06 – Neural Language Translation

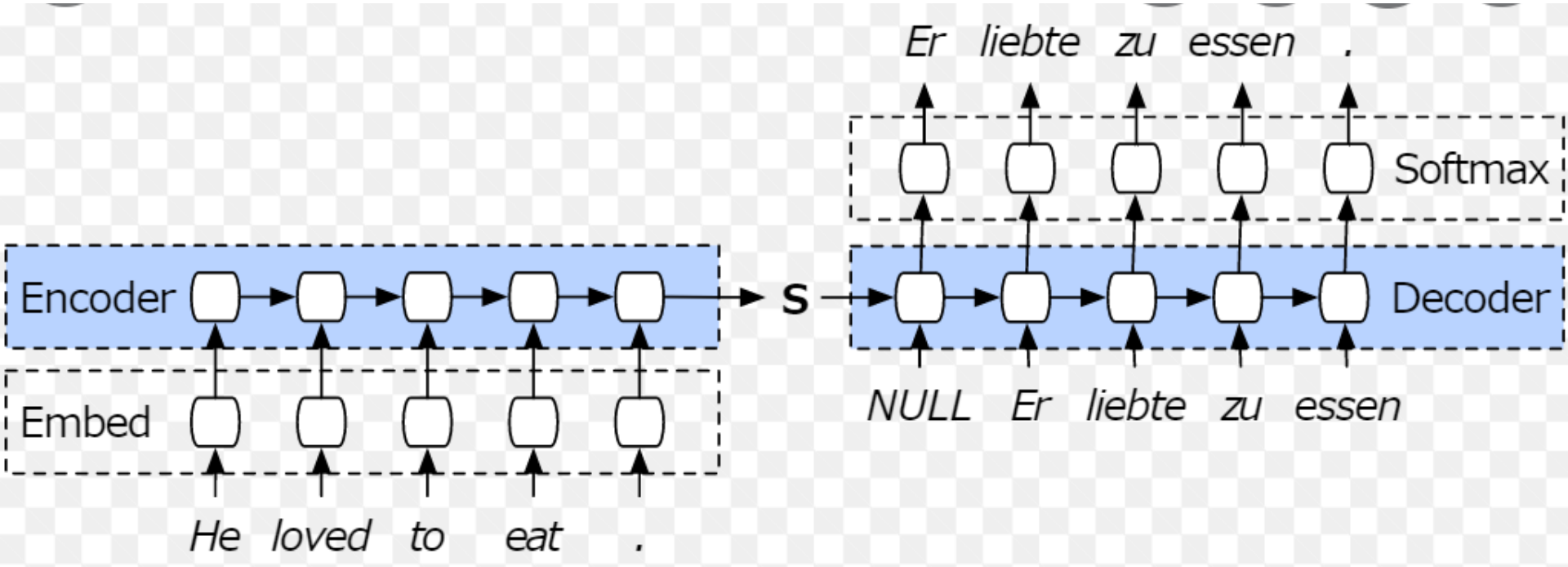
Incheon Paik
Intelligent Data Analytics Lab.
University of Aizu

Contents

- RNN Encoder - Decoder
- Statistical Machine Translation
- Sequence-to-Sequence Model
- Sequence-to-Sequence with Attention
- Transformer Based Translation
- Word Embeddings
- Back Translation
- Augmentation Based Translation

Sequence to Sequence Model

- Attention in Seq2Seq Model



Statistical Machine Translation

- SMT Goal: To find "f"

In a commonly used statistical machine translation system (SMT), the goal of the system (decoder, specifically) is to find a translation $\mathbf{f}$ given a source sentence $\mathbf{e}$, which maximizes

$$p(\mathbf{f} \mid \mathbf{e}) \propto p(\mathbf{e} \mid \mathbf{f}) p(\mathbf{f}),$$

where the first term at the right hand side is called *translation model* and the latter *language model*

- Log-linear model with additional features & weight

$$\log p(\mathbf{f} \mid \mathbf{e}) = \sum_{n=1}^{N} w_n f_n(\mathbf{f}, \mathbf{e}) + \log Z(\mathbf{e}), \quad (9)$$

where f_n and w_n are the n-th feature and weight, respectively. $Z(\mathbf{e})$ is a normalization constant that does not depend on the weights. The weights are often optimized to maximize the BLEU score on a development set.

Introduction

- Research Motivation
  - Novel Neural Network Architecture that can be used as a part of the conventional phrase-based SMT system.
  - The architecture consists of two RNN: RNN Encoder-Decoder
  - The RNN Encoder-Decoer has a novel hidden unit - GRU

RNN Encoder-Decoder

- RNN Network
  - At each time step t, the hidden state $h_{<t>}$

$$\mathbf{h}_{\langle t \rangle} = f\left(\mathbf{h}_{\langle t-1 \rangle}, x_t\right), \qquad (1)$$

  - A multinomial distribution (1-of-K coding) can be output using a softmax activation function

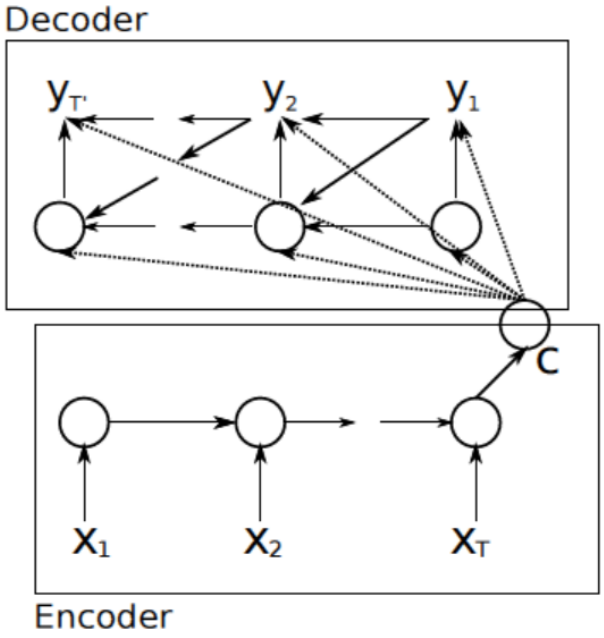
$$p(x_{t,j} = 1 \mid x_{t-1}, \ldots, x_1) = \frac{\exp\left(\mathbf{w}_j \mathbf{h}_{\langle t \rangle}\right)}{\sum_{j'=1}^{K} \exp\left(\mathbf{w}_{j'} \mathbf{h}_{\langle t \rangle}\right)}, \qquad (2)$$

  - Probability of the sequence x

$$p(\mathbf{x}) = \prod_{t=1}^{T} p(x_t \mid x_{t-1}, \ldots, x_1). \qquad (3)$$

RNN Encoder-Decoder

- RNN Encoder-Decoder



- Conditional distribution of the next symbol

- After reading the end of the sequence, the hidden state of the RNN is a summary C (Context Vector) of the whole input sequence.
- The hidden state of the decoder at time t is computed by

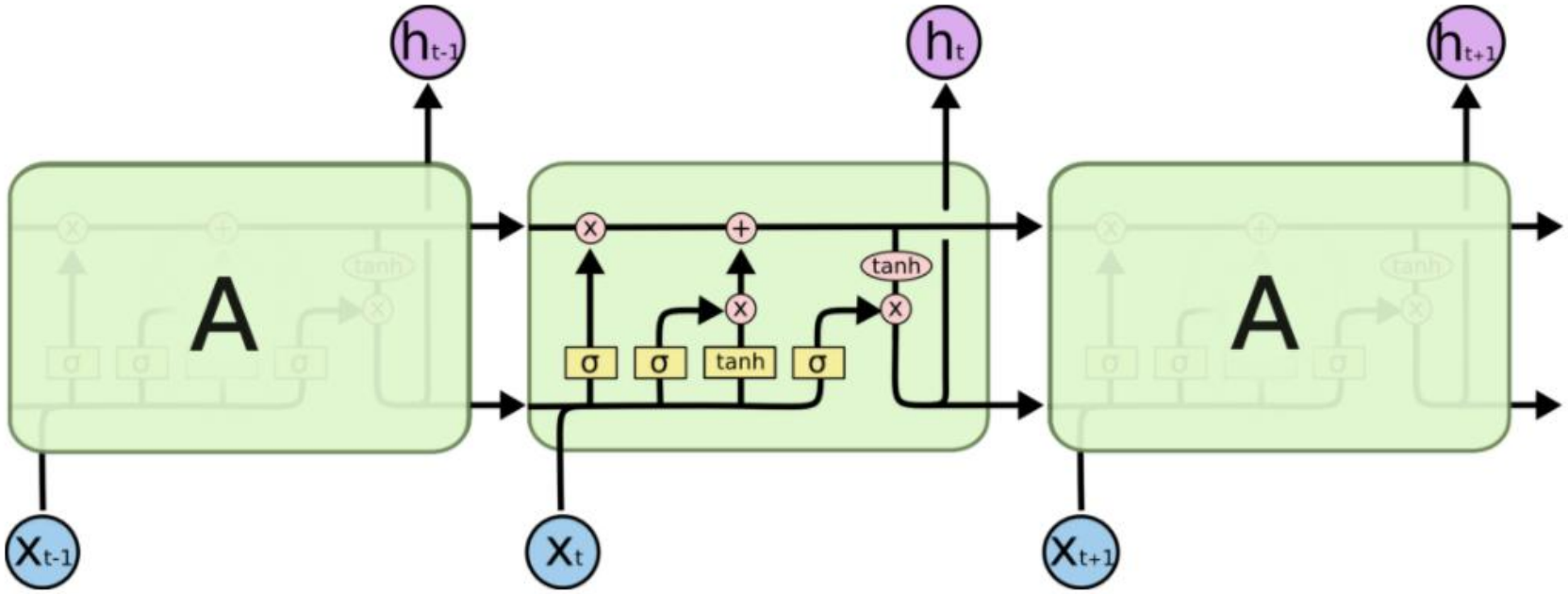
$$\mathbf{h}_{\langle t \rangle} = f\left(\mathbf{h}_{\langle t-1 \rangle}, y_{t-1}, \mathbf{c}\right),$$

and similarly, the conditional distribution of the next symbol is

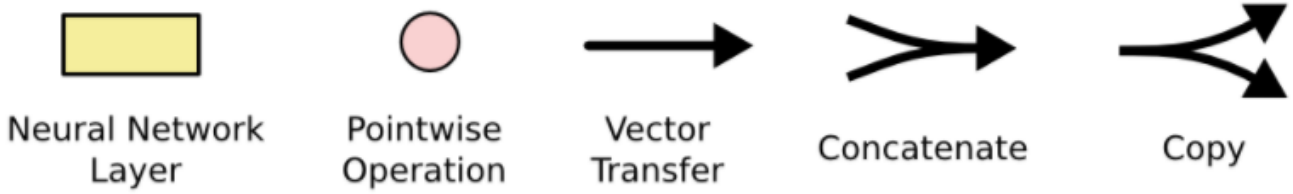
$$P(y_t|y_{t-1}, y_{t-2}, \ldots, y_1, \mathbf{c}) = g\left(\mathbf{h}_{\langle t \rangle}, y_{t-1}, \mathbf{c}\right).$$

for given activation functions f and g (the latter must produce valid probabilities, e.g. with a softmax).

LSTM Network

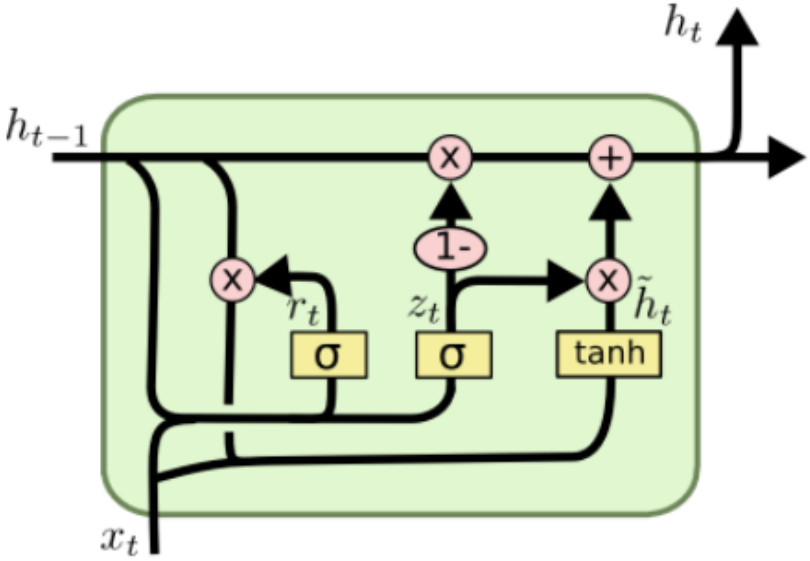


The repeating module in an LSTM contains four interacting layers.



Variants on LSTM- GRU

- A slightly more dramatic variation on the LSTM is the Gated Recurrent Unit, or GRU, introduced by Cho, et al. (2014).



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Statistical Machine Translation

- Scoring Phrase Pairs with RNN Encoder-Decoder

When train the RNN Encoder–Decoder, the (normalized) frequencies of each phrase pair in the original corpora were ignored.

This measure was taken in order

(1) to reduce the computational expense of randomly selecting phrase pairs from a large phrase table according to the normalized frequencies and

(2) to ensure that the RNN Encoder–Decoder does not simply learn to rank the phrase pairs according to their numbers of occurrences.

Experiment

- Experimental Data
  - Bilingual corpa

  Europarl(61M words), news commentary(5.5M), UN(421M)
  - Two Crawled Corpa: 90M, 780M
  - By Two Data Selection methods, 418M words from 2G words for LM.
  - For test, newstest2012 and 2013 used
  - Each se has more than 70,000 words
  - Most frequent words selected: 15,000 word
  - Baseline Phrased-based SMT: BLEU – 30.64 and 33.3

Experiment

- RNN Encoder-Decoder
  - For the recurrent weight matrices, we first sampled from a white Gaussian distribution and used its left singular vectors matrix, following (Saxe et al., 2014).
  - Used Ada delta and stochastic gradient descent to train the RNN Encoder–Decoder with hyperparameters = 10.6 and = 0:95 (Zeiler, 2012).
  - At each update, we used 64 randomly selected phrase pairs from a phrase table (which was created from 348M words).
  - The model was trained for approximately three days.
- Neural Language Model
  - Comparison with CSLM and SMT
  - Trained the CSLM model on 7-grams from the target corpus.
  - Each input word was projected into R^{512} embedding space, concatenated to form a 3027(1536+1024+512) dimensional vector.
  - Output layer is a simple softmax layer.

Experiment

- BLEU Scores
  - Development and test set

Models	BLEU	
	dev	test
Baseline	30.64	33.30
RNN	31.20	33.87
CSLM + RNN	31.48	34.64
CSLM + RNN + WP	31.50	34.54

Table 1: BLEU scores computed on the development and test sets using different combinations of approaches. WP denotes a *word penalty*, where we penalizes the number of unknown words to neural networks.

- Phrase pairs

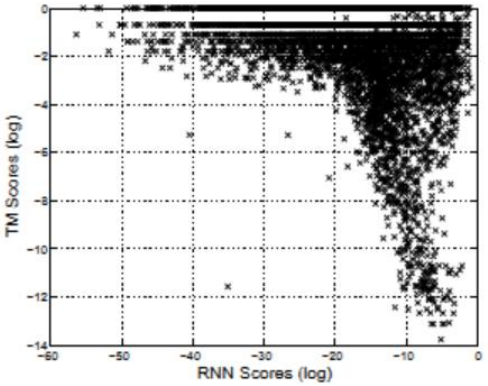


Figure 3: The visualization of phrase pairs according to their scores (log-probabilities) by the RNN Encoder–Decoder and the translation model.

Experiment

Source	Translation Model	RNN Encoder–Decoder
at the end of the	[a la fin de la] [f la fin des années] [être supprimés à la fin de la]	[à la fin du] [à la fin des] [à la fin de la]
for the first time	[**r** ⓒ pour la premi**r**ëre fois] [été donnés pour la première fois] [été commémorée pour la première fois]	[pour la première fois] [pour la première fois ,] [pour la première fois que]
in the United States and	[**?** aux **?**tats-Unis et] [été ouvertes aux États-Unis et] [été constatées aux États-Unis et]	[aux Etats-Unis et] [des Etats-Unis et] [des États-Unis et]
, as well as	[**?**s , qu'] [**?**s , ainsi que] [**?**re aussi bien que]	[, ainsi qu'] [, ainsi que] [, ainsi que les]
one of the most	[**?**t **?**l' un des plus] [**?**l' un des plus] [être retenue comme un de ses plus]	[l' un des] [le] [un des]

(a) Long, frequent source phrases

Source	Translation Model	RNN Encoder–Decoder
, Minister of Communications and Transport	[Secrétaire aux communications et aux transports :] [Secrétaire aux communications et aux transports]	[Secrétaire aux communications et aux transports] [Secrétaire aux communications et aux transports :]
did not comply with the	[vestimentaire , ne correspondaient pas à des] [susmentionnée n' était pas conforme aux] [présentées n' étaient pas conformes à la]	[n' ont pas respecté les] [n' était pas conforme aux] [n' ont pas respecté la]
parts of the world .	[ⓒ gions du monde .] [régions du monde considérées .] [région du monde considérée .]	[parties du monde .] [les parties du monde .] [des parties du monde .]
the past few days .	[le petit texte .] [cours des tout derniers jours .] [les tout derniers jours .]	[ces derniers jours .] [les derniers jours .] [cours des derniers jours .]
on Friday and Saturday	[vendredi et samedi à la] [vendredi et samedi à] [se déroulera vendredi et samedi ,]	[le vendredi et le samedi] [le vendredi et samedi] [vendredi et samedi]

(b) Long, rare source phrases

Table 2: The top scoring target phrases for a small set of source phrases according to the translation model (direct translation probability) and by the RNN Encoder–Decoder. Source phrases were randomly selected from phrases with 4 or more words. **?** denotes an incomplete (partial) character. **r** is a Cyrillic letter *ghe*.

Experiment

Source	Samples from RNN Encoder–Decoder
at the end of the	[à la fin de la] (×11)
for the first time	[pour la première fois] (×24) [pour la première fois que] (×2)
in the United States and	[aux États-Unis et] (×6) [dans les États-Unis et] (×4)
, as well as	[, ainsi que] [,] [ainsi que] [, ainsi qu'] [et UNK]
one of the most	[l' un des plus] (×9) [l' un des] (×5) [l' une des plus] (×2)

(a) Long, frequent source phrases

Source	Samples from RNN Encoder–Decoder
, Minister of Communications and Transport	[, ministre des communications et le transport] (×13)
did not comply with the	[n' tait pas conforme aux] [n' a pas respect l'] (×2) [n' a pas respect la] (×3)
parts of the world .	[arts du monde .] (×11) [des arts du monde .] (×7)
the past few days .	[quelques jours .] (×5) [les derniers jours .] (×5) [ces derniers jours .] (×2)
on Friday and Saturday	[vendredi et samedi] (×5) [le vendredi et samedi] (×7) [le vendredi et le samedi] (×4)

(b) Long, rare source phrases

Table 3: Samples generated from the RNN Encoder–Decoder for each source phrase used in Table 2. We show the top-5 target phrases out of 50 samples. They are sorted by the RNN Encoder–Decoder scores.

