CS224N 2023 | WEEK 1-2 배지섭

LECTURE 6:
NEURAL MACHINE TRANSLATION

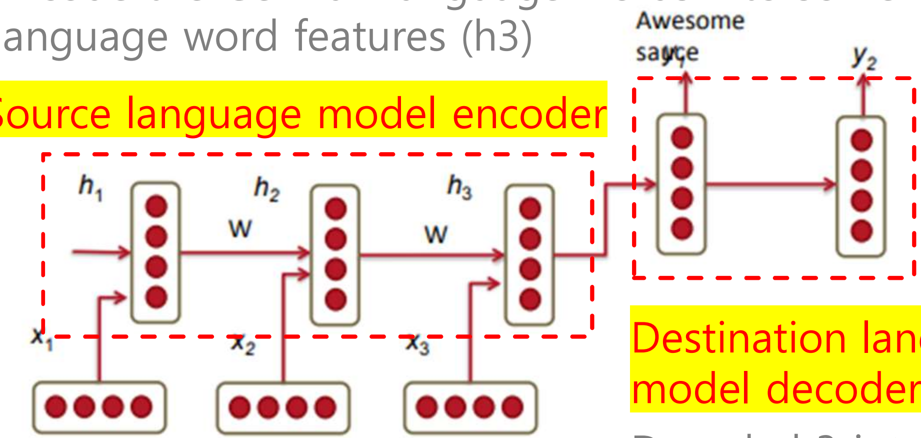
Application: RNN Translation Model

"Echt dicke Kiste"

"Awesome sauce"

Encode the German language words into some language word features (h3)

Source language model encoder



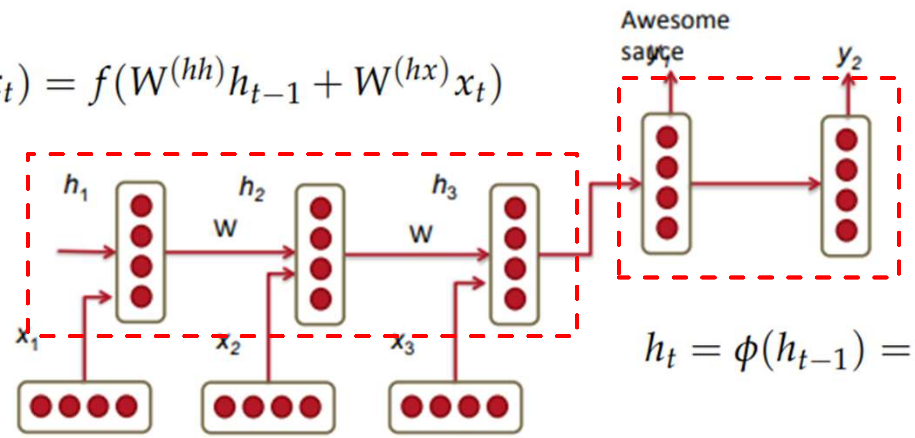
Destination language model decoder

Decode h3 into English word outputs

LECTURE 6:
NEURAL MACHINE TRANSLATION

Application: RNN Translation Model

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



$$h_t = \phi(h_{t-1}) = f(W^{(hh)}h_{t-1})$$

$$y_t = softmax(W^{(S)}h_t)$$

$$\max_{\theta} \frac{1}{N} \sum_{n=1}^{N} \log p_\theta(y^{(n)}|x^{(n)})$$

Cross entropy function

은닉 상태 가중치를 결합하여 encode & decode

Application: RNN Translation Model

Extensions:

- Trian different RNN weights for encoding and decoding
- Compute every hidden state in the decoder using three different inputs
- Deeper layers often improve prediction accuracy due to their higher learning capacity
- Train bi-directional encoders to improve accuracy
- Reversing the order of the input words can help reduce the error rate in generating the output phrase

LECTURE 6:
NEURAL MACHINE TRANSLATION

GATED RECURRENT UNITS

RNN은 long-term dependencies의 실제 포착이 어려움.

