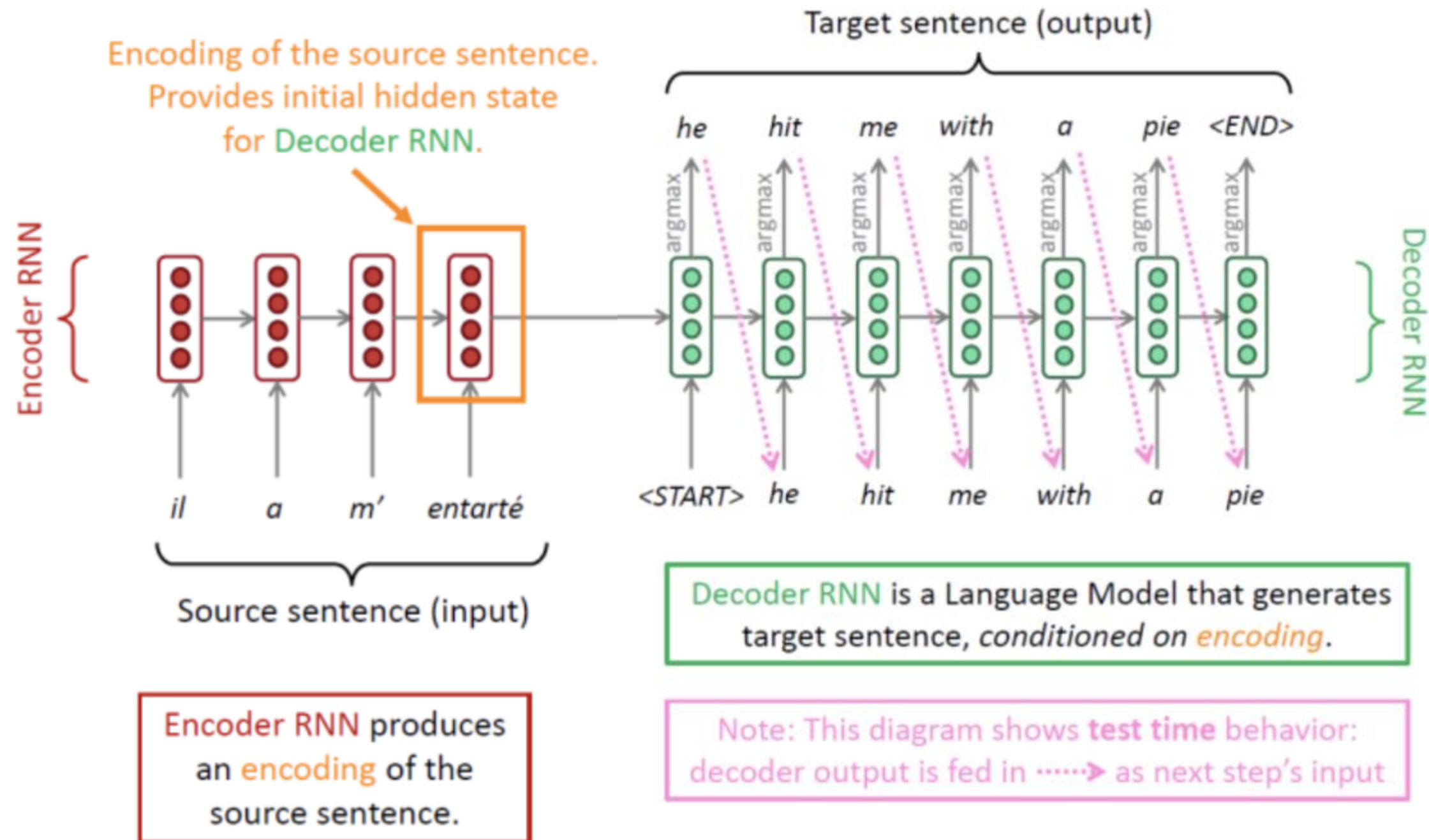
입력 문장을 문장 벡터로 변환하는 인코더와 출력 문장을 생성해내는 디코더는 인공신경망으로 구성

확률론적 관점에서 번역은 원 문장  $x$ 가 주어진 상태에서 타겟 문장  $y$ 의 조건부 확률을 최대화하는  $y$ 를 찾는 것,  
즉  $\operatorname{argmax}_y p(y|x)$



# 2. Background

sequence-to-sequence (seq2seq)