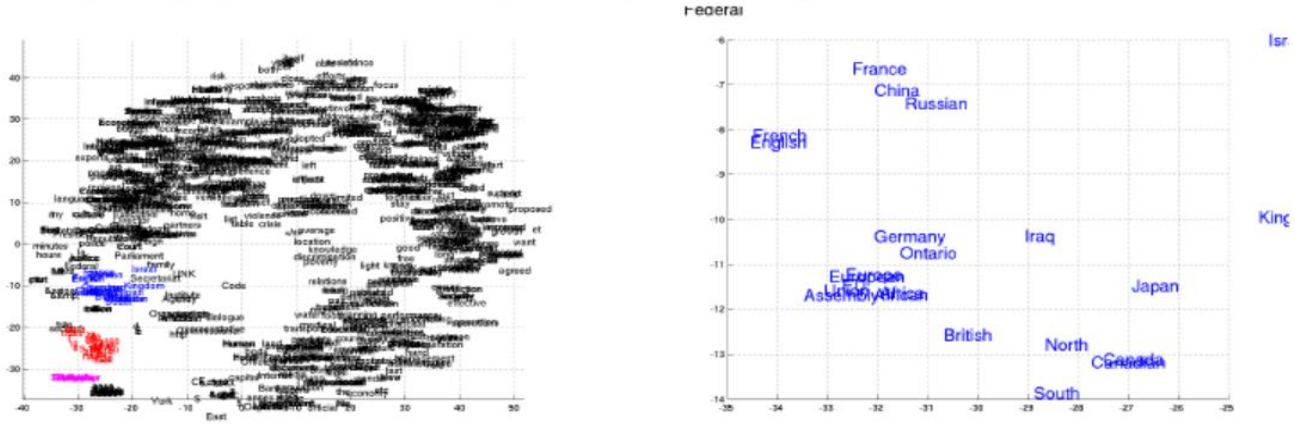
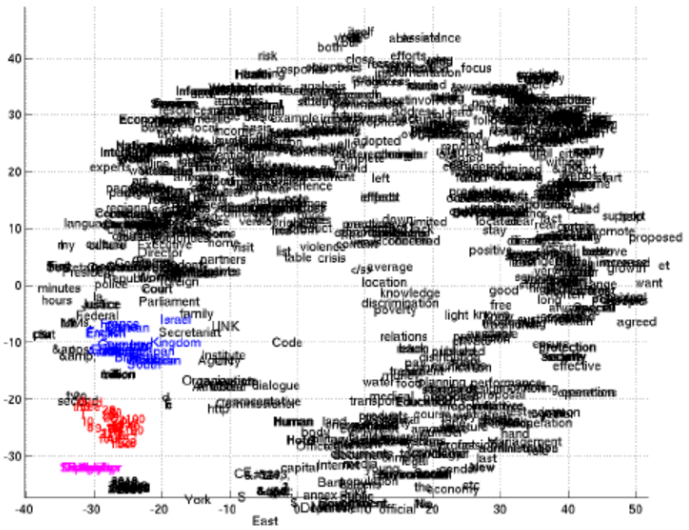
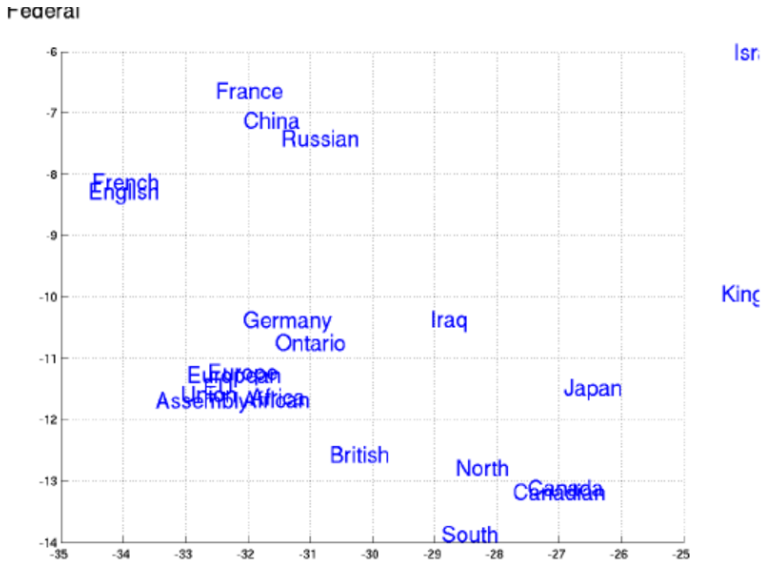
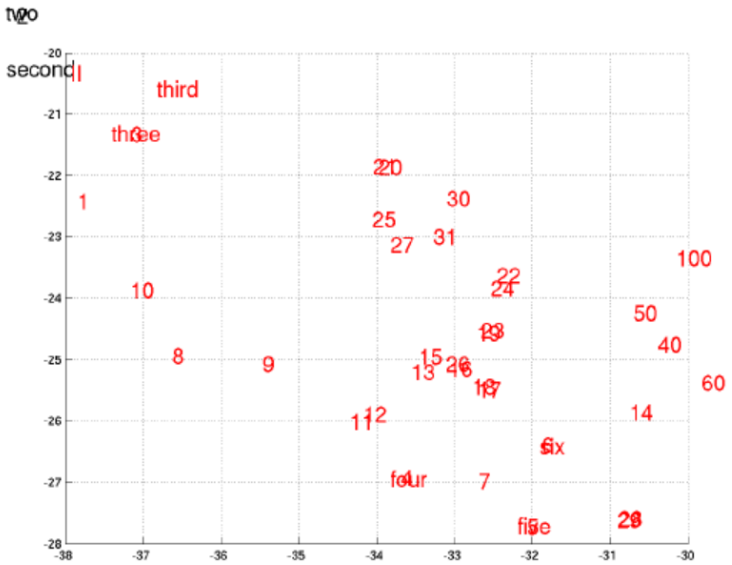


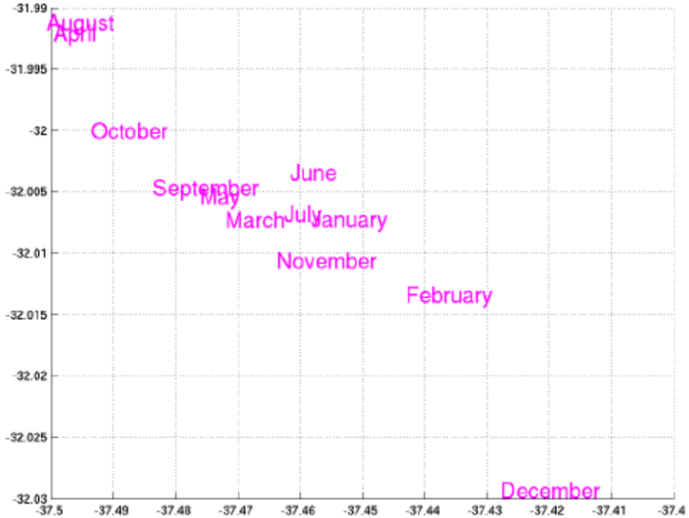
Figure 4: 2–D embedding of the learned word representation. The left one shows the full embedding space, while the right one shows a zoomed-in view of one region (color–coded). For more plots, see the supplementary material.

Experiment




Federal
France
China
Russian
French
English
Germany
Iraq
Ontario
Japan
British
North
South


two
second
third
three
one
ten
six
four
five


August
April
October
September
May
June
March
July
January
November
February
December

Experiment

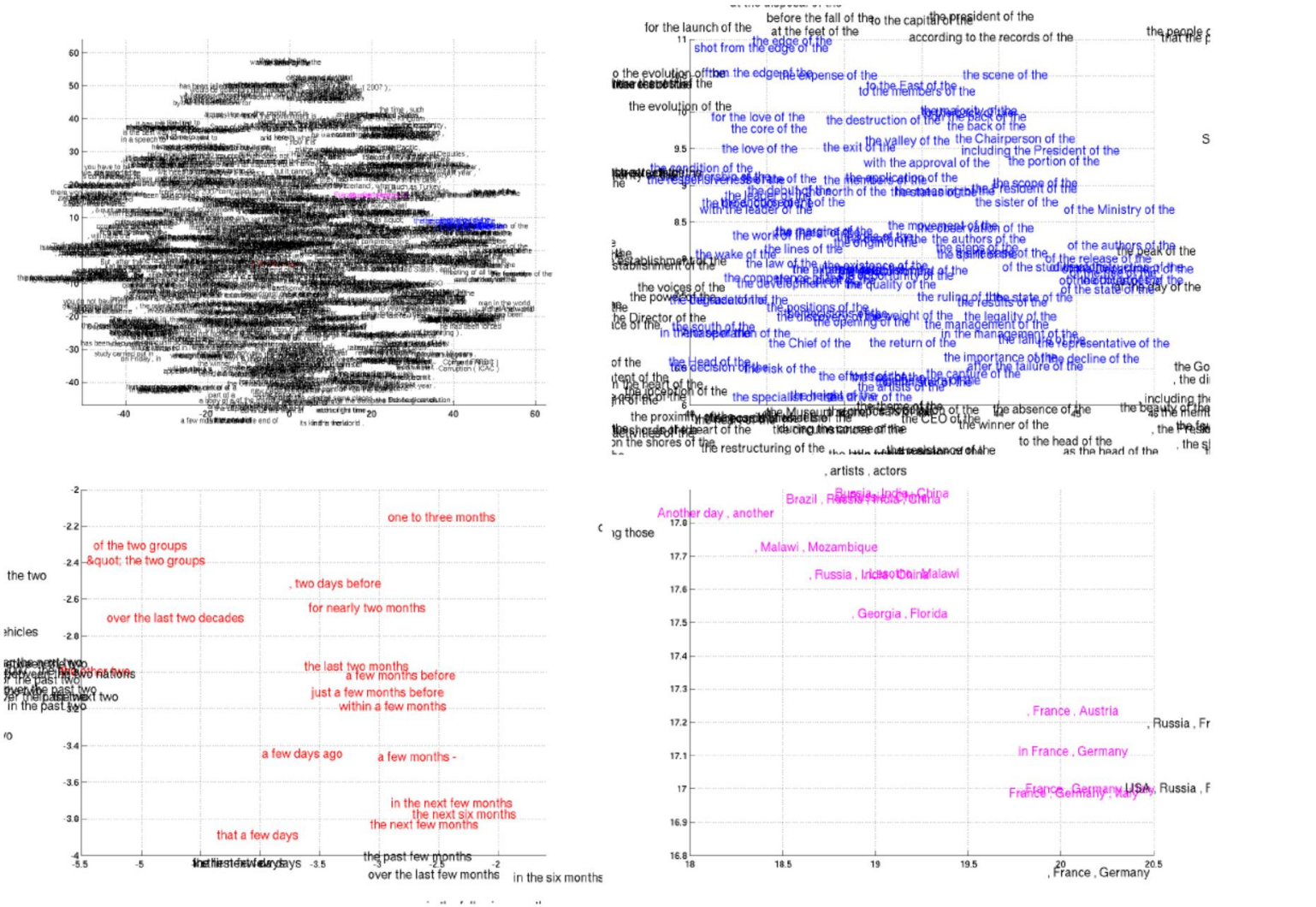


Figure 7: 2–D embedding of the learned phrase representation. The top left one shows the full representation space (1000 randomly selected points), while the other three figures show the zoomed-in view of specific regions (color–coded).

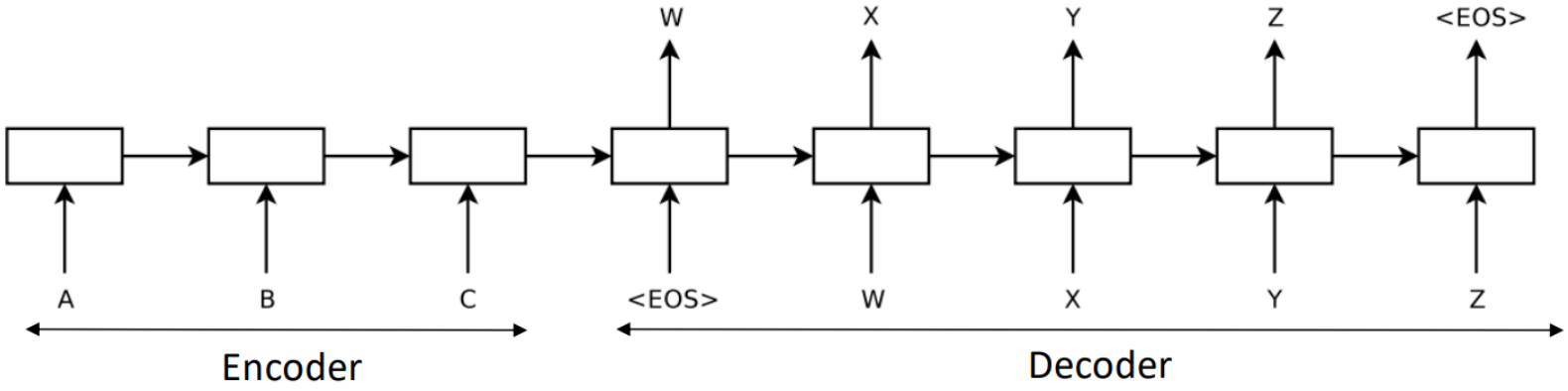
Conclusion

- Developed RNN Encoder-Decoder: mapping from a sequence to another sequence.
- Propose GRU
- New model is able to capture linguistic regularities in the phrase pairs well and it can propose well-formed target phrases.
- It improve the overall translation performance in terms of BLEU score

What is Attention

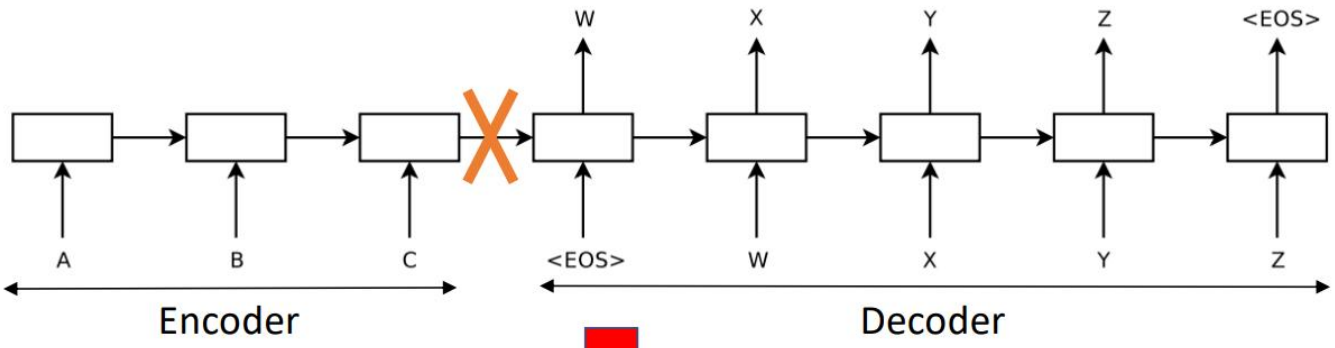
- Seq2Seq Model

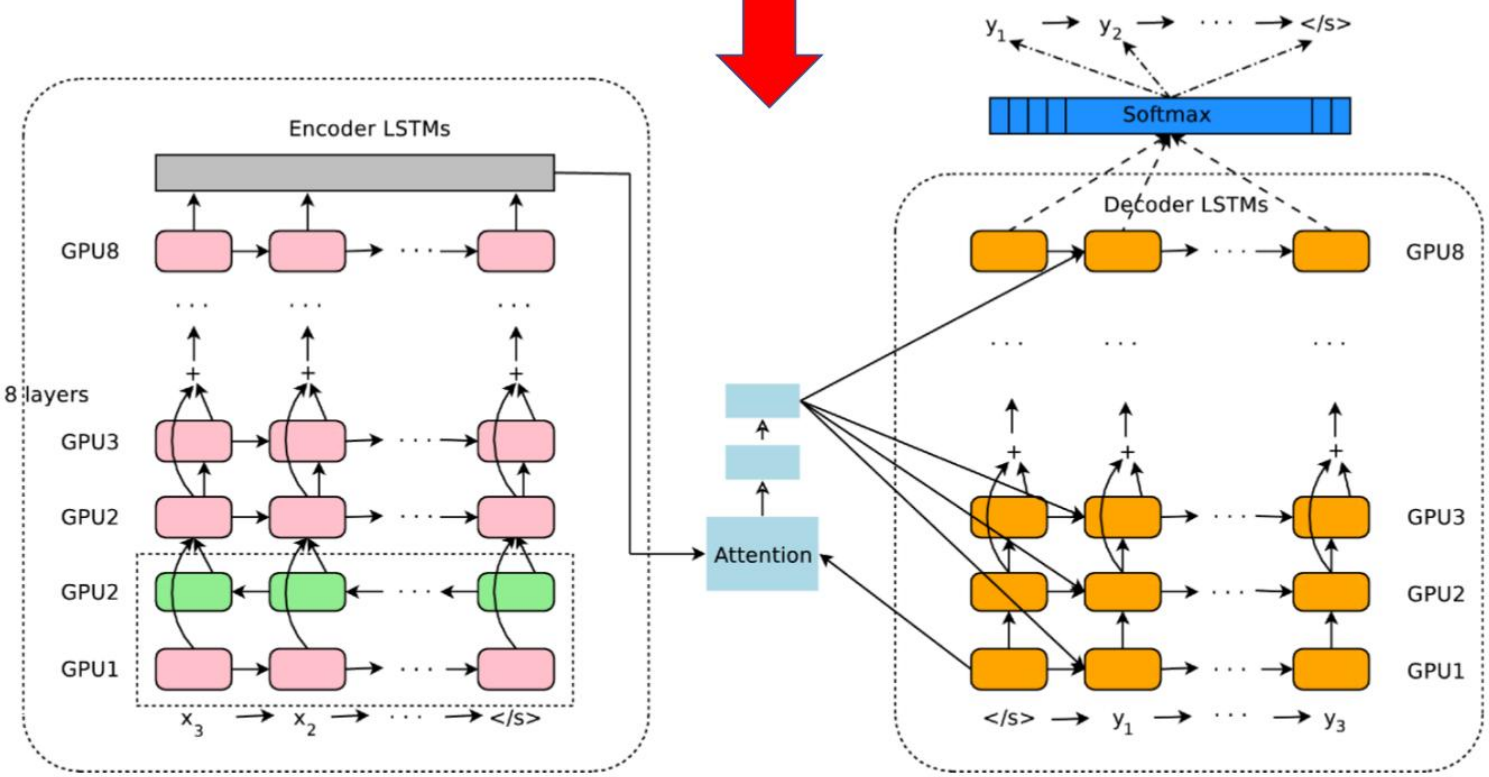
- A neural network that transforms a sequence of elements (words, etc.) to another sequence.
- Use: neural machine translation, chatbot, speech recognition, etc.
- The first seq2seq model was introduced in Sutskever et al. (2014) and purely used RNN.



Note: The model shown above uses only **2 RNN cells**. The rectangular boxes illustrate different **states** of **unrolled** RNN cells.

How to compute output using a context Vector





Google's Neural Machine Traslation System
Wu et al. (2016)

What is Attention

- Concept

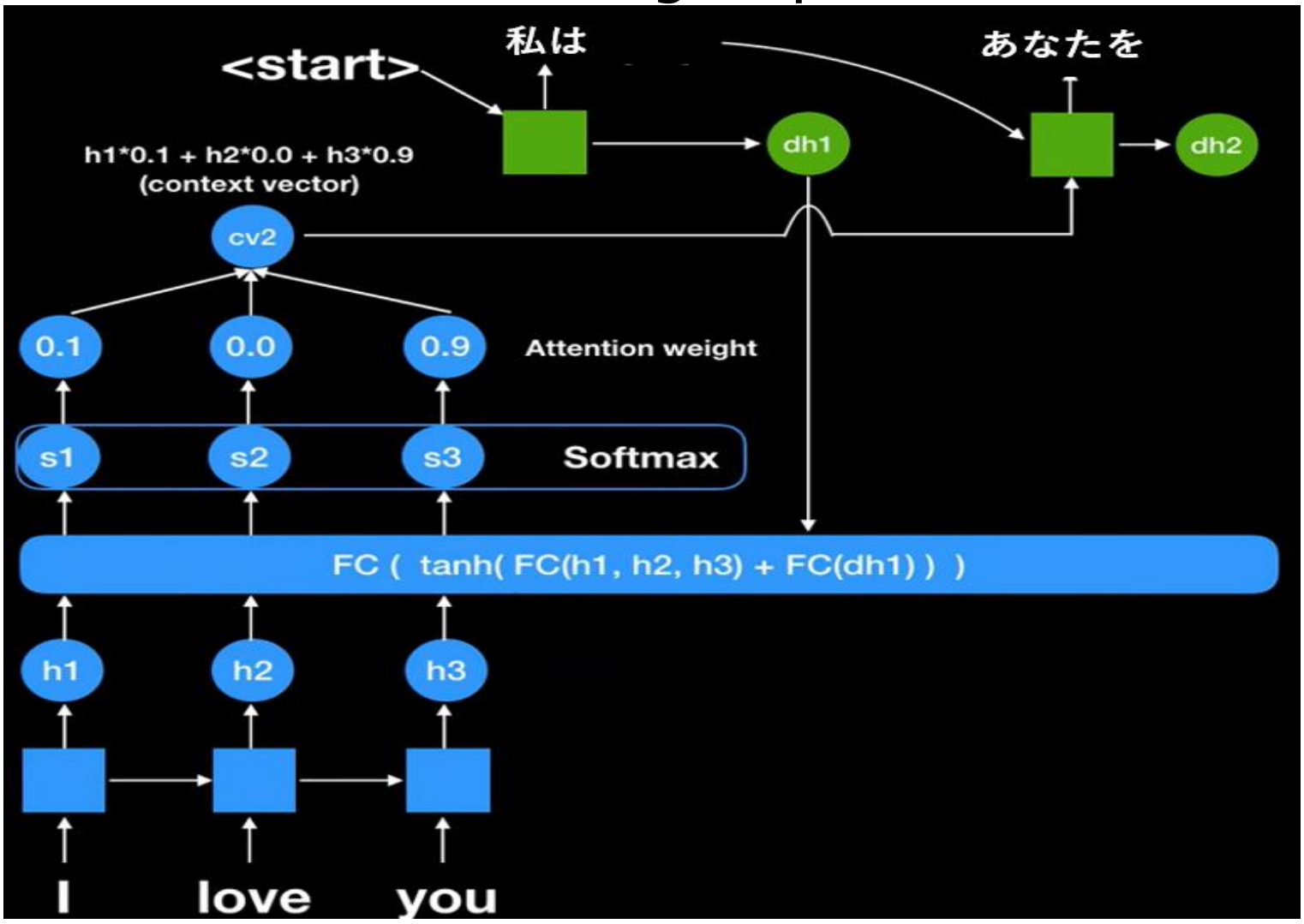
- Attention mechanism was introduced in Bahdanau et al. (2015)
- Use: to jointly *align* and *translate* words.
- *Alignment score* describes to what extent each source word is relevant to a target word.
- Two major Attention mechanisms:

a. Bahdanau Attention (additive)

b. Luong Attention (multiplicative) introduced in Luong et al. (2015)

What is Attention

- Attention in Seq2Seq Model: To overcome problem of information loss of long sequence



What is Attention

- Effect of Attention (Bahdanau Attention)

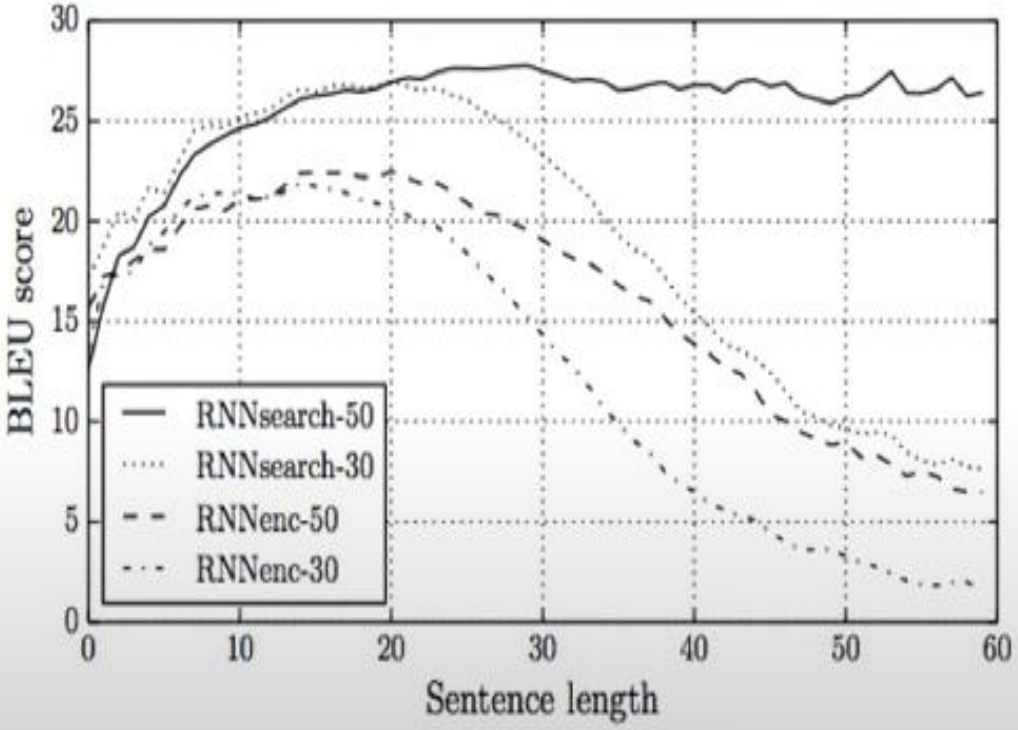


Figure 2: The BLEU scores of the generated translations on the test set with respect to the lengths of the sentences. The results are on the full test set which includes sentences having unknown words to the models.

