KOREA UNIVERSITY

DSBA CS224n 2021 Study

[Lecture 07]
Machine Translation, Sequence-to-Sequence and Attention

DSBA
Data Science & Business Analytics

고려대학교 산업경영공학과

Data Science & Business Analytics Lab

발표자 : 정용기

00

발표 목차

Contents

1. Machine Translation
2. Sequence-to-Sequence
3. Attention

"Machine Translation, which is a major use-case of sequence-to-sequence, which is improved by attention."

01 Machine Translation

Pre-Neural Machine Translation

Machine Translation(기계번역)

x: *L'homme est né libre, et partout il est dans les fers*

French to English

y: *Man is born free, but everywhere he is in chains*

1950s
The early history of Machine Translation

1990s-2010s
Statistical Machine Translation

2014
Neural Machine Translation

01 Machine Translation

Pre-Neural Machine Translation

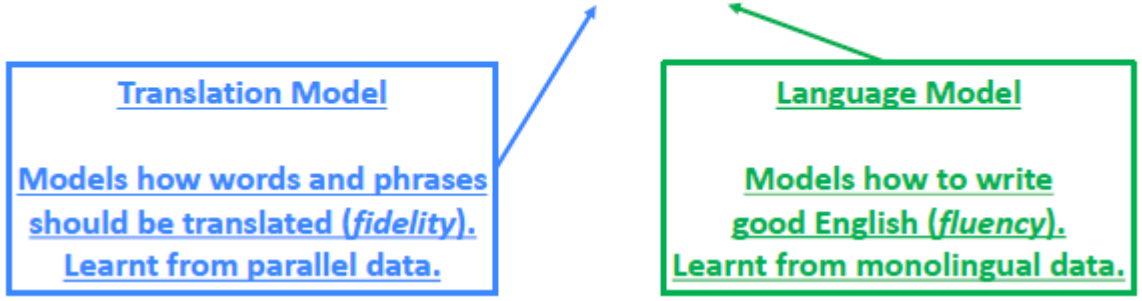
1950s : The early history of MT

- 군사 목적으로 개발되기 시작함(ex. Russian → English)
- 같은 뜻의 단어를 대체하는 단순한 방식을 사용

1990s-2010s : Statistical Machine Translation

- 확률 모델을 활용하기 시작함
- Translation Model + Language Model (Bayes Rule)
- Ex. 입력 문장 x 에 대해 조건부확률을 최대화 하는 번역 문장 y를 찾고자 함

$$\text{argmax}_y P(y|x) = \text{argmax}_y \underbrace{P(x|y)}\underbrace{P(y)}$$

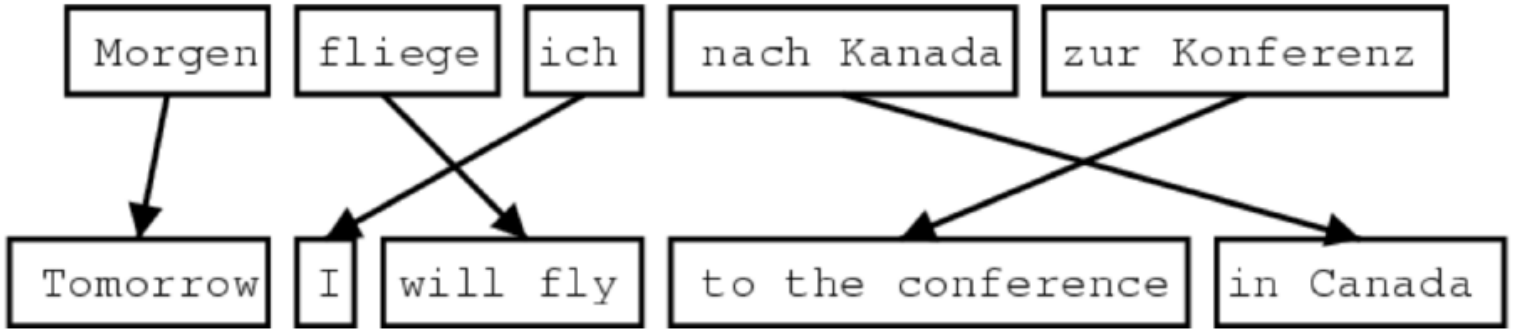


01 Machine Translation

Pre-Neural Machine Translation

Statistical Machine Translation (SMT)

- 많은 양의 parallel data 가 필요함
- Source-target sentence 간의 correspondence 를 매핑하는 alignment 사용



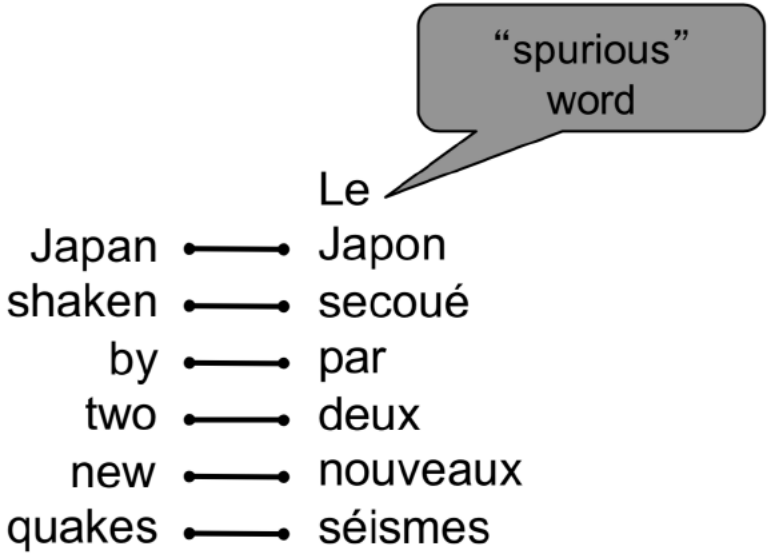
- But.. alignment 를 1:1 로 명확히 정의할 수 없는 경우가 많음 → EM algorithm 같은 학습 방법이 사용됨
  - No counterpart
  - Many-to-one
  - One-to-many
  - Many-to-many