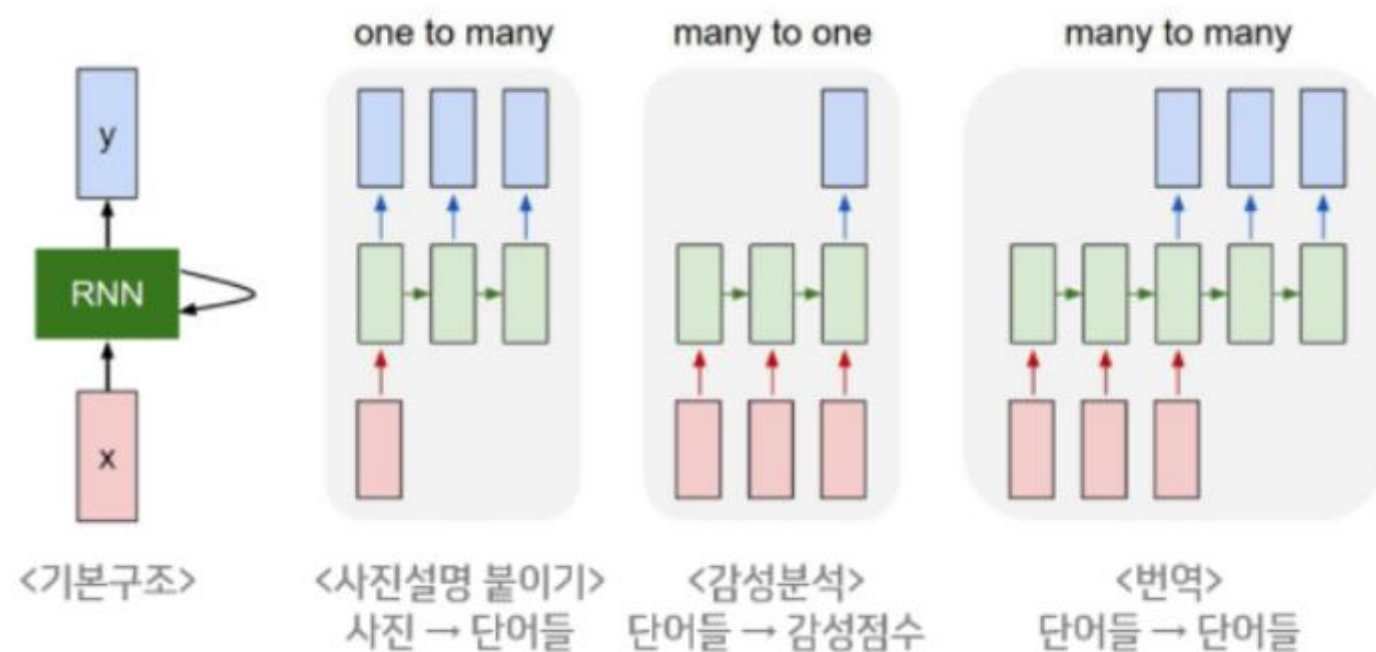


# 2. Background

## RNN의 기본 구조

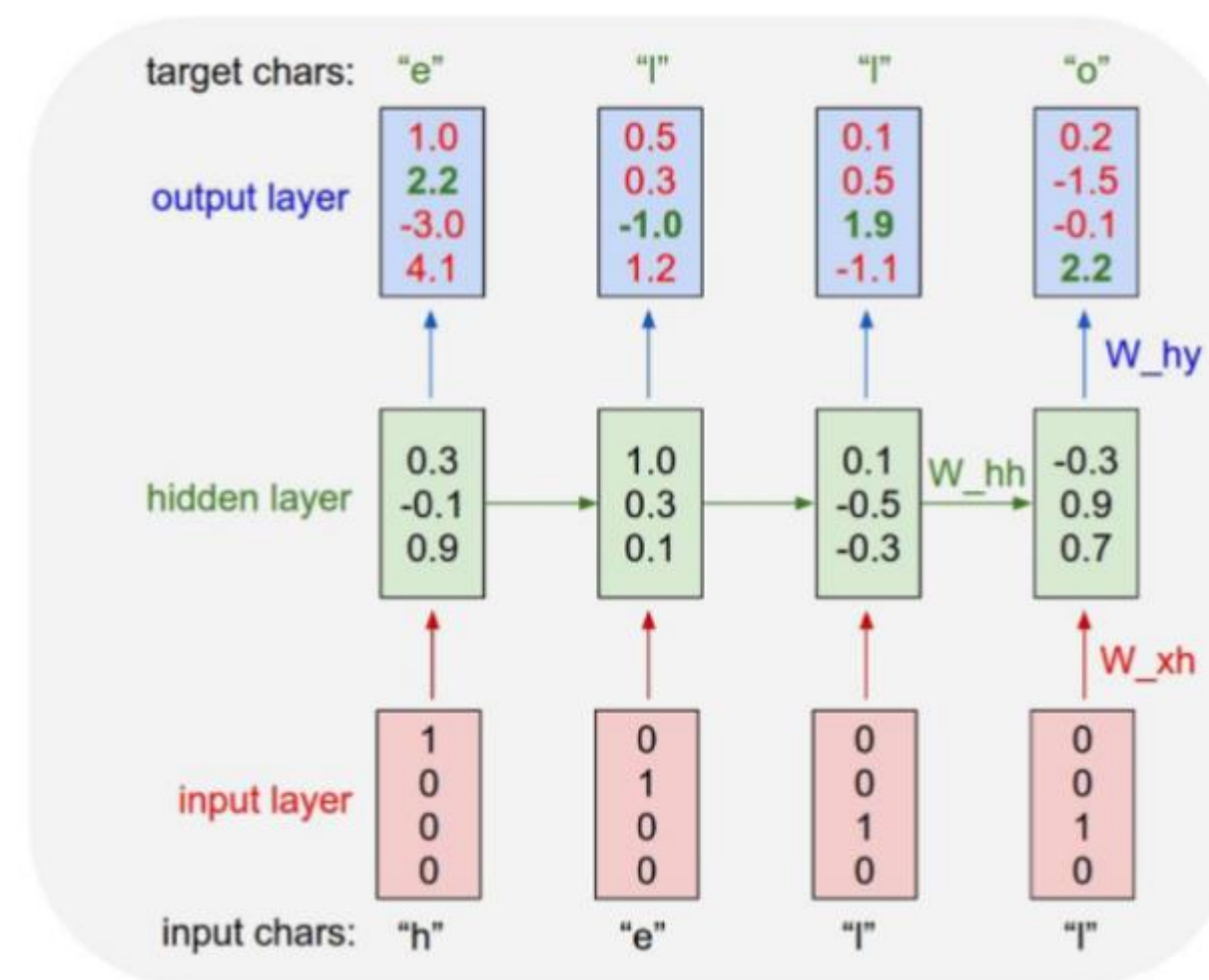
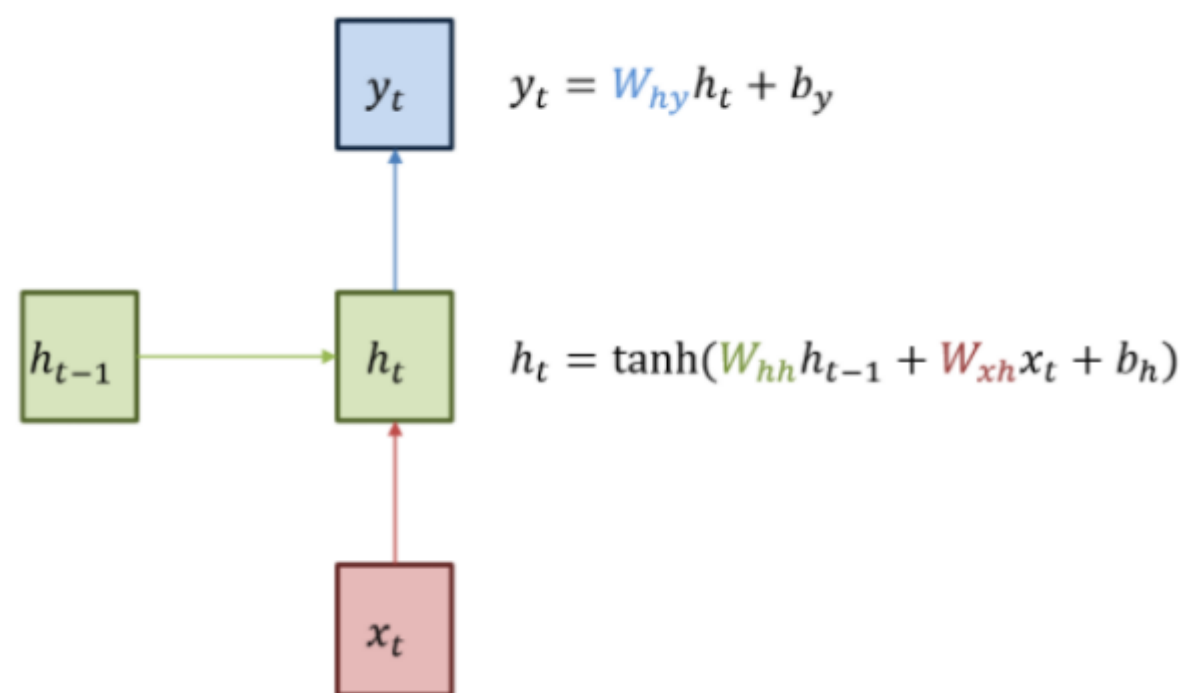
**RNN은 히든 노드가 방향을 가진 엣지로 연결돼 순환구조를 이루는(directed cycle) 인공지능망의 한 종류**

