How to Translate?

(cs224n: Translation, Seq2Seq, Attention)

자연어 심화세미나

TOBIG'S 19기 한진모

Contents

Unit 01

Statistical Translate Model

Unit 01 | Statistical Translate Model

번역이란 무엇인가?

- 영어를 한국어로 번역하는 문제를 생각하자.
- Input data(X): 영어 문장(Source language sentence)
- Output data(y): 한국어 문장(Target language sentence)

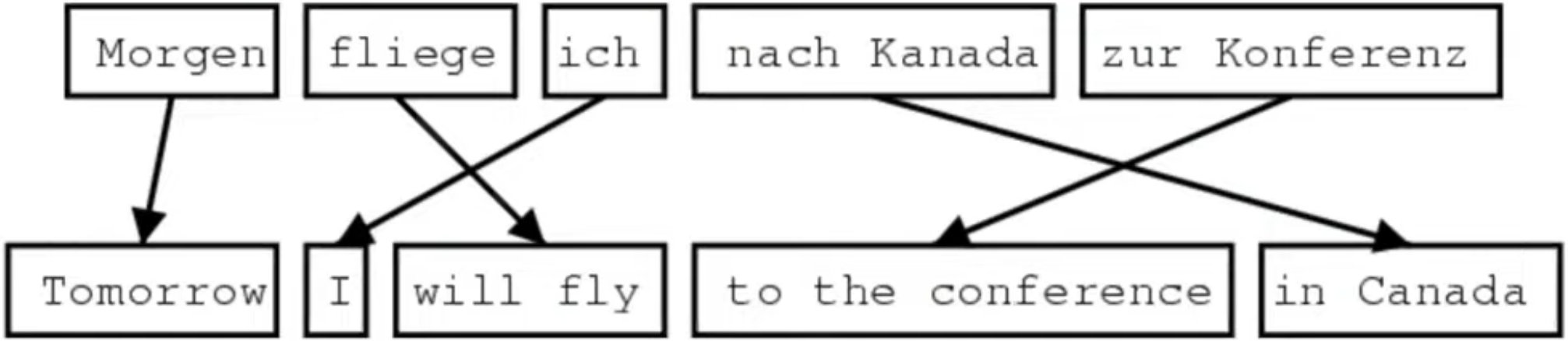
Statistical Transition Model Approach

- 목표는 $argmax_y P(y|x)$를 찾는 것이며, 이는 베이즈 정리에 의해 $argmax_y P(x|y)P(y)$를 찾는 것과 동치다.
- $P(y|x)$는 주어진 영어 문장에 대해 가능한 한국어 문장의 확률분포로, 우리가 목표하는 최종 모델이다.
- $P(x|y)$는 영어문장이 한국어문장으로 어떻게 Translate돼야 하는지에 관한 Translation Model이다.
- $P(y)$는 한 한국어문장이 등장할 확률, 즉 한국어 자체의 Fluency 등을 나타내는 Language Model이다.

Unit 01 | Statistical Translate Model

How to calculate $argmax_y P(x|y)P(y)$?

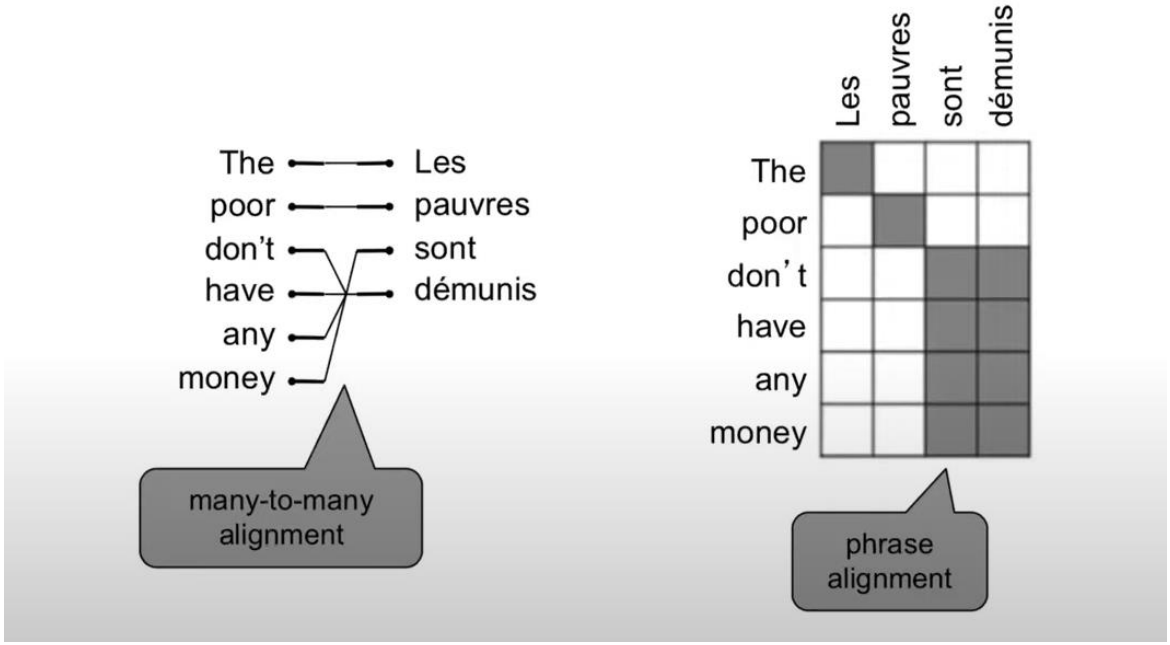
- Naïve Approach: 모든 y를 sentence인 채 나열해서 계산하기 => 계산복잡도가 너무 높다.
- y를 Word로 "Break down to pieces"
- $P(x|y)$ 를 통해 단어 단위로 가능한 모든 번역 조합을 만든 뒤, $P(y)$를 통해 어순에 맞게 나열한다.