01 Machine Translation

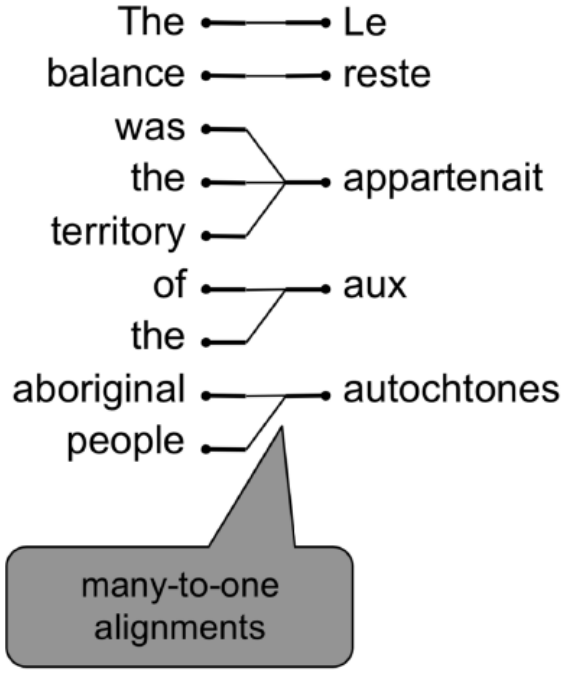
Pre-Neural Machine Translation

Statistical Machine Translation (SMT)

- No counterpart



- Many to one

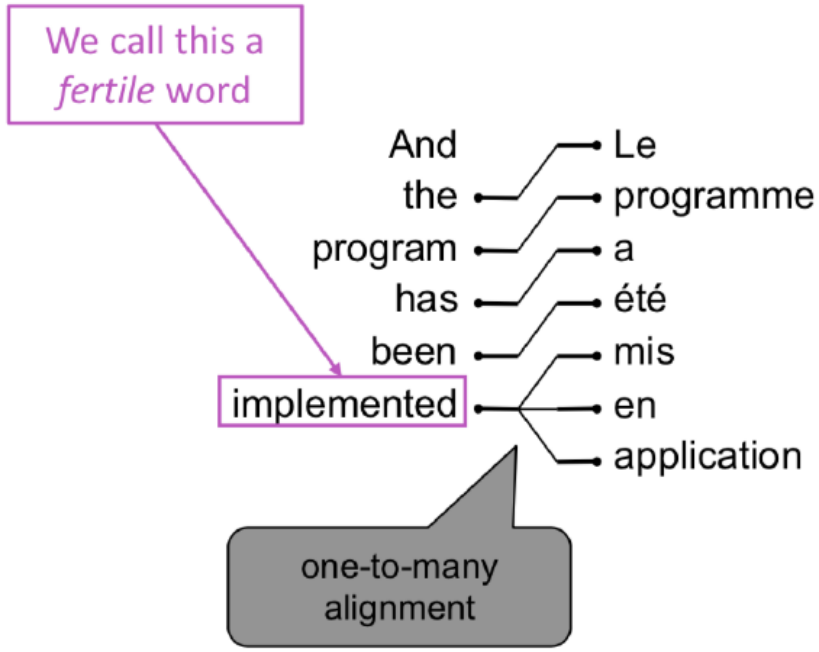


01 Machine Translation

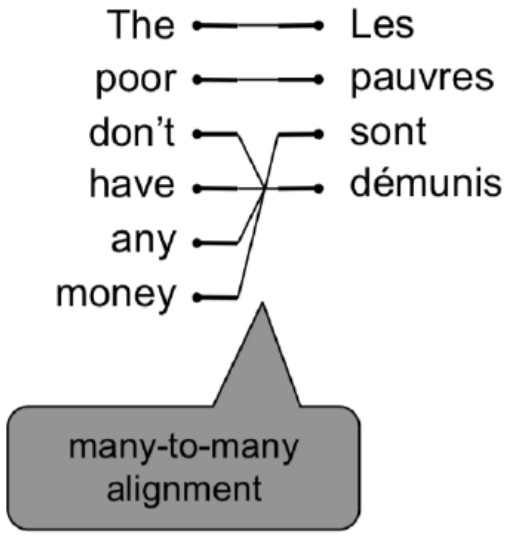
Pre-Neural Machine Translation

Statistical Machine Translation (SMT)

- One to many



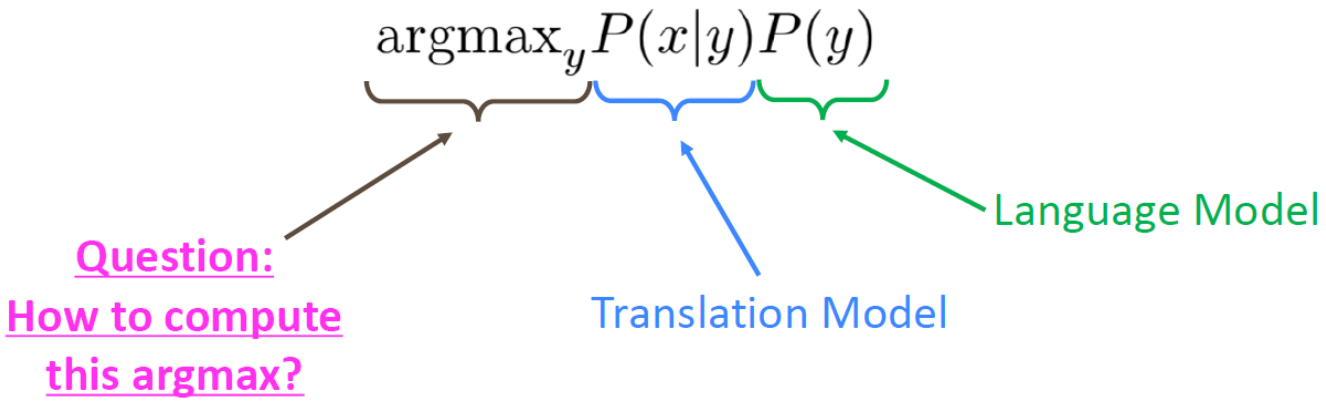
- Many to many



01 Machine Translation

Pre-Neural Machine Translation

Decoding for SMT



- 가능한 모든 y를 비교하는 방법은 너무 계산량이 많음
- 여러 factor 로 분리하고 각각의 최적해 조합으로 global optimal(가장 적절한 번역문) 을 찾는 Dynamic programing 방식으로 decoding 함