Global Attention

- Global Attention Model

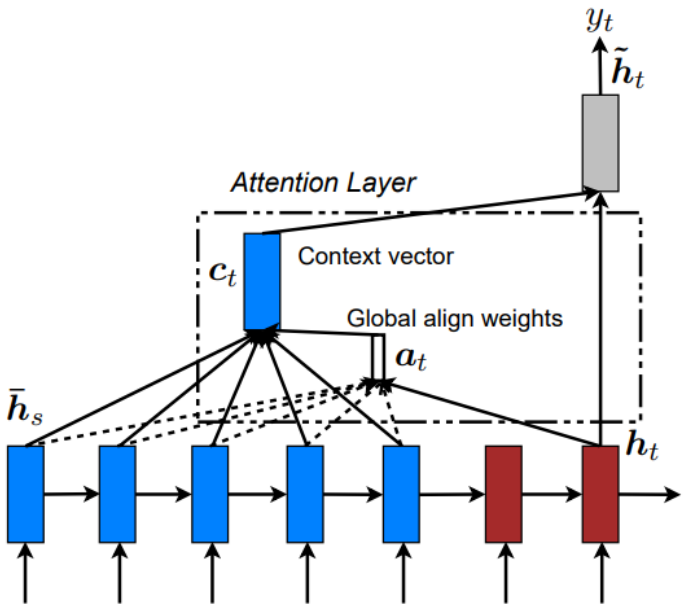


Figure 2: **Global attentional model** – at each time step t, the model infers a *variable-length* alignment weight vector $\boldsymbol{a}_t$ based on the current target state $\boldsymbol{h}_t$ and all source states $\bar{\boldsymbol{h}}_s$. A global context vector $\boldsymbol{c}_t$ is then computed as the weighted average, according to $\boldsymbol{a}_t$, over all the source states.

Specifically, given the target hidden state $\boldsymbol{h}_t$ and the source-side context vector $\boldsymbol{c}_t$, we employ a simple concatenation layer to combine the information from both vectors to produce an attentional hidden state as follows:

$$\tilde{\boldsymbol{h}}_t = \tanh(\boldsymbol{W_c}[\boldsymbol{c}_t; \boldsymbol{h}_t]) \quad (5)$$

The attentional vector $\tilde{\boldsymbol{h}}_t$ is then fed through the softmax layer to produce the predictive distribution formulated as:

$$p(y_t|y_{<t}, x) = \text{softmax}(\boldsymbol{W_s}\tilde{\boldsymbol{h}}_t) \quad (6)$$

The idea of a global attentional model is to consider all the hidden states of the encoder when deriving the context vector c_t. In this model type, a variable-length alignment vector $\boldsymbol{a}_t$, whose size equals the number of time steps on the source side, is derived by comparing the current target hidden state $\boldsymbol{h}_t$ with each source hidden state $\bar{\boldsymbol{h}}_s$:

$$\begin{aligned} \boldsymbol{a}_t(s) &= \text{align}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) \quad (7) \\ &= \frac{\exp\left(\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s)\right)}{\sum_{s'} \exp\left(\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_{s'})\right)} \end{aligned}$$

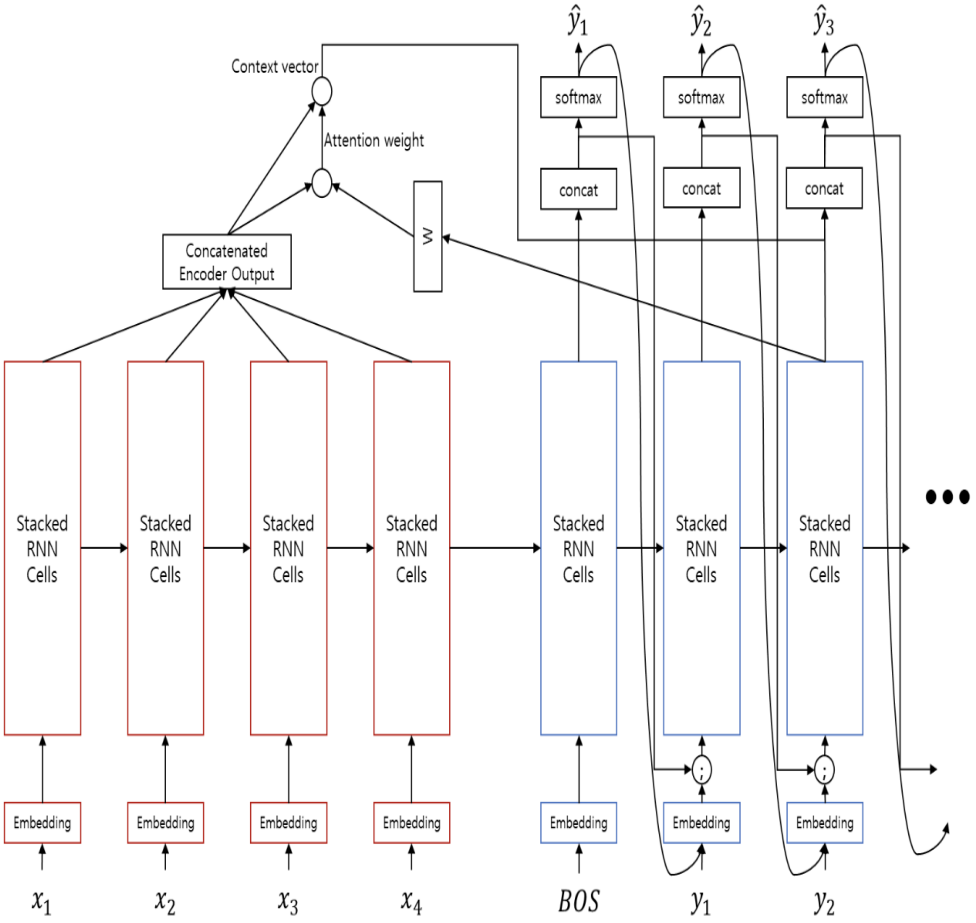
Here, score is referred as a *content-based* function for which we consider three different alternatives:

$$\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \begin{cases} \boldsymbol{h}_t^\top \bar{\boldsymbol{h}}_s & \textit{dot} \\ \boldsymbol{h}_t^\top \boldsymbol{W_a} \bar{\boldsymbol{h}}_s & \textit{general} \\ \boldsymbol{W_a}[\boldsymbol{h}_t; \bar{\boldsymbol{h}}_s] & \textit{concat} \end{cases} \quad (8)$$

$$\boldsymbol{a}_t = \text{softmax}(\boldsymbol{W_a}\boldsymbol{h}_t) \quad \textit{location} \quad (9)$$

Input Feeding

- There is information loss in the softmax phase.
- Not only feeding y_t to the next time-step, but also, feeding concatenation layer output.



$$h_t^{src} = RNN_{enc}(emb_{src}(x_t), h_{t-1}^{src})$$

$$H^{src} = [h_1^{src}; h_2^{src}; \cdots; h_n^{src}]$$

$$h_t^{tgt} = RNN_{dec}([emb_{tgt}(y_{t-1}); \tilde{h}_{t-1}^{tgt}], h_{t-1}^{tgt}) \ \ where\ h_0^{tgt} = h_n^{src} and\ y_0 = BOS$$

$$w = softmax(h_t^{tgtT} W \cdot H^{src})$$

$$c = H^{src} \cdot w \quad and\ c\ is\ a\ context\ vector$$

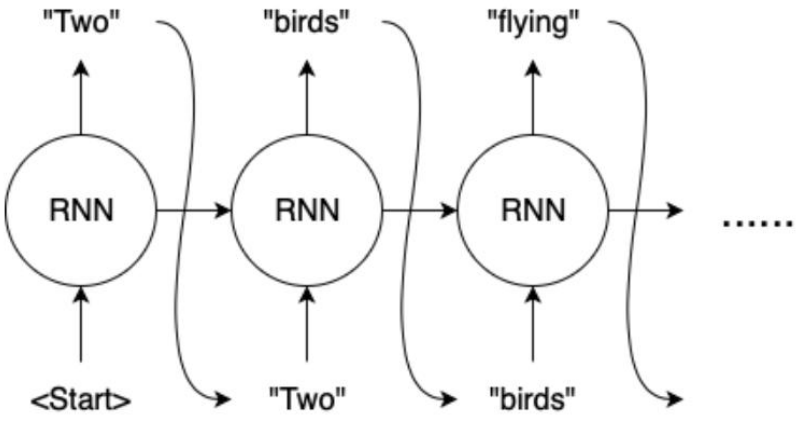
$$\tilde{h}_t^{tgt} = \tanh(linear_{2hs \to hs}([h_t^{tgt}; c]))$$

$$\hat{y}_t = softmax(linear_{hs \to |V_{tgt}|}(\tilde{h}_t^{tgt}))$$

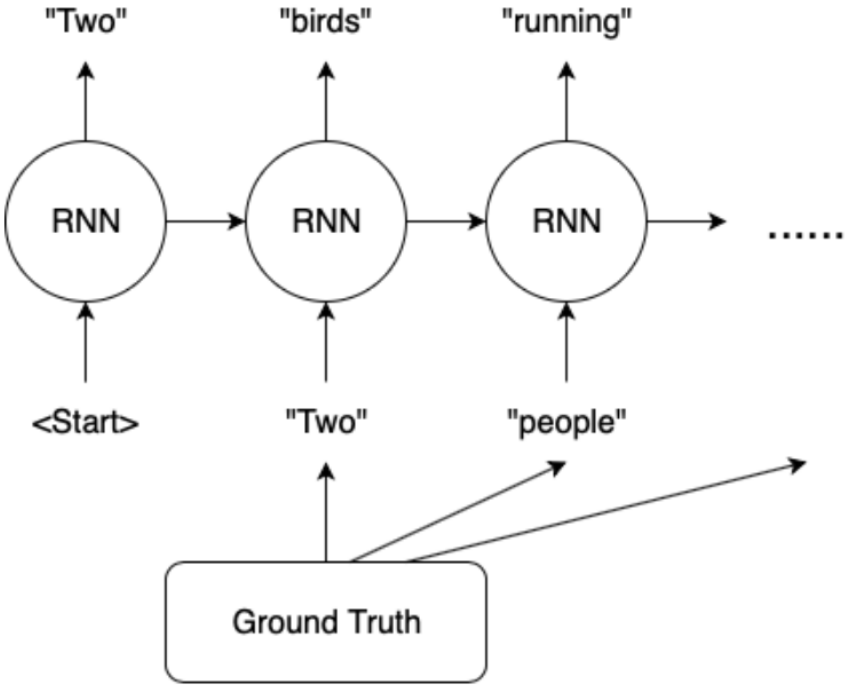
$where\ hs\ is\ hidden\ size\ of\ RNN,\ and\ |V_{tgt}|\ is\ size\ of\ output\ vocabulary.$

Teacher Forcing Method

- Wrong past inference value gives wrong input in the next time-step.
- In the training, for the decoder input, real Y value is to be input instead of decoder output at the previous time-step.

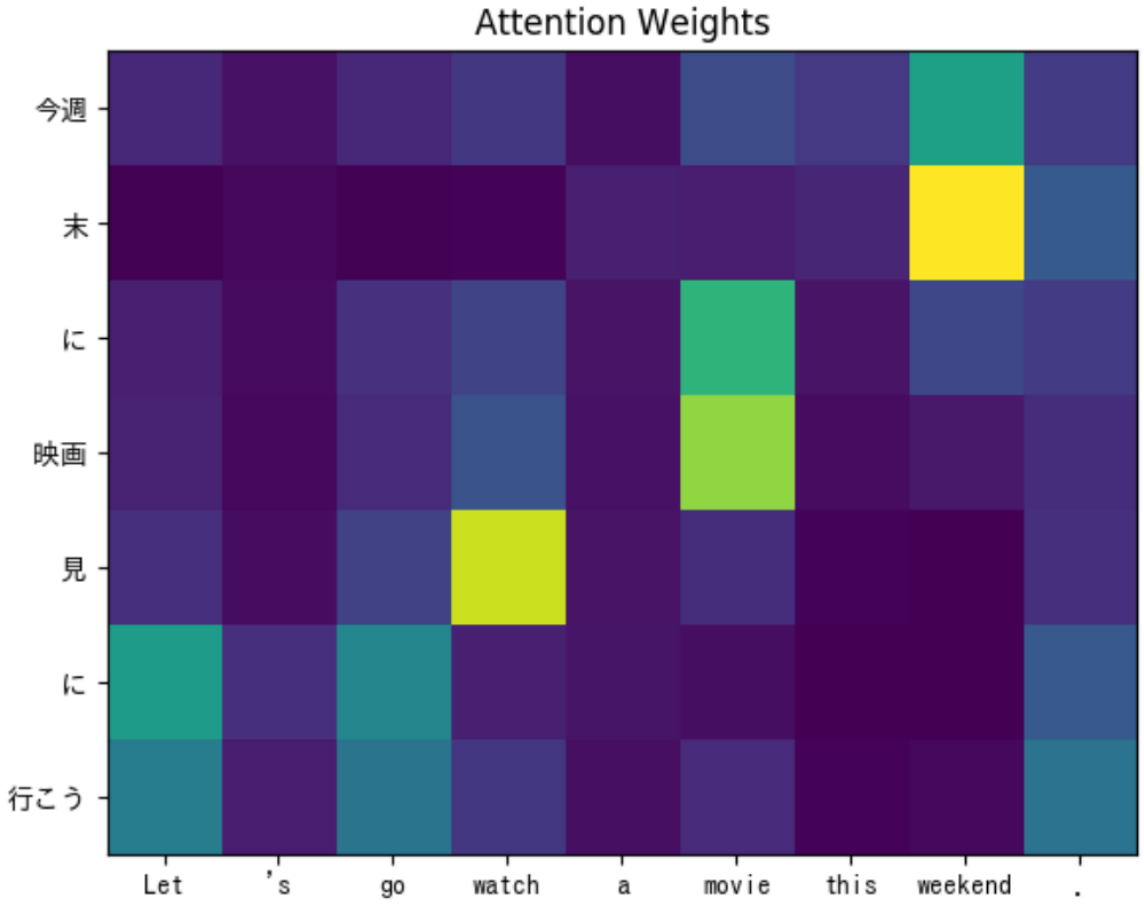


Without Teacher Forcing



With Teacher Forcing

Attention Visualization



- Attention/Alignment weights are obtained by applying *softmax* to alignment scores.
- The lighter the color, the higher the attention weight is.

Attention mask

```
# Step 2
v = tf.get_variable(name="attention_variable", shape=512,
dtype=tf.float32)
scores = tf.reduce_sum(v * tf.tanh(keys + tf.expand_dims(query, 1)),
axis=[2]) # shape [32, 10]

# Step 3
alignment_weights = tf.nn.softmax(scores + attention_mask) # shape
[32, 10]
# alignment_weights can be used to visualize attention
context = tf.matmul(tf.expand_dims(alignment_weights, 1), values) #
shape [32, 1, 512]
context = tf.squeeze(context) # shape [32, 512]
```

Bahdanau Attention

$$score(s_{t-1},\ h_i) = W_a^T\ tanh(W_b s_{t-1} + W_c h_i)$$

Attention Mask

Word sequences	ID sequences	Length	Padded sequences
Hello .	7, 3	2	7, 3, 0, 0, 0, 0, 0, 0, 0, 0
I am a student at the university of Aizu .	4, 8, 5, 10, 31, 17, 28, 9, 11, 3	10	4, 8, 5, 10, 31, 17, 28, 9, 11, 3
I have a pen .	4, 20, 5, 81, 3	5	4, 20, 5, 81, 3, 0, 0, 0, 0, 0

- When grouping sentences into batches, it is mandatory to pad shorter sequences with sequences of a number (0 is commonly used) in order to feed multiple sequences with different lengths into one tensor.
- These paddings have no meaning, so they should not contribute to the context vector.

=> *alignment weights* corresponding to these paddings must be 0.

=> *alignment scores* corresponding to these paddings must be close to *negative infinity*.

Attention Mask

```
    def get_attention_mask(self, id_seqs, neg_inf=-1e15): # shape:
[batch_size, max_len]
        padded_pos = tf.cast(tf.equal(id_seqs, self.PAD), tf.float32)
        return tf.expand_dims(padded_pos * neg_inf, axis=1) # shape
[batch_size, 1, max_len]
```

Padded sequences	Attention Mask
7, 3, 0, 0, 0, 0, 0, 0, 0, 0	0, 0, -∞, -∞, -∞, -∞, -∞, -∞, -∞, -∞
4, 8, 5, 10, 31, 17, 28, 9, 11, 3	0, 0, 0, 0, 0, 0, 0, 0, 0, 0
4, 20, 5, 81, 3, 0, 0, 0, 0, 0	0, 0, 0, 0, 0, -∞, -∞, -∞, -∞, -∞

Luong Attention (General Multiplicative)

$$\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \begin{cases} \boldsymbol{h}_t^\top \bar{\boldsymbol{h}}_s & \textit{dot} \\ \boldsymbol{h}_t^\top \boldsymbol{W}_a \bar{\boldsymbol{h}}_s & \textit{general} \\ \boldsymbol{v}_a^\top \tanh\left(\boldsymbol{W}_a [\boldsymbol{h}_t; \bar{\boldsymbol{h}}_s]\right) & \textit{concat} \end{cases}$$

```
attention_size = 512
k_layer = tf.layers.Dense(units=attention_size, activation=None,
use_bias=False)
v_layer = lambda x : x
q_layer = lambda x : x

values = v_layer(encoder_cell_outputs) # shape [32, 10, 512]
keys = k_layer(values) # shape [32, 10, 512]
query = q_layer(decoder_cell_output) # shape [32, 512]
```

Luong's Attention

- Attention Result Comparison

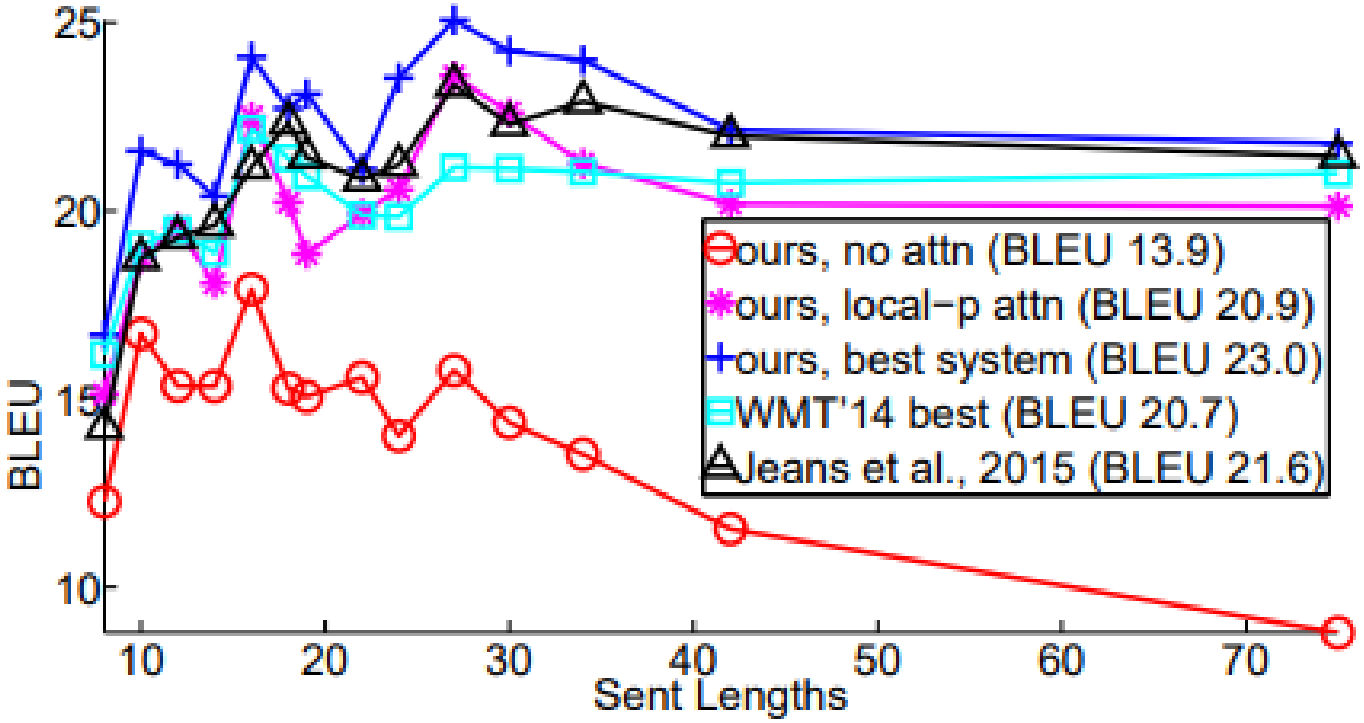


Figure 6: **Length Analysis** – translation qualities of different systems as sentences become longer.