Unit 01 | Statistical Translate Model

Statistical Transition Model's Latent Variable

- 각 언어만의 관습, 어순 등이 다르므로 단어 별 번역 정보만으로는 적절한 문장 번역을 얻을 수 없다.
- 따라서 단어들의 종합적인 alignment를 위한 **latent variable a**를 추가한다.
- 갱신된 모델: $argmax_y P(x, a|y)P(y)$



Unit 01 | Statistical Translate Model

Statistical Approach의 한계

- 가능한 조합이 너무 많이 나오기 때문에, 복잡도가 높다.
- 언어마다 고유한 현상이 있어 feature-engineering이 요구되므로 Human Effort가 많이 든다.

  => 2010년대부터 Neural Network based Translation의 부상

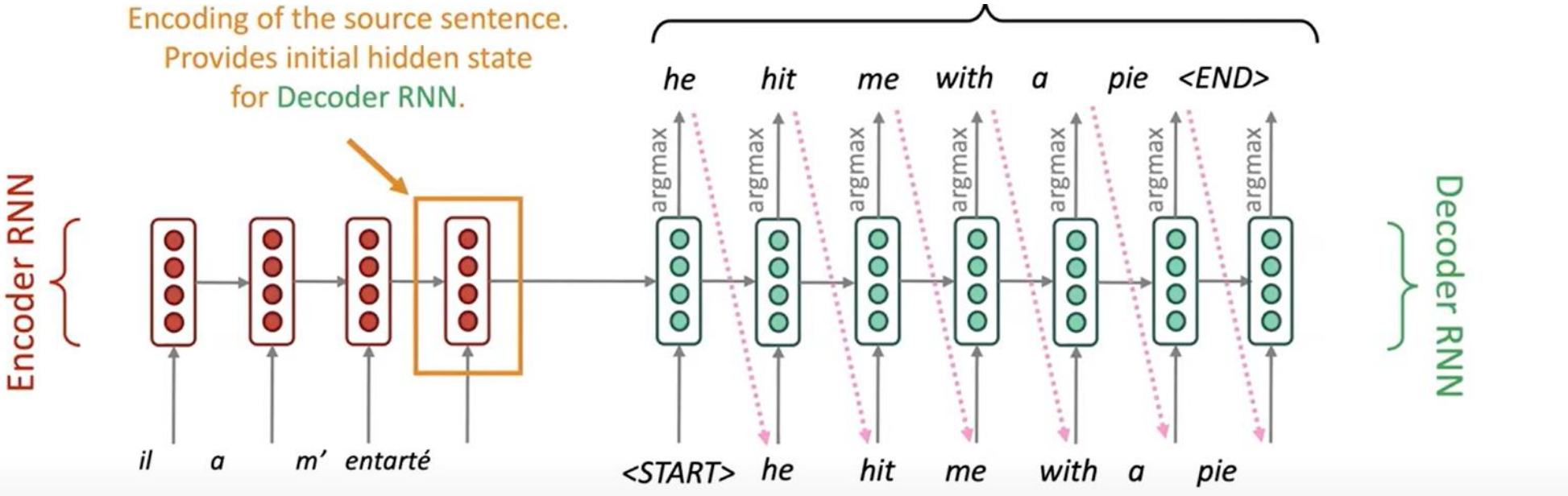
Unit 02

NN based translation: RNN

Unit 02 | NN based translation: RNN

Neural based Approach: Seq2Seq

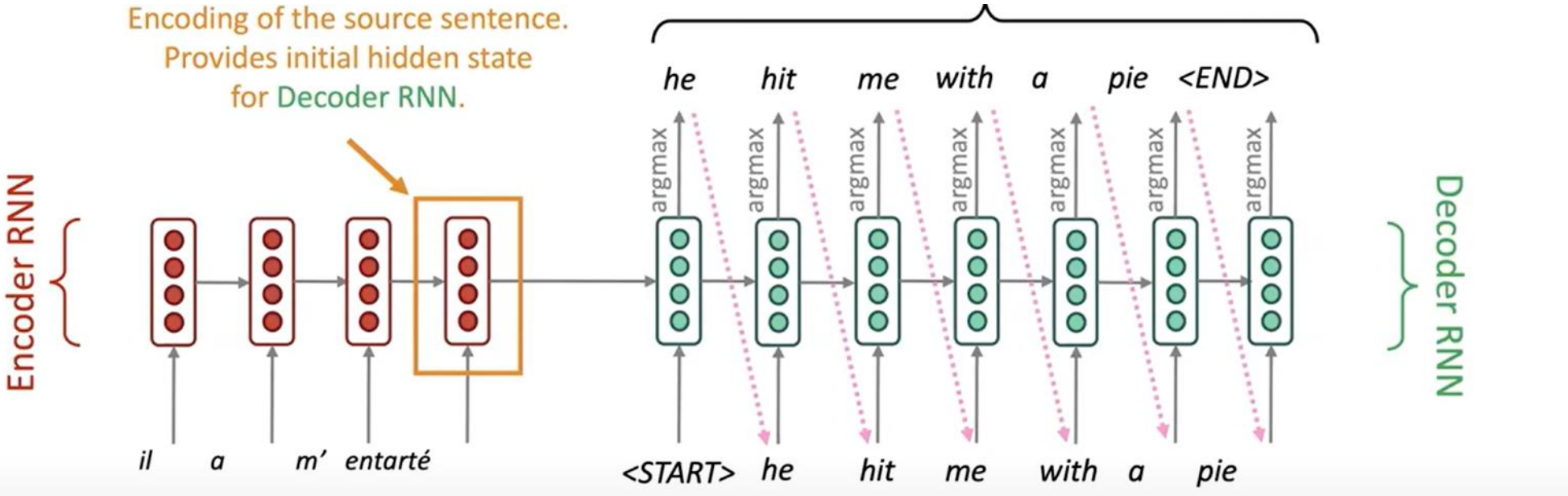
- Seq2Seq: "통째로 Sequence(Sentence)를 넣어, Sequence를 뱉는다" ⇔ Statistical Model(separation)
- RNN은 기본적으로 Seq2Seq 철학 기반이다.



Unit 02 | NN based translation: RNN

RNN(Recurrent Neural Network)의 구조

- Encoder RNN(자연어->hidden vector), Decoder RNN(hidden vector -> 자연어)로 구성
- Encoder RNN에 source(영어 문장)을 넣으면 Decoder RNN이 target sentence(한국어 문장)을 뱉음
- Decoder의 첫 hidden vector: "source sentence x에 대한 의미 요약 벡터"
- Decoder의 hidden vector h_t : 이전 step hidden vector h_{t-1} 과 이전 step 단어 output y_{t-1}을 먹인 결과



Unit 02 | NN based translation: RNN

Conditional Language Model로서의 RNN

- Source sentence x를 given condition으로 갖고, 이전의 word를 바탕으로 다음 word를 예측하는 LM
- Note. Decoder의 hidden vector엔는 x와 이전까지 뱉어진 y_i(단어들)에 대한 정보가 압축돼있다.

