COMP5046
Natural Language Processing

Lecture 11: Advanced NLP: Machine Translation and Transformer

Semester 1, 2020
School of Computer Science
The University of Sydney, Australia

Dr Caren Han
Caren.Han@sydney.edu.au

0 LECTURE PLAN

Lecture 11: Machine Translation and Transformer

1. Machine Translation
2. Statistical Machine Translation
3. Neural Machine Translation
4. Attention and Transformer for MT
5. The Rise of the Pre-trained Model

1 Machine Translation

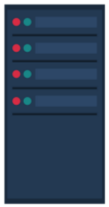
Machine Translation

*“translate a sentence x from one language (**the source language**) to a sentence y in another language (**the target language**).”*

Source language

Target language

Sentence x 生命短暂	***Sentence y*** *life is short*

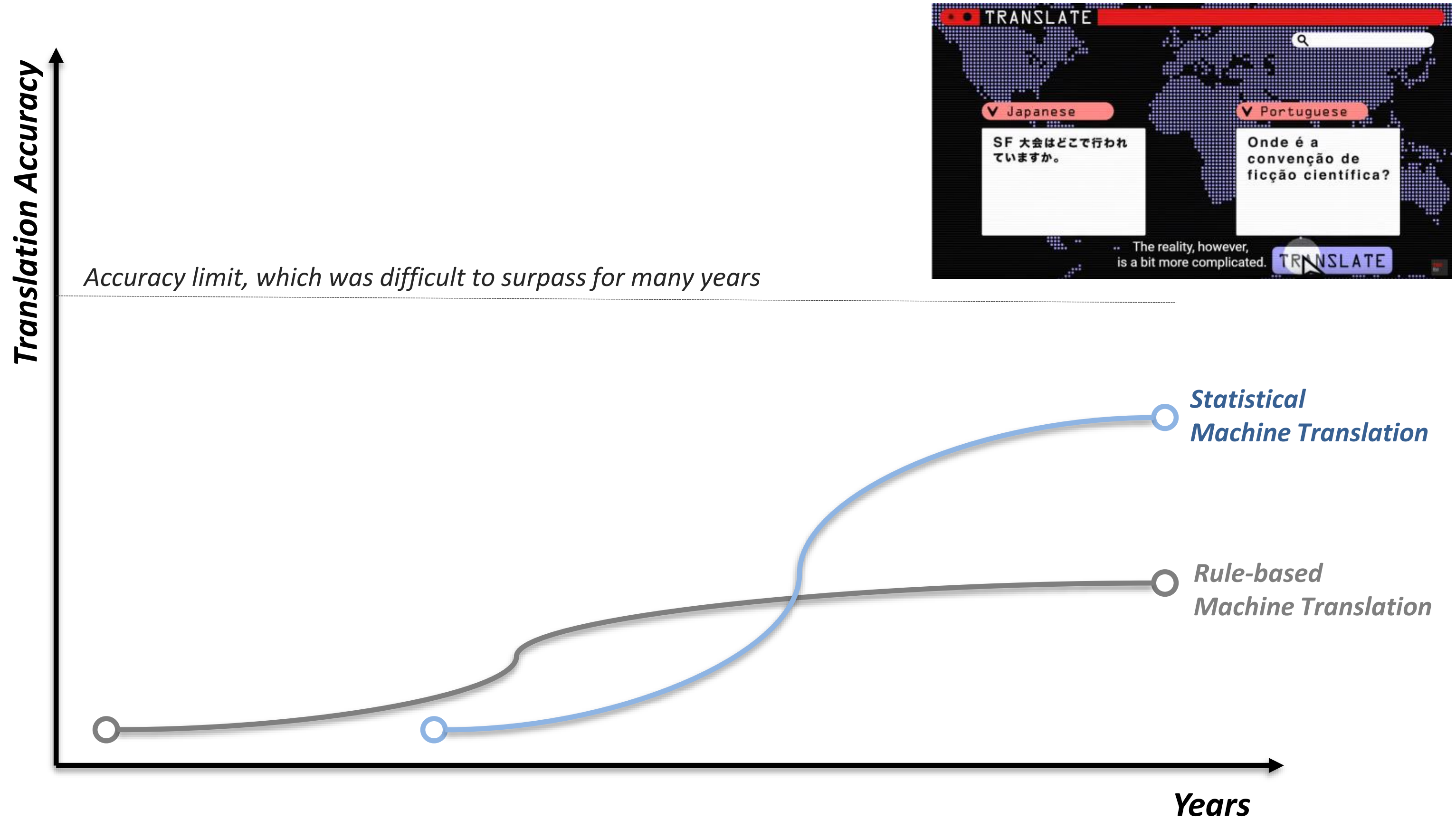


Machine

1 Machine Translation

Machine Translation progress over time

https://youtu.be/X4BmV2t83SM



2 Statistical Machine Translation

Statistical Machine Translation

*"Learning a **probabilistic model** from data"*

Source language (x)

Target language (y)

Sentence x 生命短暂 → ***Sentence y*** *life is short*

Best translation?

$$\mathrm{argmax}_y P(y|x)$$

How to learn translation model $P(x|y)$ *?*

Statistical Machine Translation

*"Learning a **probabilistic model** from data"*

Source language (x)

Sentence x 生命短暂

→

Target language (y)

Sentence y *life is short*

Best translation?

$$\mathrm{argmax}_y P(y|x)$$

Bayes Rule $$\mathrm{argmax}_y \underbrace{P(x|y)}_{\text{Translation Model}} \underbrace{P(y)}_{\text{Language Model}}$$

Translation Model (fidelity)
Models how words and phrases should be translated

Language Model (fluency)
Models to write good English

2 Statistical Machine Translation

Statistical Machine Translation

"Learning a ***probabilistic model*** *from data"*

Source language (x)

Sentence x 生命短暂

Target language (y)

Sentence y *life is short*

Best translation?

$$\underline{\mathrm{argmax}_y P(y|x)}$$

Bayes Rule

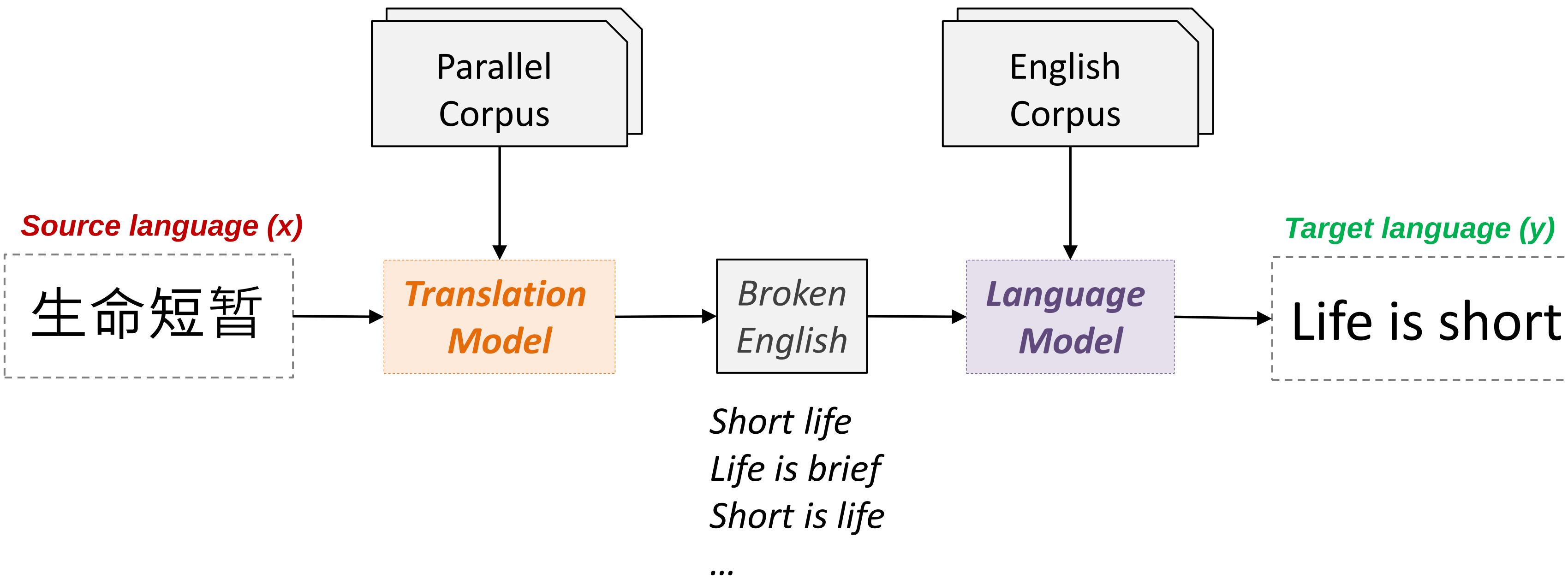
$$\mathrm{argmax}_y \underline{P(x|y)}\,\underline{P(y)}$$

Translation Model (fidelity)
Learnt from ***parallel data***

Language Model (fluency)
Learnt from ***monolingual data***

2 Statistical Machine Translation

How to learn translation model with *parallel corpus*?



Bayes Rule $\text{argmax}_y \underbrace{P(x|y)}\,\underbrace{P(y)}$

Translation Model (fidelity)
Learnt from ***parallel data***

Language Model (fluency)
Learnt from ***monolingual data***

2 Statistical Machine Translation

Parallel corpus and Alignment

*How to learn translation model **from the** parallel corpus?*

i.e. pairs of human-translated Chinese/English sentences

OPUS ... the open parallel corpus

OPUS is a growing collection of translated texts from the web. In the OPUS project we try to convert and align free online data, to add linguistic annotation, and to provide the community with a publicly available parallel corpus. OPUS is based on open source products and the corpus is also delivered as an open content package. We used several tools to compile the current collection. All pre-processing is done automatically. No manual corrections have been carried out.