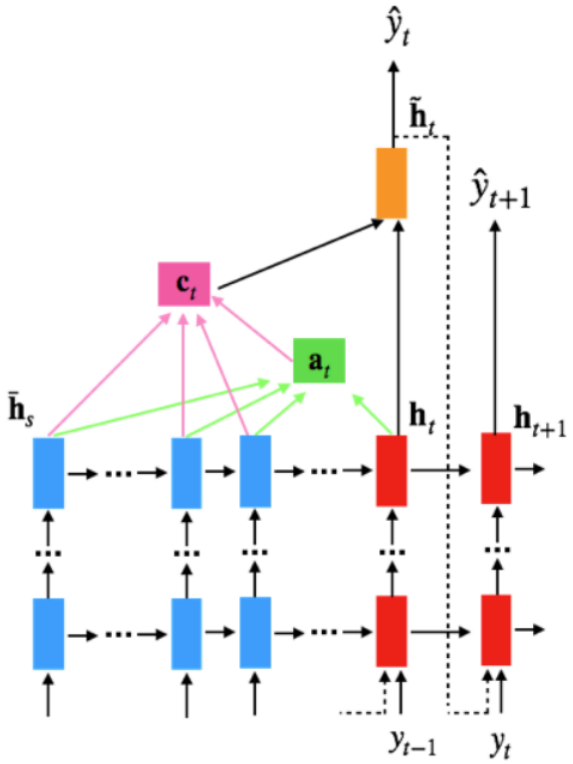
Attentions

Name	Score Function	Defined by
dot	$score(s_t,\ h_i) = s_t^T h_i$	Luong et al. (2015)
$scaled\ dot$	$score(s_t,\ h_i) = \frac{s_t^T h_i}{\sqrt{n}}$	Vaswani et al. (2017)
$general$	$score(s_t,\ h_i) = s_t^T W_a h_i$	Luong et al. (2015)
$concat$	$score(s_t,\ h_i) = W_a^T\ tanh(W_b[s_t; h_i])score(s_t,\ h_i) = W_a^T\ tanh(W_b s_t + W_c h_i)$	Bahdanau et al. (2015)
$location - base$	$\alpha_t = softmax(W_a s_t)$	Luong et al. (2015)

How to compute output using a context Vector

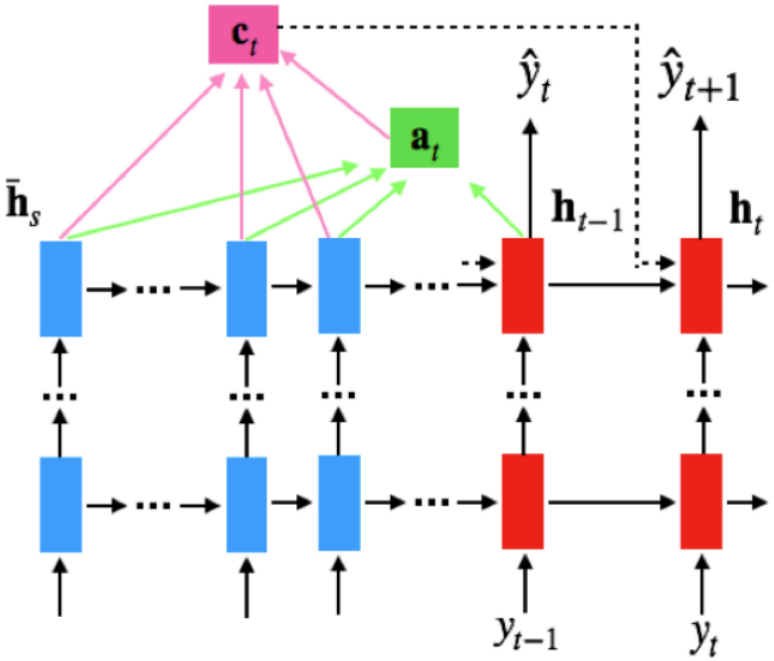
Luong Attention Mechanism

$$\mathbf{h}_t \rightarrow \mathbf{a}_t \rightarrow \mathbf{c}_t \rightarrow \tilde{\mathbf{h}}_t$$



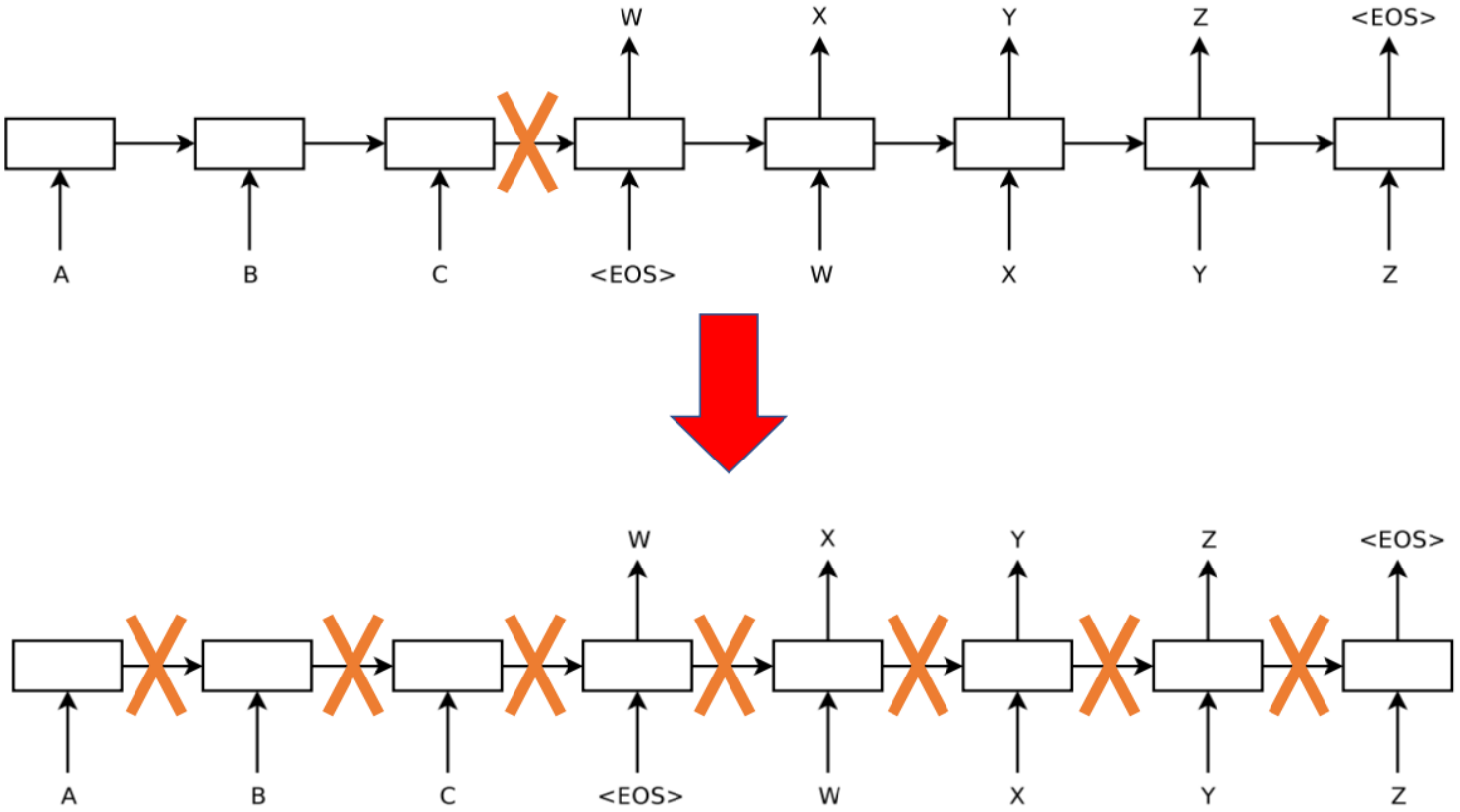
Bahdanau Attention Mechanism

$$\mathbf{h}_{t-1} \rightarrow \mathbf{a}_t \rightarrow \mathbf{c}_t \rightarrow \mathbf{h}_t$$



Self Attention

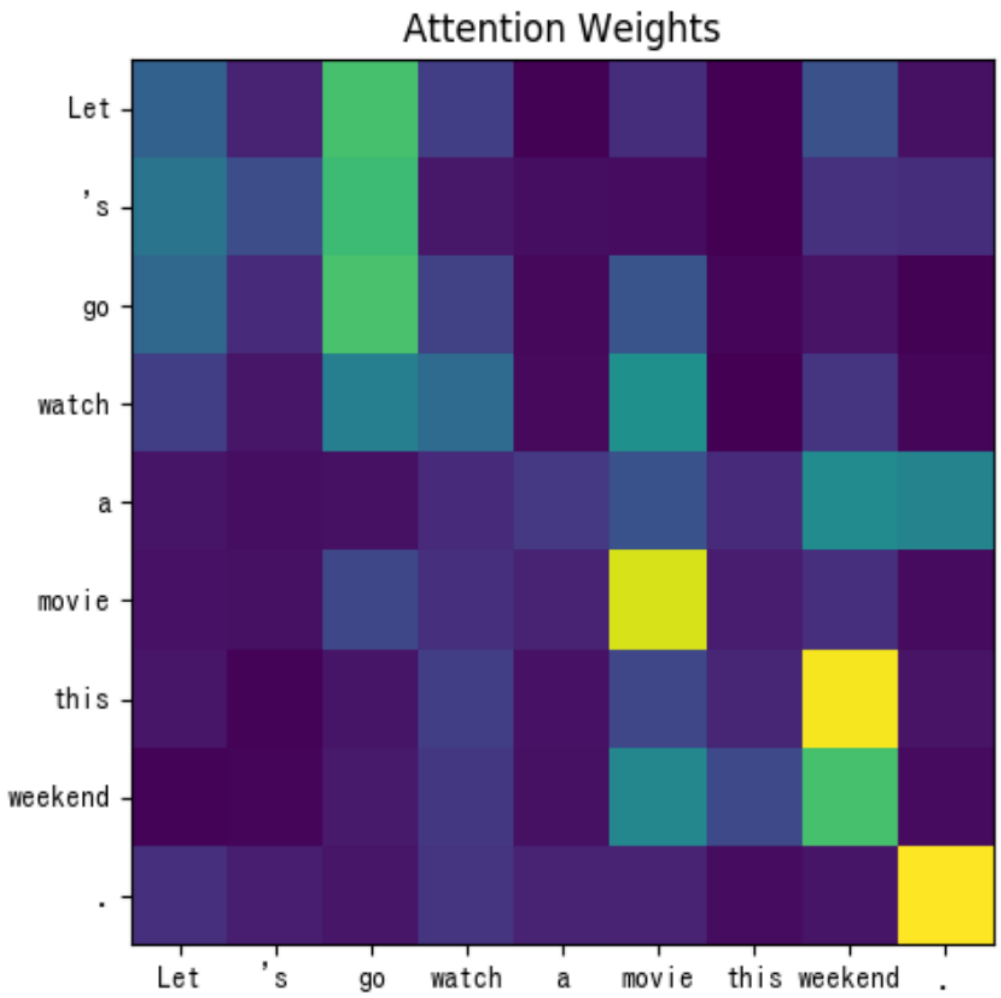
- How about?



It means that there is no RNN at all!

Self Attention

- Keys, values and queries come from the *same* place.
- Encoder: each position attends to all positions of a source word sequence.
- Decoder: each position attends to all positions of target word sequence *up to* and *including* itself.



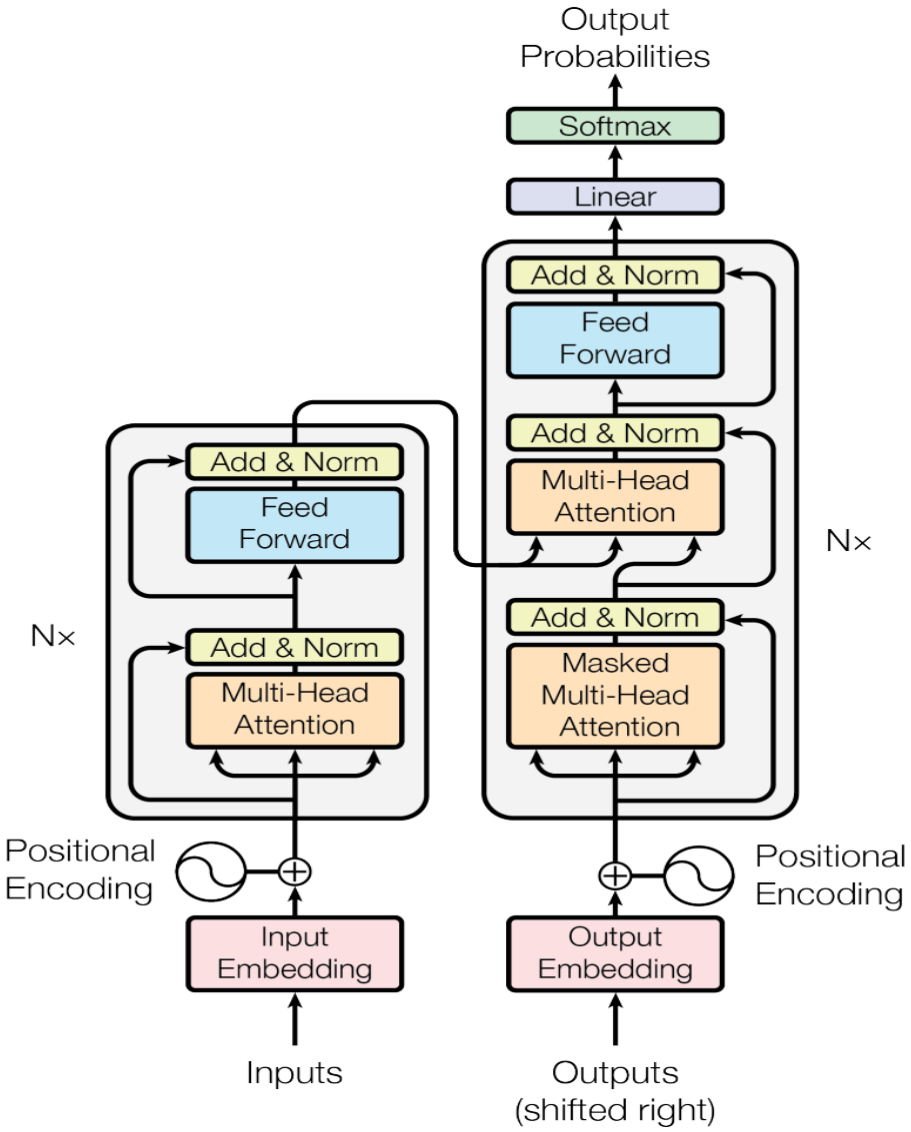
Why does self-attention surpass RNN?

Table 1: Maximum path lengths, per-layer complexity and minimum number of sequential operations for different layer types. n is the sequence length, d is the representation dimension, k is the kernel size of convolutions and r the size of the neighborhood in restricted self-attention.

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

- One key factor affecting the ability to learn long-term dependencies is the length of the paths forward and backward signals have to traverse in the network.
- The shorter these paths, the easier it is to learn long-term dependencies.

Transformer



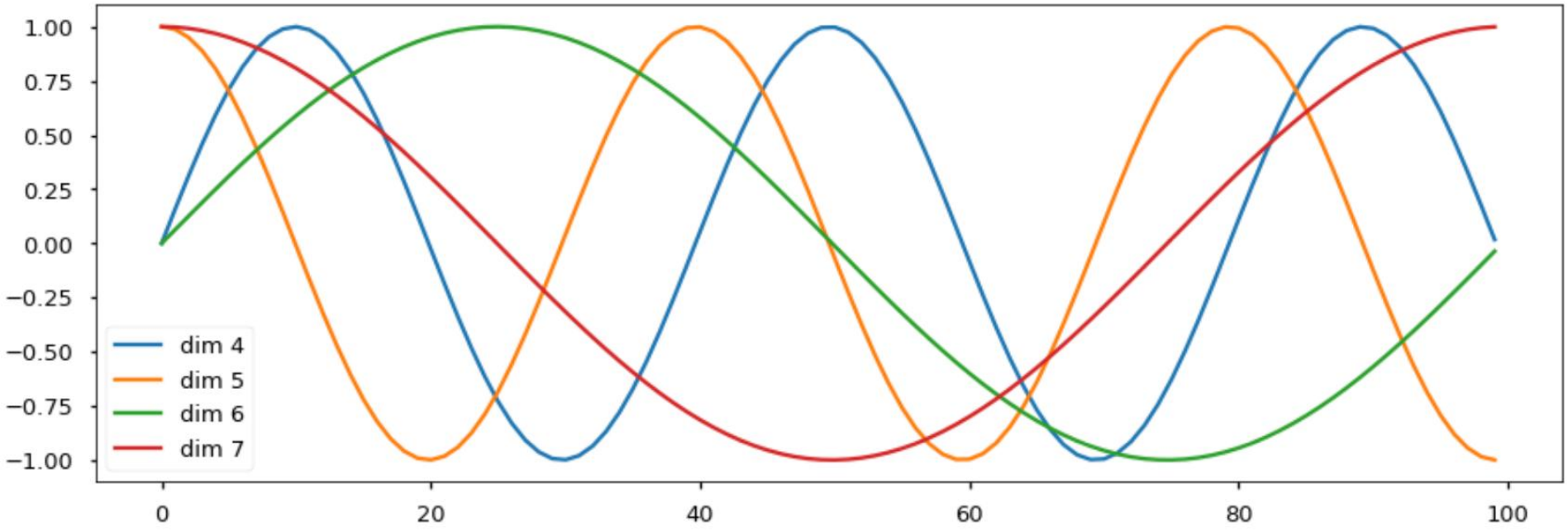
Positional Encoding

- Because no RNN is used, it is important to add some information about the relative or absolute positions of the tokens in each sequence.
- Two choices of positional encodings: learned and fixed.
- Fixed positional encodings using sine and cosine functions:

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

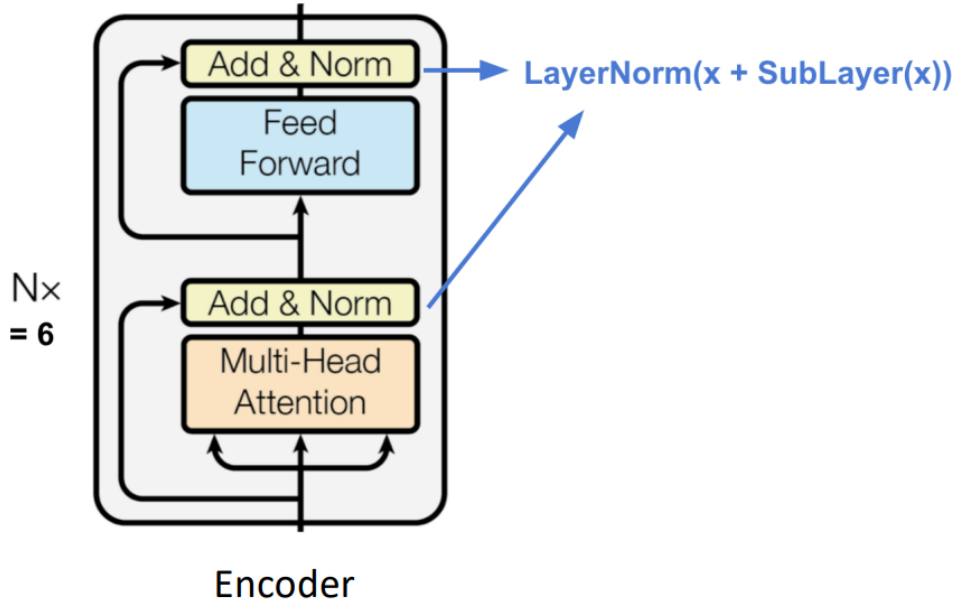
where pos is position, i is dimension index, d_model is the dimension of the model.

Positional Encoding Illustration



Encoder Stack

- A stack of N layers.
- The output of each layer becomes the input of the next layer. The output of the last layer goes to the decoder.
- Each layer has 2 sub-layers that adopt residual connections and layer normalization.
- All sub-layers output data of the same dimension d_model.