- 음성, 문자 등 순차적으로 등장하는 데이터 처리에 적합한 모델로 Convolutional Neural Networks(CNN)과 더불어 각광 받고 있는 알고리즘
- 시퀀스 길이에 관계없이 인풋과 아웃풋을 받아들일 수 있는 네트워크 구조이기 때문에 필요에 따라 다양하고 유연하게 구조를 만들 수 있다는 점이 RNN의 가장 큰 장점



# 2. Background

## RNN의 기본 구조



# 2. Background

## RNN Encoder-Decoder

- 인코더는 입력 문장, 즉 벡터 시퀀스  $x=(x_1,...,x_{T_x})$ 를 벡터  $c$ 의 형태로 읽어 냄.  
가장 일반적인 접근법은 아래와 같은 형태의 RNN을 사용하는 것
- $t$ 번째 단어의 representation과  $t-1$ 번째 단어의 hidden state를 어떠한 비선형 함수  $f$ 에 입력으로 넣어  $t$ 번째의 hidden state를 계산
  - 이 때  $h_t \in R^n$ 은 시간  $t$ 에서의 hidden state,  $c$ 는 hidden state의 시퀀스에서 생성된 context vector
  - $f$ 와  $q$ 는 어떠한 비선형 함수

$$h_t = f(x_t, h_{t-1})$$

$$c = q(h_1, \dots, h_{T_x})$$

$$\begin{cases} h_t & : \text{hidden state at time } t \\ q & : \text{non linear function (like LSTM)} \end{cases}$$

# 2. Background

## RNN Encoder-Decoder

- 디코더는 주로 다음 단어  $y_t$ 를 예측하기 위해 주어진 context 벡터  $c$ 와 이전의 모든 예측된 단어들  $y_1, \dots, y_{t-1}$ 을 이용하여 학습됨
- 즉 디코더는  $y=(y_1, \dots, y_T)$ 일 때, 다음과 같은 조건 하에서 결합확률(joint probability)을 분해하여 번역 문장  $y$ 에 대한 확률을 정의

$$p(y) = \prod_{t=1}^T p(y_t | y_1, \dots, y_{t-1}, c)$$

$$p(y_t | y_1, \dots, y_{t-1}, c) = g(y_{t-1}, s_t, c)$$

$$\begin{cases} y & : (y_1, \dots, y_t) \\ g & : \text{non linear function} \\ s_t & : \text{hidden state at time } t \end{cases}$$