01 Machine Translation

Neural Machine Translation

Statistical Machine Translation (SMT)

- 많은 하위 시스템으로 전체 시스템을 설계해야함
- Feature engineering 이 필요함
- 성능을 위해 추가 resource 가 필요 (ex. phrase table)
- 사람의 노력이 많이 필요함 (paired dataset, subcomponent 의 학습 등)

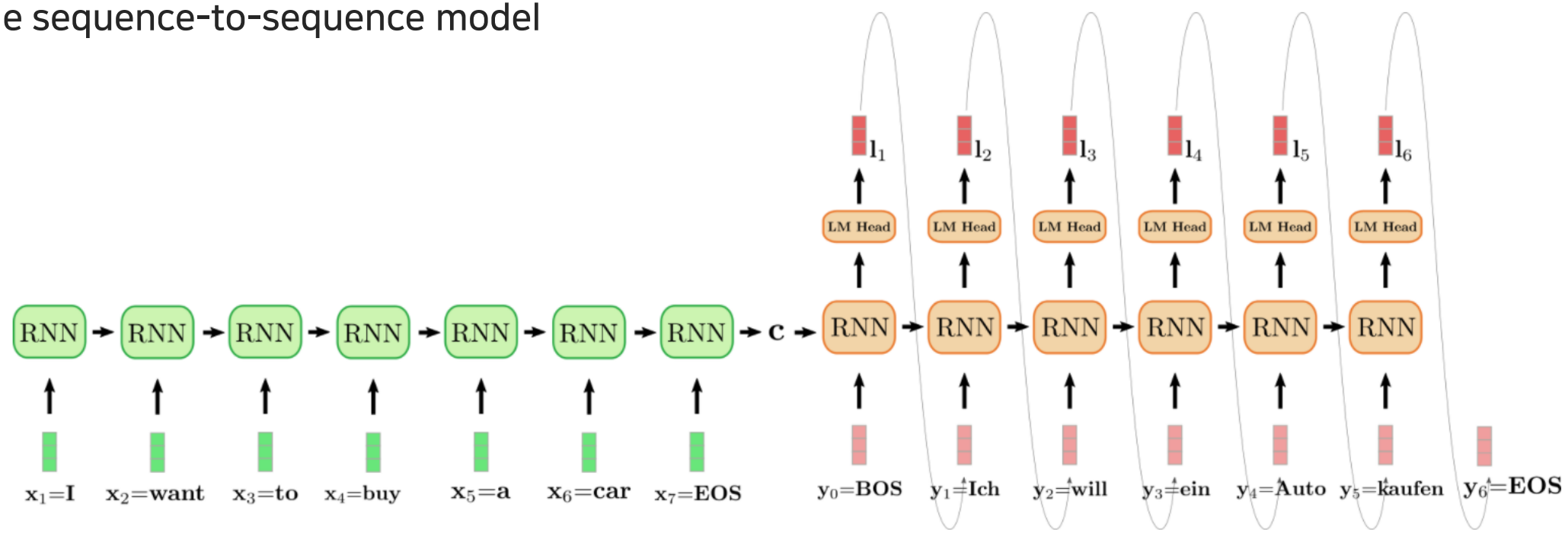
Neural Machine Translation (NMT, 2014~)

- Single end-to-end neural network → Subcomponents 별로 따로 최적화할 필요가 없음
- 2개의 RNN 으로 구성된 sequence-to-sequence model
- SMT 보다 뛰어난 성능
- Human resource 가 줄어듦 (No feature engineering, 모든 source-target pair 에 대해 동일한 method 사용 가능)
- Interpretability 가 부족
- Rule, guideline 을 적용하기 쉽지 않음

02 Sequence-to-sequence

Neural Machine Translation

The sequence-to-sequence model



https://colab.research.google.com/github/patrickvonplaten/notebooks/blob/master/Encoder_Decoder_Model.ipynb

- Encoder RNN 은 source sentence 를 encoding 함
- Decoder RNN 은 encoding 을 condition 으로 target sentence 를 생성하는 language model
  - Source sentence 의 마지막 RNN cell output 은 Decoder RNN 의 초기 hidden state 로 사용됨
  - Train : Target sentence 의 timestep 별 word 가 다음 timestep 의 word 를 예측하기 위한 입력으로 사용됨
  - Test : Decoder RNN 의 timestep 별 output 이 다음 timestep 의 입력으로 사용됨

02 Sequence-to-sequence

Neural Machine Translation

The sequence-to-sequence model

Machine Translation, which is a major use-case of sequence-to-sequence …

- MT ∈ Seq2seq (summarization, dialogue, parsing, code generation, time series, voice generation etc.)
- NMT 는 conditional probability 를 직접 계산 (w/o Bayes Rule)