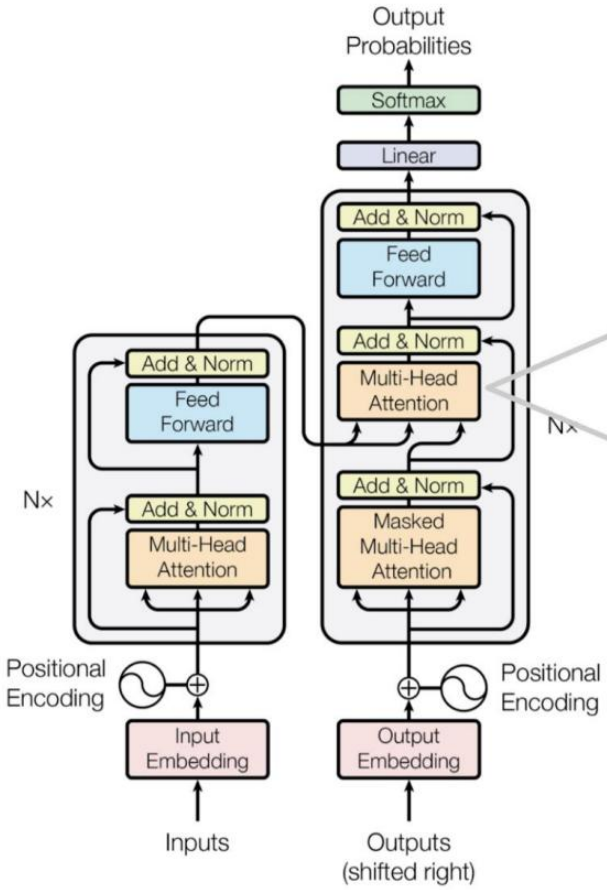
Layer Normalization

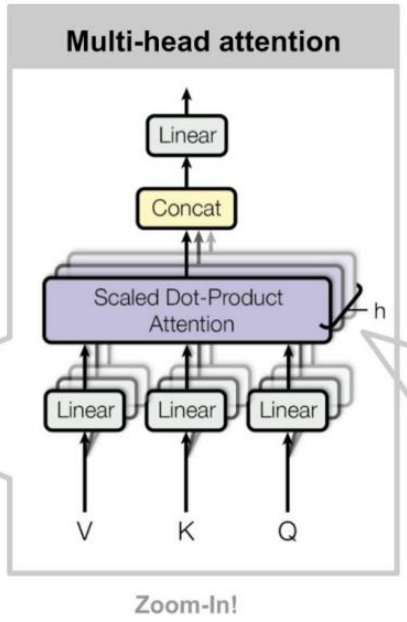
- Reading: Batch Normalization

```
class Layer_Norm:
    def __init__(self):
        self.gain = tf.get_variable(name="norm_gain", initializer=1.0)
        self.bias = tf.get_variable(name="norm_bias", initializer=0.0)
        self.epsilon = 1e-15

    def __call__(self, x):
        mean, var = tf.nn.moments(x, [-1], keep_dims=True)
        norm = (x - mean) * tf.rsqrt(var + self.epsilon)
        norm = self.gain * norm + self.bias
        return norm
```

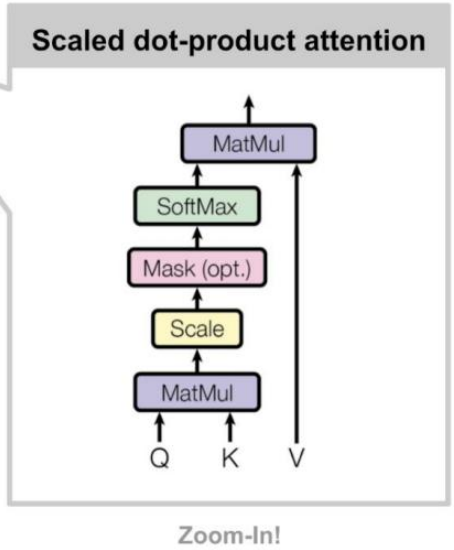
Attention



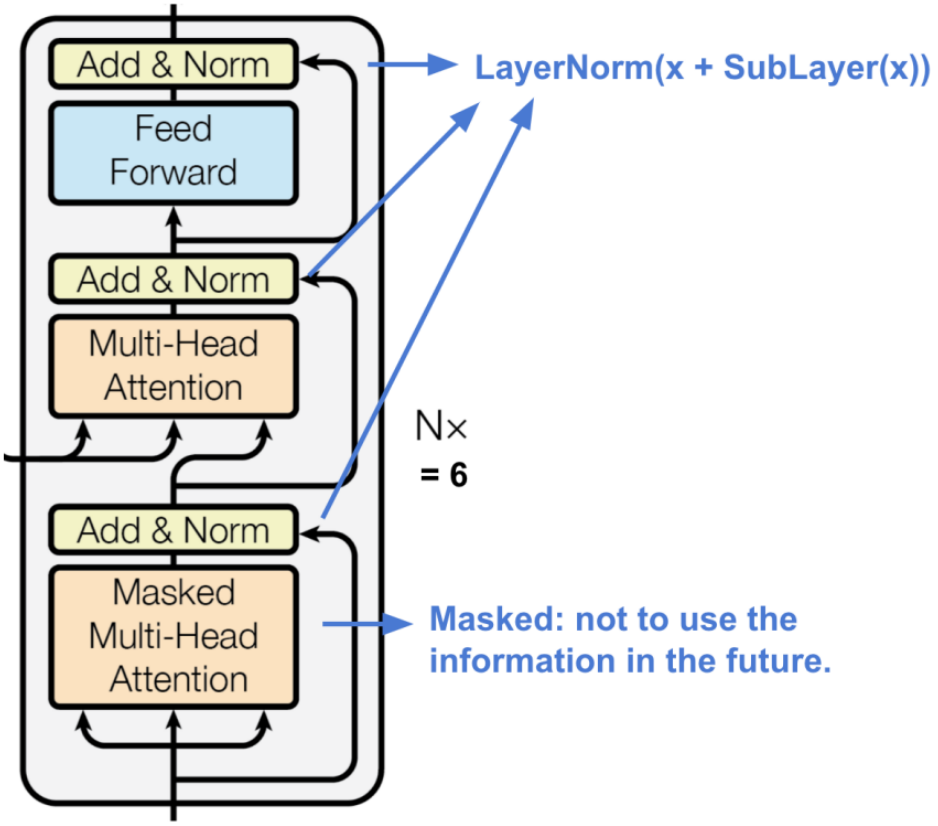


Multi-Head Attention consists of several attention layers running in parallel.

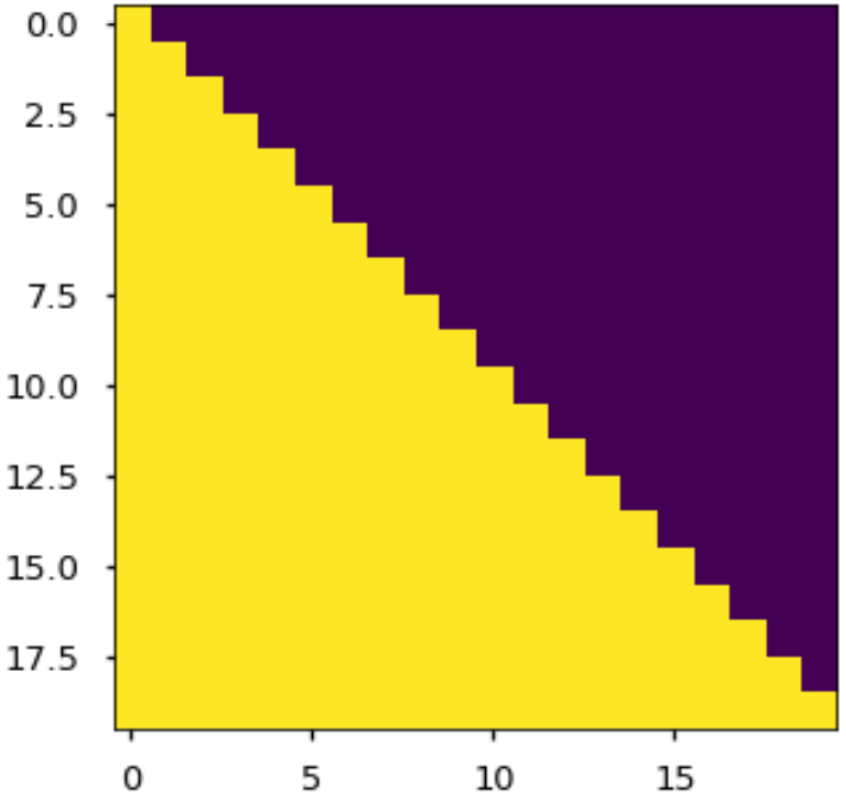
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$



Decoder Stack



Decoder



Decoder attention mask
yellow: 0, purple: -∞

BLEU Score Comparison

Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

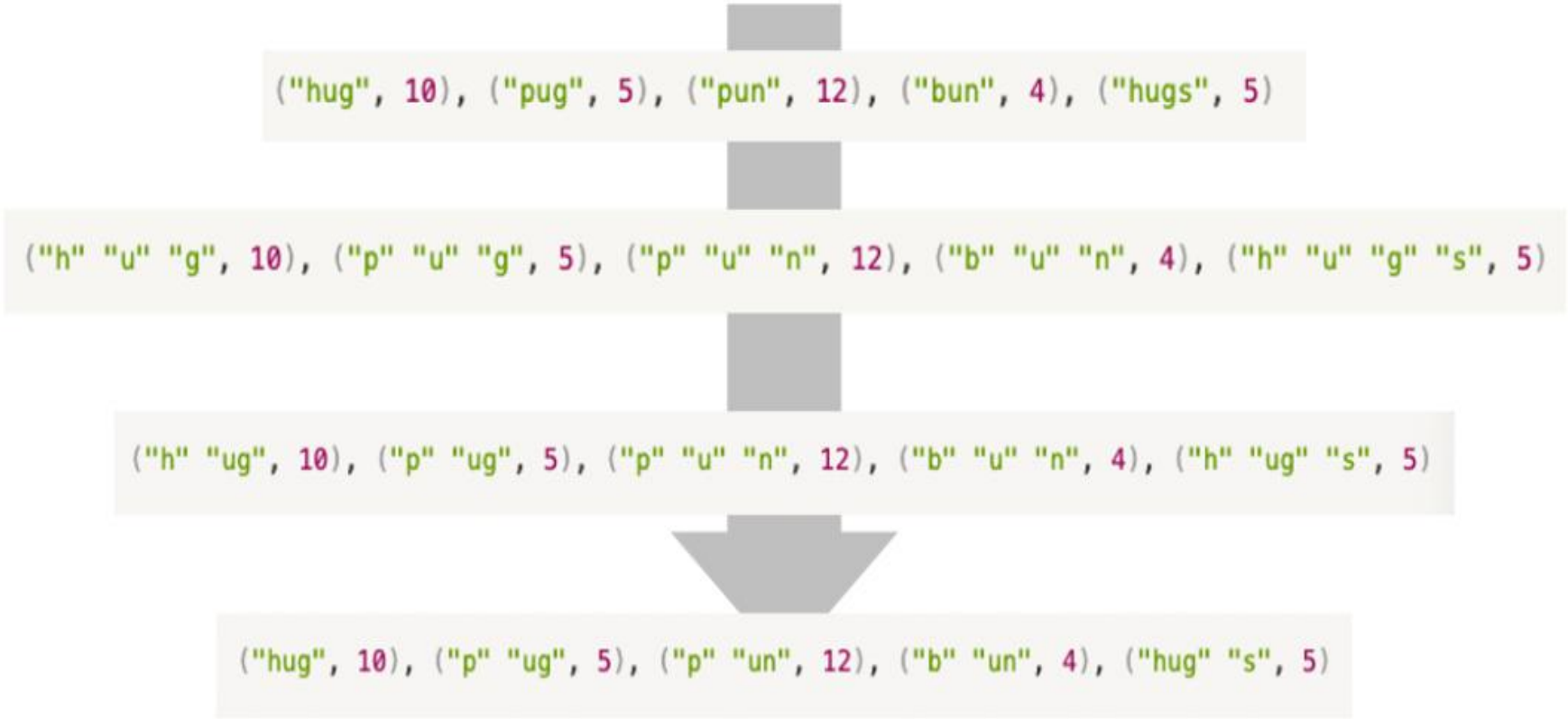
Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	**$3.3 \cdot 10^{18}$**	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	

Byte-Pair Encoding (BPE)

BPE is a simple form of data compression algorithm in which the most common pair of consecutive bytes of data is replaced with a byte that does not occur in that data.

Suppose we have data **aaabdaaabac** which needs to be encoded (compressed). The byte pair **aa** occurs most often, so we will replace it with **Z** as **Z** does not occur in our data. So we now have **ZabdZabac** where **Z = aa**. The next common byte pair is **ab** so let's replace it with **Y**. We now have **ZYdZYac** where **Z = aa** and **Y = ab**. The only byte pair left is **ac** which appears as just one so we will not encode it. We can use recursive byte pair encoding to encode **ZY** as **X**. Our data has now transformed into **XdXac** where **X = ZY, Y = ab,** and **Z = aa**. It cannot be further compressed as there are no byte pairs appearing more than once. We decompress the data by performing replacements in reverse order.

Byte-Pair Encoding (BPE)



Word Embeddings

- Word Embedding Methods
  - Word2Vec
  - Glove
  - BERT Word Embedding

Word Embeddings

- Representations of Words
  - Sparse Representation
    - One-hot coding
    - Cannot describe similarity between words
  - Distributed Representation
    - Distributional hypothesis: Words on similar context have similar meaning.
    - A word can be described in a vector form:
    - (Ex) dog= [0.2 0.3 0.5 0.7 0.2 ... 0.2]
  - Word Embedding Methods
    - Word2Vec
    - Glove
    - BERT Embedding

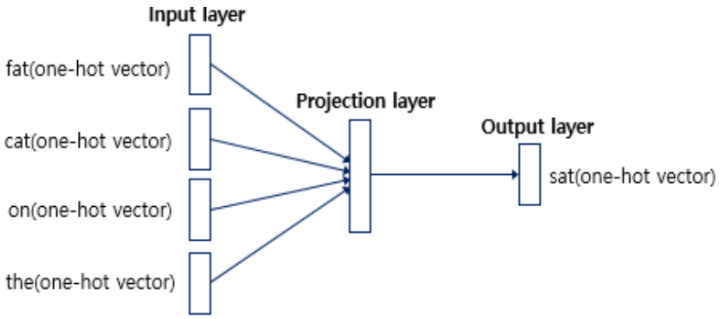
Word2Vec

- Approaches
  - CBOW(Continuous Bag of Words)
    - Predict a center word using input of neighbor words
  - Skip-Gram
    - Predict neighbor words using input of a center word
- CBOW

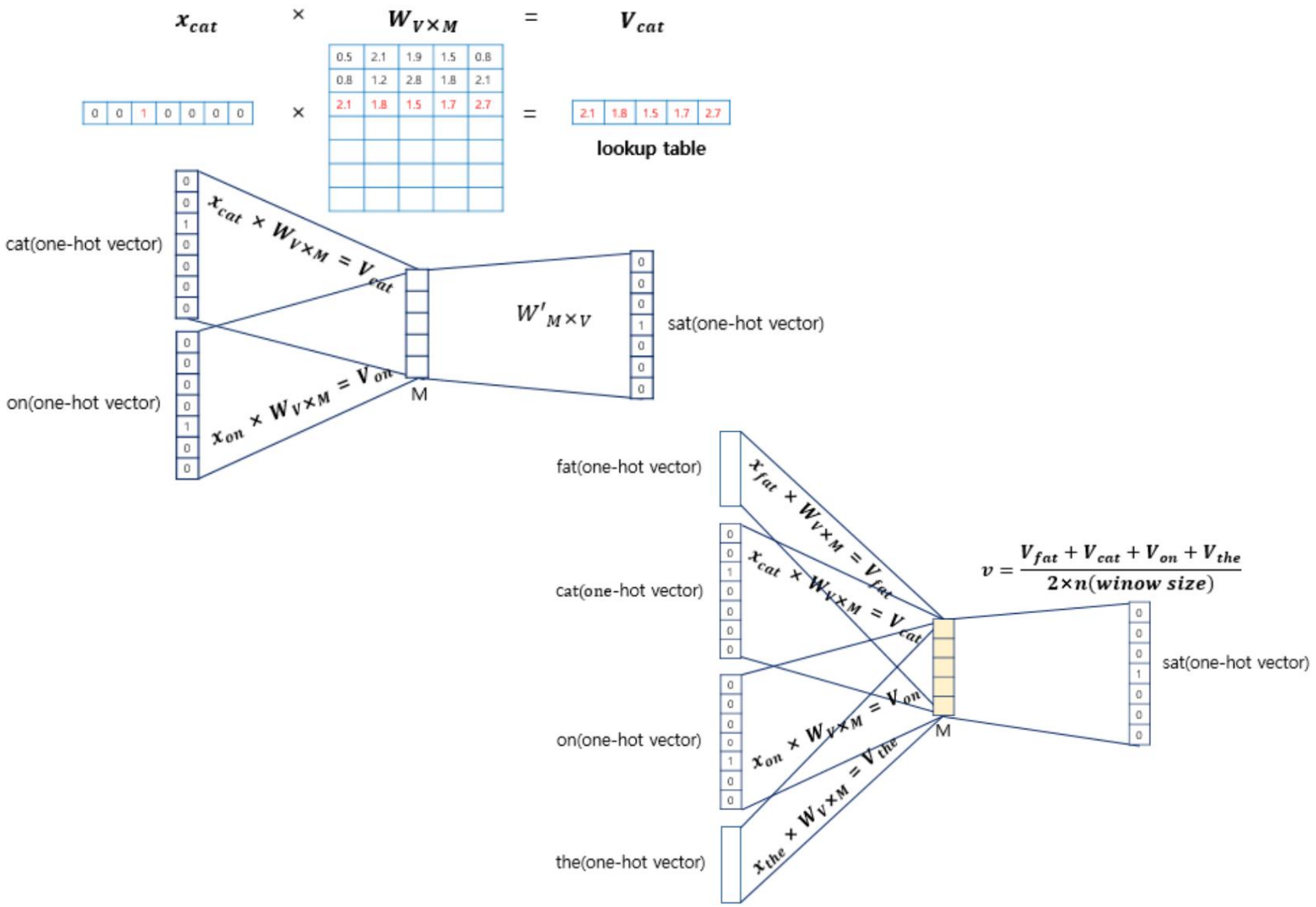
center word — neighbor word

The fat cat sat on the mat
The fat cat sat on the mat
The fat cat sat on the mat
The fat cat sat on the mat
The fat cat sat on the mat
The fat cat sat on the mat
The fat cat sat on the mat

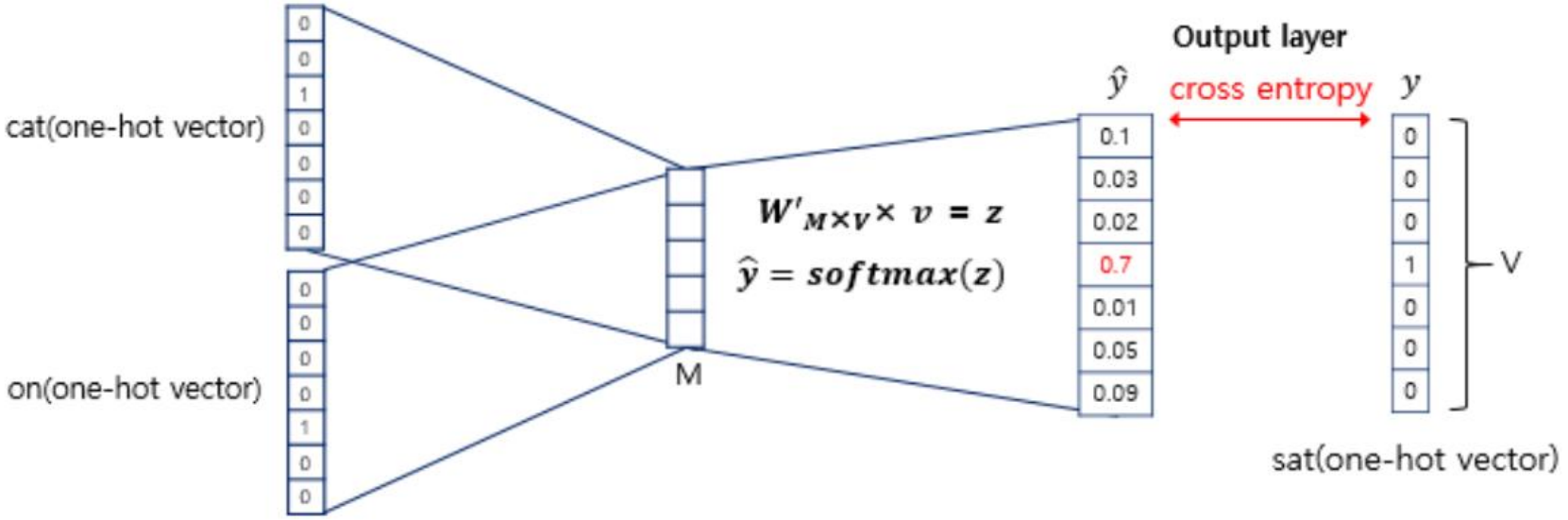
center word	neighbor word
[1, 0, 0, 0, 0, 0]	[0, 1, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0]
[0, 1, 0, 0, 0, 0]	[1, 0, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0], [0, 0, 0, 1, 0, 0]
[0, 0, 1, 0, 0, 0]	[1, 0, 0, 0, 0, 0], [0, 1, 0, 0, 0, 0], [0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 1, 0]
[0, 0, 0, 1, 0, 0]	[0, 1, 0, 0, 0, 0], [0, 0, 1, 0, 0, 0], [0, 0, 0, 0, 1, 0], [0, 0, 0, 0, 0, 1]
[0, 0, 0, 0, 1, 0]	[0, 0, 1, 0, 0, 0], [0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 0, 1], [0, 0, 0, 0, 0, 0]
[0, 0, 0, 0, 0, 1]	[0, 0, 0, 1, 0, 0], [0, 0, 0, 0, 1, 0], [0, 0, 0, 0, 0, 1]
[0, 0, 0, 0, 0, 0]	[0, 0, 0, 0, 1, 0], [0, 0, 0, 0, 0, 1], [0, 0, 0, 0, 0, 0]



Word2Vec (CBOW)