The OPUS collection is growing! Check this page from time to time to see new data arriving ...
Contributions are very welcome! Please contact <jorg.tiedemann@helsinki.fi >

Search & download resources: en (English) ▾ zh (Chinese) ▾ >1M ▾

Language resources: click on [tmx | moses | xces | lang-id] to download the data! (raw = untokenized, ud = parsed with universal dependencies, alg = word alignments and phrase tables)

corpus	doc's	sent's	en tokens	zh tokens	XCES/XML	raw	TMX	Moses	mono	raw	ud	alg	dic	freq			other files
MultiUN v1	67167	10.5M	288.2M	80.0M	xces en zh	en zh	tmx	moses	en zh	en zh		alg		en zh	query	sample	
OpenSubtitles v2016	9829	10.3M	80.6M	71.7M	xces en zh	en zh	tmx	moses	en zh	en zh		alg	dic	en zh		sample	
OpenSubtitles v2011	714	0.7M	6.1M	6.2M	xces en zh	en zh										sample	
News-Commentary v11	7107	0.1M	6.6M	1.6M	xces en zh	en zh	tmx	moses	en zh	en zh		alg smt	dic	en zh	query	sample	
Tanzil v1	30	0.2M	5.6M	1.7M	xces en zh	en zh	tmx	moses	en zh	en zh		alg smt	dic	en zh	query	sample	
UN v20090831	1	74.1k	3.7M	1.2M	xces en zh	en zh	tmx	moses	en zh	en zh		alg smt		en zh	query	sample	
News-Commentary v9.1	1	91.6k	3.4M	0.8M	xces en zh	en zh	tmx	moses	en zh	en zh		alg smt		en zh		sample	
News-Commentary v9.0	1	91.6k	3.1M	0.8M	xces en zh	en zh	tmx	moses	en zh	en zh				en zh		sample	
TED2013 v1.1	1	0.2M	3.1M	0.9M	xces en zh	en zh	tmx	moses	en zh	en zh		alg smt	dic	en zh	query	sample	
total	**84851**	**22.2M**	**400.4M**	**164.9M**	**22.2M**		**21.5M**	**21.5M**									

http://opus.nlpl.eu/

2 Statistical Machine Translation

Parallel corpus and Alignment

How to align these sentence (Open subtitles)

(trg)="1"> 片名 ： 解放 的 潘 多 拉

(src)="1"> My name is Alice .
(trg)="2"> 我的 名字 是 阿 ? 丽 斯 。

(src)="2"> Alice Bonnard ...
(trg)="3"> 阿 ? 丽 斯 ...

(src)="3"> like my father and mother .
(trg)="4"> 象我 的 父母 。

(src)="4"> I hate people .
(trg)="5"> 我 恨 周 ? 围 的 人 。

(src)="5"> They oppress me .
(trg)="6"> 他 ?? 压 迫 我 。

(src)="6"> All year , I was away at school .
(trg)="7"> 整年 我 都 是 去 ? 学 校 。

(src)="7"> I only came home for end- of- term holidays
(trg)="8"> 我 只 有 ? 学 期 近 ? 结 束 ? 时 回家

(src)="8"> Summer holidays were the worst .
(trg)="9"> 暑假 最麻 ? 烦 。

(src)="9"> They were endless .
(trg)="10"> ? 没 完 ? 没 了 。

(src)="10"> I' m a little girl .
(trg)="11"> 我 是 一 ? 个 小女孩 。

(src)="11"> I don' t know , no , I don' t know .
(trg)="12"> 我 不知道 , 不 , 我 不知道 。

Statistical Machine Translation

How to learn translation model?

How to learn translation model ***from the*** ***parallel corpus****?*

$$P(x|y)$$

$$P(x, a|y)$$

i.e. pairs of human-translated Chinese/English sentences

Source language (x) ***Target language (y)***

Sentence x *life is short* → *Sentence y* 生命短暂

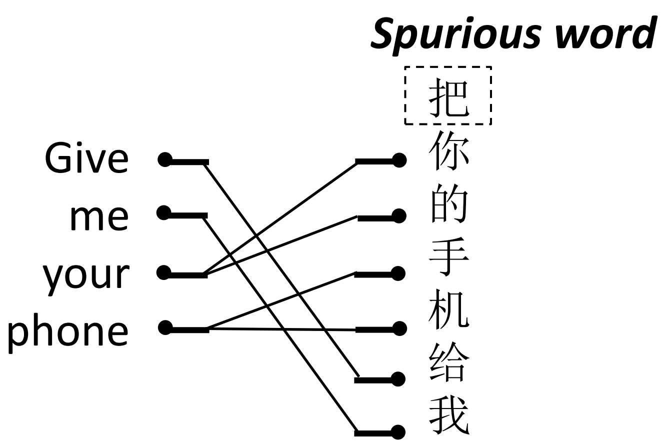
a is the ***alignment***

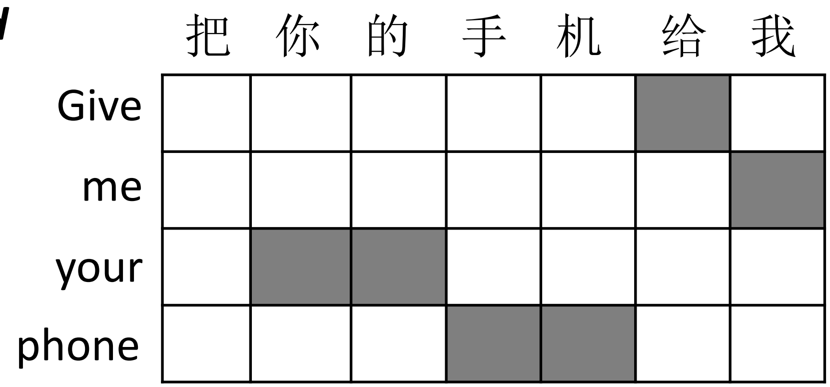
Alignment is the correspondence between particular words in the translated sentence pair.
(i.e. word-level correspondence between *source sentence x* and *target sentence y*)

2 Statistical Machine Translation

What is Alignment *a*?

“The correspondence between particular words in the translated sentence pair”

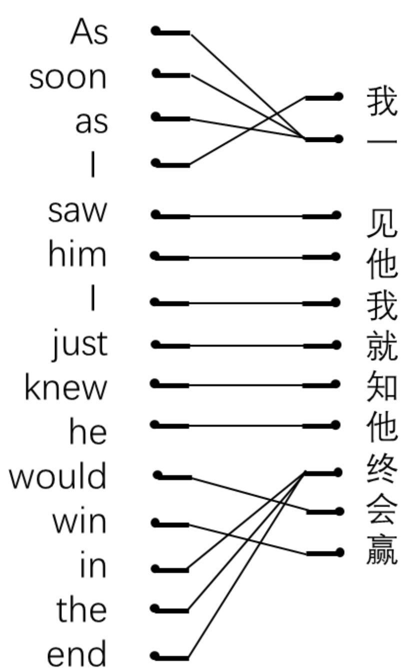


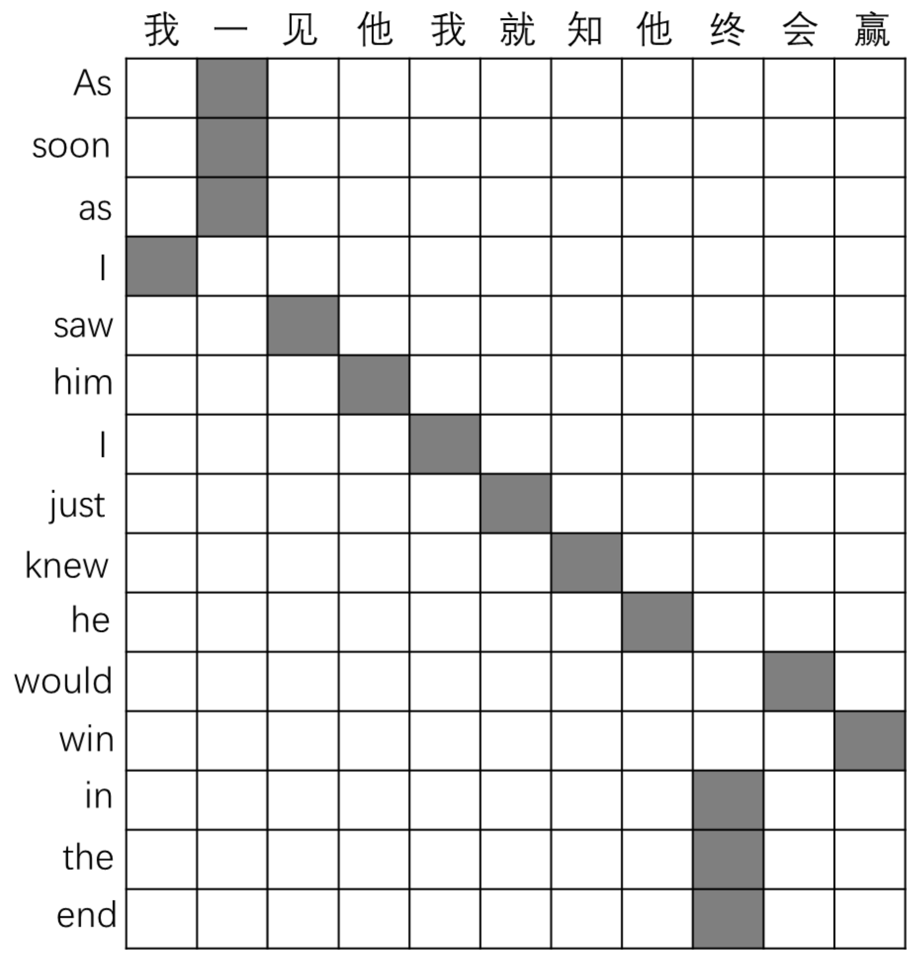


2 Statistical Machine Translation

What is Alignment *a*?