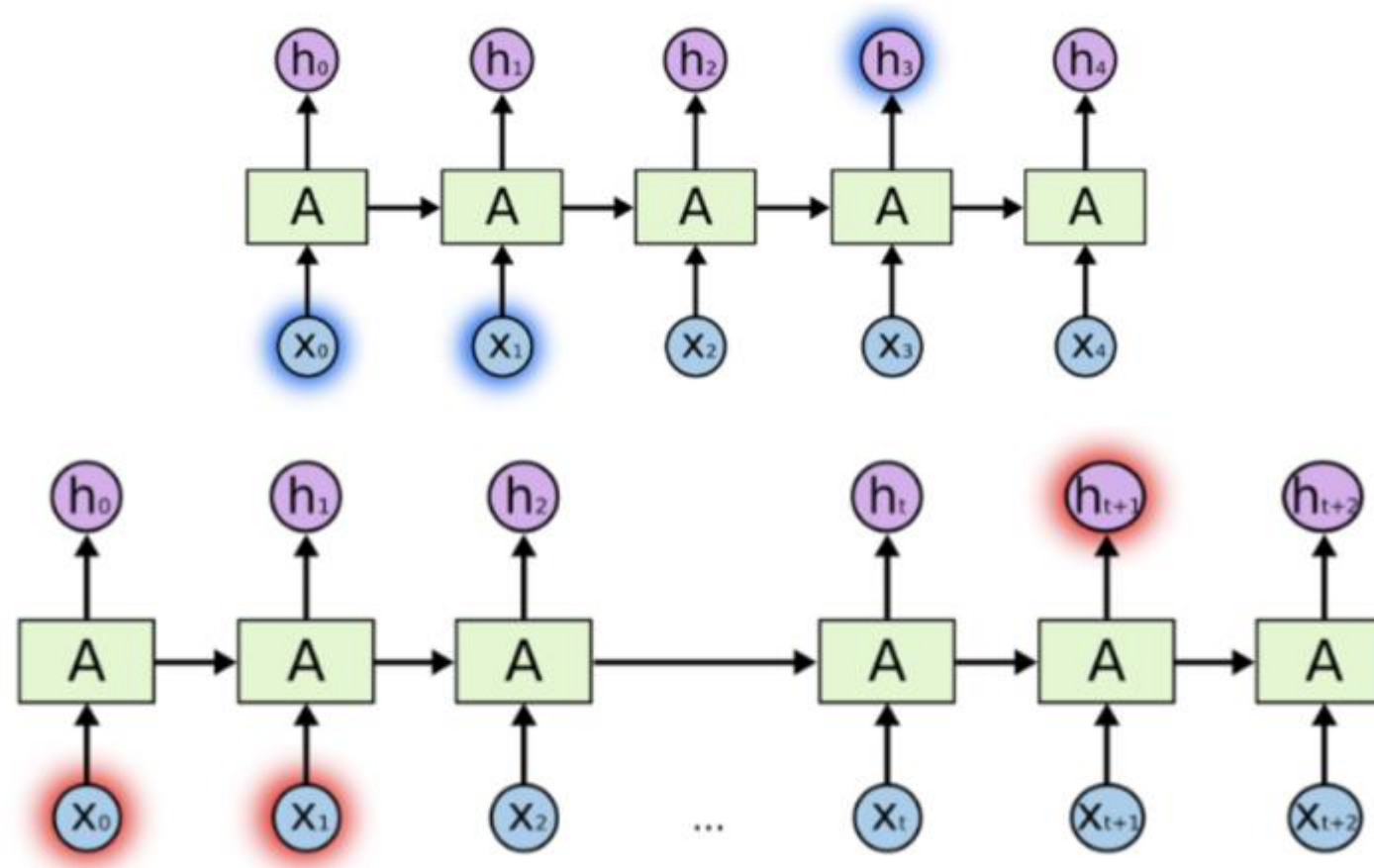


# 2. Background

## LSTM 네트워크 구조

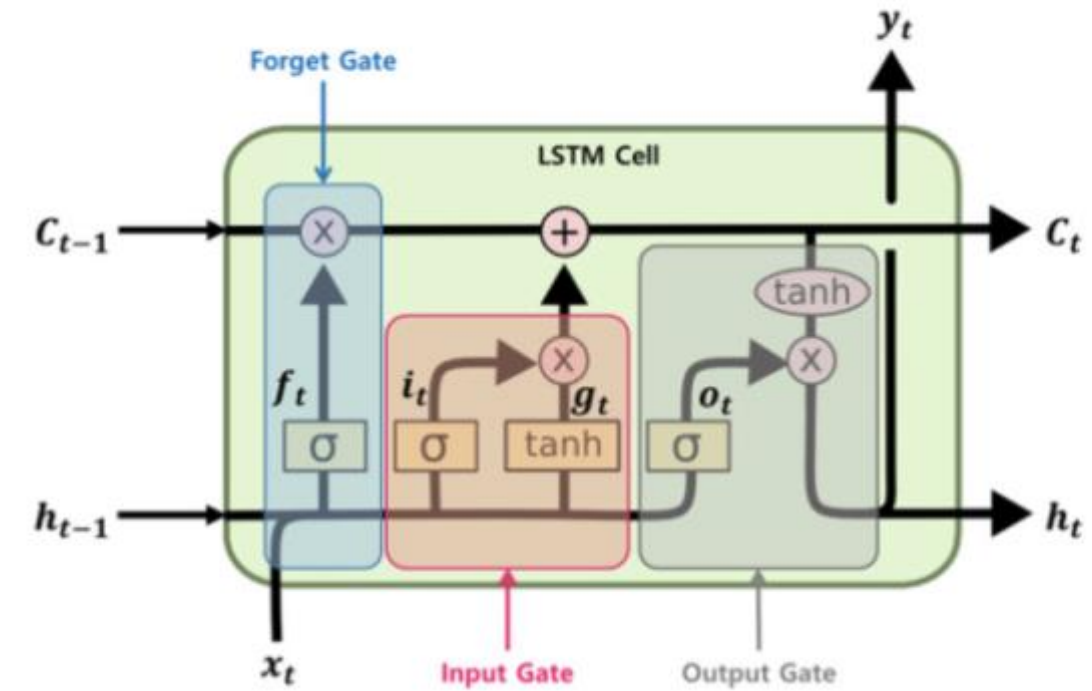
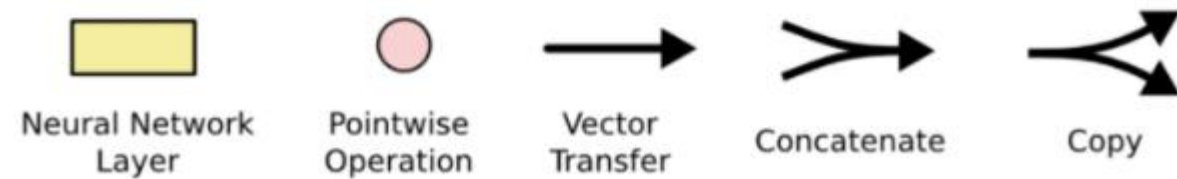
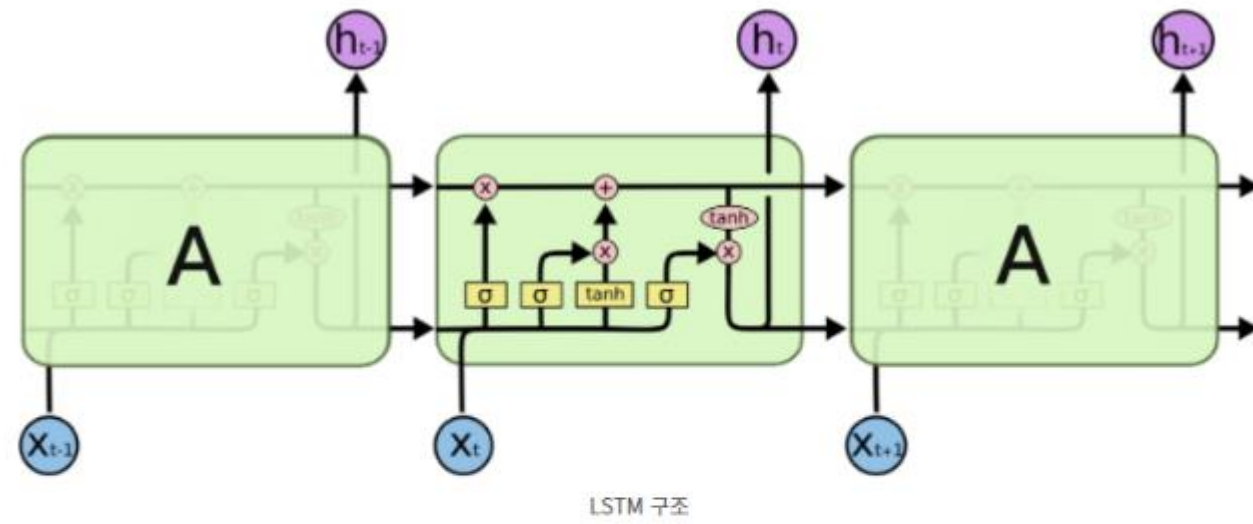
RNN은 관련 정보와 그 정보를 사용하는 지점 사이 거리가 멀 경우 역전파시 그래디언트가 점차 줄어 학습능력이 크게 저하(vanishing gradient problem)

이 문제를 극복하기 위해서 LSTM (RNN의 히든 state에 cell-state를 추가한 구조)



# 2. Background

## LSTM 네트워크 구조



# What is NMT

- **NMT consists of small sub-components that are tuned separately**
- **NMT models belong to a family of "Encoders-Decoders".**
- **Maximize the probability of a correct translation given a source sentence.**

**Fixed-length vector compress all the necessary information.**

- **Difficult for the NMT to cope with long sentences, especially those are longer than training corpus.**