GRU는 RNN의 long-term dependencies의 포착을 쉽게 해줌.

$$z_t = \sigma(W^{(z)} x_t + U^{(z)} h_{t-1}) \qquad \text{(Update gate)}$$

$$r_t = \sigma(W^{(r)} x_t + U^{(r)} h_{t-1}) \qquad \text{(Reset gate)}$$

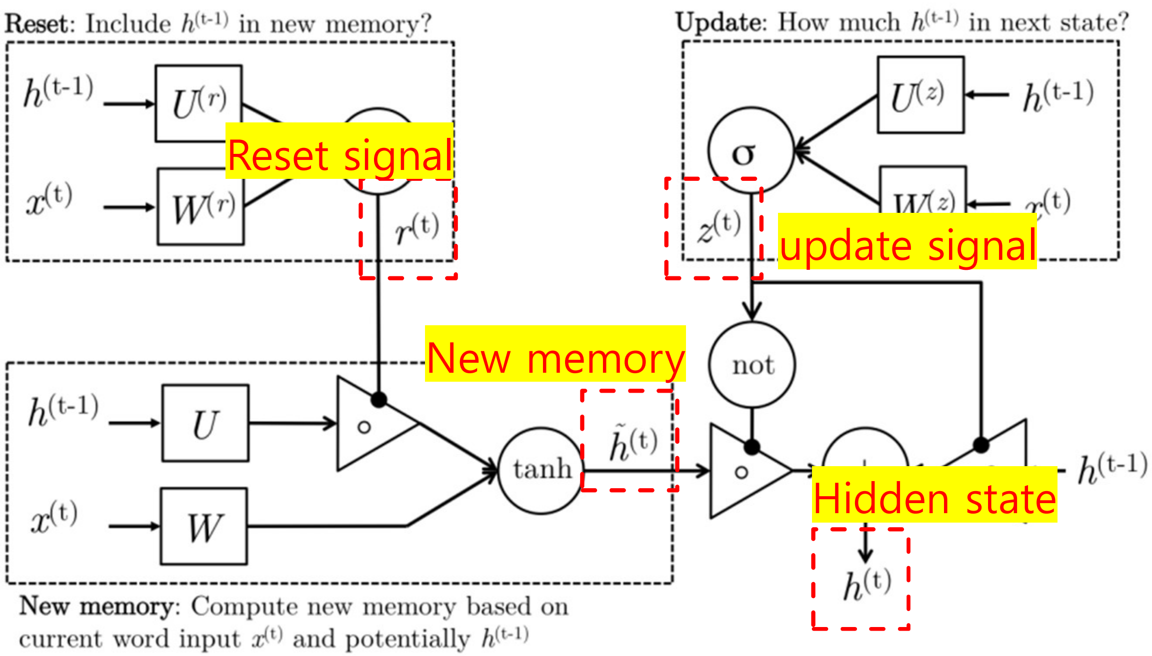
$$\tilde{h}_t = \tanh(r_t \circ U h_{t-1} + W x_t) \qquad \text{(New memory)}$$

$$h_t = (1 - z_t) \circ \tilde{h}_t + z_t \circ h_{t-1} \qquad \text{(Hidden state)}$$