Many-to-One Alignment

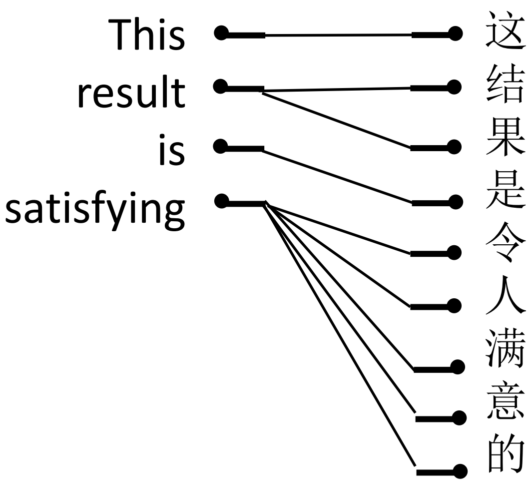


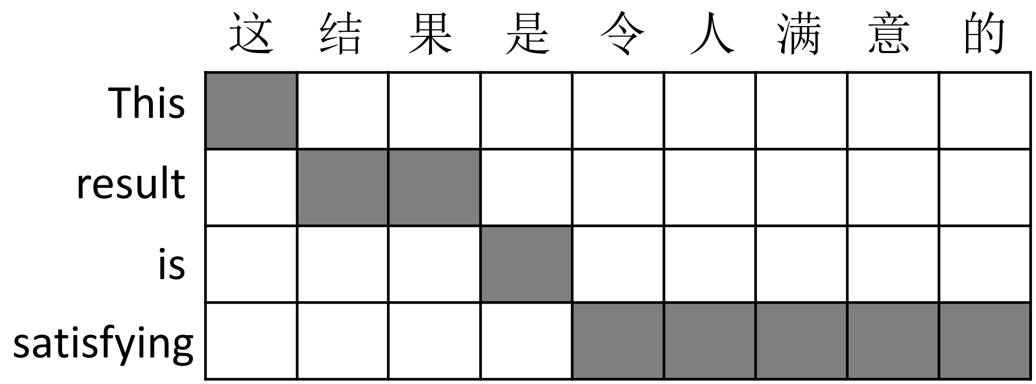


2 Statistical Machine Translation

What is Alignment *a*?

One-to-Many Alignment

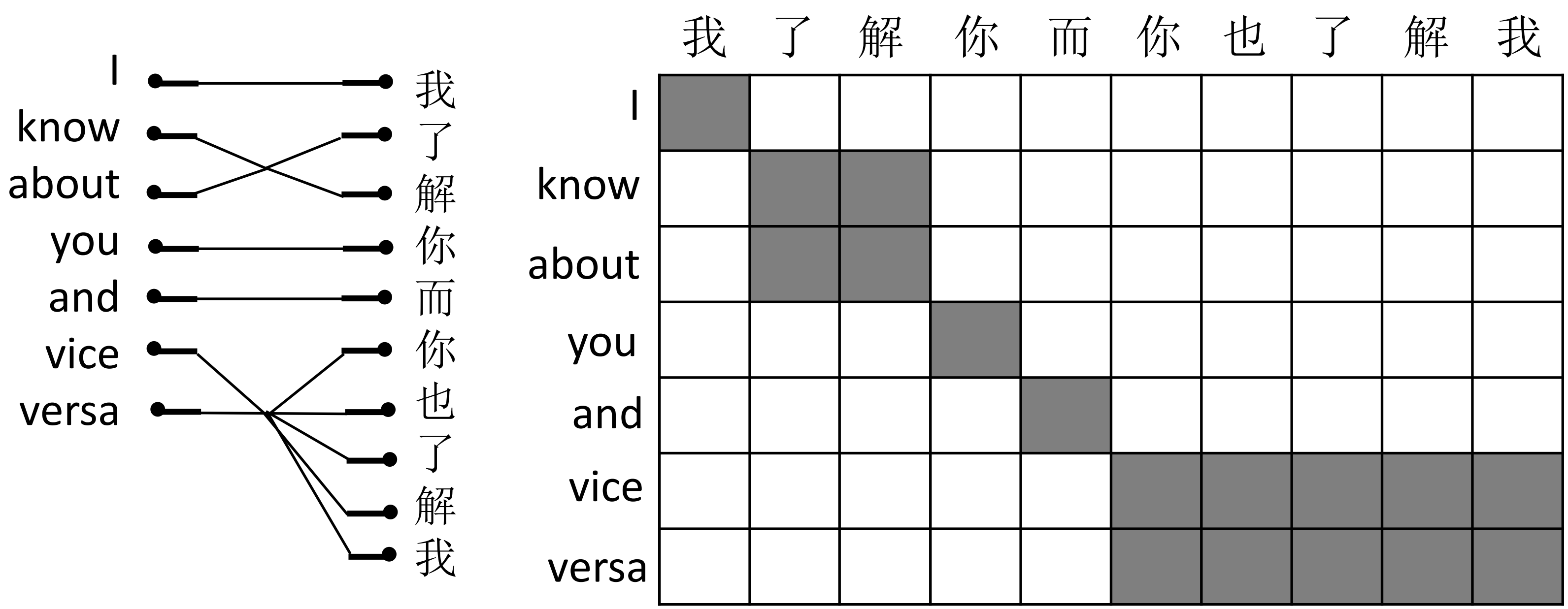




Statistical Machine Translation

What is Alignment *a*?

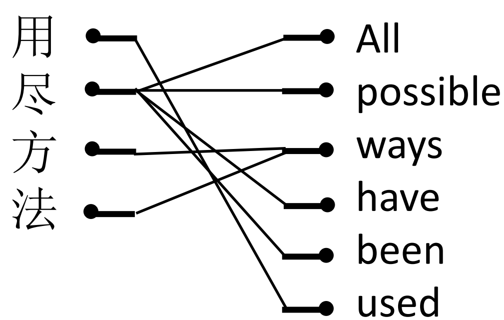
Many-to-many Alignment

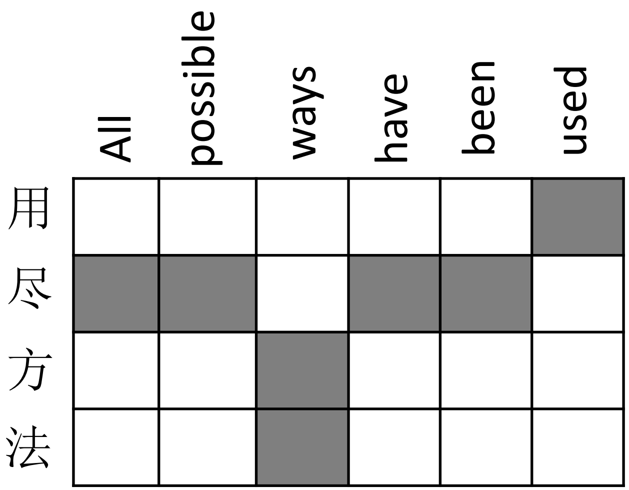


2 Statistical Machine Translation

What is Alignment *a*?

Some word has no single-word equivalent in English





2 Statistical Machine Translation

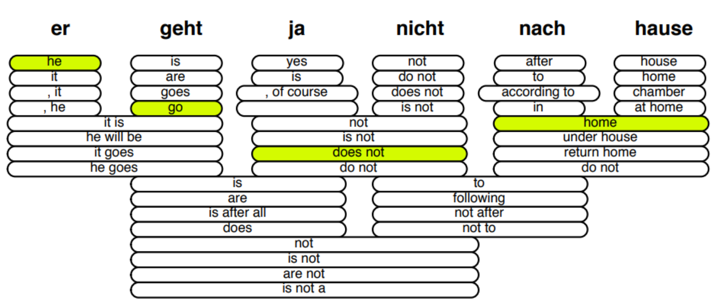
Decoding for SMT

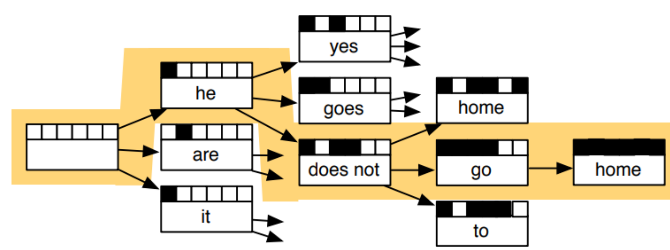
$$\operatorname{argmax}_y \underbrace{P(x|y)}_{} \underbrace{P(y)}_{}$$

Translation Model (fidelity)
Learnt from ***parallel data***

Language Model (fluency)
Learnt from ***monolingual data***

- We could enumerate every possible y and calculate the probability? Too expensive!
- Answer: Use a ***heuristic search algorithm*** to ***search for the best*** translation, discarding hypotheses that are too low-probability





backtrack from highest scoring complete hypothesis

2 Statistical Machine Translation

Statistical Machine Translation

The Best System

*SMT was a **huge research field** and **Extremely complex System***

Hundreds of important details (haven't mentioned here)