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

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

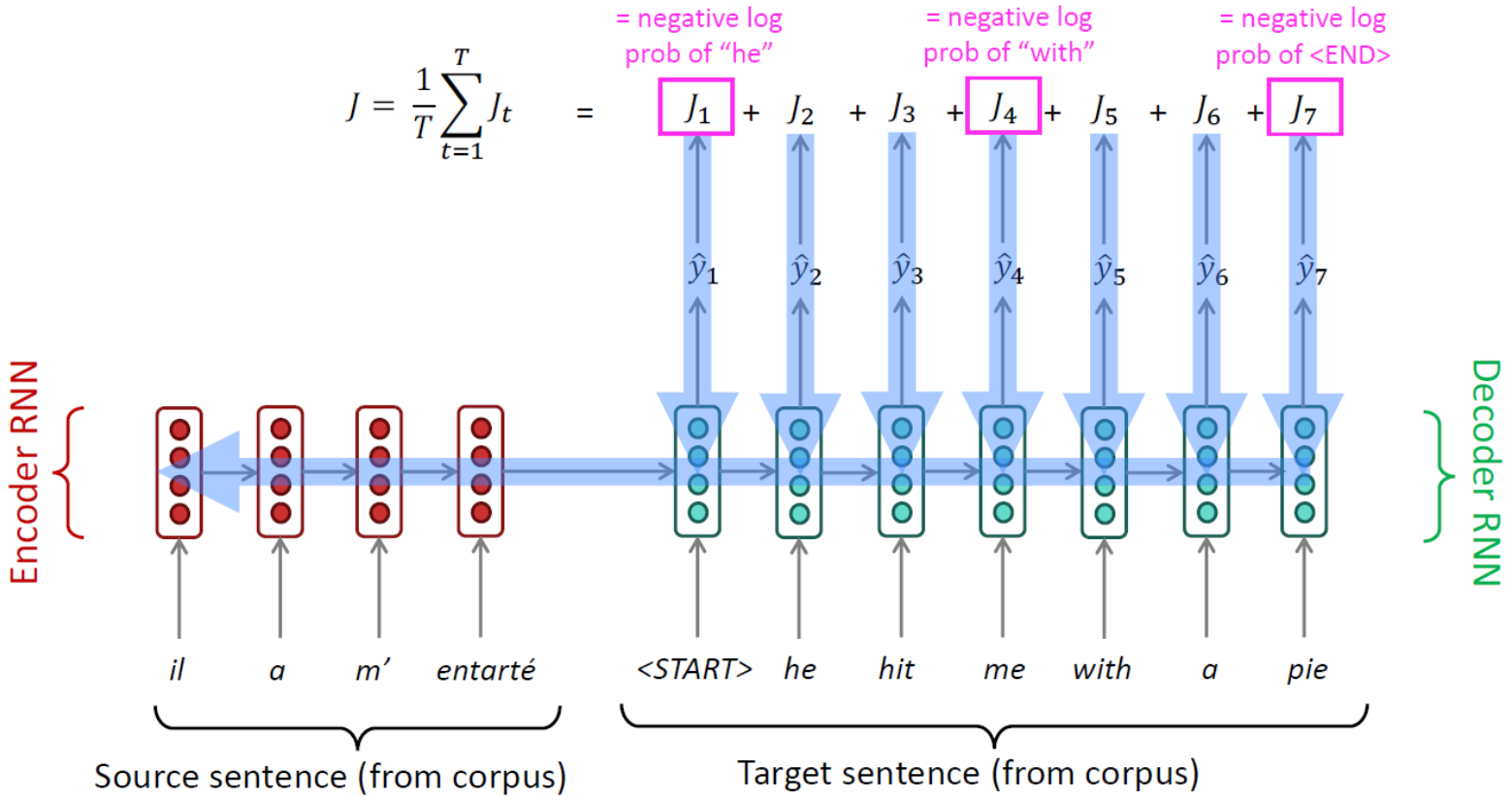
- End-to-end 학습이 가능
- But, 여전히 parallel corpus 가 필요

02 Sequence-to-sequence

Neural Machine Translation

Training a Neural Machine Translation system

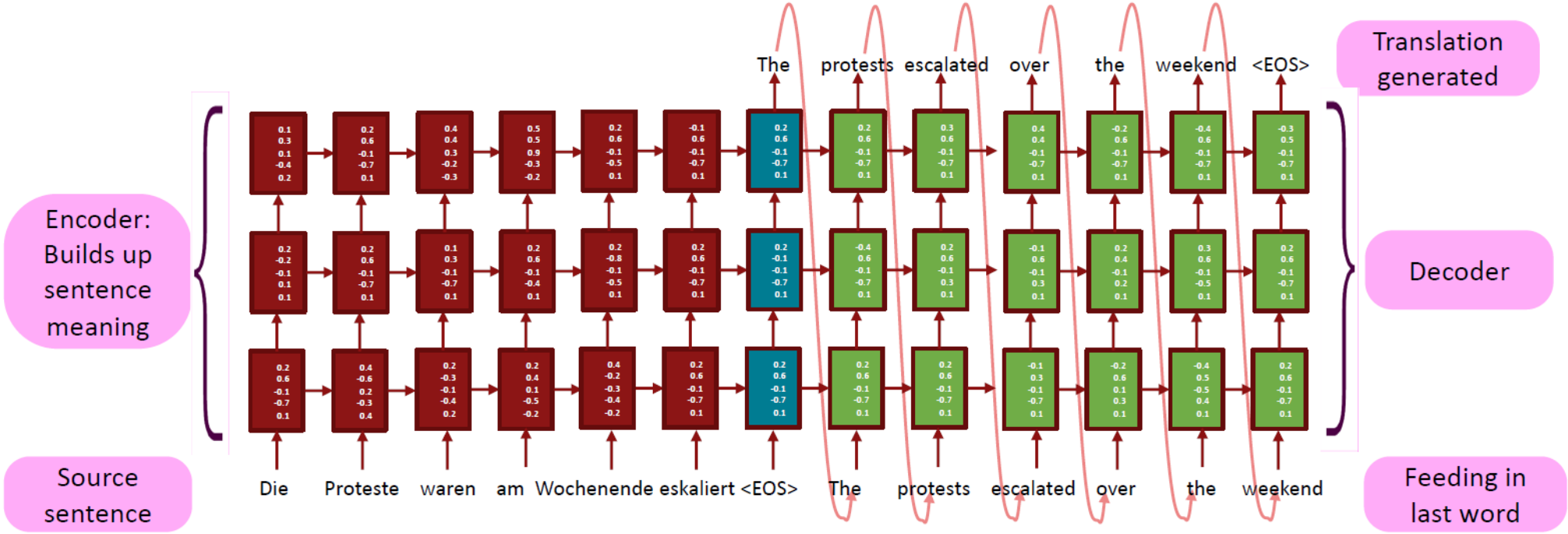
- NLL loss : $\boldsymbol{L(y) = -\frac{1}{K}\sum_{k=1}^{K} \log(y)}$, probability 를 $[0, \infty]$의 범위로 매핑하여 loss 최소화 문제로 정의함