LECTURE 6:
NEURAL MACHINE TRANSLATION

GATED RECURRENT UNITS

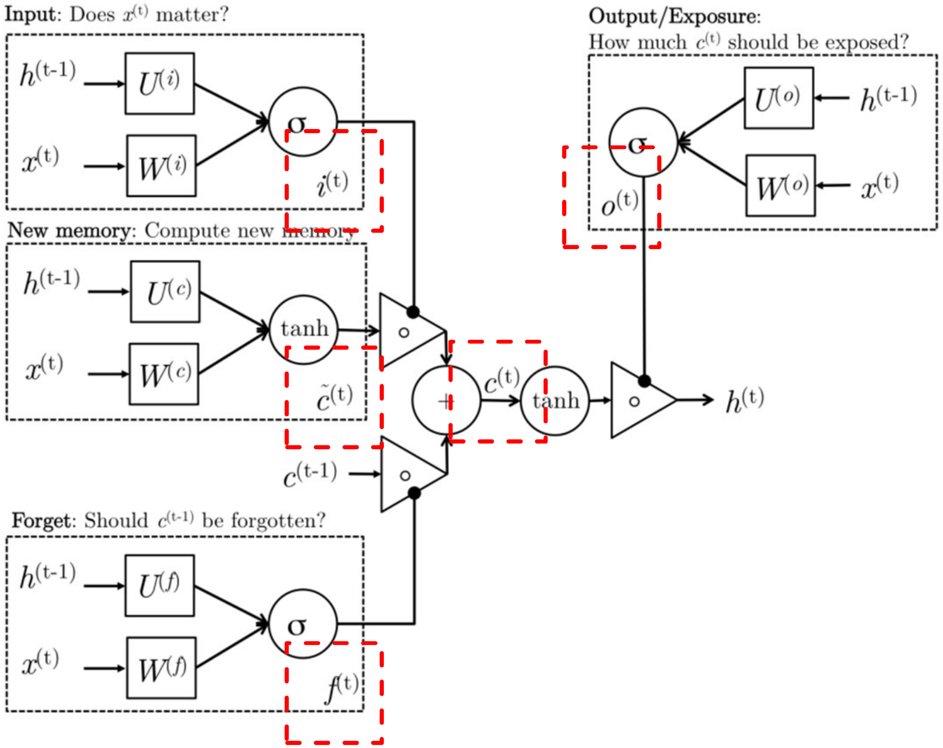
$z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1})$	(Update gate)
$r_t = \sigma(W^{(r)}x_t + U^{(r)}h_{t-1})$	(Reset gate)
$\tilde{h}_t = \tanh(r_t \circ Uh_{t-1} + Wx_t)$	(New memory)
$h_t = (1 - z_t) \circ \tilde{h}_t + z_t \circ h_{t-1}$	(Hidden state)



LECTURE 6:
NEURAL MACHINE TRANSLATION

LONG-SHORT-TERM-MEMORIES

$$i_t = \sigma(W^{(i)}x_t + U^{(i)}h_{t-1}) \qquad \text{(Input gate)}$$
$$f_t = \sigma(W^{(f)}x_t + U^{(f)}h_{t-1}) \qquad \text{(Forget gate)}$$
$$o_t = \sigma(W^{(o)}x_t + U^{(o)}h_{t-1}) \qquad \text{(Output/Exposure gate)}$$
$$\tilde{c}_t = \tanh(W^{(c)}x_t + U^{(c)}h_{t-1}) \qquad \text{(New memory cell)}$$
$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \qquad \text{(Final memory cell)}$$
$$h_t = o_t \circ \tanh(c_t)$$



LECTURE 7:
SELF ATTENTION AND TRANSFORMERS

NEURAL MACHINE TRANSLATION WITH SEQ2SEQ

이전까지는 주어진 문장에서 다음 단어를 예측하는 단일 결과에 대해 논의함.
이것은 아래의 문제들에 적용될 수 없음.

- **Translation:** taking a sentence in one language as input and outputting the same sentence in another language.
- **Conversation:** taking a statement or question as input and responding to it.
- **Summarization:** taking a large body of text as input and outputting a summary of it.

Sequence-to-sequence model이 해당 문제를 다룰 수 있음.

LECTURE 7:
SELF ATTENTION AND TRANSFORMERS

NEURAL MACHINE TRANSLATION WITH SEQ2SEQ

Word-based system

이전 변역 시스템은 아래 요소들로 구성되는 확률모델에 기반했다.