# 3. Learning to align and translate

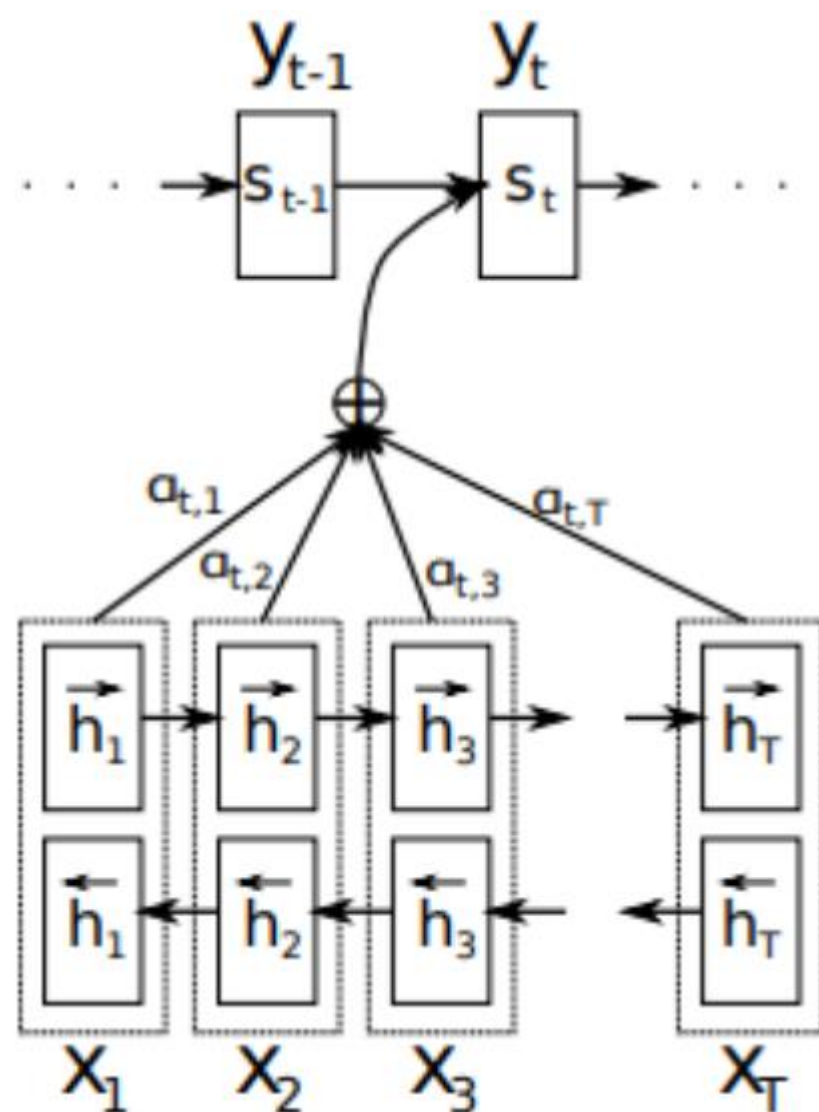


Figure 1: The graphical illustration of the proposed model trying to generate the  $t$ -th target word  $y_t$  given a source sentence  $(x_1, x_2, \dots, x_T)$ .

BiRNN : 순방향(forward)와 역방향(backward) RNN

순방향 RNN  $\vec{f}$  sequence를 처음부터 순서대로 읽고 forward hidden state  $(\vec{h}_1, \dots, \vec{h}_{T_x})$  를 계산

역방향 RNN  $\overleftarrow{f}$  는 sequence를 역방향으로 마지막부터 처음까지 읽고 backward hidden state  $(\overleftarrow{h}_1, \dots, \overleftarrow{h}_{T_x})$  를 계산

각 단어  $x_j$ 에 대해서 forward hidden state 와 backward hidden state를 concatenate

$$h_j = \left[ \vec{h}_j^\top; \overleftarrow{h}_j^\top \right]^\top$$

이 방법을 통해 annotation  $h_j$  는  $j$ 번째 단어 앞뒤의 정보를 모두 포함

# 3. Learning to align and translate

Previous

Encoder

$$h_t = f(x_t, h_{t-1}) \\ = \text{sigm}(W^{hx}x_t + W^{hh}h_{t-1})$$



Paper

Encoder

$$h_t = f(x_t, h_{t-1}) \\ = \text{sigm}(W^{hx}x_t + W^{hh}h_{t-1})$$

$$c = q(\{h_1, h_2, \dots, h_{T_x}\}) \\ = \prod_{t=1}^{T_x} p(h_t | \{h_1, \dots, h_{t-1}\})$$



$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \\ \alpha_{ij} = \frac{\exp e_{ij}}{\sum_{k=1}^{T_x} \exp e_{ik}} \\ e_{ij} = a(s_{i-1}, h_j)$$

$a$  : alignment model

Decoder

$$p(\mathbf{y}) = \prod_{t=1}^T p(y_t | y_1, \dots, y_{t-1}, c)$$

$$p(y_t | y_1, \dots, y_{t-1}, c) = g(y_{t-1}, s_t, c)$$

$$\begin{cases} \mathbf{y} & : (y_1, \dots, y_t) \\ g & : \text{non linear function} \\ s_t & : \text{hidden state at time } t \end{cases}$$



Decoder

$$p(y_t | y_1, \dots, y_{t-1}, \mathbf{x}) = g(y_{t-1}, s_i, c_i)$$

$$s_i = f(s_{i-1}, y_{i-1}, c_i)$$

# 3. Learning to align and translate

## Decoder

$$C = q(\{h_1, h_2, \dots, h_{T_x}\}) \\ = \prod_{t=1}^{T_x} p(h_t | \{h_1, \dots, h_{t-1}\})$$



$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \\ \alpha_{ij} = \frac{\exp e_{ij}}{\sum_{k=1}^{T_x} \exp e_{ik}} \\ e_{ij} = a(s_{i-1}, h_j) \\ a : \text{alignment model}$$

### A.2.2 DECODER

The hidden state  $s_i$  of the decoder given the annotations from the encoder is computed by

$$s_i = (1 - z_i) \circ s_{i-1} + z_i \circ \tilde{s}_i,$$

where

$$\tilde{s}_i = \tanh(W E y_{i-1} + U[r_i \circ s_{i-1}] + C c_i)$$

$$z_i = \sigma(W_z E y_{i-1} + U_z s_{i-1} + C_z c_i)$$

$$r_i = \sigma(W_r E y_{i-1} + U_r s_{i-1} + C_r c_i)$$

$E$  is the word embedding matrix for the target language.  $W, W_z, W_r \in \mathbb{R}^{n \times m}$ ,  $U, U_z, U_r \in \mathbb{R}^{n \times n}$ , and  $C, C_z, C_r \in \mathbb{R}^{n \times 2n}$  are weights. Again,  $m$  and  $n$  are the word embedding dimensionality and the number of hidden units, respectively. The initial hidden state  $s_0$  is computed by  $s_0 = \tanh(W_s \overleftarrow{h}_1)$ , where  $W_s \in \mathbb{R}^{n \times n}$ .