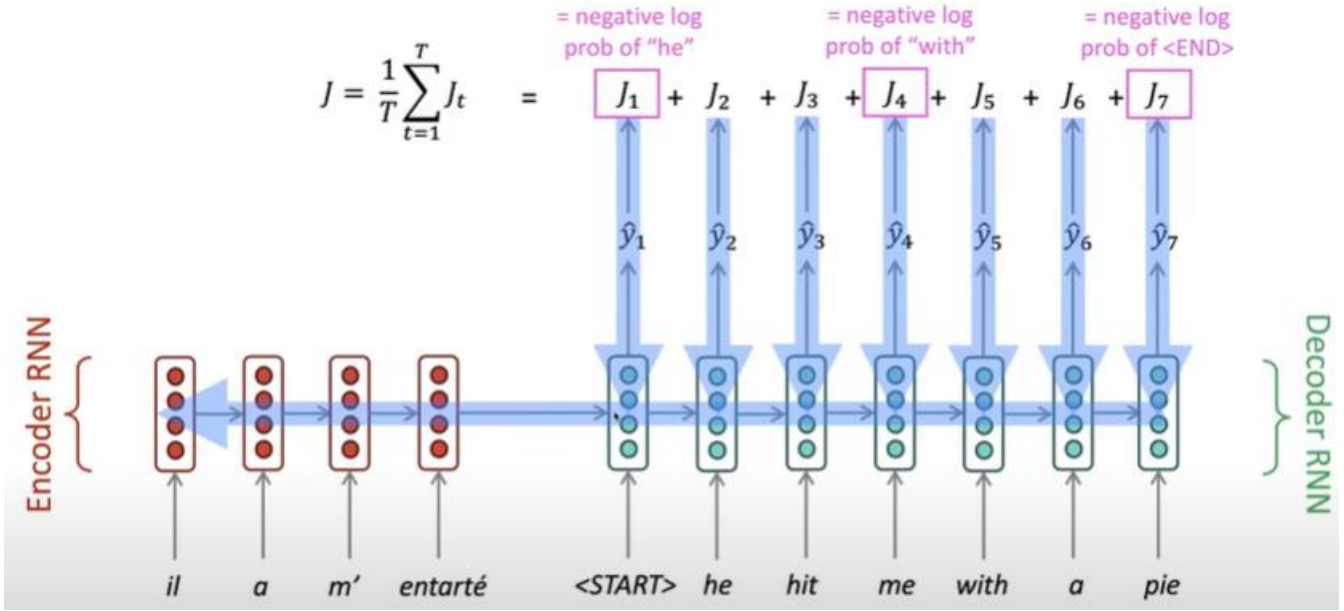
$$P(y|x) = P(y_1|x)\,P(y_2|y_1,x)\,P(y_3|y_1,y_2,x)\ldots\underbrace{P(y_T|y_1,\ldots,y_{T-1},x)}$$

Probability of next target word, given target words so far and source sentence x

Unit 02 | NN based translation: RNN

How to train RNN?(손실함수, Backpropagation)

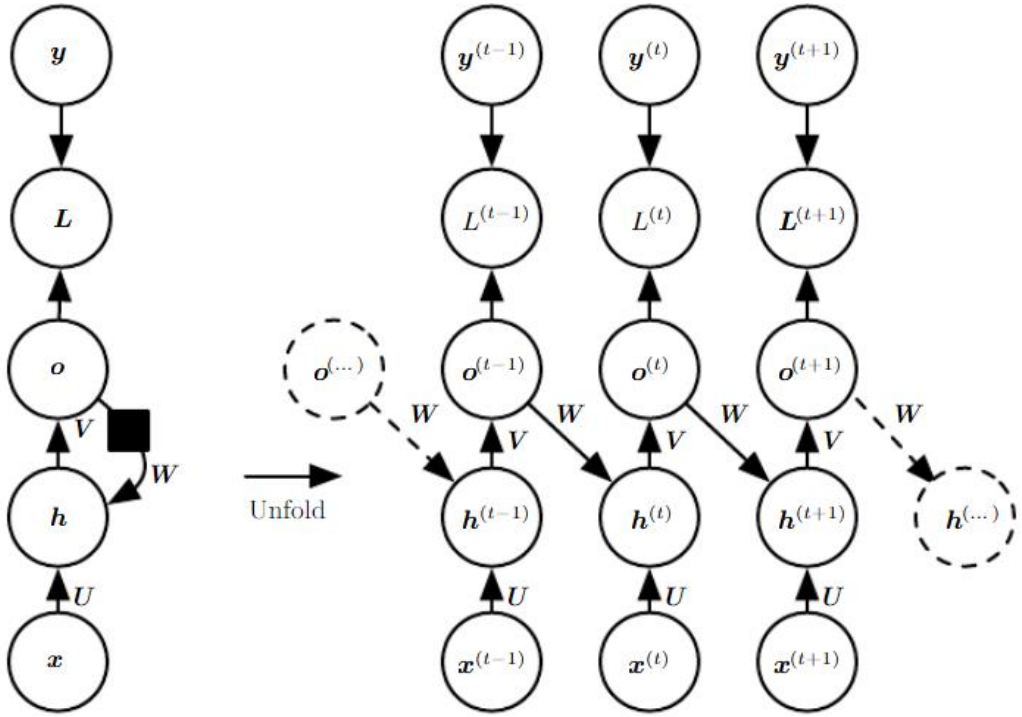
- Decoder는 time step t에서 초기에 설정된 임의의 가중치로 나름의 한국어 단어 output o_t를 뱉어본다.
- Ground truth y_t가 "고양이"인데 o_t가 "개 " 라면, 이에 대한 J_t: Negative Log likelihood error를 계산한다.
- 전체 손실함수 J는 모든 time step에 대한 J_t들의 산술평균이다.
- Block Diagram에서 나타나듯 미분 가능하지 않은 과정이 없으므로, Backpropagation이 가능하다.