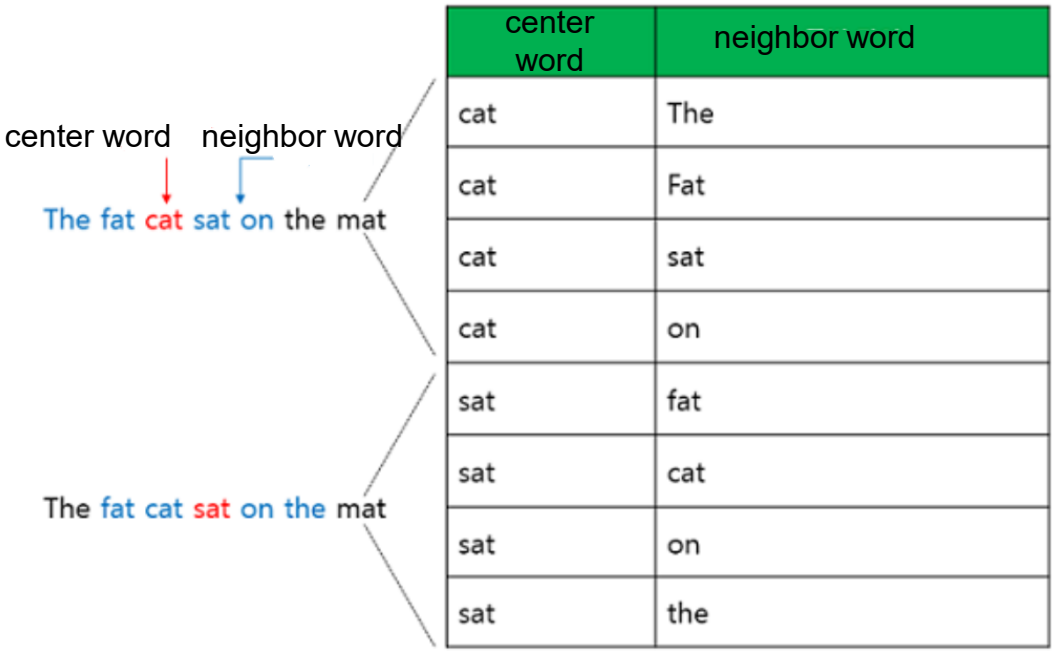
Word2Vec (CBOW)



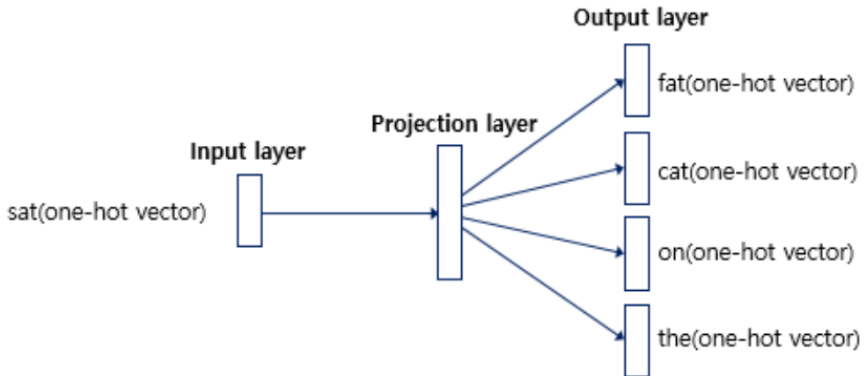
Size of Word Set

$$cost(\hat{y}, y) = -\sum_{j=1}^{V} y_j \, log(\hat{y}_j)$$

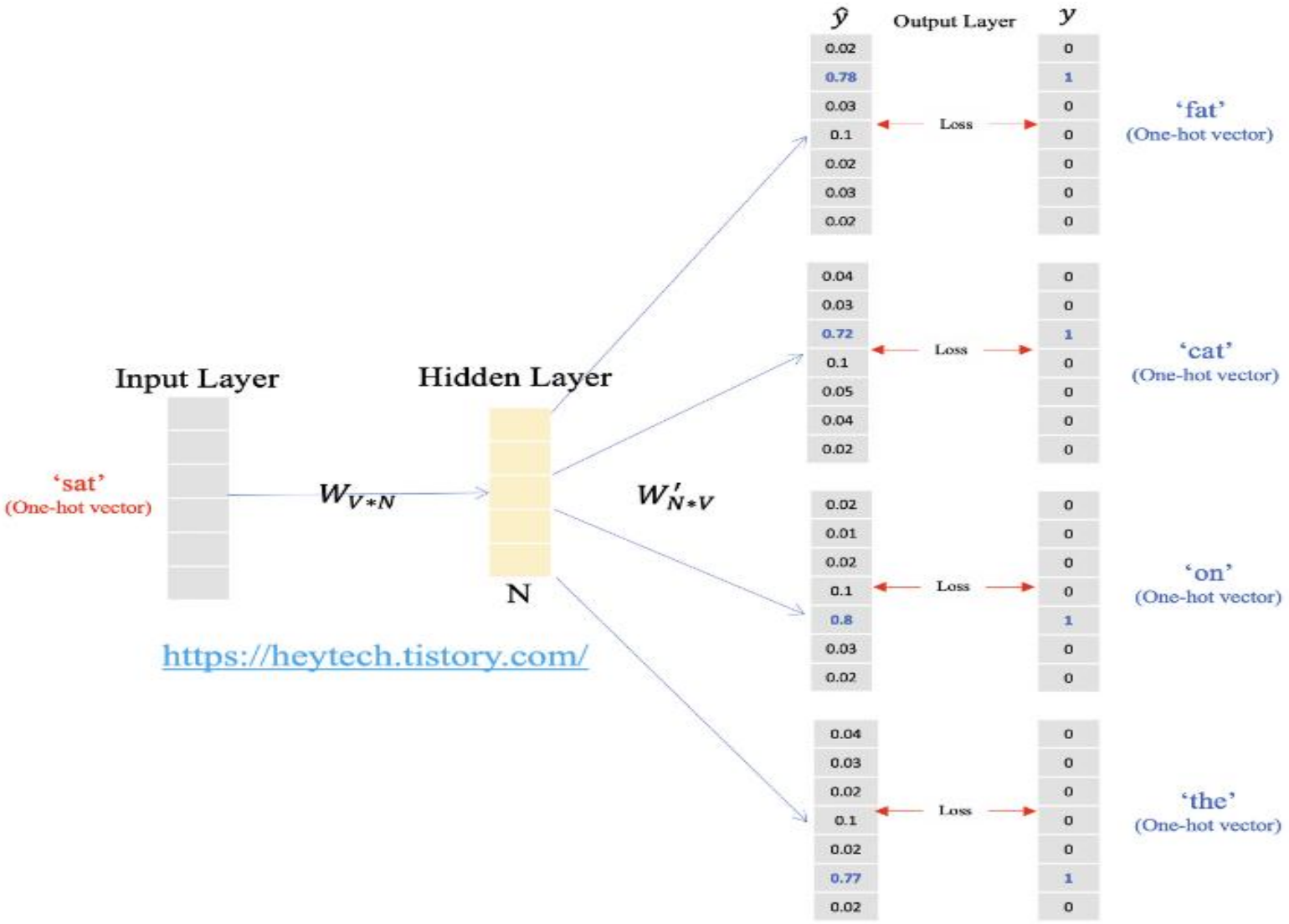
Word2Vec (SKIP Gram)



center word	neighbor word
cat	The
cat	Fat
cat	sat
cat	on
sat	fat
sat	cat
sat	on
sat	the



SKIP Gram – Learning Process



GloVe (Global Vectors for Word Representation)

- The distributional word representation is learned from count-based methods.
- co-occurrence statistics of corpus.

CO-OCCURRENCE PROBABILITY

$$P_{ij} = P(j|i) = \frac{X_{ij}}{X_i} = \frac{X_{ij}}{\sum_k X_{ik}},$$

X_{ij} = number of times word j occurs in the context of word i.

- https://towardsdatascience.com/emnlp-what-is-glove-part-ii-9e5ad227ee0

GloVe

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

- P(solid|ice)/P(solid|steam) = 8.9
- P(gas|ice)/P(gas|steam) = 0.0085

● Goal:

When a special word K is given, inner product of embedded two words to be proposition of probability of co-occurance of the two words.

GloVe

- Finding function F

NAIVE MODEL

$$F(w_i, w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}.$$

VECTOR DIFFERENCE MODEL

$$F(w_i - w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}.$$

- https://towardsdatascience.com/emnlp-what-is-glove-part-ii-9e5ad227ee0

$$F(w_{ice}, w_{steam}, w_{solid}) = \frac{P_{ice,solid}}{P_{steam,solid}} = \frac{P(solid|ice)}{P(solid|steam)} = \frac{1.9 \times 10^{-4}}{2.2 \times 10^{-5}} = 8.9$$

GloVe

- Changing the function F

$$F(w_i - w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}$$

$$F((w_i - w_j)^T \tilde{w}_k) = \frac{P_{ik}}{P_{jk}}$$

$$F((w_i - w_j)^T \tilde{w}_k) = \frac{F(w_i^T \tilde{w}_k)}{F(w_j^T \tilde{w}_k)}$$

$$F(w_i^T \tilde{w}_k - w_j^T \tilde{w}_k) = \frac{F(w_i^T \tilde{w}_k)}{F(w_j^T \tilde{w}_k)}$$

- F must satisfy the condition – Exponential function

$$w_i \longleftrightarrow \tilde{w}_k$$

$$X \longleftrightarrow X^T$$

$$F(X - Y) = \frac{F(X)}{F(Y)}$$

GloVe

- Exp → F

$$exp(w_i^T \tilde{w}_k - w_j^T \tilde{w}_k) = \frac{exp(w_i^T \tilde{w}_k)}{exp(w_j^T \tilde{w}_k)}$$

$$w_i^T \tilde{w}_k = \log P_{ik} = \log X_{ik} - \log X_i$$

$$w_i^T \tilde{w}_k = \log X_{ik} - b_i - \tilde{b_k}$$

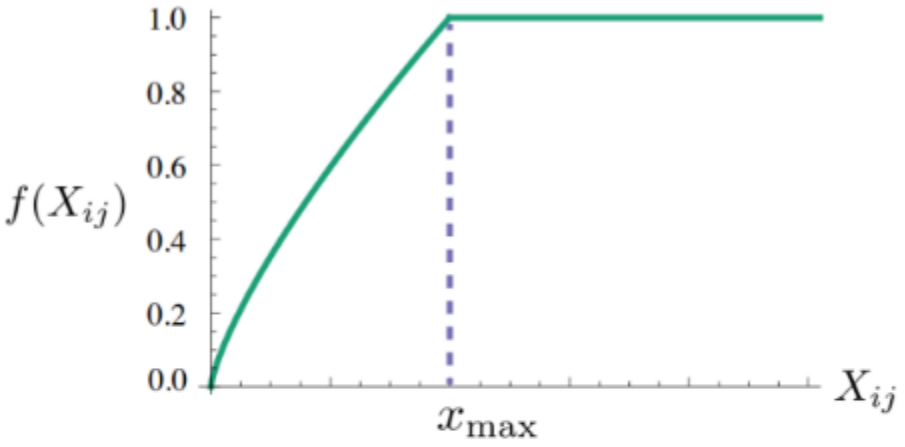
$$w_i^T \tilde{w}_k + b_i + \tilde{b_k} = \log X_{ik}$$

- Minimization of objective function J

$$J = \sum_{i,j=1}^{V} \left(w_i^T \tilde{w}_j + b_i + \tilde{b_j} - \log X_{ij} \right)^2$$

GloVe

- Final Objective Function



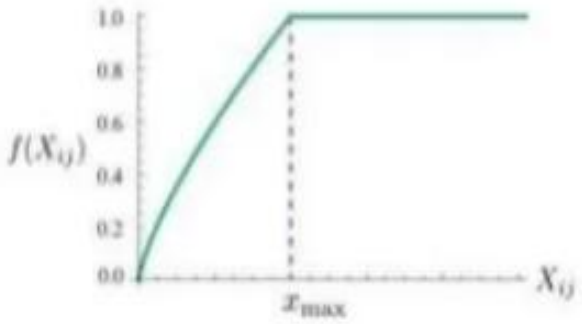
$$J = \sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2$$

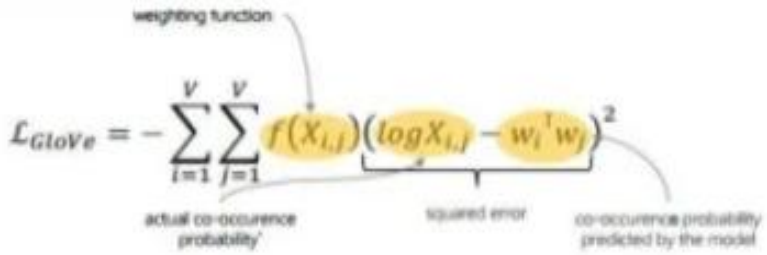
$$where \quad f(x) = \begin{cases} \left(\frac{x}{x_{max}}\right)^{\alpha} \\ 1 \quad otherwise \end{cases} if \quad x < x_{max}$$

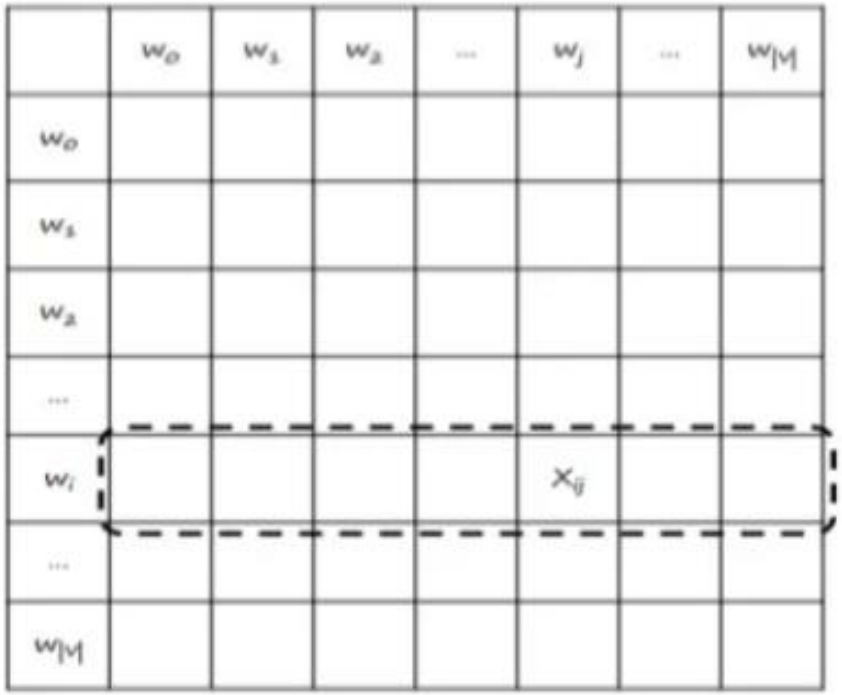
GloVe

- global log-bilinear regression model

Global statistics of co-occurrence probability

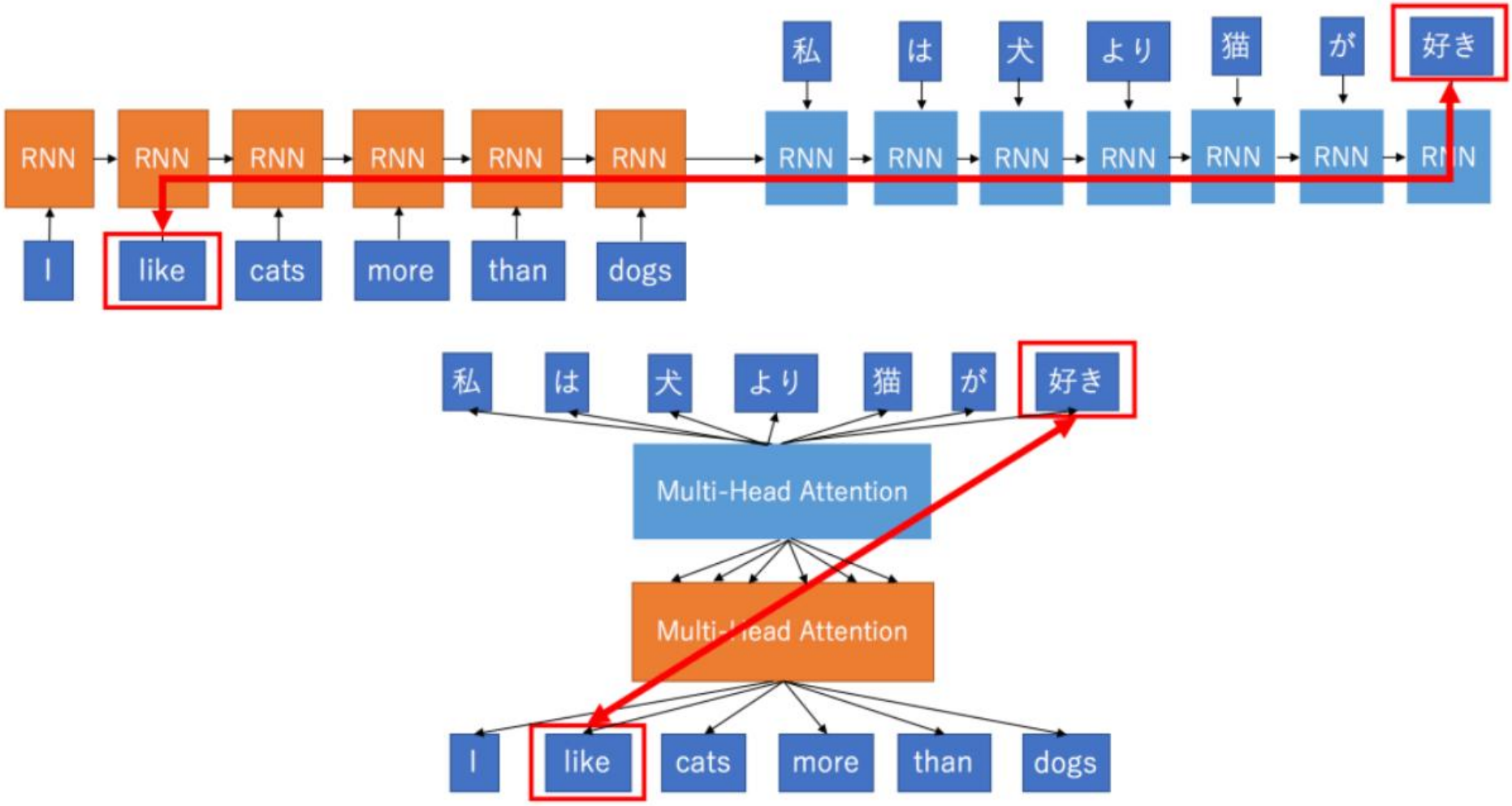






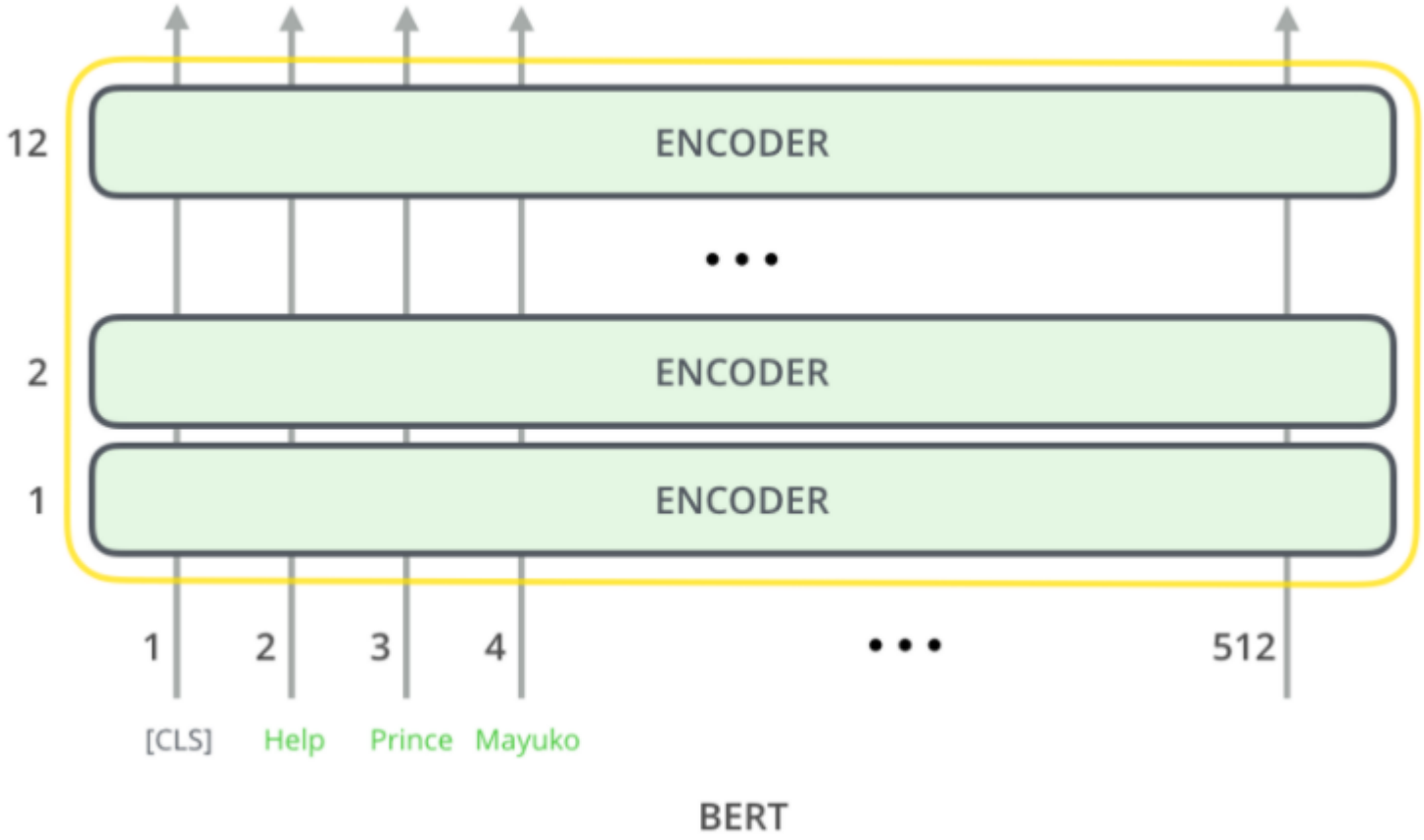
https://www.slideshare.net/HyunYoungLee3/word2vec-119367230