- a **translation model**, telling us what a sentence/phrase in a source language most likely translates into
- a **language model**, telling us how likely a given sentence/phrase is overall.

이 시스템 (naïve word-based system)은 언어들간 다른 어순의 포착을 실패한다.

Phrase-based system

구(phrase)의 단어 순서를 입출력에서 고려해 더욱 복잡한 구문은 다룰 수 있다.

LECTURE 7:
SELF ATTENTION AND TRANSFORMERS

NEURAL MACHINE TRANSLATION WITH SEQ2SEQ

Seq2Seq

LSTM의 사용으로

1. 자의적인 결과 순서를 생성
2. 특정 부분에 자동으로 집중

하는 것이 가능하다.

- an *encoder*, which takes the model's input sequence as input and encodes it into a fixed-size "context vector", and
- a *decoder*, which uses the context vector from above as a "seed" from which to generate an output sequence.

NEURAL MACHINE TRANSLATION WITH SEQ2SEQ

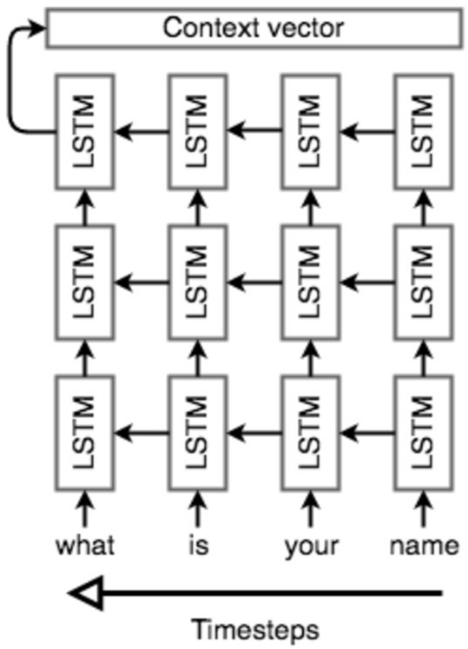
Seq2Seq architecture - encoder

Stacked LSTM

Input 시퀀스의 반전

"the last thing that the encoder sees will (roughly) corresponds to the first thing that the model outputs."

"network is unrolled"

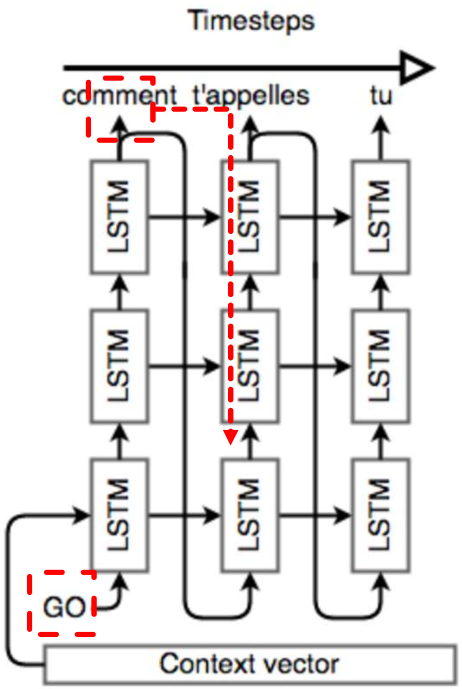


LECTURE 7:
SELF ATTENTION AND TRANSFORMERS

NEURAL MACHINE TRANSLATION WITH SEQ2SEQ

Seq2Seq architecture - decoder

Seq2Seq는 매우 긴 입력에 효과적이지 않다.
(LSTM의 실용적 한계)