The context vector  $c_i$  are recomputed at each step by the alignment model:

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j,$$

where

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} \\ e_{ij} = v_a^\top \tanh(W_a s_{i-1} + U_a h_j),$$

and  $h_j$  is the  $j$ -th annotation in the source sentence (see Eq. (7)).  $v_a \in \mathbb{R}^{n'}$ ,  $W_a \in \mathbb{R}^{n' \times n}$  and  $U_a \in \mathbb{R}^{n' \times 2n}$  are weight matrices. Note that the model becomes RNN Encoder-Decoder (Cho *et al.*, 2014a), if we fix  $c_i$  to  $\overrightarrow{h}_{T_x}$ .

With the decoder state  $s_{i-1}$ , the context  $c_i$  and the last generated word  $y_{i-1}$ , we define the probability of a target word  $y_i$  as

$$p(y_i | s_i, y_{i-1}, c_i) \propto \exp(y_i^\top W_o t_i),$$

where

$$t_i = [\max \{\tilde{t}_{i,2j-1}, \tilde{t}_{i,2j}\}]_{j=1, \dots, l}^\top$$

and  $\tilde{t}_{i,k}$  is the  $k$ -th element of a vector  $\tilde{t}_i$  which is computed by

$$\tilde{t}_i = U_o s_{i-1} + V_o E y_{i-1} + C_o c_i.$$

$W_o \in \mathbb{R}^{K_y \times l}$ ,  $U_o \in \mathbb{R}^{2l \times n}$ ,  $V_o \in \mathbb{R}^{2l \times m}$  and  $C_o \in \mathbb{R}^{2l \times 2n}$  are weight matrices. This can be understood as having a deep output (Pascanu *et al.*, 2014) with a single maxout hidden layer (Goodfellow *et al.*, 2013).



# 3. Learning to align and translate

Encoder

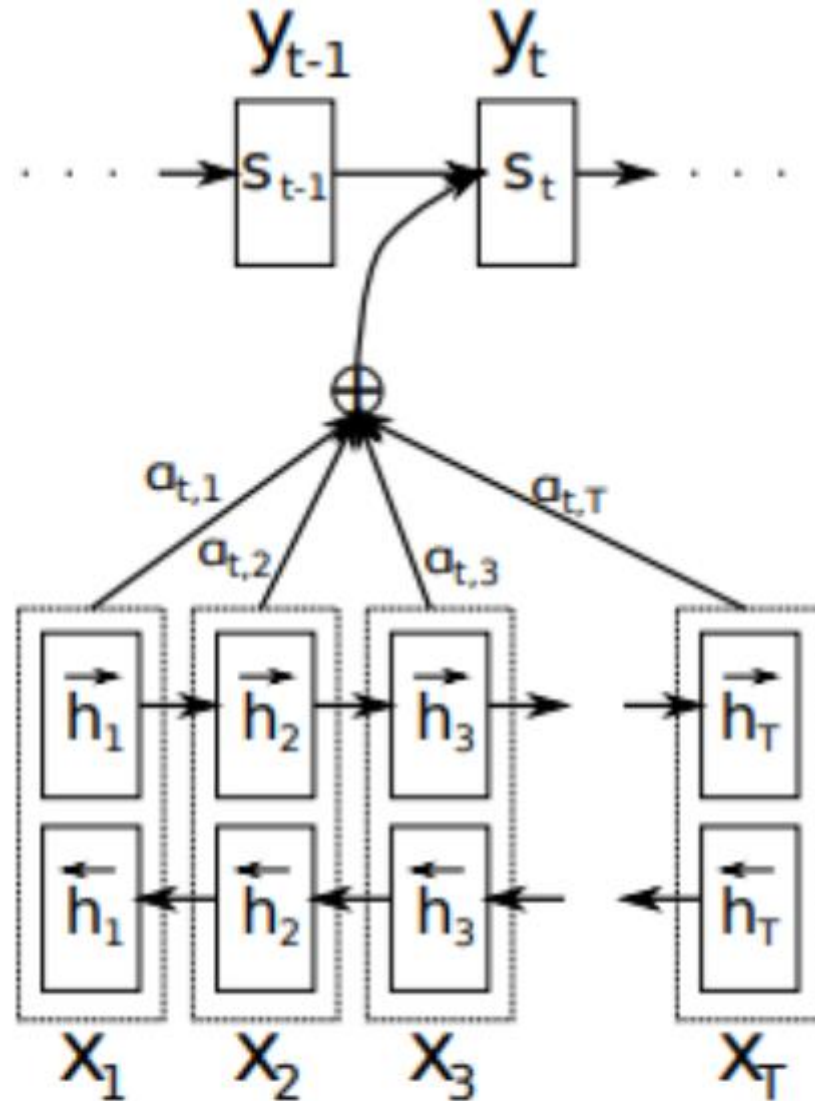


Figure 1: The graphical illustration of the proposed model trying to generate the  $t$ -th target word  $y_t$  given a source sentence  $(x_1, x_2, \dots, x_T)$ .

$$\mathbf{x} = (x_1, \dots, x_{T_x}), x_i \in \mathbb{R}^{K_x}$$

$$\mathbf{y} = (y_1, \dots, y_{T_y}), y_i \in \mathbb{R}^{K_y}$$

$$\vec{h}_i = \begin{cases} (1 - \vec{z}_i) \circ \vec{h}_{i-1} + \vec{z}_i \circ \vec{h}_i & , \text{if } i > 0 \\ 0 & , \text{if } i = 0 \end{cases}$$

where

$$\vec{h}_i = \tanh(\vec{W} \vec{E} x_i + \vec{U} [\vec{r}_i \circ \vec{h}_{i-1}])$$

$$\vec{z}_i = \sigma(\vec{W}_z \vec{E} x_i + \vec{U}_z \vec{h}_{i-1})$$

$$\vec{r}_i = \sigma(\vec{W}_r \vec{E} x_i + \vec{U}_r \vec{h}_{i-1}).$$

$E$  = embedding matrix

$W, U$  = Weight matrices



# 4. Experiment Settings

- Target task : English-French translation -> bilingual parallel corpora ACL WMT 14
- Datasest(850M) : 61M words Europarl, 5.5M news comments, 421M UN, 90M & 272.5M crawled corpora
- Usage(348M) : data selection metod(Axelrod, "Pseudo-in-domain subcorpora")
- Portion : news-test-2012,2013 (validation set), news-test-2014 (test sest)
- Preprocessing : UNK token, create 30,000 frequent words list (No lowercasing, No stemming)

# 4. Experiment Settings

## Models

each model trained twice

### 1. RNN Encoder - Decoder (RNNencdec)

- up to 30 words sentence
- up to 50 words sentence
- 1000 hidden units of encoder when forward + 1000 hidden units of encoder when backward