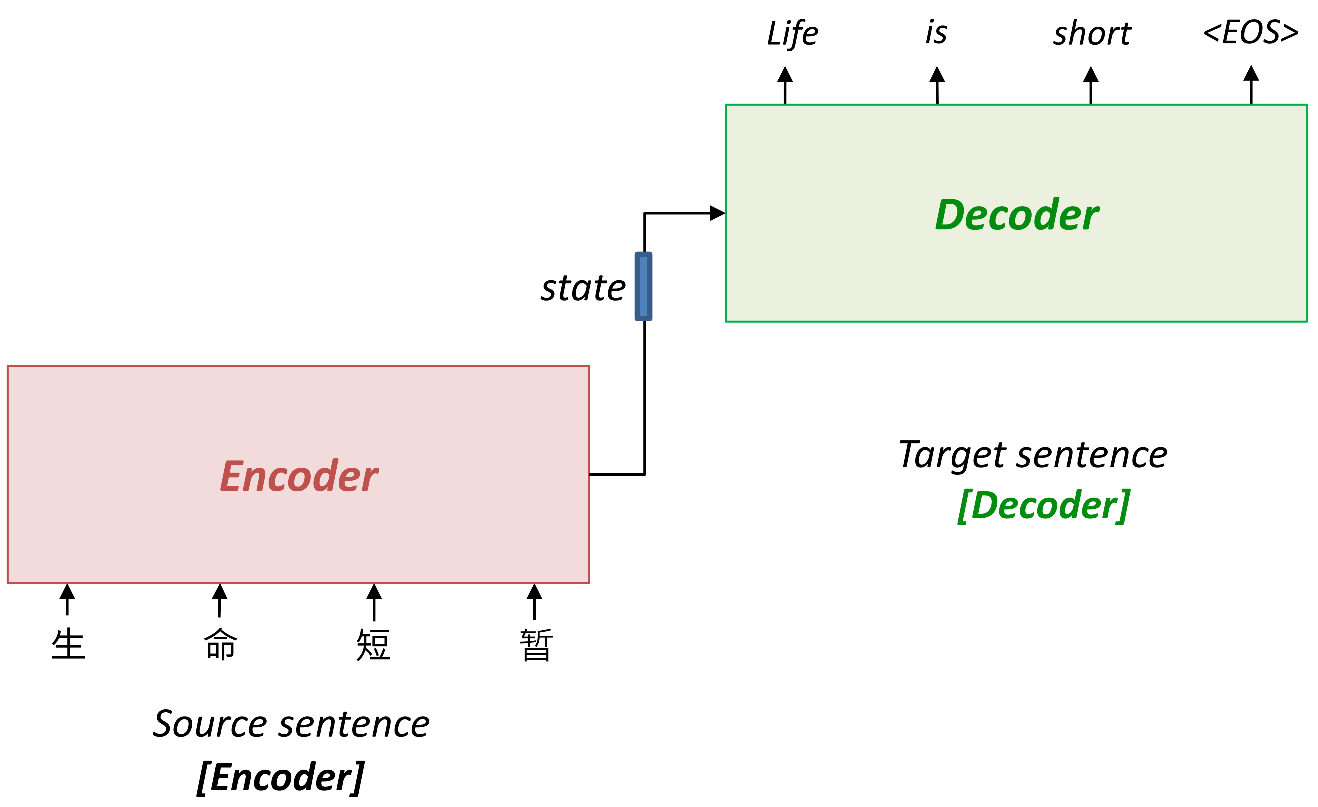
- *Systems had many separately-designed subcomponents*
- *Lots of feature engineering*
  - *Need to design features to capture particular language phenomena*
- *Require compiling and maintaining extra resources*
  - *Like tables of equivalent phrases*
- *Lots of human effort to maintain*
  - *Repeated effort for each language pair!*

3 Neural Machine Translation

Neural Machine Translation with Seq2Seq

"a way to do Machine Translation with a single neural network (NN)"

- *The NN architecture is called **seq2seq** and involves **two RNNs.***

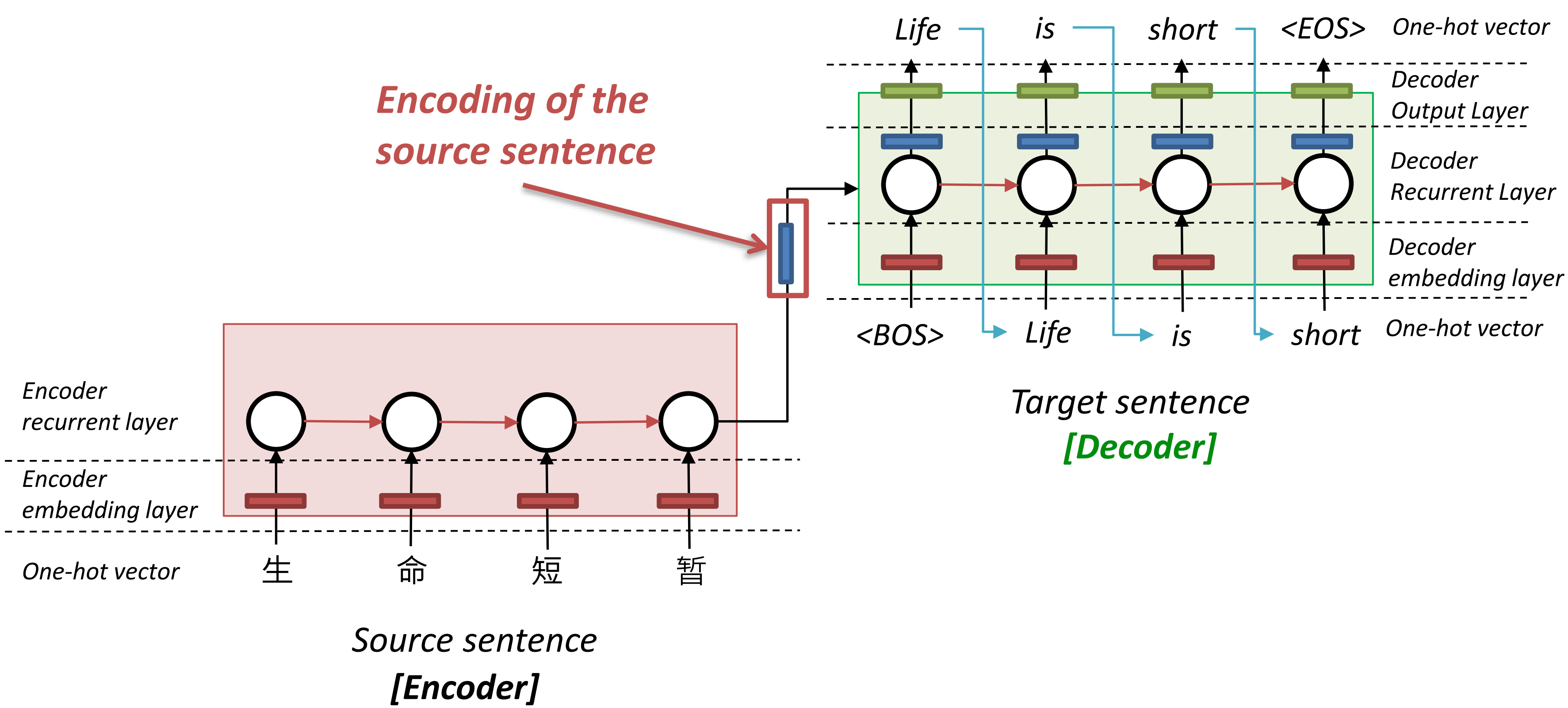


3 Neural Machine Translation

Neural Machine Translation with Seq2Seq

"a way to do Machine Translation with a single neural network (NN)"