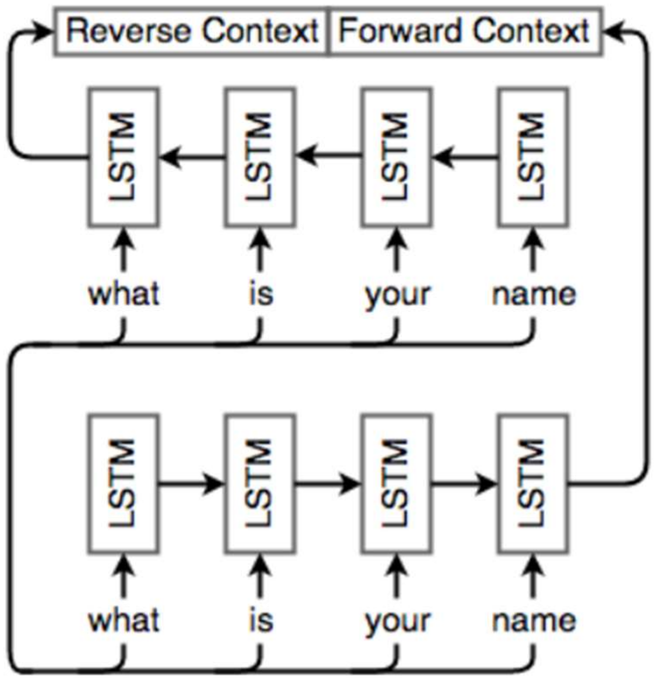
LECTURE 7:
SELF ATTENTION AND TRANSFORMERS

RECAP & BASIC NMT EXAMPLE

$\left[o_t^{(f)} \quad o_t^{(b)}\right]$ 는 CONCATENATED VECTOR이다.

$o_t^{(f)}$ is the output of the forward-direction RNN on word t

$o_t^{(b)}$ is the corresponding output from the reverse-direction RNN.



LECTURE 7:
SELF ATTENTION AND TRANSFORMERS

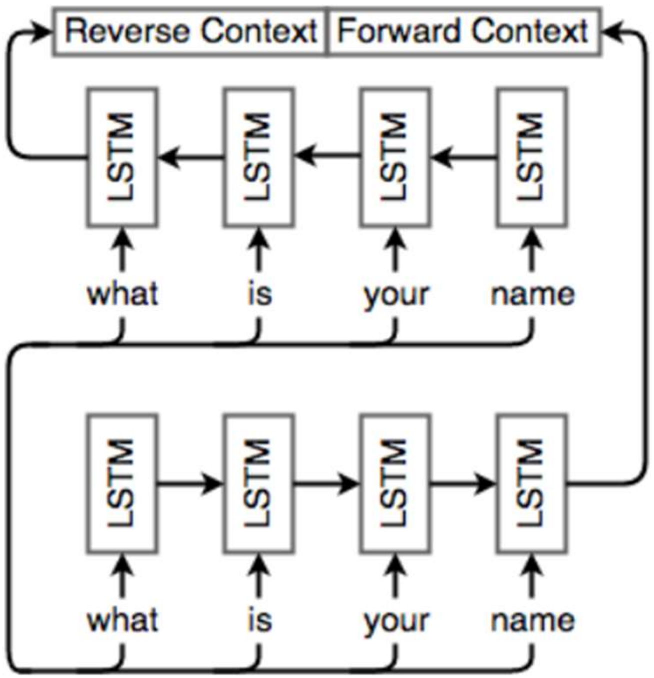
RECAP & BASIC NMT EXAMPLE

$\left[o_t^{(f)} \quad o_t^{(b)}\right]$ 는 CONCATENATED VECTOR이다.

$o_t^{(f)}$ is the output of the forward-direction RNN on word t

$o_t^{(b)}$ is the corresponding output from the reverse-direction RNN.

Bidirectional LSTM은 기존의 종속성에 대한 문제를 해결할 수 있다.



LECTURE 7:
SELF ATTENTION AND TRANSFORMERS

ATTENTION MECHANISM

"the ball is on the field."

"the ball is on the field."

Attention mechanism은 decoder가 모든 input seq를 매 단계마다 확인하도록 함.