BERT Embedding



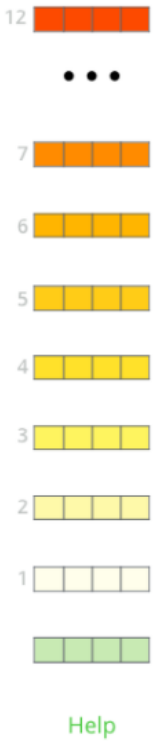
Source: http://mlexplained.com/2017/12/29/attention-is-all-you-need-explained/

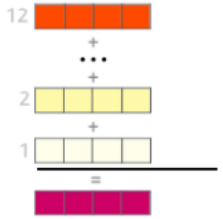
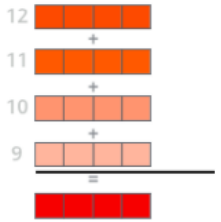
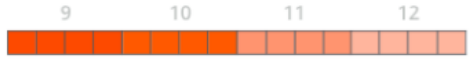
BERT Embedding



BERT Embedding

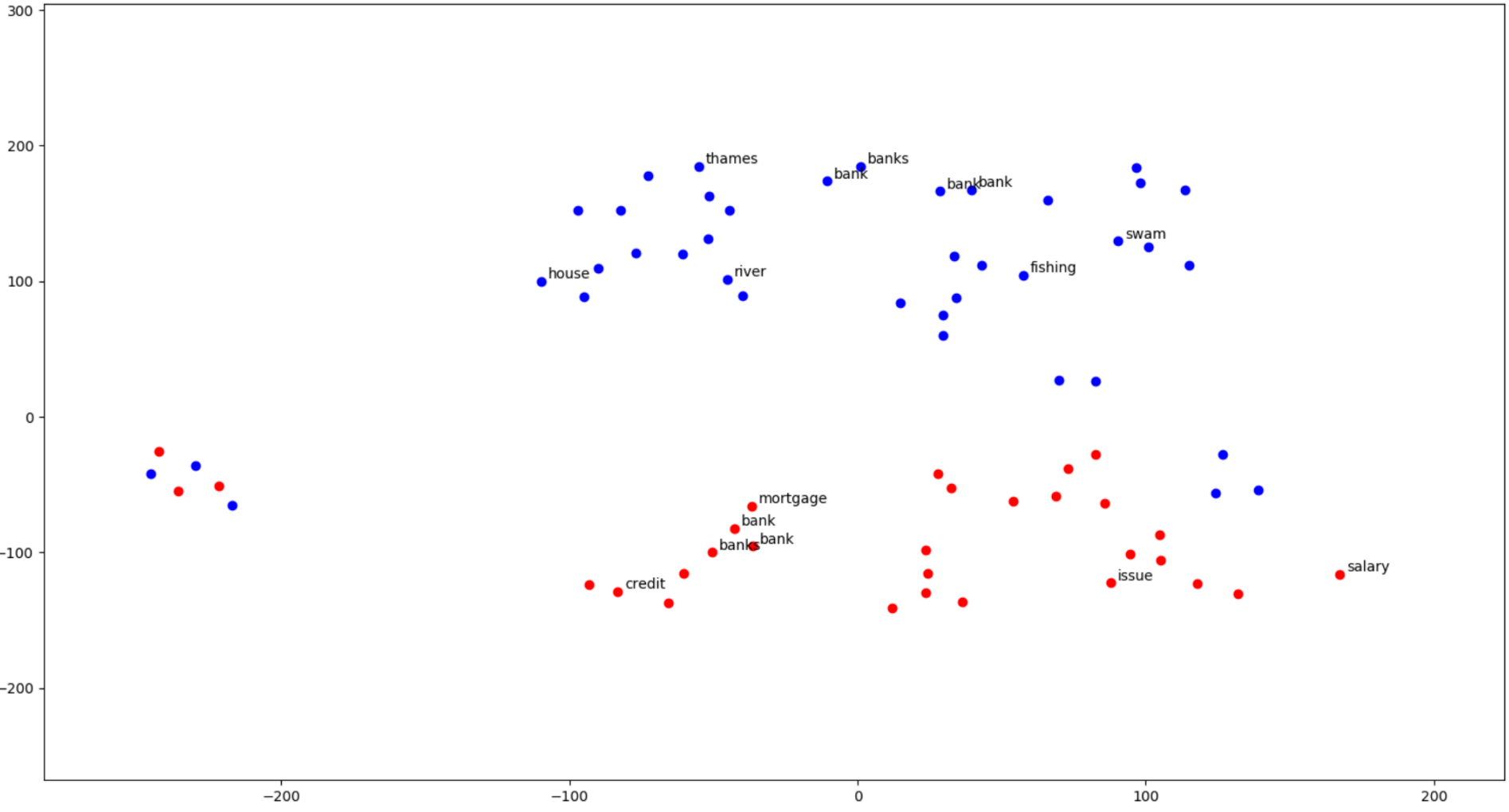
What is the best contextualized embedding for "Help" in that context?
For named-entity recognition task CoNLL-2003 NER



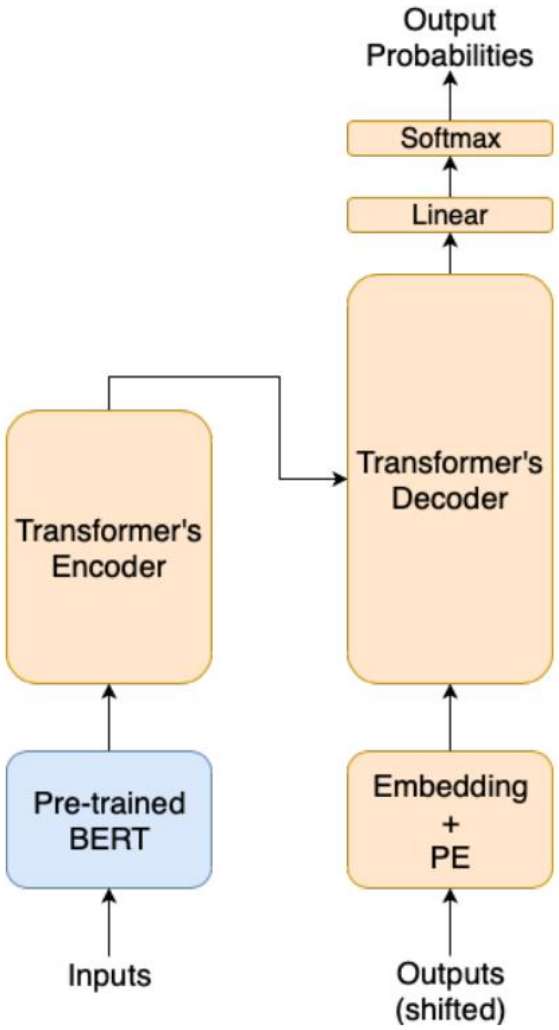
		Dev F1 Score
First Layer		91.0
Last Hidden Layer		94.9
Sum All 12 Layers		95.5
Second-to-Last Hidden Layer		95.6
Sum Last Four Hidden		95.9
Concat Last Four Hidden		96.1

Source: http://jalammar.github.io/illustrated-bert/

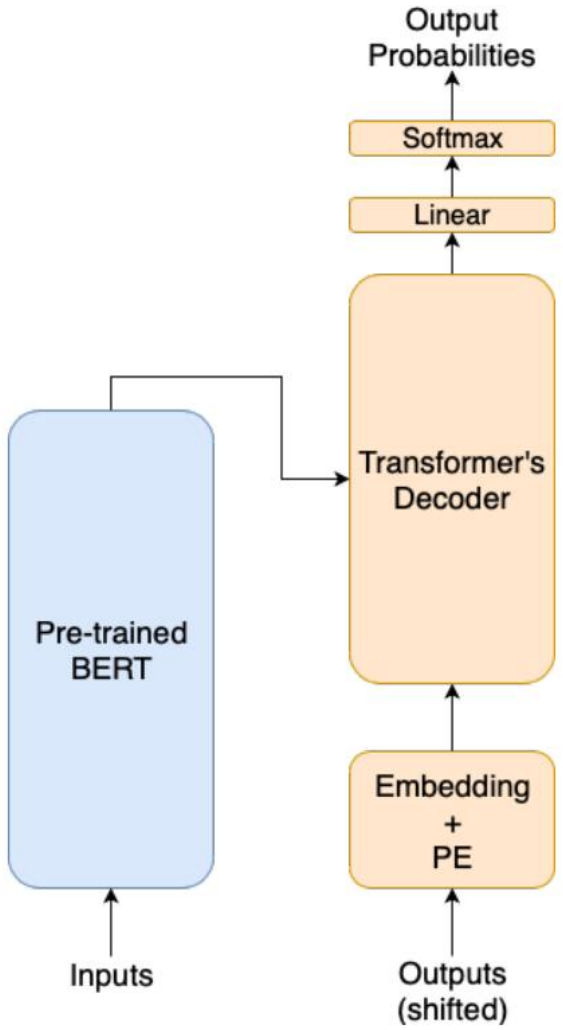
BERT Embedding



BERT Embedding



BERT – WE



BERT – Encoder

BERT Embedding

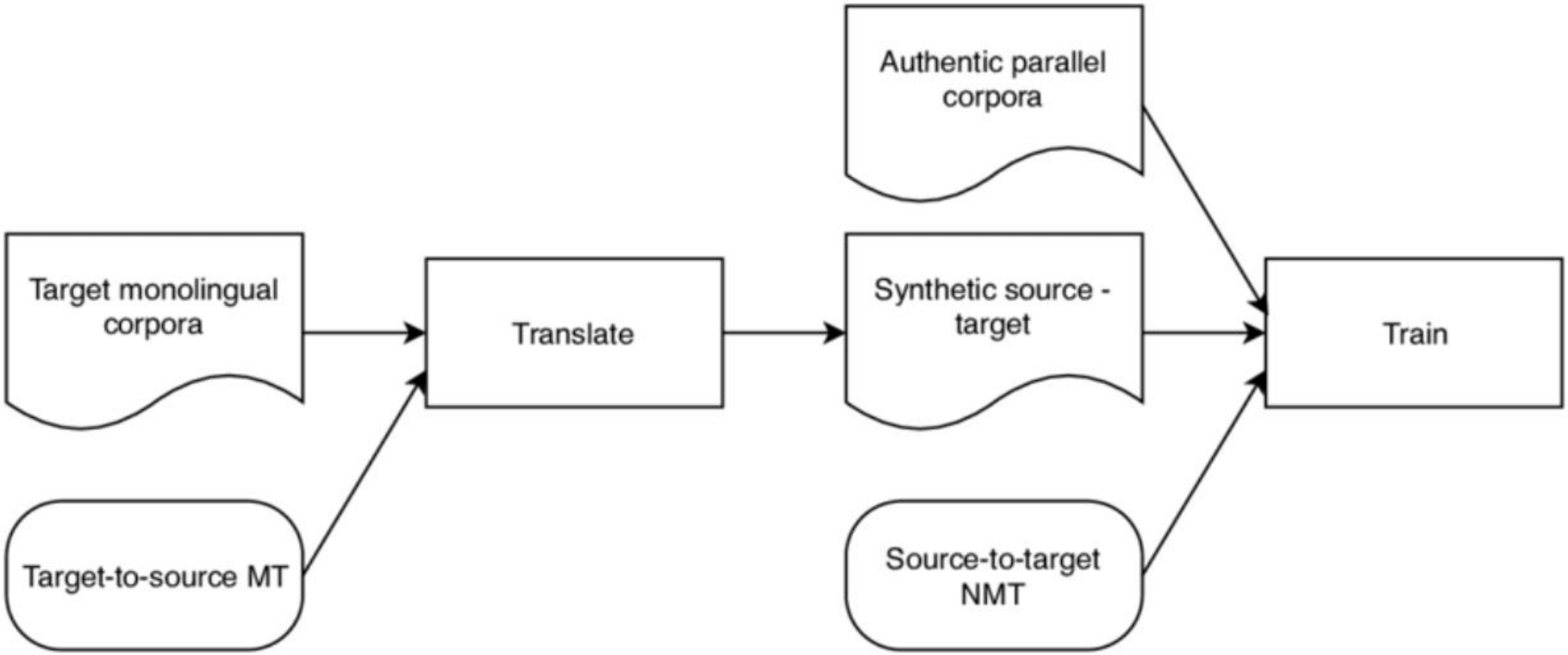
	JESC (general)	IWSLT 2017 (ambiguity)
Transformer	18	7.23
Transformer$_{\mathbf{BERT-WE}}$	20.31	8.67
Transformer$_{\mathbf{BERT-Encoder}}$	17.78	8.16

Table 1: BLEU score evaluation of three translation models on two different test sets

Source sentences	1. A fisherman is sitting on the **bank**.
	2. He is swimming to the opposite **bank**.
Google Translate	1. 漁師が銀行に座っています。
	2. 彼は反対側の銀行に泳いでいます。
Transformer	1. 漁師が銀行に座っています
	2. 彼は正反対の銀行に向かって泳いでいます
Transformer$_{\mathbf{BERT-WE}}$	1. 漁師が土手に座っている
	2. 反対側の岸に泳いでいる
Transformer$_{\mathbf{BERT-Encoder}}$	1. 漁師が土手に座っています
	2. 彼は反対側の岸に向かって泳いでいる

Table 2: Sample translations produced by different machine translation systems. We highlight the homographs in source sentences in bold, the corresponding wrongly translated words in red and the corresponding correctly translated words in blue and green.

Back Translation



Back Translation

	news13	news14	news15
bitext	36.97	42.90	39.92
+sampling	**37.85**	**45.60**	**43.95**

Table 4: Tokenized BLEU on various test sets for WMT English-French translation.

	news13	news14	news15
bitext	35.30	41.03	38.31
+sampling	**36.13**	**43.84**	**40.91**

Table 5: De-tokenized BLEU (sacreBLEU) on various test sets for WMT English-French.

	En–De	En–Fr
a. Gehring et al. (2017)	25.2	40.5
b. Vaswani et al. (2017)	28.4	41.0
c. Ahmed et al. (2017)	28.9	41.4
d. Shaw et al. (2018)	29.2	41.5
DeepL	33.3	**45.9**
Our result	**35.0**	45.6
detok. sacreBLEU[3]	*33.8*	*43.8*

Table 6: BLEU on newstest2014 for WMT English-German (En–De) and English-French (En–Fr). The first four results use only WMT bitext (WMT'14, except for b, c, d in En–De which train on WMT'16). DeepL uses proprietary high-quality bitext and our result relies on back-translation with 226M newscrawl sentences for En–De and 31M for En–Fr. We also show detokenized BLEU (SacreBLEU).

	news17	news18
baseline	29.36	42.38
+BT	32.66	44.94
+ensemble	33.31	46.39
+filter copies	**33.35**	**46.53**
% of source copies	0.56%	0.53%

Table 7: De-tokenized case-insensitive sacreBLEU on WMT English-German newstest17 and newstest18.

Back Translation

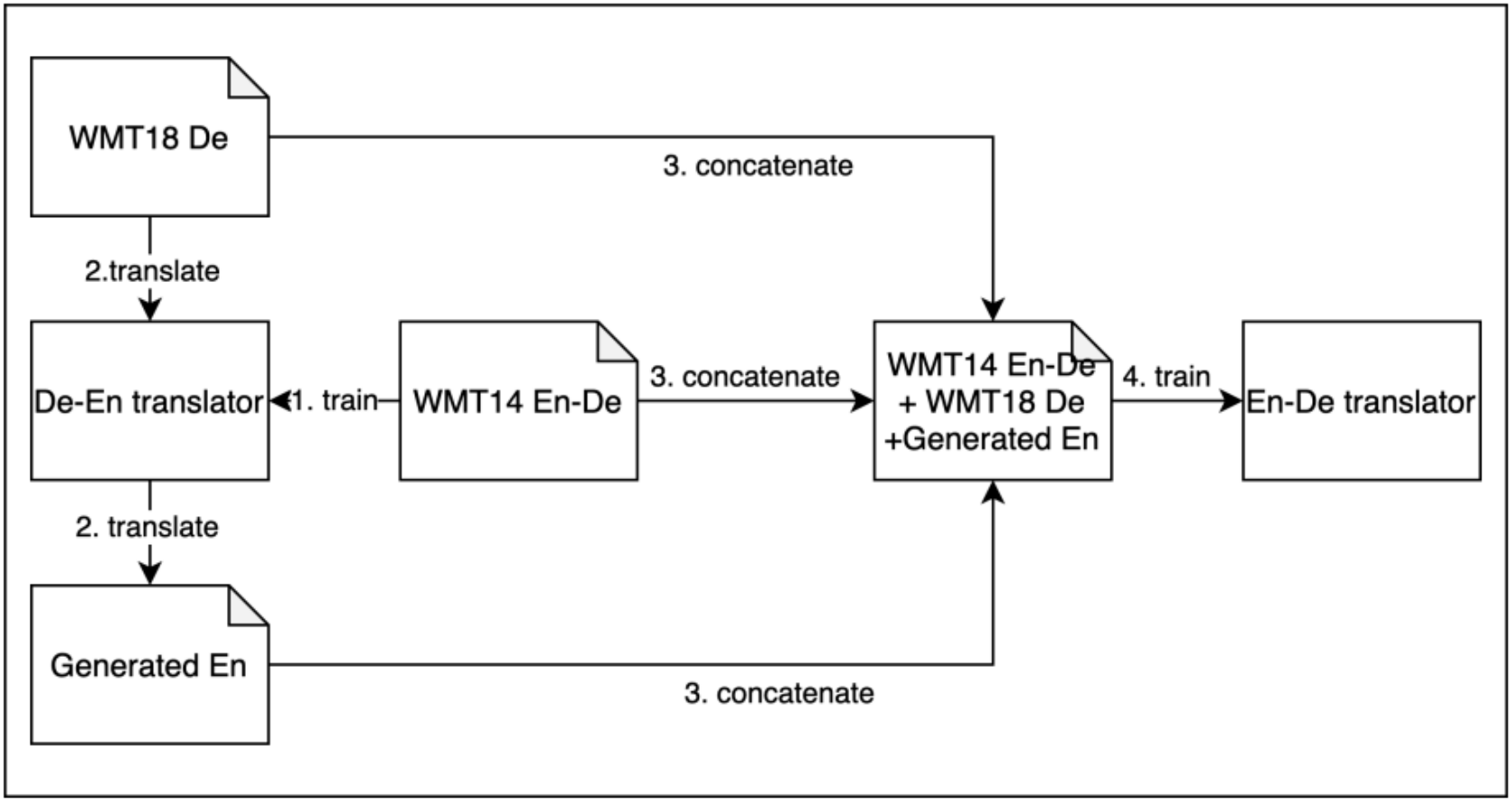


Figure 1. Back-translation is a method that uses additional data to improve translation performance. There are four major steps in the learning process. In recent years, it has begun to be used to evaluate state of the art models. This research is based on back-translation.

Augmentation Based Translation

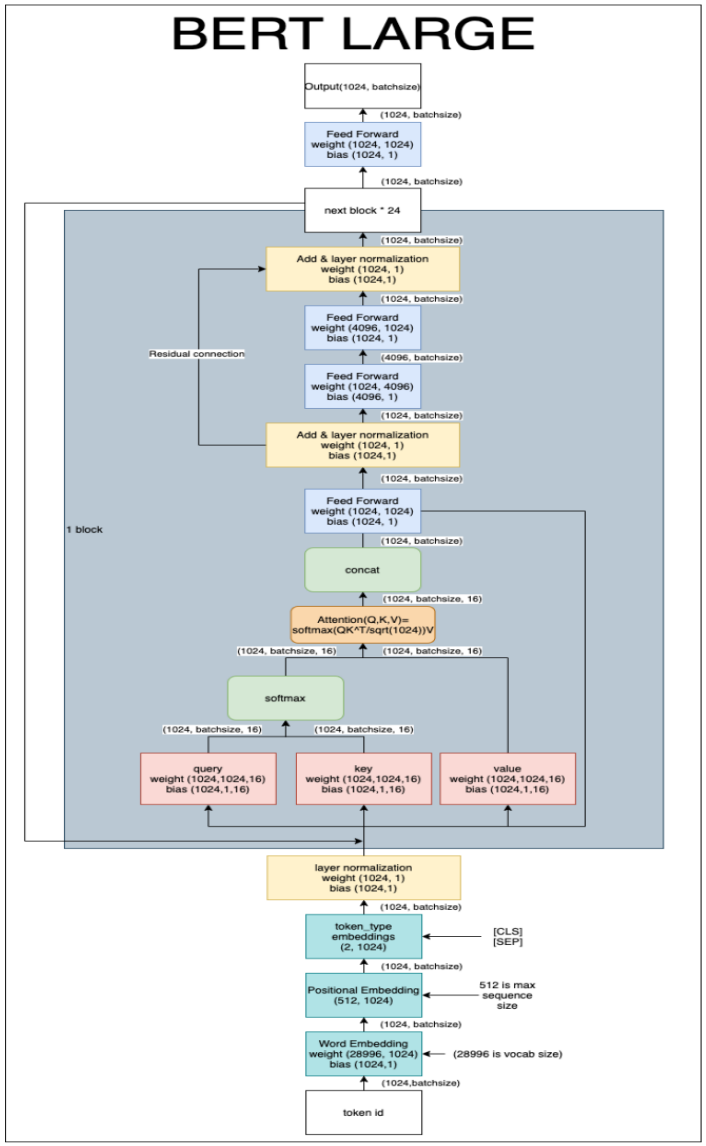


Figure 3. In the BERT Large model, the numbers next to the weight and bias indicate the size of the tensor, and the numbers with arrows indicate the size of the tensor flowing through it.