02 Sequence-to-sequence

Neural Machine Translation

Multi-layer RNNs

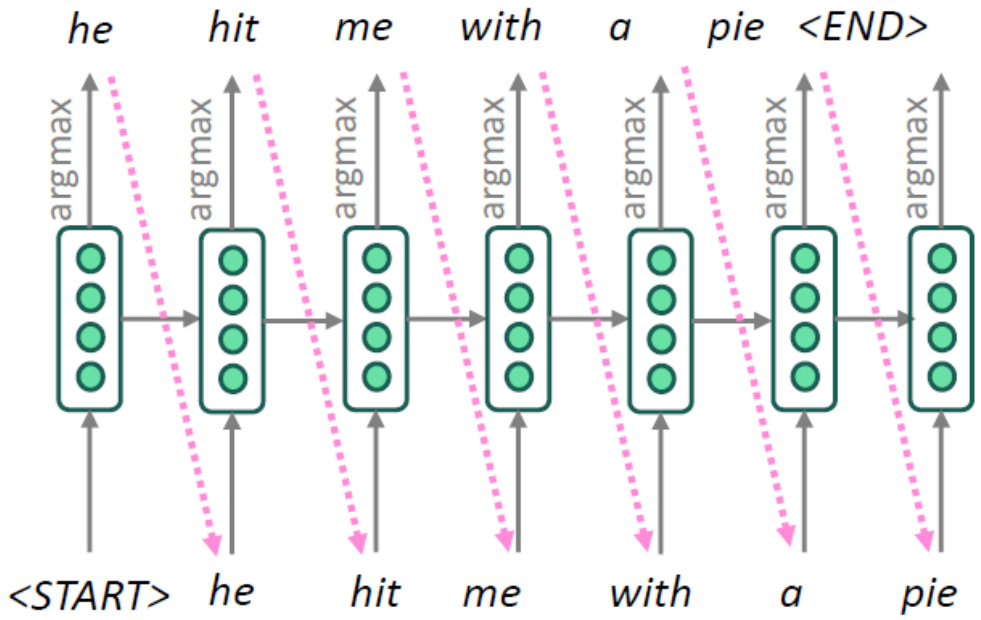


- Timestep 의 진행과 평행하게 layer 를 추가하여 좀 더 higher-level feature 를 학습할 수 있게 함
- 좋은 성능을 내는 RNN 모델은 대부분 Multi-layer RNNs 구조를 사용
  - Encoder with 2-4 layers, Decoder with 4 layers
  - Transformer-based : 12 or 24 layers

02 Sequence-to-sequence

Neural Machine Translation

Greedy decoding



- Input: *il a m'entarté* *(he hit me with a pie)*
- → *he* ____
- → *he hit* ____
- → *he hit a* ____ *(whoops! no going back now…)*

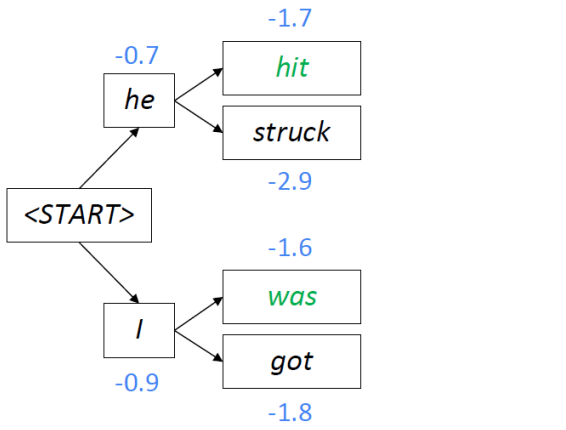
- Exhaustive search 방식보다 효율적
- Step 별 argmax word 를 선택하는 방식
- 부자연 스러운 문장이 생성될 수 있음 (최적해를 보장하지 않음)