(-> 어느 시점에 어떤 input word가 중요한지 결정)

LECTURE 7:
SELF ATTENTION AND TRANSFORMERS

ATTENTION MECHANISM

Bahdanau et al. NMT model

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

Context vector만 사용한 seq2seq와 구분

s_{i-1} : the previous hidden vector
y_{i-1} : generated word at the previous step
c_i : a context vector

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

h_j : hidden vector

$$\alpha_{i,j} = \frac{exp(e_{i,j})}{\sum_{k=1}^{n} exp(e_{i,k})}$$

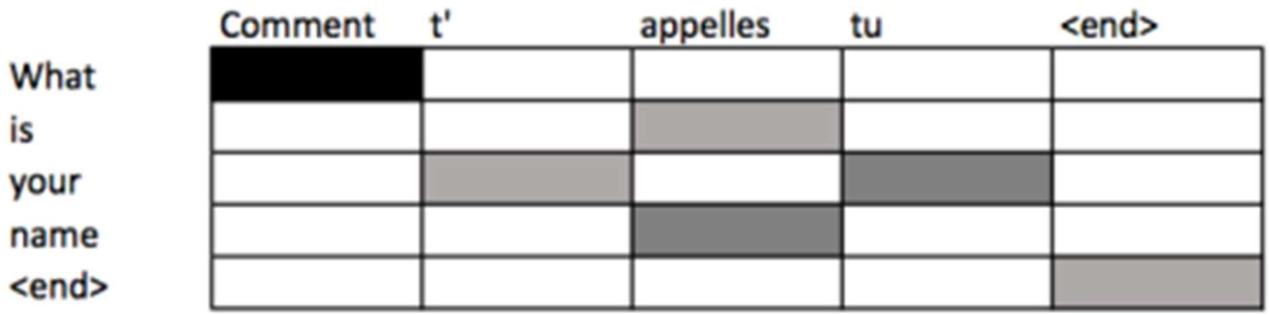
Softmax 층을 사용해 점수를 벡터 $\alpha_i = (\alpha_{i,1}, \ldots, \alpha_{i,n})$ 로 Nomalize

$$c_i = \sum_{j=1}^{n} \alpha_{i,j} h_j$$

Context vector는 decoder의 i번째 단계에 대해 원래 문장의 연관된 문맥 정보를 포착

ATTENTION MECHANISM

Connection with translation alignment



Decoding step I, Attention score는 소스 문장의 단어가 타겟 단어 I 와 나란히 정렬됨을 나타낸다.

OTHER MODELS

Huong et al. NMT model

- Global attention

$$score(h_i, \bar{h}_j) = \begin{cases} h_i^T \bar{h}_j \\ h_i^T W \bar{h}_j \\ W[h_i, \bar{h}_j] \end{cases} \in \mathbb{R} \qquad c_i = \sum_{j=1}^{n} \alpha_{i,j} h_j \qquad \alpha_{i,j} = \frac{exp(score(h_j, \bar{h}_i))}{\sum_{k=1}^{n} exp(score(h_k, \bar{h}_i))} \qquad \boxed{\tilde{h}_i} = f([\bar{h}_i, c_i])$$

최종 예측을 위해서 decode에 input으로 사용

- Local attention

  정렬 위치를 예측, 이 위치 중심의 window를 사용해 context vector를 계산

다양한 예측 방법이 있음.

LECTURE 7:
SELF ATTENTION AND TRANSFORMERS

SEQUENCE MODEL DECODERS

$$\bar{s}* = \text{argmax}_{\bar{s}}(\mathbb{P}(\bar{s}|s))$$

Idea는 $(\bar{s}|s)$에 대한 확률분포로 가장 적합한 문장을 얻는다.