Augmentation Based Translation

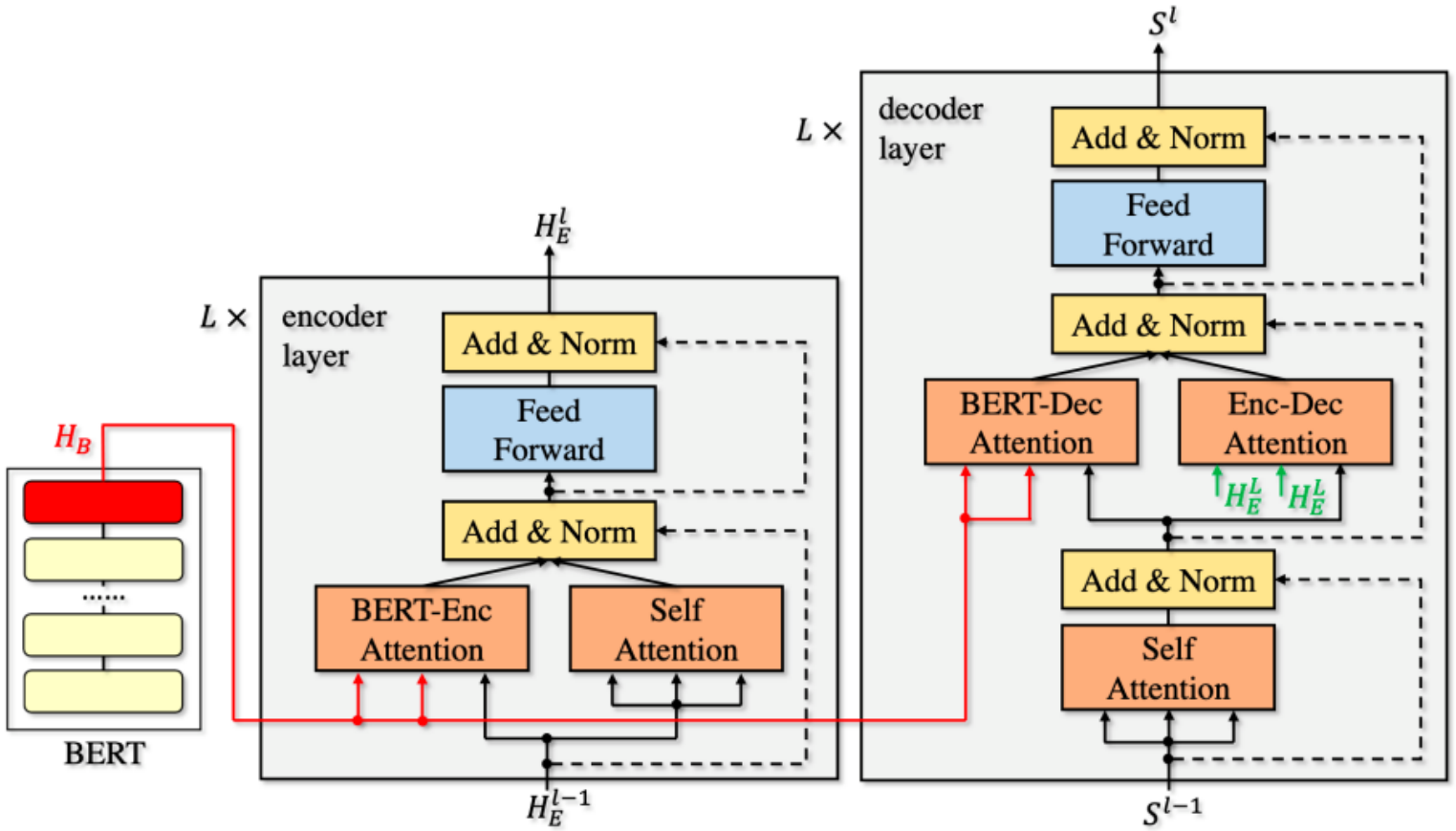
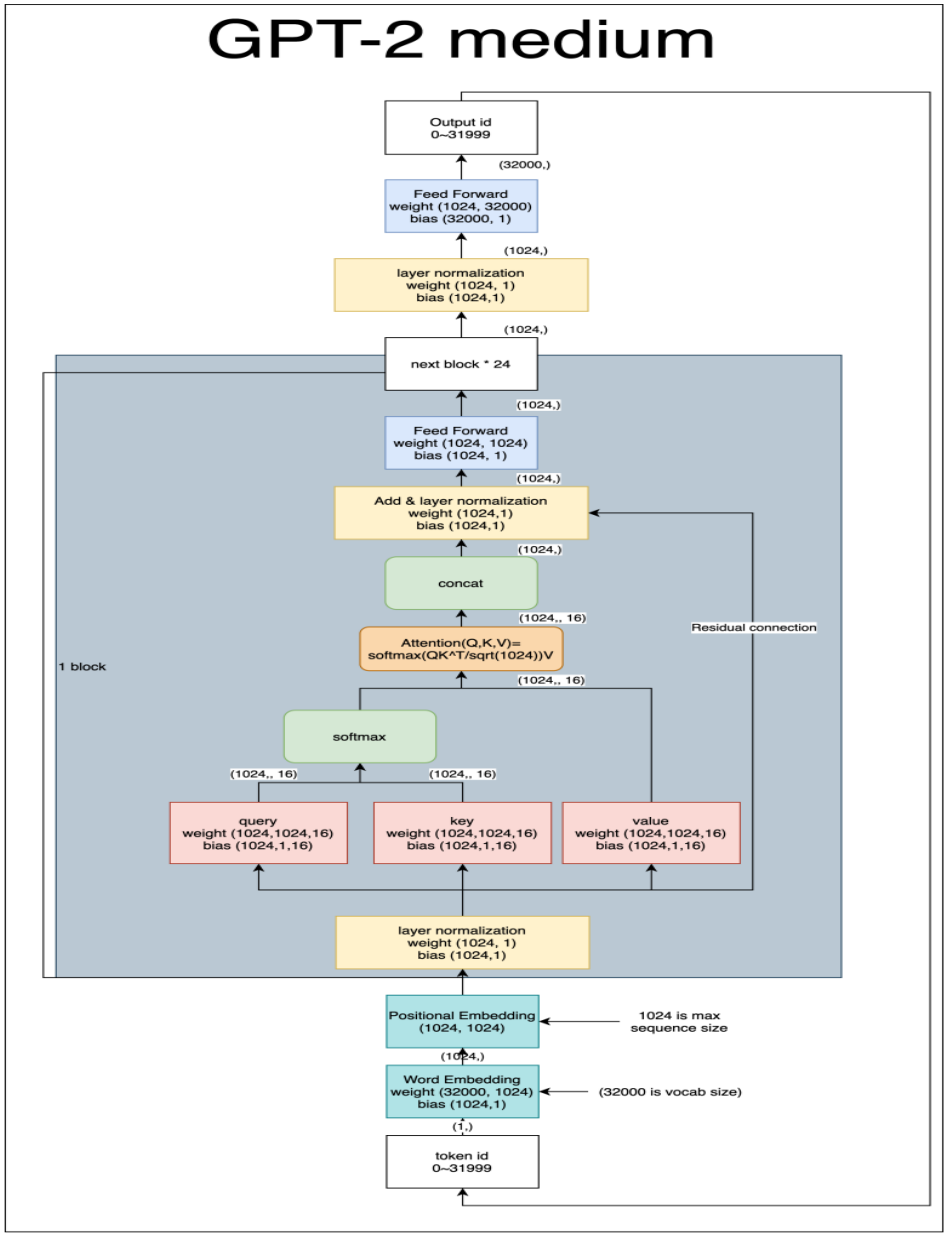


Figure 4. The BERT-fused model is a seq-seq model based on the transformer, and like the transformer, it has an encoder and a decoder. This is the model we use in the experiments of this study.

Augmentation Based Translation



Augmentation Based Translation

Table 2. An example of GPT-2 predicting the continuation of a sentence of tatoebaEn-Ja and WMT14En-De. In the case of the tatoeba corpus, a novel-like sentence is generated, while in the case of the WMT corpus, a news-like sentence is generated.

Original Sentence (tatoebaEn-Ja)
My mother set the table for dinner.
Generated by GPT-2
"I'm just going to start out with the turkey and then it'll get a bit more elaborate." As the meal was being served, I saw his mother looking at me with a curious look of confusion. "I asked you to be the hostess?" The look made the girl seem confused by the idea. Her face was completely red. "Well... what kind of hostess do you think I am?" Her mother, who I guess was her wife, looked at me. Her eyebrows were raised. After hearing the words, her eyes widened and she opened them wide in shock. "What do you mean? Are you telling me you can be the hostess? I'm in a hurry. I'm going to be late, I'm going to be late!" And it was all over. My mother was standing at the table after the dinner and my aunt and uncle were looking at me with a curious expression. I was a bit amazed.
Original Sentence(WMT14En-De)
This is all in accordance with the principles that we have always upheld .
Generated by GPT-2
"We will continue to work to preserve the Constitution of the United States, and we will do so in accordance with the rule of law" and "to respect our fellow citizens no matter where they live or how many different languages we all speak, or what country they live in". At a news conference today, the president also urged Congress to continue to work on "a balanced approach" that would also allow for the possibility of a bi-national test. "We have to look at what we would do with people coming in from other countries that are hostile to the U.S.", he said. The New York City police department said it stopped arresting hundreds of people a day in 2012 because its officers didn't think their fellow officers were doing their jobs properly.

Augmentation Based Translation

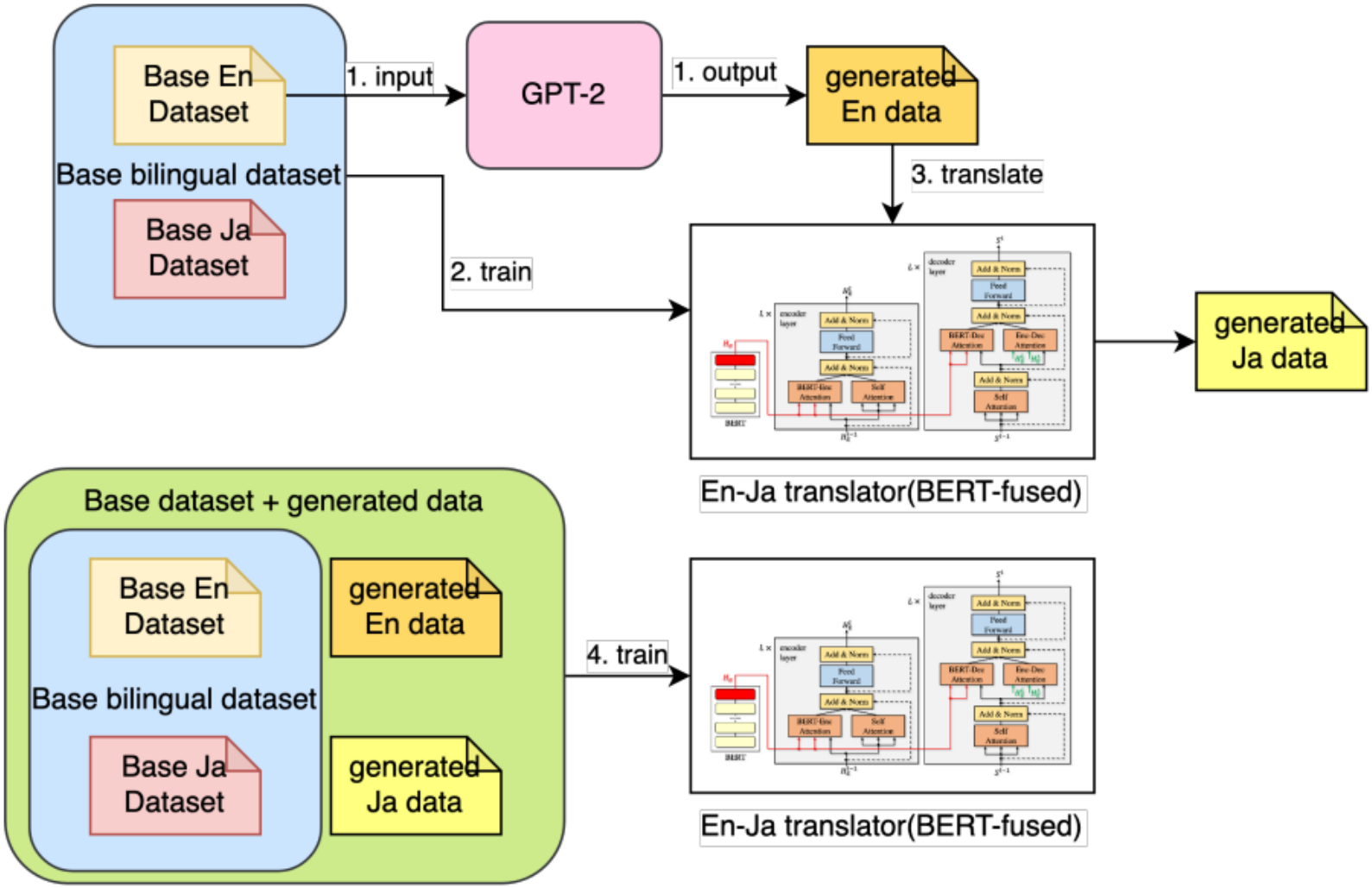


Figure 6. This figure describes how to increase the data and which data to use to train the translator.

Augmentation Based Translation

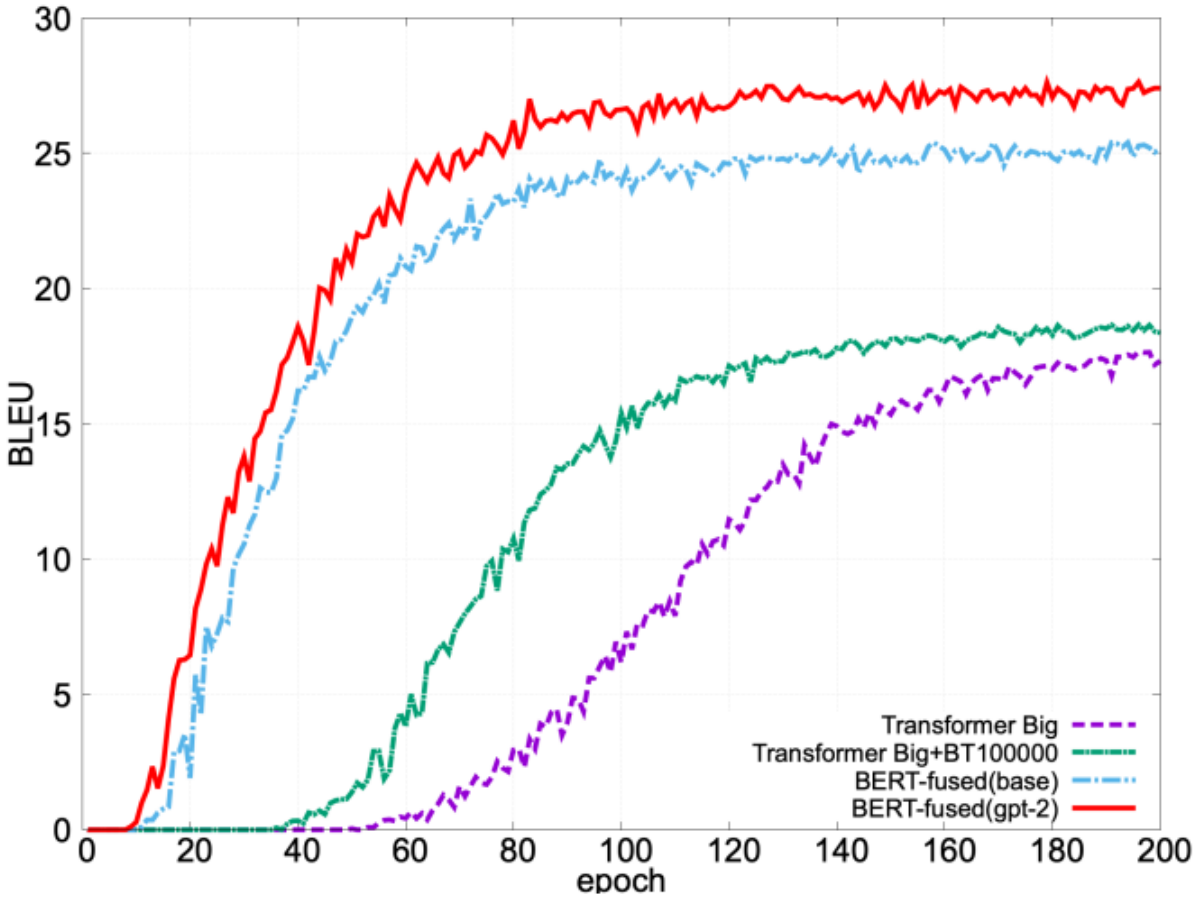


Figure Results of the tatoebaEn-Ja experiment. tatoebaEn-Ja requires a larger number of epochs than the other experiments due to the small amount of data.

Augmentation Based Translation

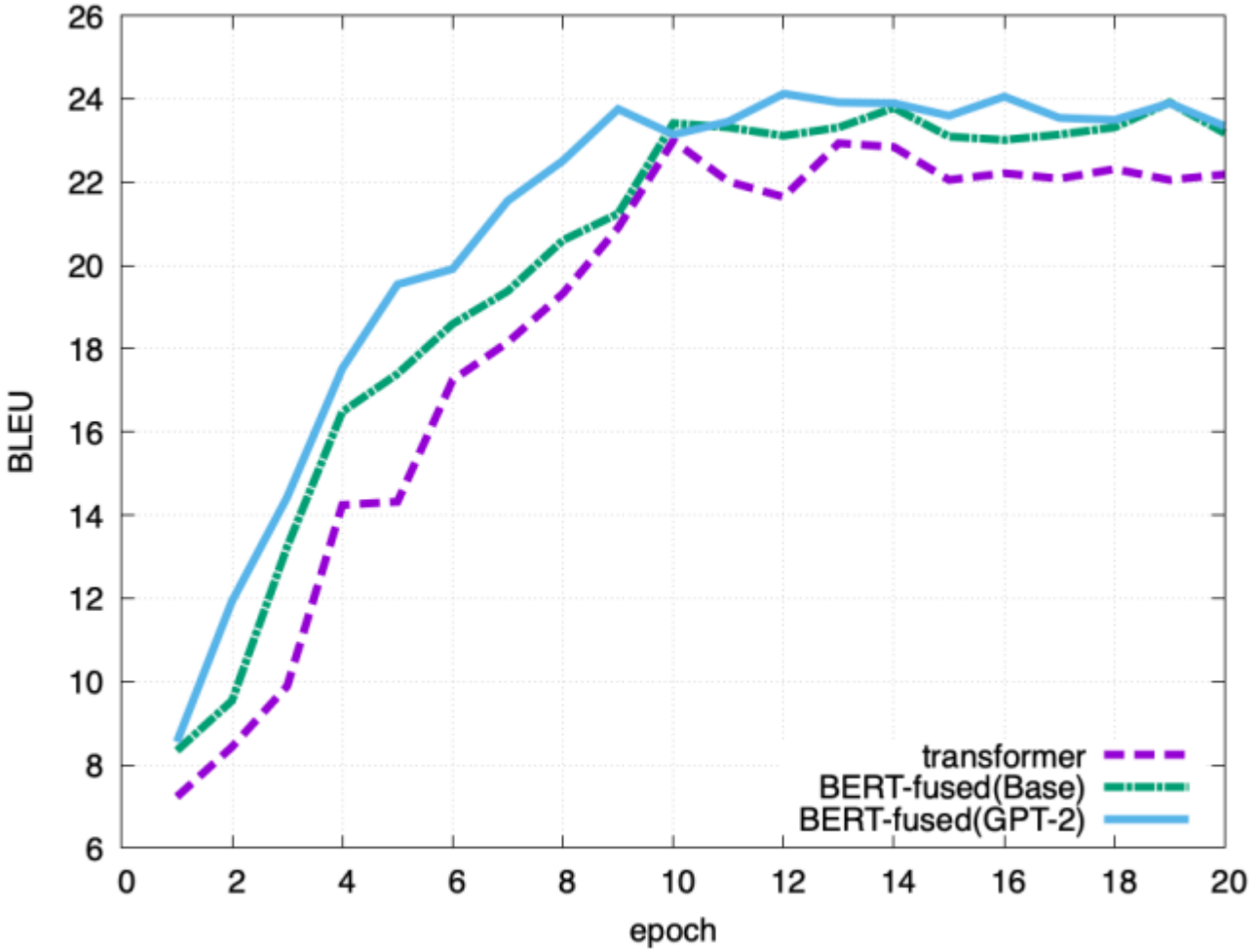


Figure Results of the WMT18En-Ch experiment.

Augmentation Based Translation

Model	tatoebaEn-Ja	WMT14En-De	WMT18En-Ch
Transformer	19.67	28.4 [8]	22.99
Back-Translation	19.41	29.31	
MAT	22.96	29.9 [10]	
BERT-fused		30.75 [15]	
BERT-fused (base)	25.36	29.80	23.91
BERT-fused (GPT-2)	27.50	30.14	24.12

Thank you!

Questions?