- *The NN architecture is called **seq2seq** and involves **two RNNs.***

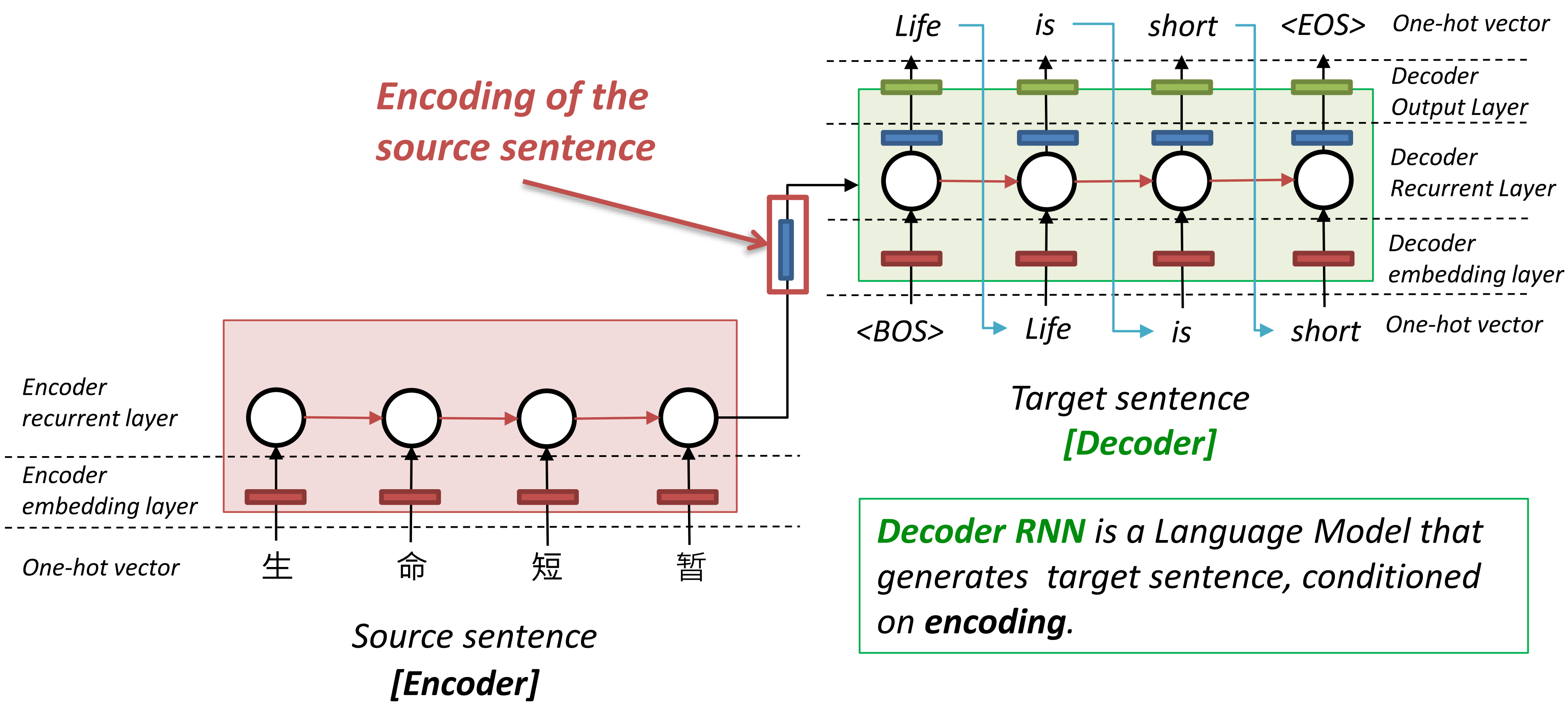


3 Neural Machine Translation

Neural Machine Translation with Seq2Seq

"a way to do Machine Translation with a single neural network (NN)"

- *The NN architecture is called **seq2seq** and involves **two RNNs.***

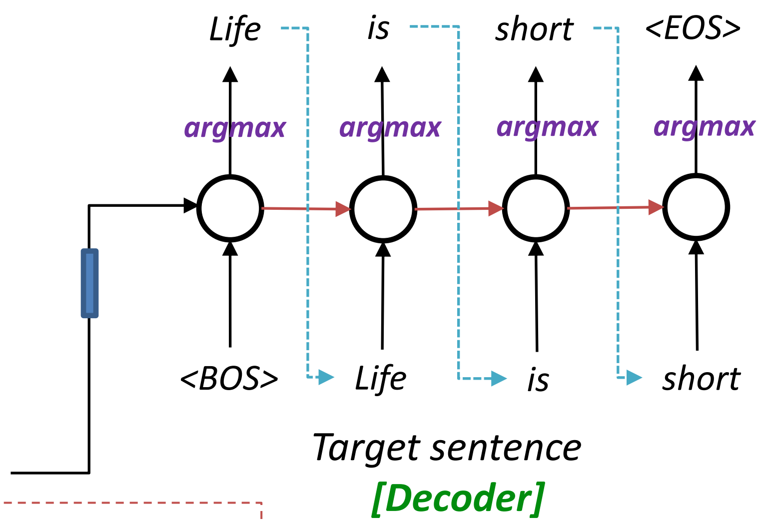


__Decoder RNN__ is a Language Model that generates target sentence, conditioned on __encoding__.

Neural Machine Translation: Greedy Decoding *[Recap]*

Language Model Decoding: Recap

- Generate the sentence by taking ***argmax*** (the most probable word) on each step
- Use that as the next word, and feed it as input on the next step
- Keep going until you produce *<EOS>*



Greedy decoding has no way to undo decisions!!
(Ungrammatical, unnatural)

Solution..? try computing all possible sequences

Neural Machine Translation: Beam Search Decoding *[Recap]*

Language Model Decoding: Recap

- A search algorithm which aims to find a high-probability sequence (not necessarily the optimal sequence, though) by tracking multiple possible sequences at once.

- On each step of decoder, keep track of the k most probable partial sequences (which we call hypotheses)
- K is the beam size (in practice around 5 to 10)

- After you reach some stopping criterion, choose the sequence with the highest probability (factoring in some adjustment for length)

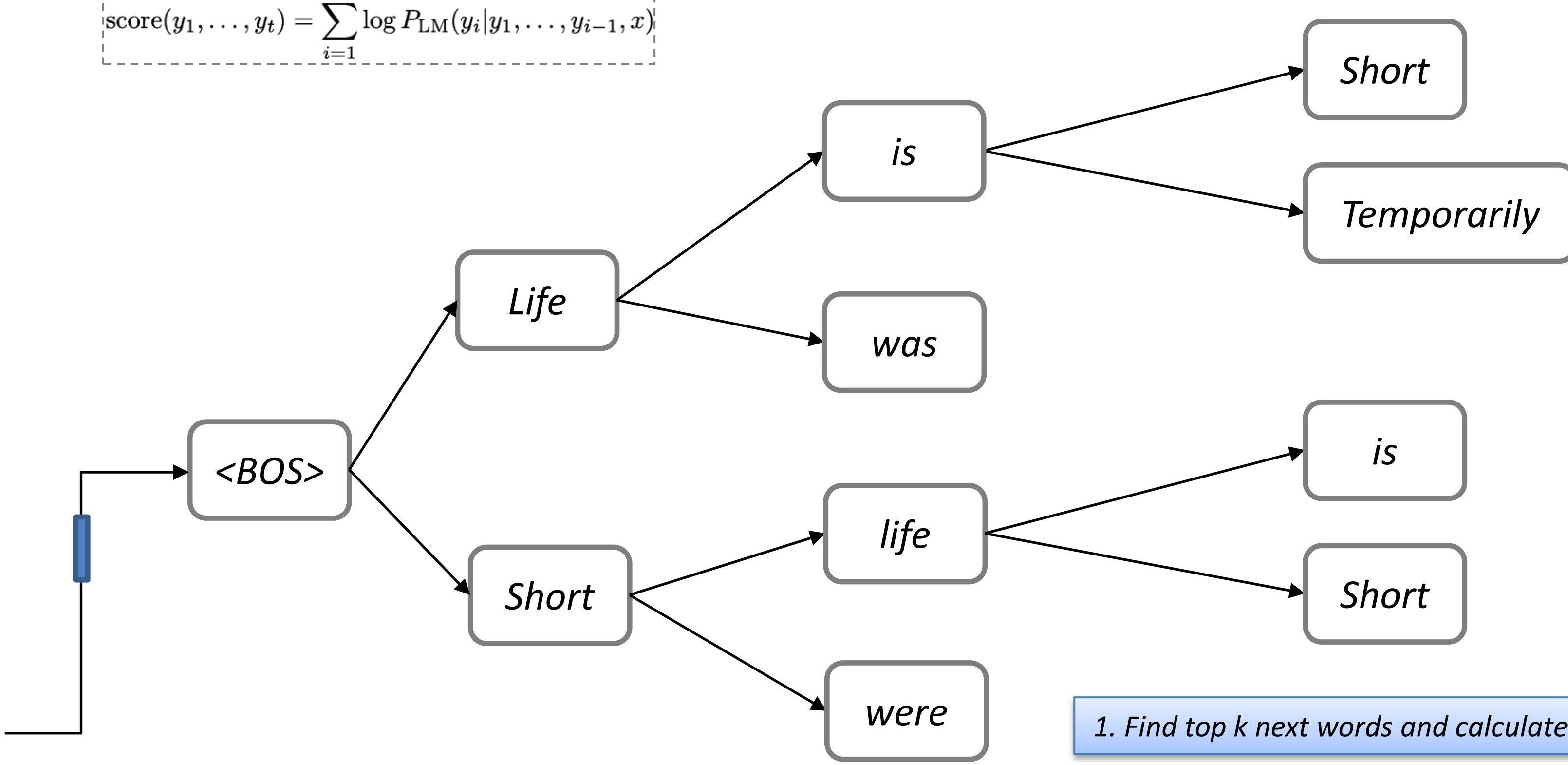
3 Neural Machine Translation

Neural Machine Translation: Beam Search Decoding

Language Model Decoding: Recap

Assume that k(beam size)=2

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



1. Find top k next words and calculate scores

2. Of these k^2 hypotheses, keep only highest k

Evaluate Machine Translation

BLEU *(Bilingual Evaluation Understudy)*

*“Compares the **machine-written translation** to one or several **human-written translation**(s), and computes a similarity score based on”*