- Exhaustive search
- Ancestral sampling
- Greedy search 기하급수적 크기의 출력, 낮은 성능과 많은 변수 그리고 실수 한번으로 크게 영향

- Beam search

$$\tilde{\mathcal{H}}_{t+1} = \bigcup_{k=1}^{K} \mathcal{H}_{t+1}^{\tilde{k}}$$

최고의 K개 후보를 유지

NMT에 가장 흔히 사용되는 기술이다.

$$\mathcal{H}_{t+1}^{\tilde{k}} = \{(x_1^k, \ldots, x_t^k, v_1), \ldots, (x_1^k, \ldots, x_t^k, v_{|V|})\}$$

LECTURE 7:
SELF ATTENTION AND TRANSFORMERS

EVLAUATION OF MACHINE TRANSLATION SYSTEMS

두 개의 기초적인 평가와 BLEU

- Human Evaluation
- Evaluation against another task
- Bilingual Evaluation Understudy(BLUE)

A there are many ways to evaluate the quality of a translation, like comparing the number of n-grams between a candidate translation and reference.

B the quality of a translation is evaluate of n-grams in a reference and with translation.

p_n = # matched n-grams / # n-grams in candidate translation

$$\beta = e^{\min(0, 1 - \frac{\text{len}_{\text{ref}}}{\text{len}_{\text{MT}}})}$$

$$\text{BLEU} = \beta \prod_{i=1}^{k} p_n^{w_n}$$

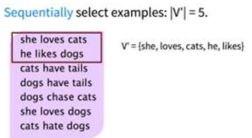
Corpus 수준에서만 잘 작동한다.

기하평균은 하나의 값만 0이 되어도 전체 결과가 0이다.

LECTURE 7:
SELF ATTENTION AND TRANSFORMERS

DEALING WITH THE LARGE OUTPUT VOCABULARY

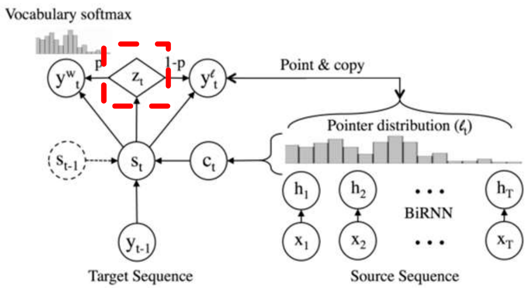
- Scaling soft mas
  - Noise Contrastive Estimation
  - Hierarchiacal Softmas
- Reducing vocabulary



$$Q(y_t) = \begin{cases} \frac{1}{|V'|}, if y_t \in |V'| \\ 0, otherwise \end{cases}$$

The challenge is that the correct target word is unknown and we have to "guess" what the target word might be.

- Handling unknown words



WORD AND CHARACTER-BASED MODELS

Word segmentation

data set contains 4 words with their frequencies

selected most frequent n-gram pair (e,s,9)

adding current most frequent n-gram pair (es,t,9).

Figure 10: Byte Pair Encoding

start with a vocabulary of characters and keep extending the vocabulary with most frequent n-gram pairs in the data set

repeated until all n-gram pairs are selected or vocabulary size reaches some threshold

Once the vocabulary is built, NMT system with some seq2seq architecture can be directly trained on these word segments

WORD AND CHARACTER-BASED MODELS

Character-based model

this model iterates over all characters c1, c2 . . . c_m to look up
the character embeddings e1,e2 . . .e_m

fed into a biLSTM to get the final hidden states h_f , h_b for forward and backward directions

$$e_w = W_f H_f + W_b H_b + b$$

The final word embedding is computed by an affine transformation of two hidden states

LECTURE 7:
SELF ATTENTION AND TRANSFORMERS

WORD AND CHARACTER-BASED MODELS

Hybrid NMT

The system translates mostly at word-level and consults the character components for rare words

On a high level, the character-level recurrent neural networks compute source word representations and recover unknown target words when needed

- Word-based Translation as a Backbone
- Source Character-based Representation
- Target Character-level Generation