Unit 02 | NN based translation: RNN

19기 자연어 심화세미나
TOBIG'S 19기 한진모

Furthermore. Backpropagation rule(Retrieved from Bengio, DeepLearning)



$$\frac{\partial L}{\partial L^{(t)}} = 1.$$

$$\nabla_{\boldsymbol{h}^{(\tau)}} L = \boldsymbol{V}^\top \nabla_{\boldsymbol{o}^{(\tau)}} L.$$

$$(\nabla_{\boldsymbol{o}^{(t)}} L)_i = \frac{\partial L}{\partial o_i^{(t)}} = \frac{\partial L}{\partial L^{(t)}} \frac{\partial L^{(t)}}{\partial o_i^{(t)}} = \hat{y}_i^{(t)} - \mathbf{1}_{i=y^{(t)}}.$$

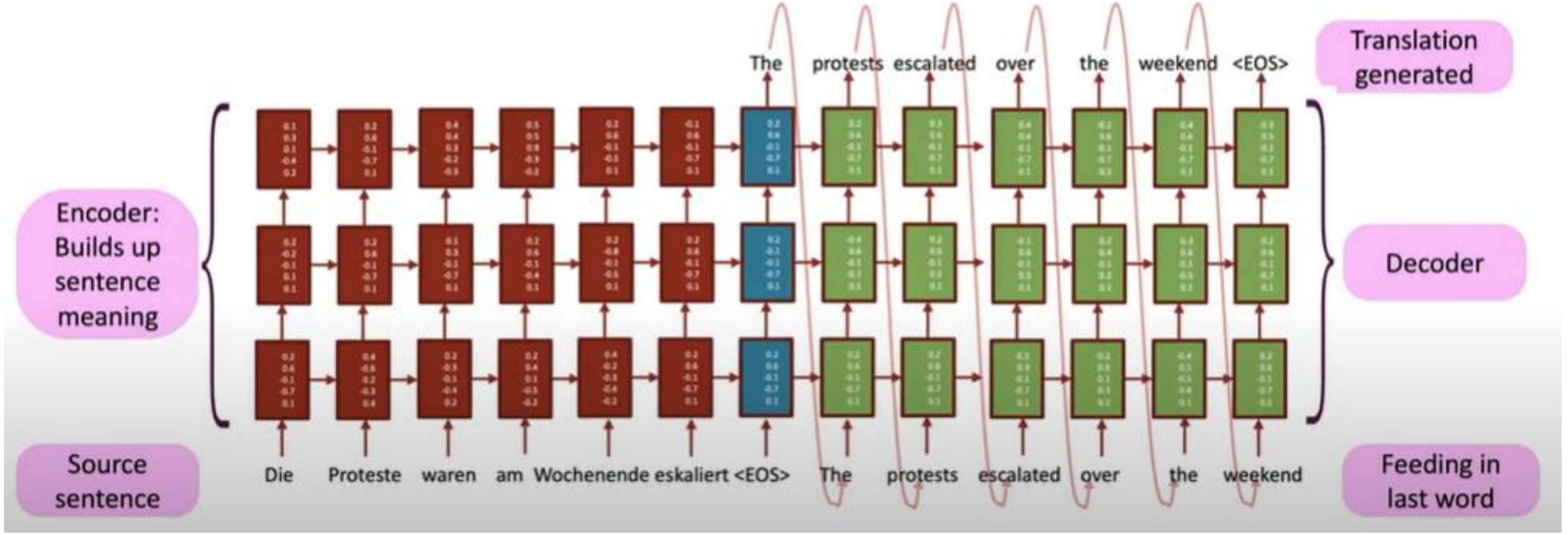
$$\begin{aligned}\nabla_{\boldsymbol{h}^{(t)}} L &= \left(\frac{\partial \boldsymbol{h}^{(t+1)}}{\partial \boldsymbol{h}^{(t)}}\right)^\top (\nabla_{\boldsymbol{h}^{(t+1)}} L) + \left(\frac{\partial \boldsymbol{o}^{(t)}}{\partial \boldsymbol{h}^{(t)}}\right)^\top (\nabla_{\boldsymbol{o}^{(t)}} L) \\ &= \boldsymbol{W}^\top \operatorname{diag}\left(1 - \left(\boldsymbol{h}^{(t+1)}\right)^2\right) (\nabla_{\boldsymbol{h}^{(t+1)}} L) + \boldsymbol{V}^\top (\nabla_{\boldsymbol{o}^{(t)}} L)\end{aligned}$$