### 2. Attention (RNNsearch)

- up to 30 words sentence
- up to 50 words sentence
- 1000 hidden units of encoder when forward + 1000 hidden units of encoder when backward
- 1000 hidden units of decoder

### 3. Optimizer

- AdaDelta( $\epsilon = 10^{-6}$ ,  $\rho = 0.95$ )
- Mini\_batch = 80 sentences

# 4. Experiment Settings

## Models

### 4. Initialization

- L2-norm normalization for gradient of cost function
- Stochastic gradient descent (SGD)
- hidden layer( $n$ ) = 1000, word-embedding-dim( $m$ ) = 620
- RNN weight matrices as "Random or thogonal matrices"
- Sampling elements from "Gaussian distribution" (mean = 0, variance =  $0.001^2$ )

## Quantitative Results

BLEU score를 기반으로 translation 성능을 평가

RNNsearch 가 모든 경우 더 뛰어난 성능을 제공

기존의 phrase-based translation system(Moses) 와 유사한 성능

- Moses의 모델의 경우 추가적인 monolingual corpus를 사용
- RNNsearch모델이 뛰어난 성능을 제공한다는 것을 알 수 있음

Model	All	No UNK <sup>o</sup>
RNNencdec-30	13.93	24.19
RNNsearch-30	21.50	31.44
RNNencdec-50	17.82	26.71
RNNsearch-50	26.75	34.16
RNNsearch-50*	28.45	36.15
Moses	33.30	35.63

Table 1: BLEU scores of the trained models computed on the test set. The second and third columns show respectively the scores on all the sentences and, on the sentences without any unknown word in themselves and in the reference translations. Note that RNNsearch-50\* was trained much longer until the performance on the development set stopped improving. (o) We disallowed the models to generate [UNK] tokens when only the sentences having no unknown words were evaluated (last column).

## Quantitative Results

### BLEU (Bilingual Evaluation Understudy)

- 언어를 번역하는 기계 번역에서 모델의 성능 평가하는 탁월한 알고리즘
- 생성된 결과가 사람 수준의 번역에 가까울수록 점수가 높음
- 현재까지는 모델 출려과 사람 수준의 출력을 비교하는 데 가장 널리 사용되는 측정법

### 원리

- 생성된 텍스트 캡션을 정답 캡션의 세트로 평가하는 것(일반적으로 하나의 캐션을 하나 또는 그 이상의 캡션으로 평가)
- 점수는 각 캡션에 대해 계산한 후 전체 코퍼스에 평균화해 전반적인 품질 평가를 얻음
- 0에서 1까지의 범위이며 점수가 1에 가까울수록 고품질 번역



# 5. Results

## Qualitative Analysis

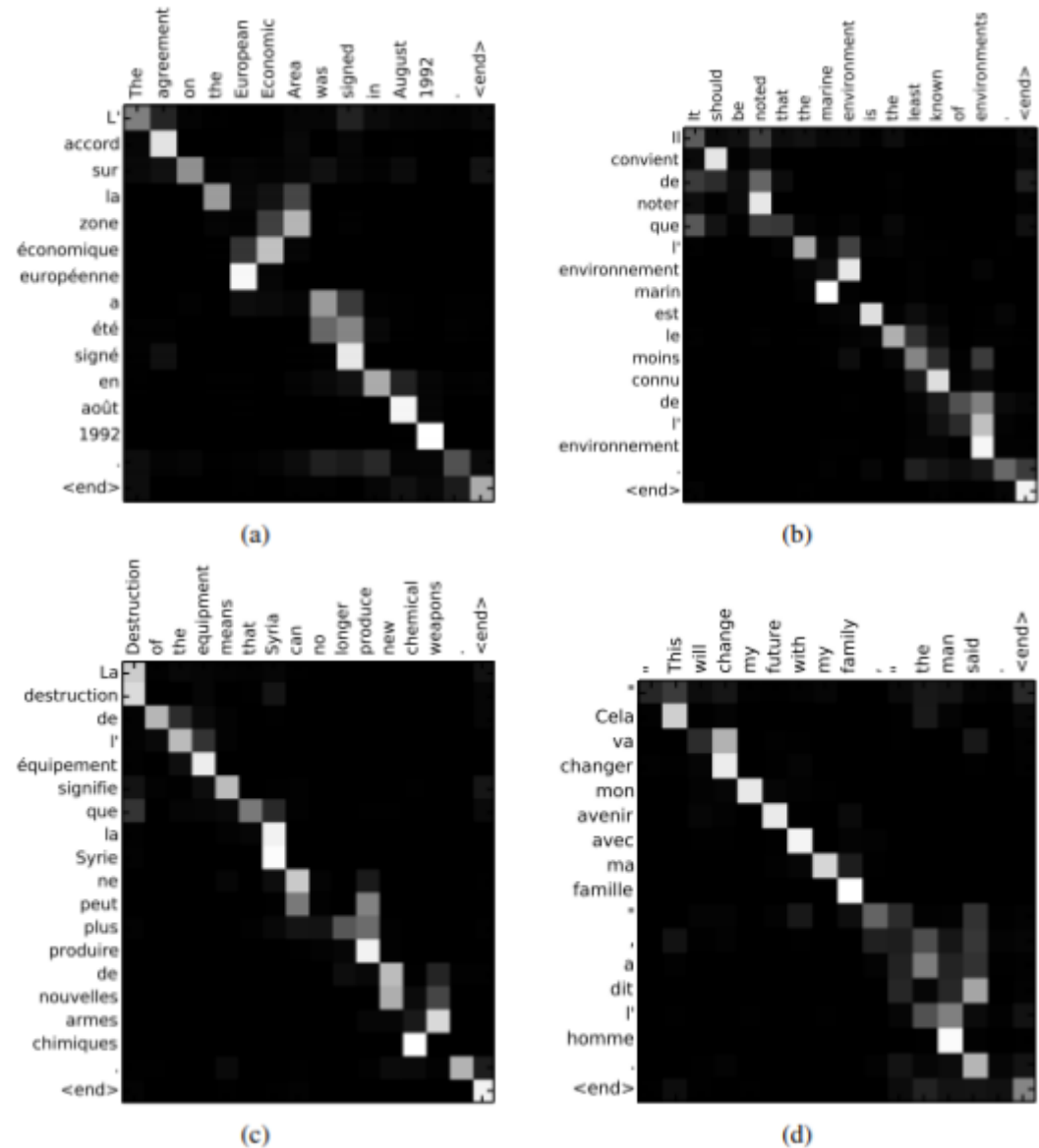


Figure 3: Four sample alignments found by RNNsearch-50. The x-axis and y-axis of each plot correspond to the words in the source sentence (English) and the generated translation (French), respectively. Each pixel shows the weight  $\alpha_{ij}$  of the annotation of the  $j$ -th source word for the  $i$ -th target word (see Eq. (6)), in grayscale (0: black, 1: white). (a) an arbitrary sentence. (b–d) three randomly selected samples among the sentences without any unknown words and of length between 10 and 20 words from the test set.