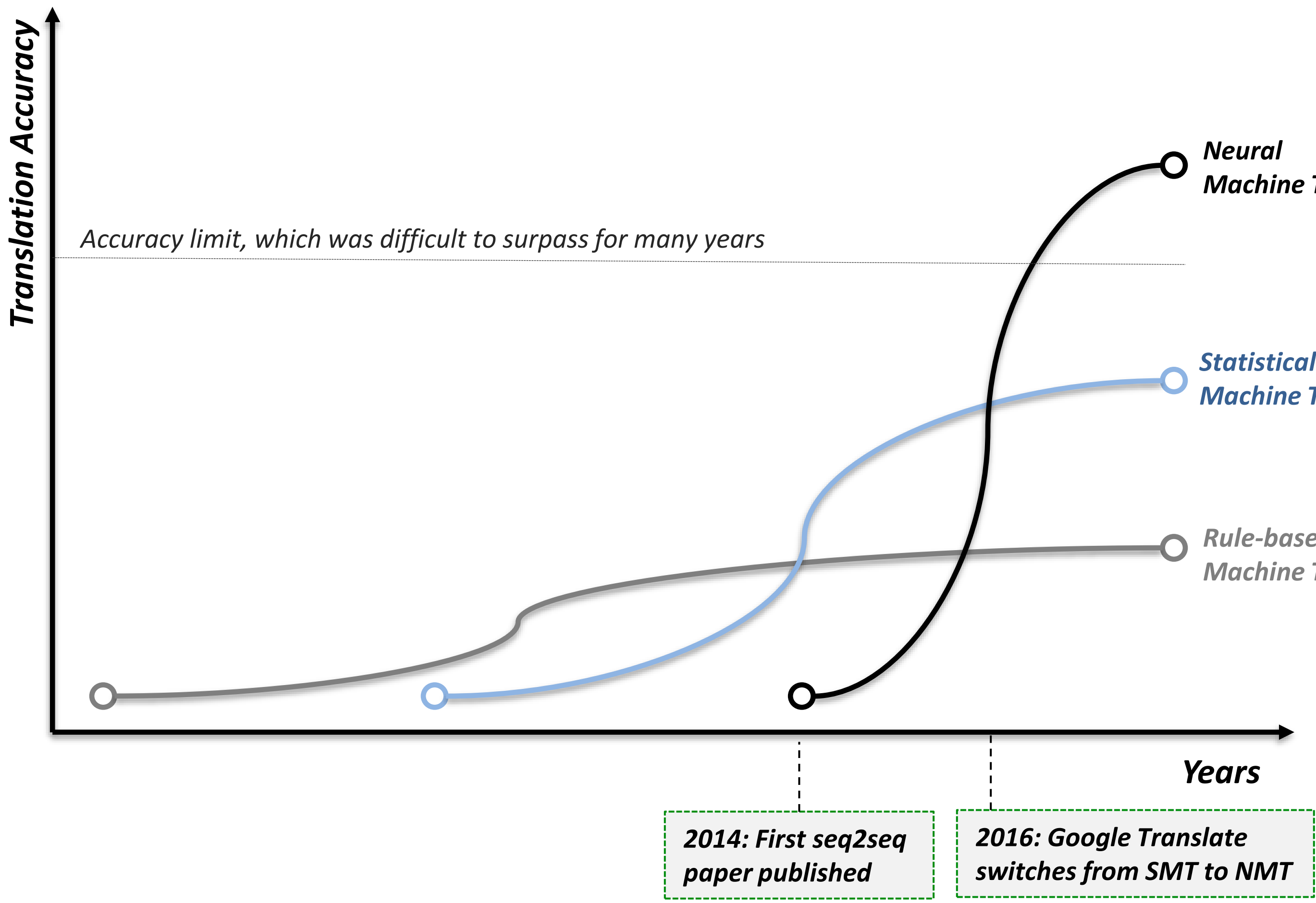
- *n-gram precision (usually for 1 to 4-grams)*
- *Plus a penalty for too-short system translations*

BLEU *is useful but imperfect*

- *Many valid ways to translate a sentence*
- *So a good translation can get a poor BLEU score because it has low n-gram overlap with the human translation*

Neural Machine Translation

Machine Translation progress over time



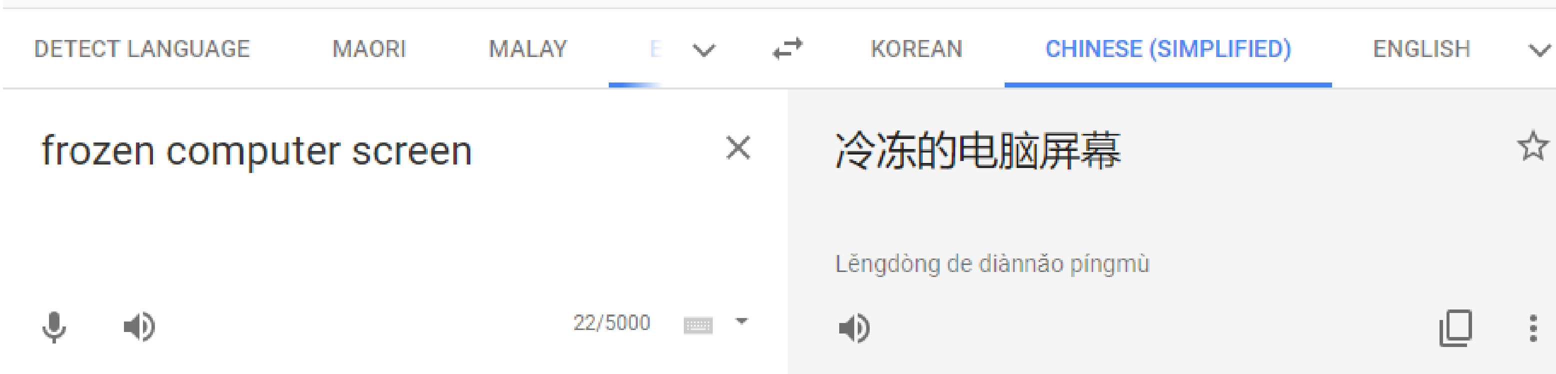
3 Neural Machine Translation

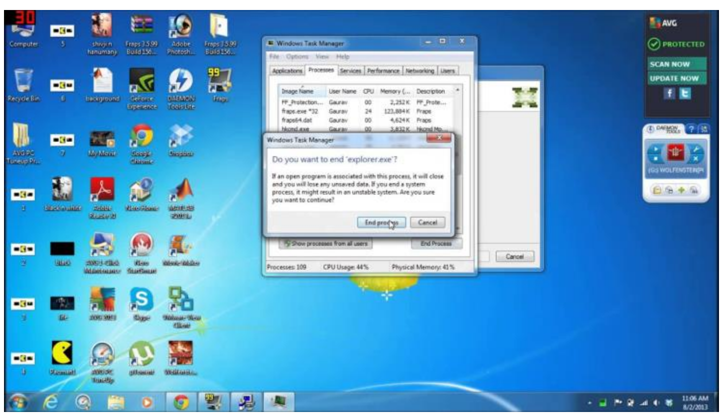
However, there are still several difficulties…

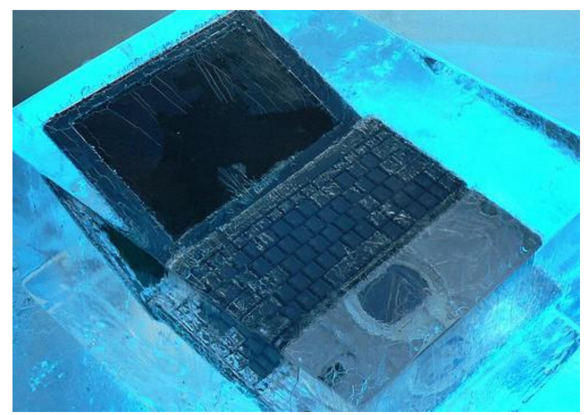
- ***Out-of-vocabulary (OOV)*** *words*
- *Domain* ***mismatch*** *between train and test data*
- *Maintaining context over* ***longer text***
- ***Low-resource*** *language pairs*

