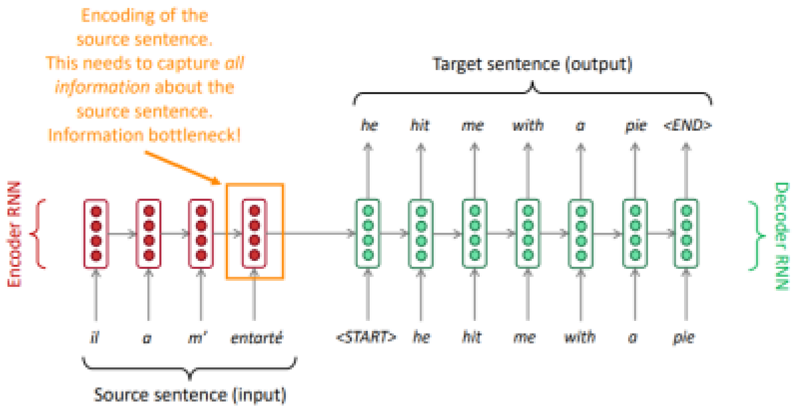
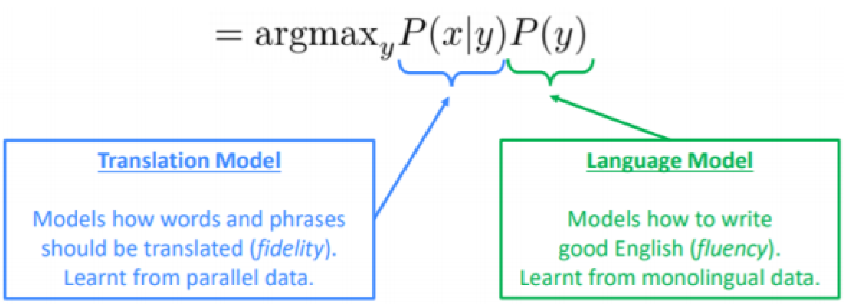
Week 10

태그	
열	

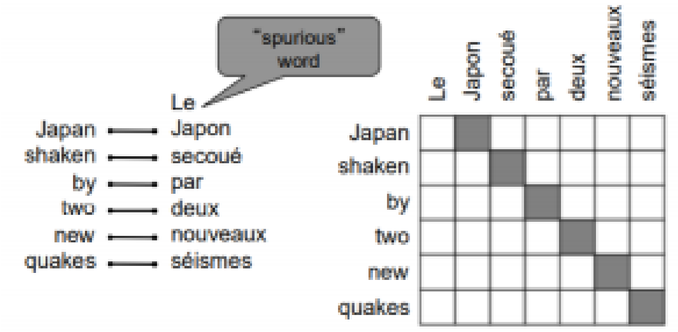


Machine Translation

- Definition: 한 언어의 문장을 다른 언어의 문장으로 번역하는 것
- Machine translation is a major use-case of a new neural architecture (seq-to-seq)
- seq-to-seq is improved by attention