$$\begin{aligned}
\nabla_{\boldsymbol{c}} L &= \sum_t \left(\frac{\partial \boldsymbol{o}^{(t)}}{\partial \boldsymbol{c}}\right)^\top \nabla_{\boldsymbol{o}^{(t)}} L = \sum_t \nabla_{\boldsymbol{o}^{(t)}} L, \\
\nabla_{\boldsymbol{b}} L &= \sum_t \left(\frac{\partial \boldsymbol{h}^{(t)}}{\partial \boldsymbol{b}^{(t)}}\right)^\top \nabla_{\boldsymbol{h}^{(t)}} L = \sum_t \operatorname{diag}\left(1 - \left(\boldsymbol{h}^{(t)}\right)^2\right) \nabla_{\boldsymbol{h}^{(t)}} L, \\
\nabla_{\boldsymbol{V}} L &= \sum_t \sum_i \left(\frac{\partial L}{\partial o_i^{(t)}}\right) \nabla_{\boldsymbol{V}^{(t)}} o_i^{(t)} = \sum_t (\nabla_{\boldsymbol{o}^{(t)}} L)\, \boldsymbol{h}^{(t)\top}, \\
\nabla_{\boldsymbol{W}} L &= \sum_t \sum_i \left(\frac{\partial L}{\partial h_i^{(t)}}\right) \nabla_{\boldsymbol{W}^{(t)}} h_i^{(t)} \\
&= \sum_t \operatorname{diag}\left(1 - \left(\boldsymbol{h}^{(t)}\right)^2\right) (\nabla_{\boldsymbol{h}^{(t)}} L)\, \boldsymbol{h}^{(t-1)\top}, \\
\nabla_{\boldsymbol{U}} L &= \sum_t \sum_i \left(\frac{\partial L}{\partial h_i^{(t)}}\right) \nabla_{\boldsymbol{U}^{(t)}} h_i^{(t)} \\
&= \sum_t \operatorname{diag}\left(1 - \left(\boldsymbol{h}^{(t)}\right)^2\right) (\nabla_{\boldsymbol{h}^{(t)}} L)\, \boldsymbol{x}^{(t)\top},
\end{aligned}$$

Unit 02 | NN based translation: RNN

Multi-layer RNN

- Single RNN layer를 수직 방향으로 여러 개 쌓아 Multi-layer RNN을 만든다.
- 2000개의 hidden vector node를 수평으로 두는 것보다, 500개씩 4 layer로 쌓는 게 낫다.
- Lower layer: Lower-level features(straightforward) ⇔ Higher layer: Higher-level features(overall structure)
- Layer를 너무 많이 쌓으면 Gradient vanish/explode가 생길 수 있으므로 skip/dense connection을 둔다.



Unit 02 | NN based translation: RNN

19기 자연어 심화세미나
TOBIG'S 19기 한진모

Revision of Greedy Search(1)

- Greedy search: Decoder의 각(국소적인) time step에서 최적인 단어를 골라서 뱉음
- 따라서 전체적 맥락에서 더 좋은 번역을 찾지 못할 수 있음, undo-decision 불가
- Sol: Beam Search; 각 time step마다 **k개의 가설을 explore**하여, '종합적으로' 더 나은 번역을 따라가자.