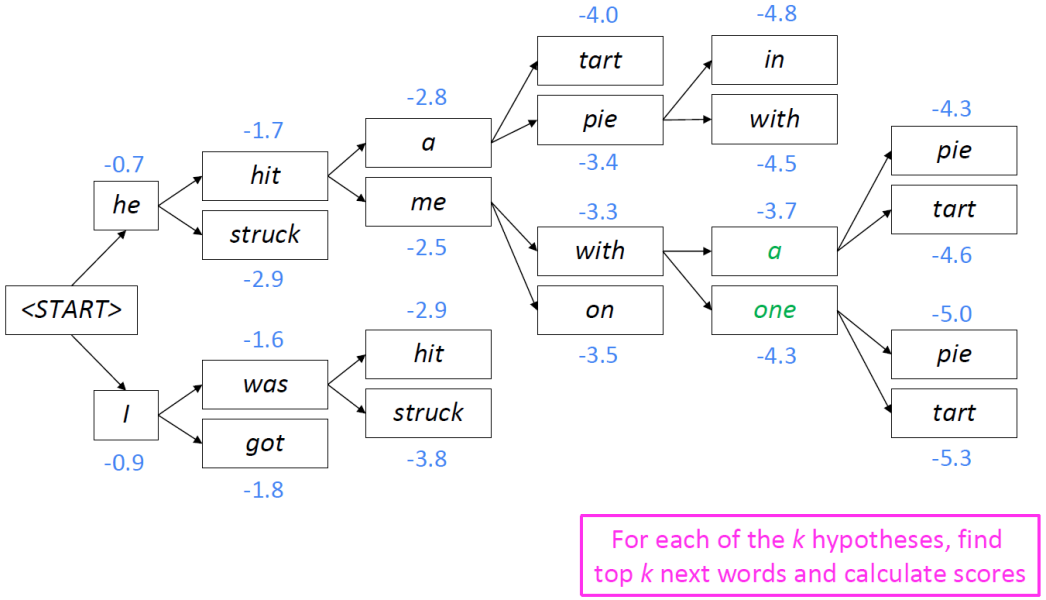
02 Sequence-to-sequence

Neural Machine Translation

Beam search decoding

- k 개의 best translation 을 유지하며 decoding 하는 방식 (k : beam size, usually 5-10)
- 최적해를 보장하진 않지만 exhaustive search 보다 효율적이고 greedy decoding 보다 더 자연스러운 문장을 생성함
- k^2의 hypotheses 를 비교하고 k 개의 best hypotheses 를 선택하는 과정을 반복
- EOS 토큰의 등장이나 사전 정의된 max timestep T, 또는 n 개의 완성된 문장을 종료조건으로 함
- NLL loss 합을 생성된 문장 길이로 나눠서 normalize 함





02 Sequence-to-sequence

Neural Machine Translation

Evaluation for Machine Translation : BLEU

- Target sentence 와 machine-written translation 을 비교
- N-gram precision 기반 similarity score + penalty for too short translation

- `예측된 sentence` : **빛이 쐬는** 노인은 **완벽한** 어두운곳에서 **잠든 사람과 비교할 때** 강박증이 **심해질** 기회가 **훨씬 높았다**
- `true sentence` : **빛이 쐬는** 사람은 **완벽한** 어둠에서 **잠든 사람과 비교할 때** 우울증이 **심해질** 가능성이 **훨씬 높았다**

$$BLEU = min(1, \frac{output\ length(\text{예측 문장})}{reference\ length(\text{실제 문장})})(\prod_{i=1}^{4} precision_i)^{\frac{1}{4}}$$

짧은 문장에 대한 penalty

N-gram precision 의 기하평균

https://donghwa-kim.github.io/BLEU.html

- 1-gram precision: $\frac{\text{일치하는}1-gram\text{의 수(예측된 }sentence\text{중에서)}}{\text{모든}1-gram\text{쌍 (예측된 }sentence\text{중에서)}} = \frac{10}{14}$
- 2-gram precision: $\frac{\text{일치하는}2-gram\text{의 수(예측된 }sentence\text{중에서)}}{\text{모든}2-gram\text{쌍 (예측된 }sentence\text{중에서)}} = \frac{5}{13}$
- 3-gram precision: $\frac{\text{일치하는}3-gram\text{의 수(예측된 }sentence\text{중에서)}}{\text{모든}3-gram\text{쌍 (예측된 }sentence\text{중에서)}} = \frac{2}{12}$
- 4-gram precision: $\frac{\text{일치하는}4-gram\text{의 수(예측된 }sentence\text{중에서)}}{\text{모든}4-gram\text{쌍 (예측된 }sentence\text{중에서)}} = \frac{1}{11}$

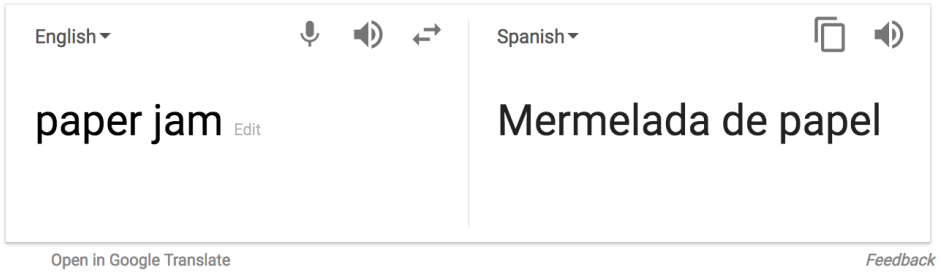
$$(\prod_{i=1}^{4} precision_i)^{\frac{1}{4}} = (\frac{10}{14} \times \frac{5}{13} \times \frac{2}{12} \times \frac{1}{11})^{\frac{1}{4}}$$