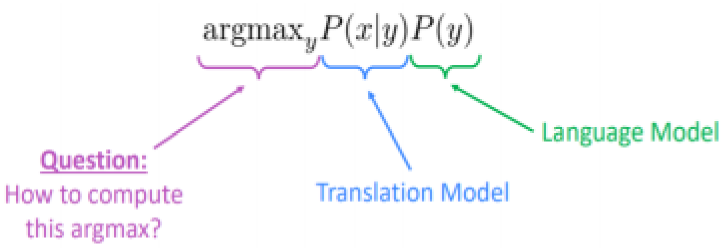


- Translation: 작은 단어와 구의 번역
- Language model: 좋은 문장, 좋은 구조 도출
- Alignment



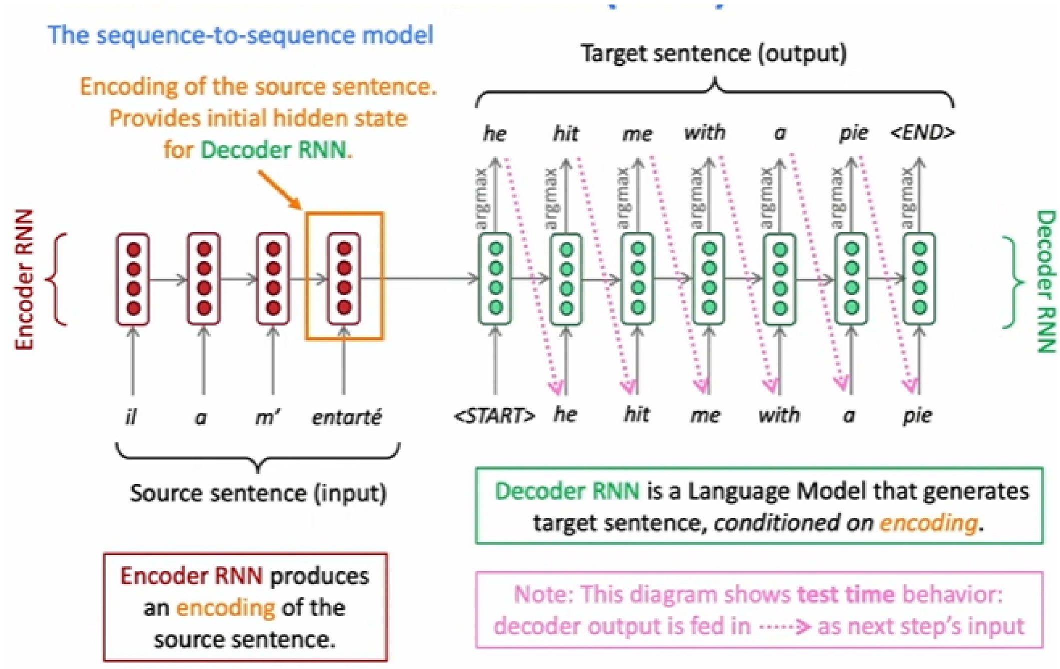
- 정렬이란, 두 문장 사이에서 특정 단어쌍들의 대응
- 어떤 단어들은 일대일 대응이 되지 않기도 함
- 혹은 many-to-one, one-to-many, many-to-many 관계가 있기도 함

- Decoding for SMT (Statistical Machine Translation)



  - 해결 방법
    - 무차별 대입 솔루션
    - Heuristic 알고리즘
  - 모든 가능성을 고려하고 가장 가능성이 높은 방향을 선택해 나감
  - 특징
    - 좋은 성능을 내지만 매우 복잡한 구조
    - 각 system은 sub-system들이 모여있는 형태
    - 많은 feature engineering이 필요
    - 추가적인 많은 자료 필요
    - 사람의 손을 많이 거쳐야함

Neural Machine Translation (Sequence to Sequence)

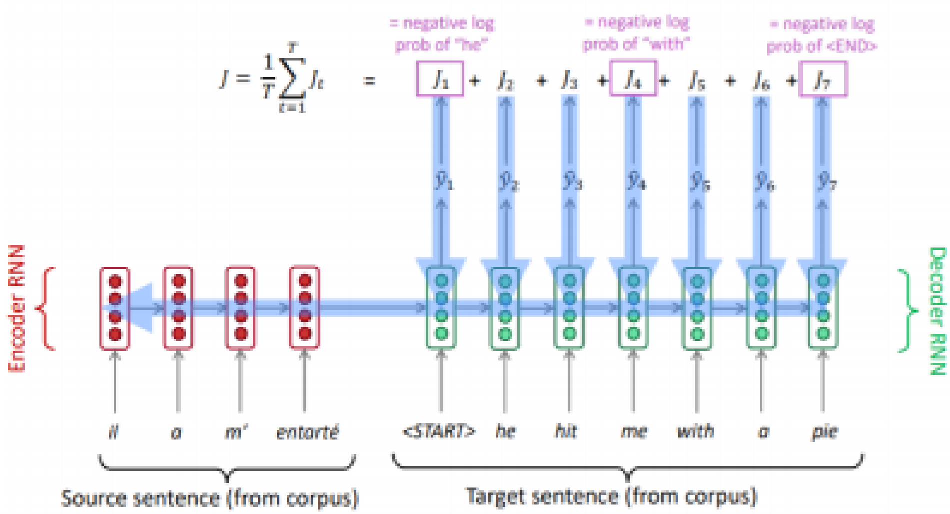


- Seq to seq model은 conditional language model의 예이다.

  - **NMT directly calculates** $P(y|x)$:

$$P(y|x) = P(y_1|x)\,P(y_2|y_1,x)\,P(y_3|y_1,y_2,x)\ldots\underbrace{P(y_T|y_1,\ldots,y_{T-1},x)}_{\text{Probability of next target word, given target words so far and source sentence } x}$$

- Training



- 하나의 loss에 대해 시스템 전체가 학습하므로 end-to-end 시스템이라고 한다.

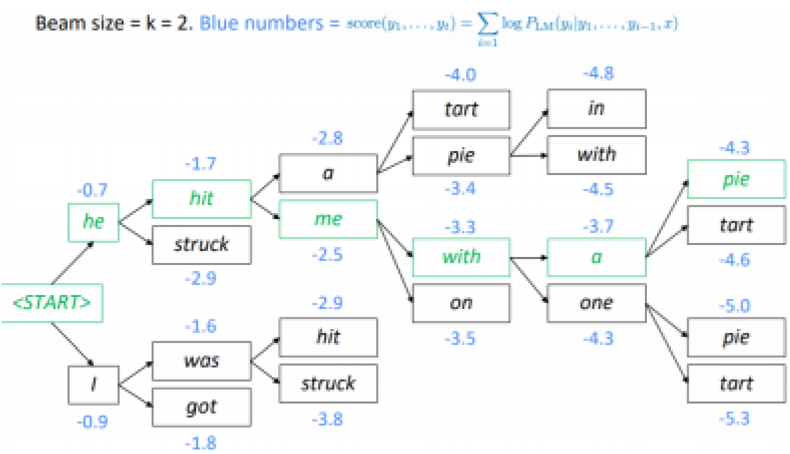
- Greedy decoding의 문제
  - 초반이 잘못되면 뒷부분도 다 망치게 된다
  - How to solve it?
    - Beam search decoding
- Beam search decoding