3 Neural Machine Translation

Machine Translation is not PERFECT…



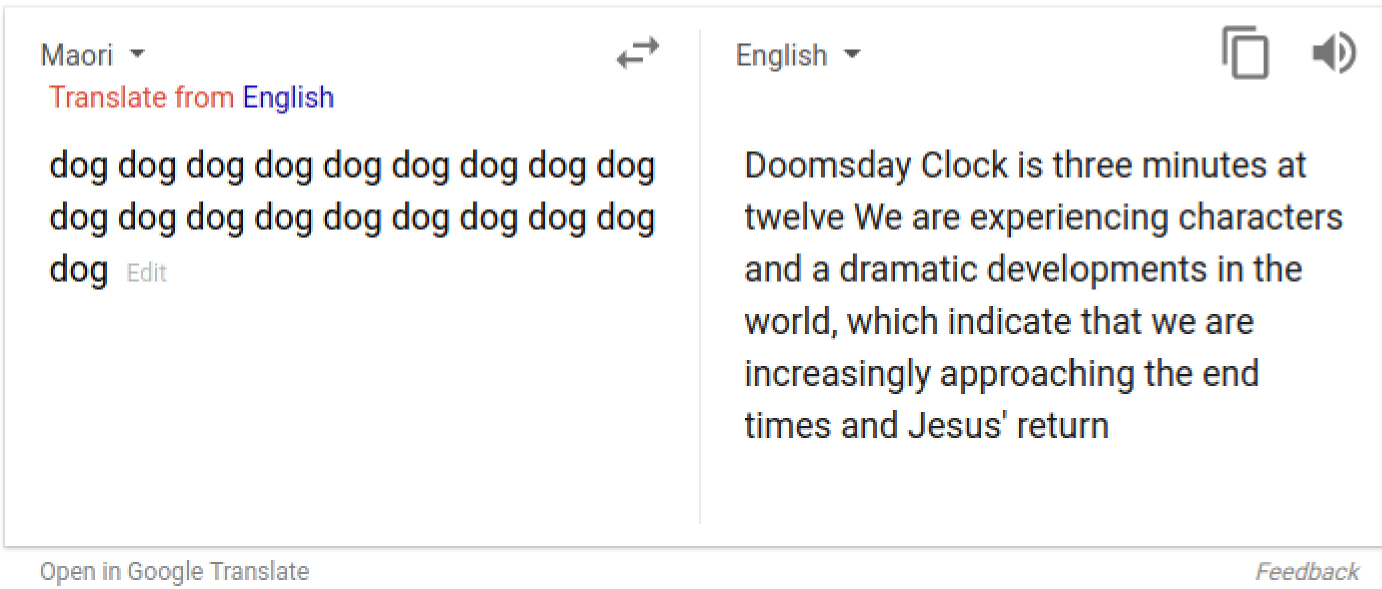




Using common sense is still hard and **NMT** picks up biases in training data

3 Neural Machine Translation

Machine Translation is not PERFECT…



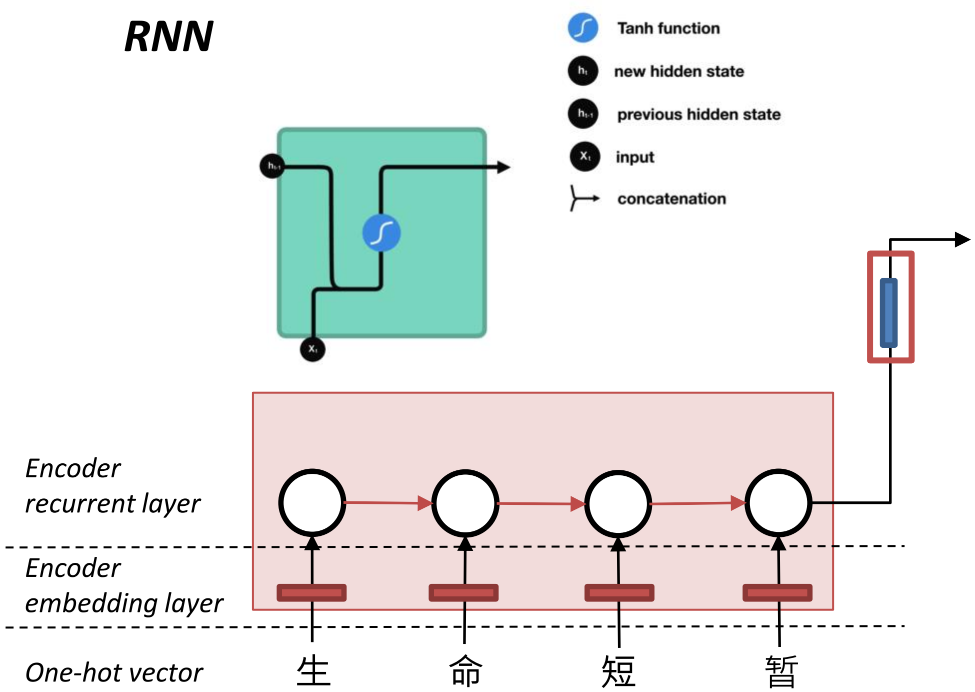
Uninterpretable systems do strange things

https://www.vice.com/en_us/article/j5npeg/why-is-google-translate-spitting-out-sinister-religious-prophecies

3 Neural Machine Translation

Neural Machine Translation with Seq2Seq

RNN-based neural MT was sort of successful! But…



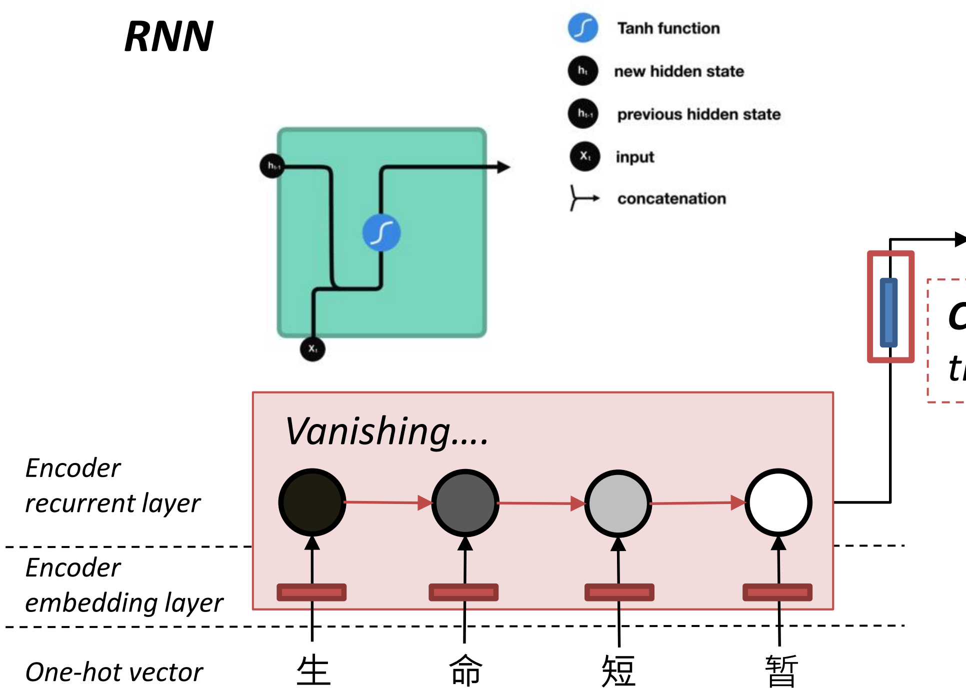
Source sentence

[Encoder]

3 Neural Machine Translation

Neural Machine Translation with Seq2Seq

RNN-based neural MT was sort of successful! But…

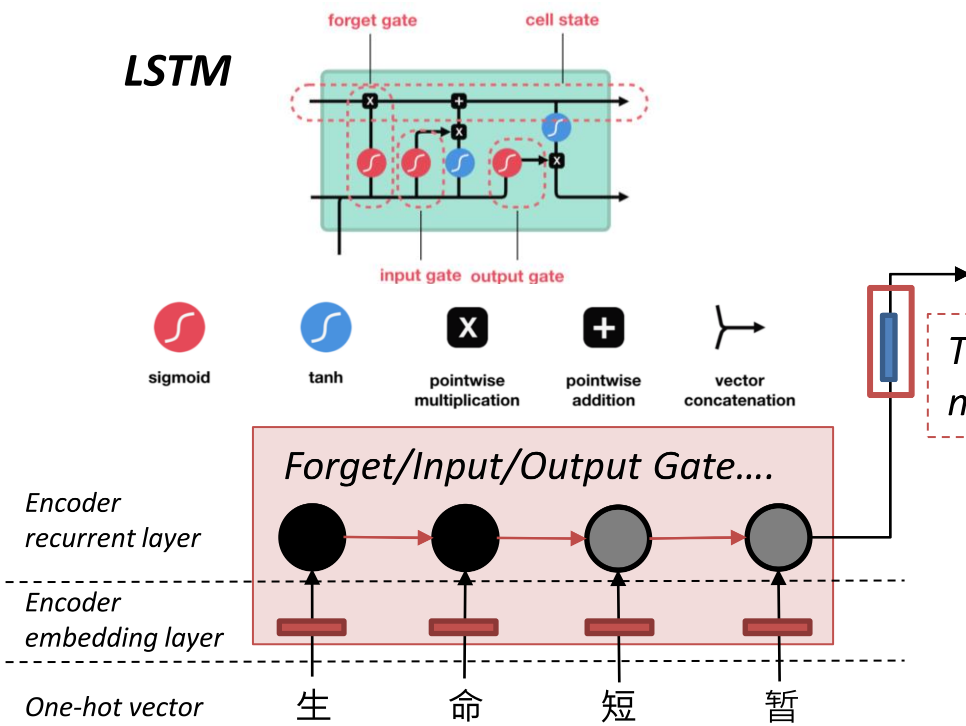


Cannot remember *all information about the source sentence*

Source sentence
[Encoder]

Neural Machine Translation with Seq2Seq

RNN-based neural MT was successful! But…



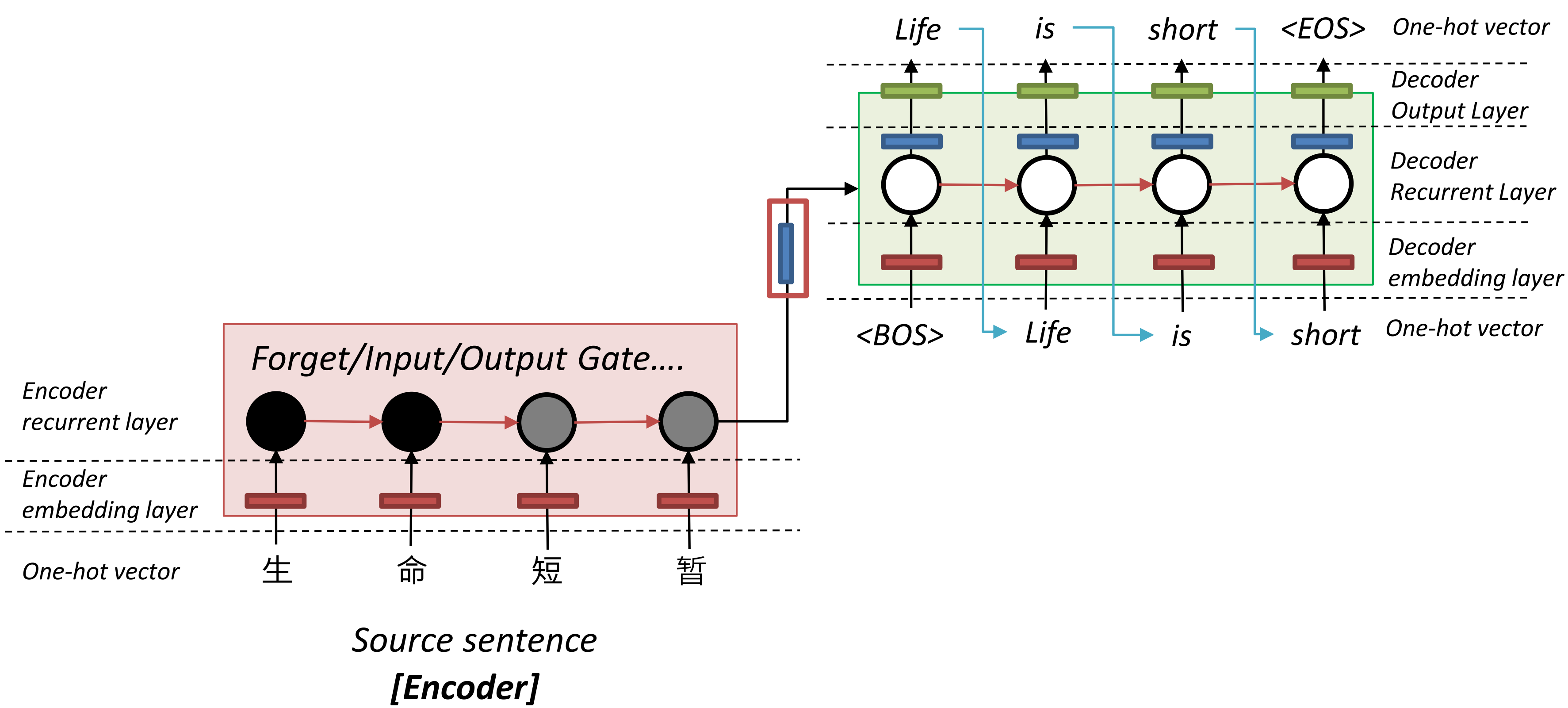
*They **can remember sequences of 100s**, not 1000s or 10,000s or more.*