02 Sequence-to-sequence

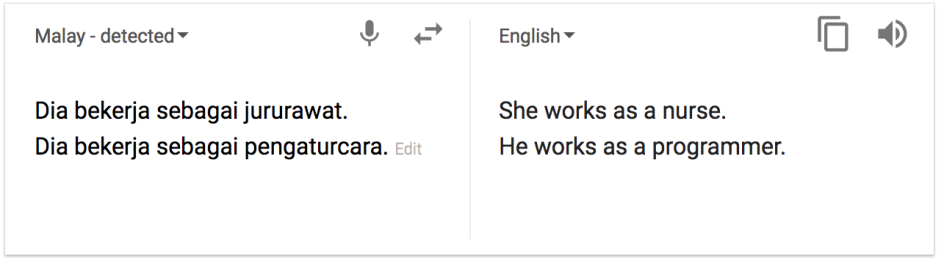
Neural Machine Translation

Remained difficulties

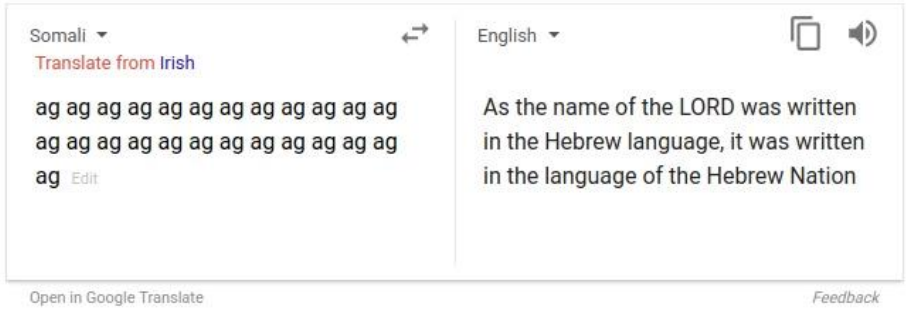
- 관용표현



- Bias 학습



- Uninterpretable

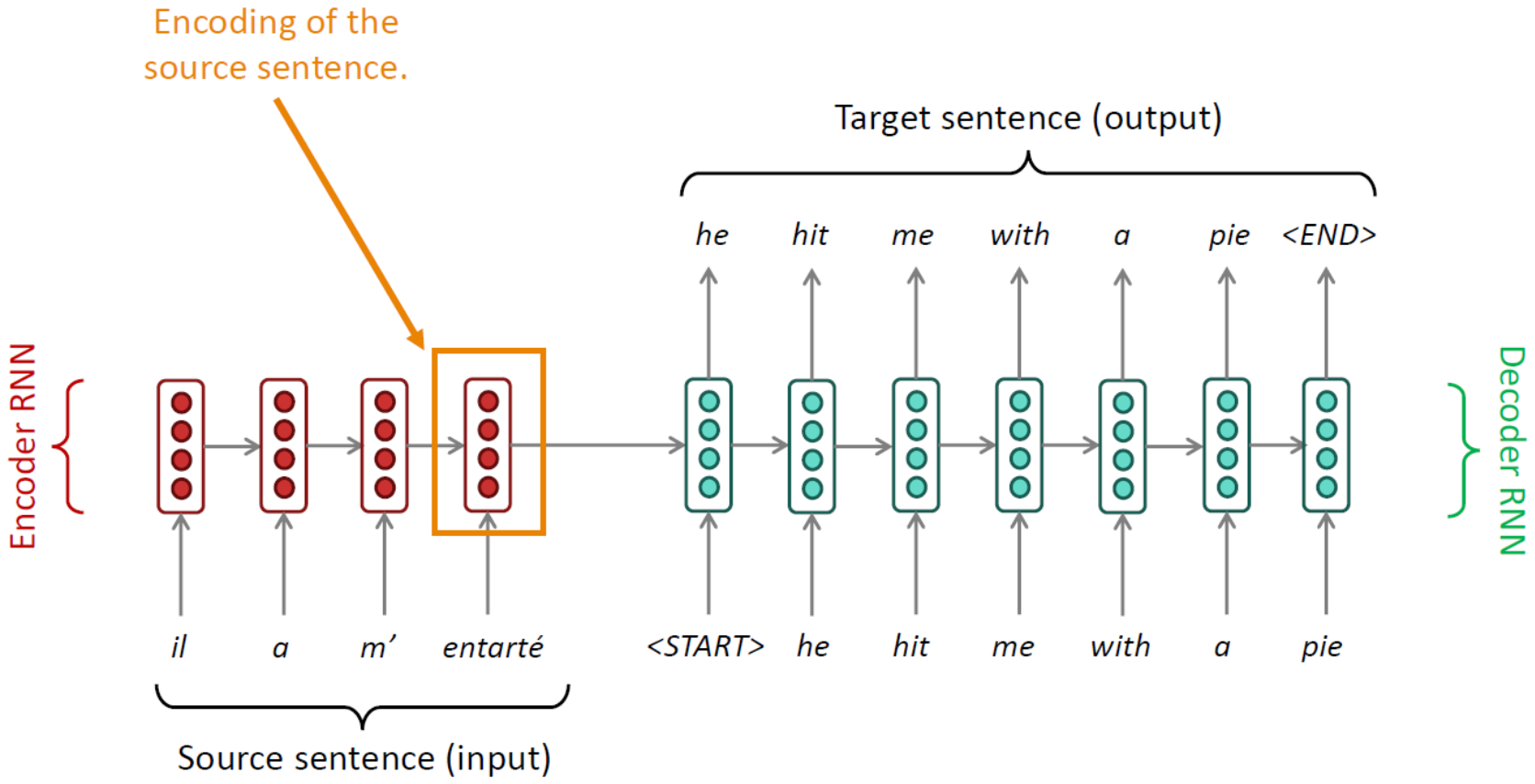


03 Attention

Sequence-to-Sequence with Attention

The bottleneck problem

- Source sentence 의 마지막 encoding 만을 입력으로 사용

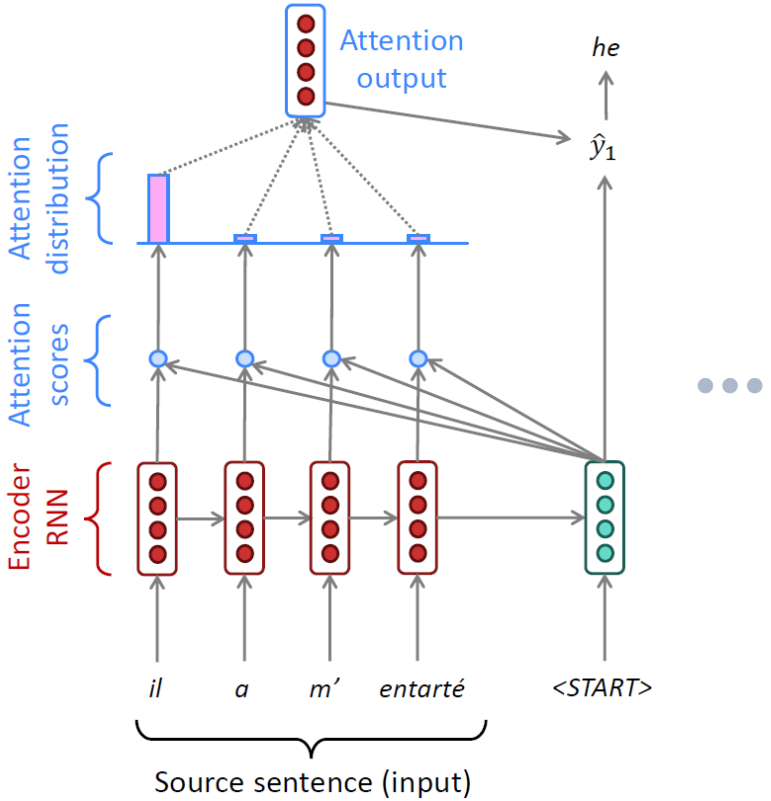


03 Attention

Sequence-to-Sequence with Attention

Attention

- Bottleneck 문제를 해결하기 위해 제안됨
- Decoder 의 각 step 에서 source sentence 의 특정 부분에 집중할 수 있도록 connection 을 추가



1. Encoder 의 step 별 encoding 과 Decoder hidden state 의 similarity 를 계산하여 attention score 와 attention weight(probability) 를 계산한다.

2. Attention distribution 를 사용해 encoder hidden state 를 가중합하여 attention output 을 계산한다.

3. Attention output 과 decoder hidden state 를 concat 하고 해당 step 의 word 를 생성한다.

03 Attention

Sequence-to-Sequence with Attention

Attention : in equations

Encoder hidden state : $h_1, \ldots, h_N \in \mathbb{R}^h$

Decoder hidden state : $s_t \in \mathbb{R}^h$

1. Encoder 의 step 별 encoding 과 Decoder hidden state 의 similarity 를 계산하여 attention score 와 attention weight(probability) 를 계산한다.

Attention score : $\boldsymbol{e}^t = [\boldsymbol{s}_t^T \boldsymbol{h}_1, \ldots, \boldsymbol{s}_t^T \boldsymbol{h}_N] \in \mathbb{R}^N$

Attention weight : $\alpha^t = \mathrm{softmax}(\boldsymbol{e}^t) \in \mathbb{R}^N$

2. Attention distribution 를 사용해 encoder hidden state 를 가중합하여 attention output 을 계산한다.

Attention output : $\boldsymbol{a}_t = \sum_{i=1}^{N} \alpha_i^t \boldsymbol{h}_i \in \mathbb{R}^h$

3. Attention output 과 decoder hidden state 를 concat 하고 해당 step 의 word 를 생성한다.

Concat with attention : $[\boldsymbol{a}_t; \boldsymbol{s}_t] \in \mathbb{R}^{2h}$

03 Attention

Sequence-to-Sequence with Attention

Attention is great

- NMT 의 성능향상 : decoder 가 source sentence 의 특정 영역에 집중할 수 있도록 함
- Source sentence 의 모든 embedding 을 참고하는 구조를 사용하여 일반적인 Encoder-Decoder 구조의 bottleneck 문제를 해결함
- Shortcut 구조를 사용하여 Vanishing gradient 문제를 줄임