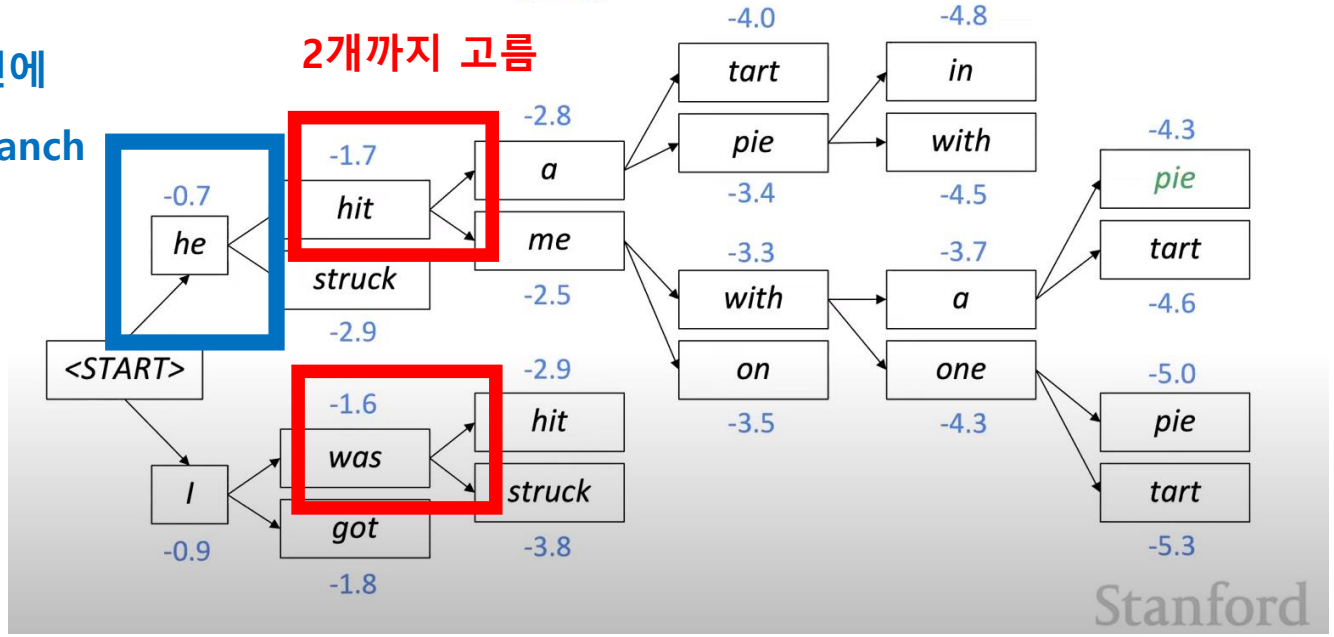
$$\text{score}(y_1, \ldots, y_t) = \sum_{i=1}^{t} \log P_{\text{LM}}(y_i | y_1, \ldots, y_{i-1}, x)$$

한 계층에서

k=2

한 번에 2개 Branch

2개까지 고름



Unit 02 | NN based translation: RNN

Revision of Greedy Search(2)

- Greedy search의 Stopping criterion(탐색 종료 조건): decode가 <END> 토큰을 뱉자마자.
- Beam Search Decoding: <END> 토큰이 최고점수여도 그 다음 k-1개 토큰을 찾으므로 바로 종료하지 않음.
- 따라서 임의로 탐색을 종료하는 time step T를 설정하거나, branch의 수 N을 설정함.

Revision of Beam Search

- Hypothesis가 길어질수록 Score가 낮아진다. 확률은 곱할수록 감소하기 때문이다.
- 따라서 길이가 짧은 문장이 뱉어질 가능성이 높다.
- 이를 방지하기 위해 Score 계산 시 문장의 길이(Word의 개수) T로 **원래의 Score 공식을 Normalize한다.**

$$\frac{1}{t}\sum_{i=1}^{t}\log P_{\mathrm{LM}}(y_i|y_1,\ldots,y_{i-1},x)$$

Unit 02 | NN based translation: RNN

How to Evaluate?

- Ideal method: 영어와 한국어를 모두 아는 사람에게 번역 점수를 평가하도록 함. 그러나 이는 비쌈
- Automatic method: BLEU(**Bi**L**ingual** **E**valuation **U**nderstudy) 이용
- BLEU: 인간 번역과 기계 번역 문장 사이 몇 단어가 겹쳤는지(n-gram precisions) 기준으로 점수를 매김
- BLEU는 유용하지만, 문장을 번역할 다양한 방법이 있으므로 "단어 " 만 기준으로 평가하는 것은 불완전함

Unit 02 | NN based translation: RNN

NN based Translation: Pros

- More fluent
- Better use of context: hidden states를 통해 x에 대한 conditionin을 하므로
- 단일 시스템으로서 Neural Network 하나만 최적화하면 됨: 구성요소별로 별개로 최적화하지 않음
- Much less human engineering: feature engineering이 필요하지 않음

NN based Translation: Cons

- Less interpretable: hard to debug
- Difficult to control: safety concerns

Unit 02 | NN based translation: RNN

Needed to be resolved

- Out-of-vocabulary words problem
- Domain mismatch(ex. Facebook-news)
- Low-resource language pairs(훈련을 위한 데이터가 부족함)
- Pronoun resolution errors(대명사가 무엇을 의미하는지?)
- 과거 데이터에 의존한 사회적 편견 (ex. It's nurse => 그녀는 간호사다. It's police => 그는 경찰이다.)