- A hypothesis $y_1, \ldots, y_t$ has a score which is its log probability:

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

  - Scores are all negative, and higher score is better
  - We search for high-scoring hypotheses, tracking top *k* on each step

  - Core idea: On each step of decoder, keep track of the k most probable partial translations (which we call hypotheses)
    - k is the beam size (in practice around 5 to 10)
  - 효율적이지만 최적의 결과를 보장하진 못함
  - Example



- NMT의 장점
  - 더 나은 성능

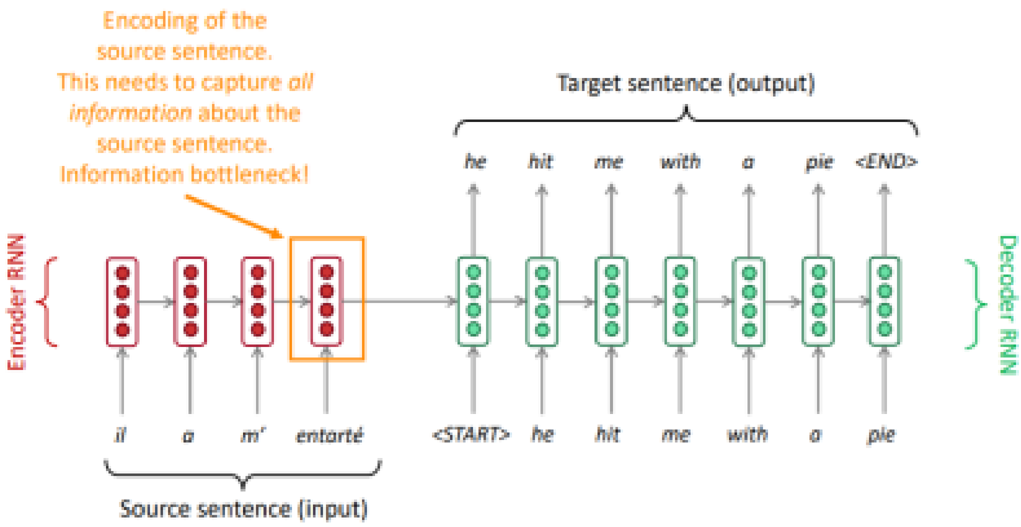
    - Single neural network to be optimized end-to-end (하부 구조가 개별적으로 optimized 될 필요 x)
    - 인간의 노력 덜 필요
- NMT의 단점
    - Hard to debug
    - Difficult to control
- How do we evaluate Machine Translation?
    - BLEU (Bilingual Evaluation Understudy)

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

$$= min(1, \frac{14}{14}) \times (\frac{10}{14} \times \frac{5}{13} \times \frac{2}{12} \times \frac{1}{11})^{\frac{1}{4}}$$

Attention

- Seq-to-seq: the bottleneck problem



    - 맨 끝에서 모든 정보를 캡쳐하기를 강요 → 너무 많은 압력 → 병목 문제

- 해결책 → **Attention**

- Core idea: on each step of the decoder, use direct connection to the encoder to focus on a particular part of the source sequence
- Attention in equations

- We have encoder hidden states $h_1, \ldots, h_N \in \mathbb{R}^h$
- On timestep *t*, we have decoder hidden state $s_t \in \mathbb{R}^h$
- We get the attention scores e^t for this step:

$$e^t = [s_t^T h_1, \ldots, s_t^T h_N] \in \mathbb{R}^N$$

- We take softmax to get the attention distribution α^t for this step (this is a probability distribution and sums to 1)

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

- We use α^t to take a weighted sum of the encoder hidden states to get the attention output a_t

$$a_t = \sum_{i=1}^{N} \alpha_i^t h_i \in \mathbb{R}^h$$

- Finally we concatenate the attention output a_t with the decoder hidden state s_t and proceed as in the non-attention seq2seq model

$$[a_t; s_t] \in \mathbb{R}^{2h}$$

73

1. Encoder hidden states
2. Decoder hidden state
3. Softmax
4. Attention output
5. Y hat

- Attention의 장점
    - NMT 성능을 향상시킴
    - 병목문제 해결
    - 기울기 소실 문제 해결
    - 추적 가능성