- Alignment 를 스스로 학습할 수 있고 attention probability 를 통해 결과에 대한 interpretability 를 제공함

Sentiment analysis example

the abundance of road rumble and engine hum will grate on you over a long trip. wind noise is tolerable, but overall, the spark lacks isolation from its surroundings, contributing to an insubstantial feel.

True : Negative(1,2) (noise)

Pred : Negative
Confidence : 0.734
Attention : an, feel, insubstantial, and, rumble

1st : wind noise is tolerable, but overall, the spark lacks isolation from its surroundings, contributing to an insubstantial feel.
2nd : the abundance of road rumble and engine hum will grate on you over a long trip.

03 Attention

Sequence-to-Sequence with Attention

Attention is general Deep Learning technique

- More general definition of attention:
  - Given a set of vector *values*, and a vector *query*, attention is a technique to compute a weighted sum of the values, dependent on the query.

- Attention output 계산을 위한 가중합은 value(encoder hidden state) 가 가지는 정보의 선별적 사용으로 해석할 수 있고, query(decoder hidden state) 는 이 정보가 어디에 집중할 지를 결정한다.
- Attention 은 query 에 dependent 한 고정된 크기의 value representation 을 얻을 수 있는 방법이다.

감사합니다

00 Appendix

- CS224N Lecture 7 강의 : https://www.youtube.com/watch?v=wzfWHP6SXxY&list=PLoROMvodv4rOSH4v6133s9LFPRHjEmbmJ&index=7
- CS224N Lecture 8 강의 : https://www.youtube.com/watch?v=gKD7jPAdbpE&list=PLoROMvodv4rOSH4v6133s9LFPRHjEmbmJ&index=8
- CS224N Lecture 7 슬라이드 : https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1214/slides/cs224n-2021-lecture07-nmt.pdf
- RNN figure : https://colab.research.google.com/github/patrickvonplaten/notebooks/blob/master/Encoder_Decoder_Model.ipynb
- BLEU : https://donghwa-kim.github.io/BLEU.html
- Other review :
  - DSBA : https://www.youtube.com/watch?v=c8y9ZAb9aks&list=PLetSlH8YjIfVdobI2IkAQnNTb1Bt5Ji9U&index=7
  - 투빅스 : https://velog.io/@tobigs-text1314/CS224n-Lecture-8-Machine-Translation-Sequence-to-sequence-and-Attention
  - https://jeongukjae.github.io/posts/cs224n-lecture-8-machine-translation,-seq2seq,-attention/
  - https://aaoossiinnaa.tistory.com/43?category=457408