Unit 03

Attention

Unit 03 | Attention

Limitation of Seq2Seq Architecture

- Encoder의 마지막 hidden vector에 모든 정보를 집어넣어야 => information bottleneck
- 따라서 Order of word와 같은 정보가 소실될 수 있음

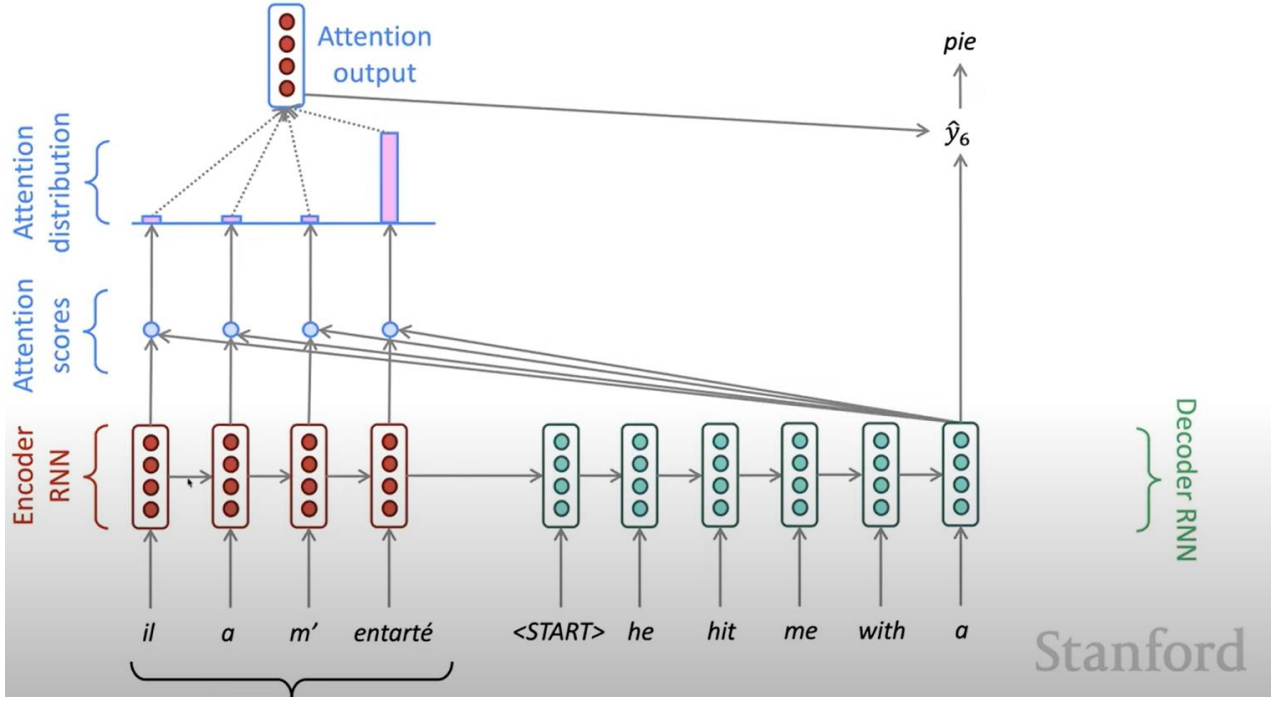
Idea of Attention

- Translation을 하면서, 실시간으로 source sentence의 특정 부분에 주목하고 싶다.
- 각 Time step마다 Decoder에서 Encoder의 특정 부분으로 가는 Direct Connection을 만들자.

Unit 03 | Attention

Brief on Attention-Adapted Architecture Architecture

- RNN의 Seq2Seq 구조를 기반으로 설명됨(Decoder에서 Encoder로의 Connection)
- Decoder의 각 hidden vector는 Encoder의 모든 hidden vector에 대해 Attention(내적 기반 연산)을 수행
- 이렇게 계산된 Attention Score를 바탕으로 Attention Output을 산출(Weighted sum related to Encoder)
- 이 Attention Output이 Decoder의 Output 생성에 추가로 관여함 => Encoder의 특정 부분 정보 반영됨



Reference

Stanford CS224N NLP with Deep Learning | Winter 2021 | Lecture 7 - Translation, Seq2Seq, Attention

: https://www.youtube.com/watch?v=wzfWHP6SXxY&list=PLoROMvodv4rOSH4v6133s9LFPRHjEmbmJ&index=7

***All Images without clarified source are retrieved on the above reference.**

감사합니다