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})},$$

$$e_{ij} = a(s_{i-1}, h_j)$$

The agreement on the European Economic Area was signed in August 1992.  
L' accord sur la zone économique européenne a été signé en août 1992.

Destruction of the equipment means that Syria can no longer produce new chemical weapons.  
La destruction de l' équipement signifie que la Syrie ne peut plus produire de Nouvelles armes chimiques.



# 5. Results

## Qualitative Analysis

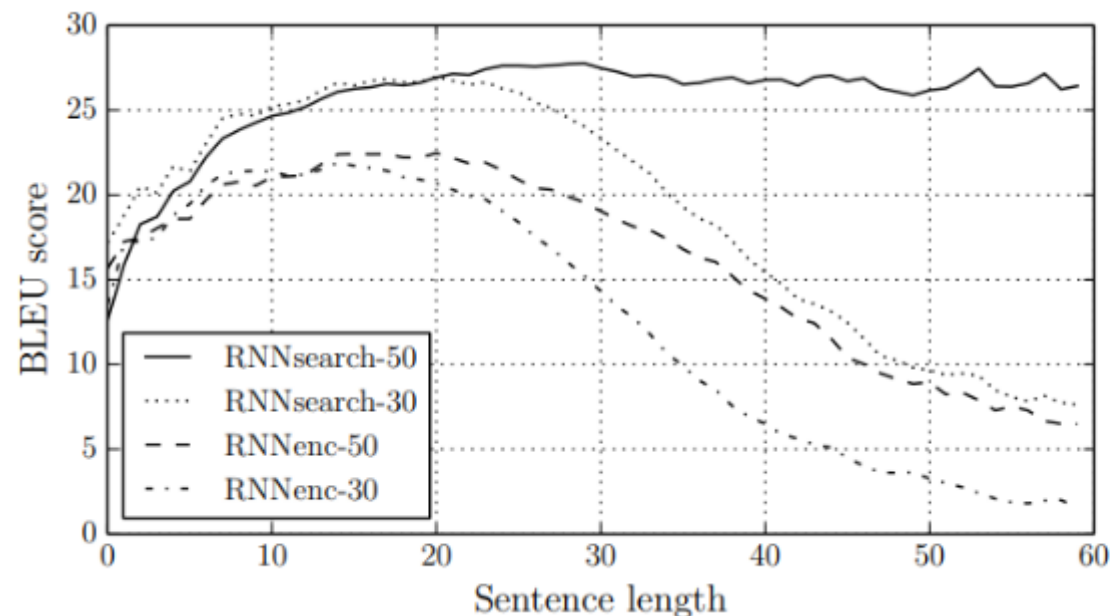


Figure 2: The BLEU scores of the generated translations on the test set with respect to the lengths of the sentences. The results are on the full test set which includes sentences having unknown words to the models.

As an example, consider this source sentence from the test set:

*An admitting privilege is the right of a doctor to admit a patient to a hospital or a medical centre to carry out a diagnosis or a procedure, based on his status as a health care worker at a hospital.*

The RNNencdec-50 translated this sentence into:

*Un privilège d'admission est le droit d'un médecin de reconnaître un patient à l'hôpital ou un centre médical d'un diagnostic ou de prendre un diagnostic en fonction de son état de santé.*

On the other hand, the RNNsearch-50 generated the following correct translation, preserving the whole meaning of the input sentence without omitting any details:

*Un privilège d'admission est le droit d'un médecin d'admettre un patient à un hôpital ou un centre médical pour effectuer un diagnostic ou une procédure, selon son statut de travailleur des soins de santé à l'hôpital.*

Let us consider another sentence from the test set:

*This kind of experience is part of Disney's efforts to "extend the lifetime of its series and build new relationships with audiences via digital platforms that are becoming ever more important," he added.*

The translation by the RNNencdec-50 is

*Ce type d'expérience fait partie des initiatives du Disney pour "prolonger la durée de vie de ses nouvelles et de développer des liens avec les lecteurs numériques qui deviennent plus complexes.*

As with the previous example, the RNNencdec began deviating from the actual meaning of the source sentence after generating approximately 30 words (see the underlined phrase). After that point, the quality of the translation deteriorates, with basic mistakes such as the lack of a closing quotation mark.

Again, the RNNsearch-50 was able to translate this long sentence correctly:

*Ce genre d'expérience fait partie des efforts de Disney pour "prolonger la durée de vie de ses séries et créer de nouvelles relations avec des publics via des plateformes numériques de plus en plus importantes", a-t-il ajouté.*

## Qualitative Analysis

- annotation weights( $a_{ij}$ )
  - strong weights along the diagonal of each matrix
  - Adjective and nouns are typically ordered differently between french and english
- The strength of the soft-alignment
  1. the  $\rightarrow$  I[le], [la]로 번역되는 경우가 있다.  $\rightarrow$  solving the model look at both [the] and [man]
  2. source and target phrases of different lengths requires a mapping some words to or from nowhere[NULL]

# 6. Related Work

## Neural networks for machine translation

\*2003(Benjo) =introduction of a neural probabilistic language model

Limitations of providing a single feature to an existing SMT to rerank a list of candidate translations

2012(Schwenk) =using feedforward neural network to compute score of a pair of source-target phrases

2013(Kalchbrenner), 2014(Devlin) = Recurrent Continuous translation models

this = Rather than using a neural network as a part of the existing system, generates a translation from a source sentence directly



# 7. Conclusion

seq2seq by letting a model soft-search for a set of input words  
annotations computed by an encoder, when generating each target word  
free from source sentence fixed length  
focus on information relevant to the generation of the next target word.

- \* left challenges : to better handle unknown, or rare words
  - > this required to be more widely used and to match the performance in MT systems in all



# References

1. <https://www.youtube.com/watch?v=l9pWT6BHpj0&list=PLkZZN0wo7X2PCnqe-uygTEYkaQoZ4JF2F&index=2>
2. <https://heiwais25.github.io/nlp/2019/06/18/neural-machine-translation-by-jointly-learning-to-align-and-translate/>
3. <https://misconstructed.tistory.com/49>
4. <https://bkshin.tistory.com/entry/NLP-14-%EC%96%B4%ED%85%90%EC%85%98Attention?category=1097026>
5. <https://ratsgo.github.io/from%20frequency%20to%20semantics/2017/10/06/attention/>
6. <https://www.youtube.com/watch?v=upskBSbA9cA>
7. 밑바닥부터 시작하는 딥러닝2

# THANK YOU

## Q & A