Source sentence
[Encoder]

Neural Machine Translation with Seq2Seq

Then, how to solve the information bottleneck issue?

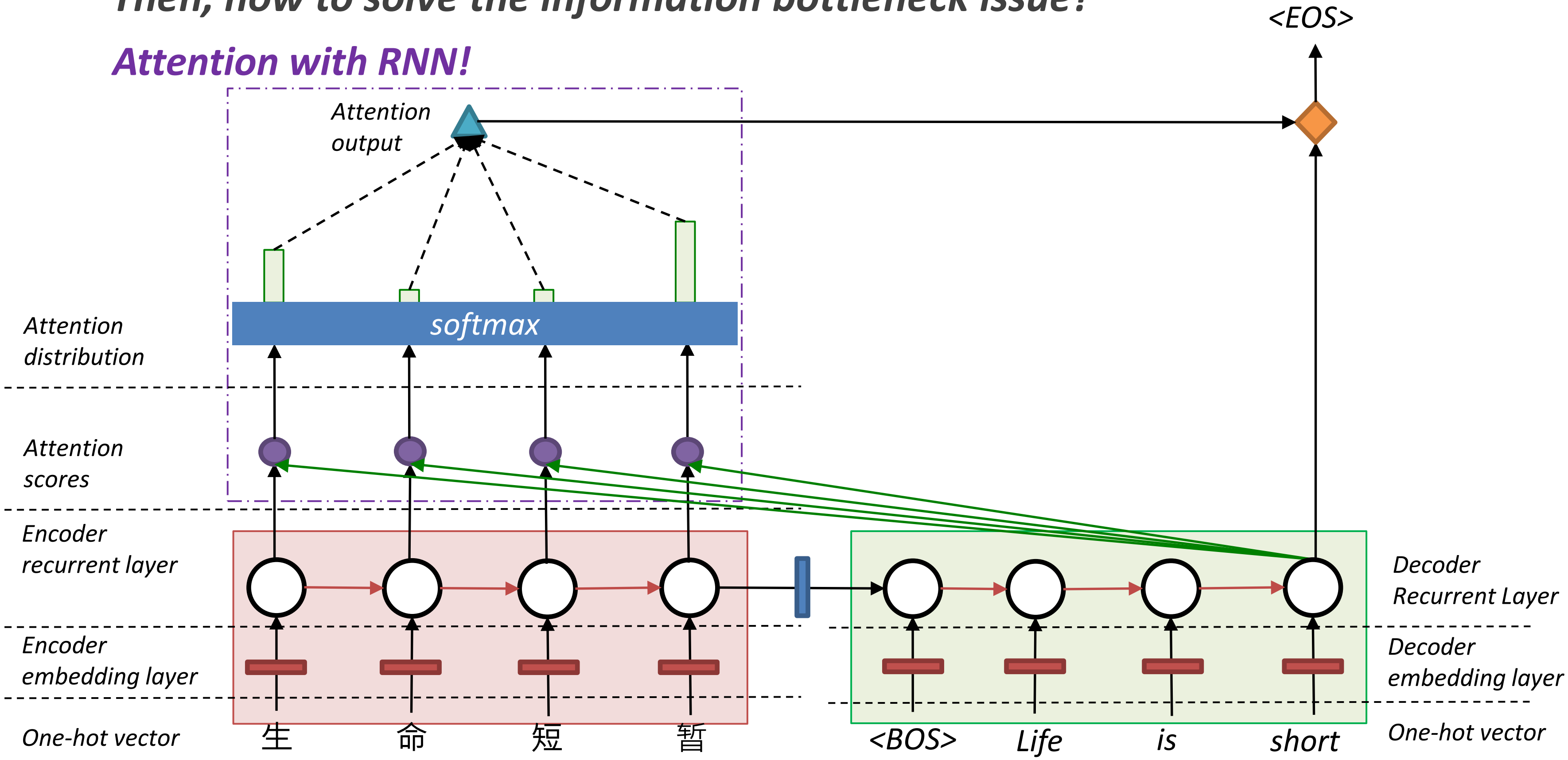
Attention!



Neural Machine Translation with RNN and Attention

Then, how to solve the information bottleneck issue?

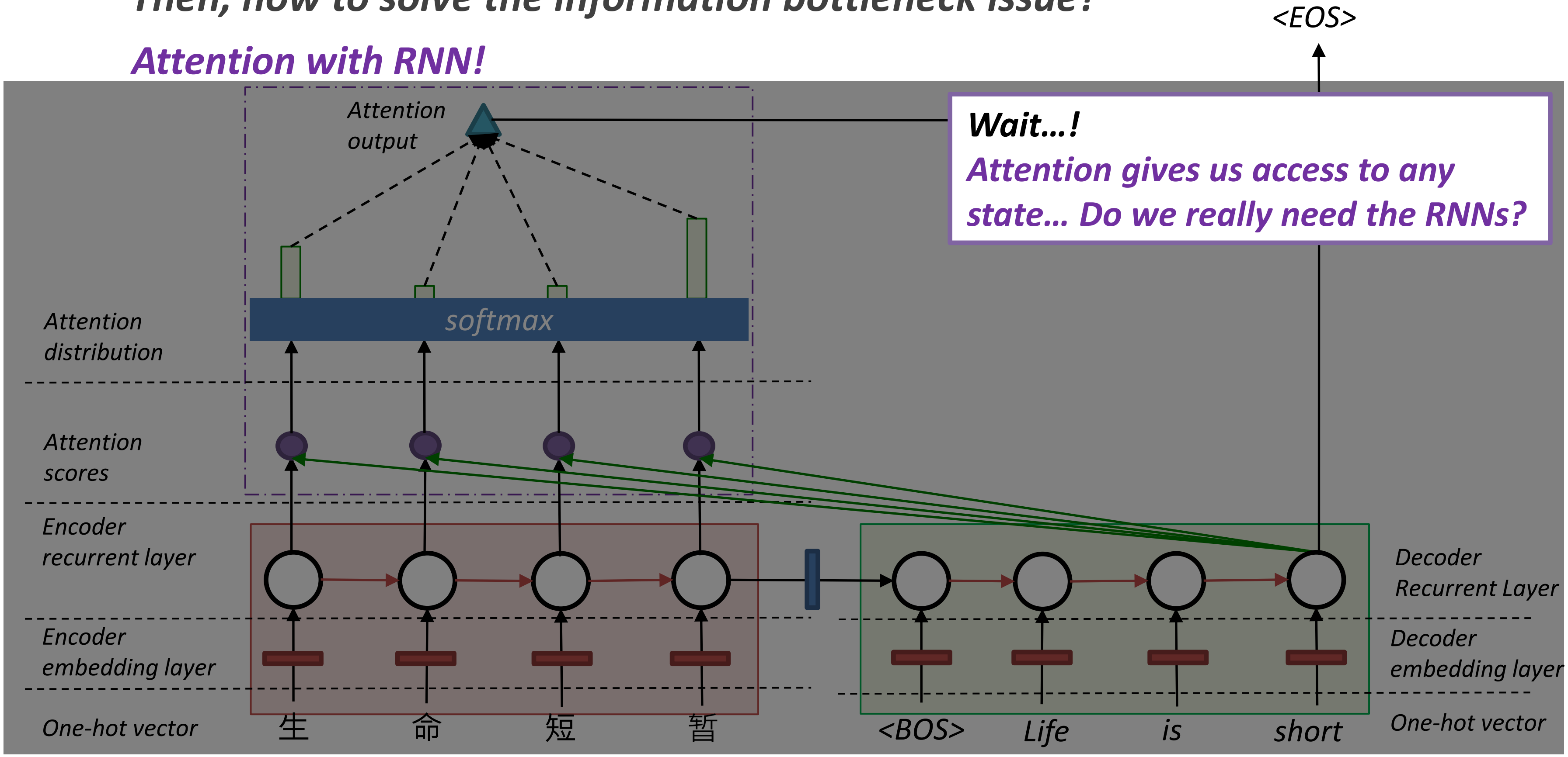
Attention with RNN!



Neural Machine Translation with RNN and Attention

Then, how to solve the information bottleneck issue?

Attention with RNN!

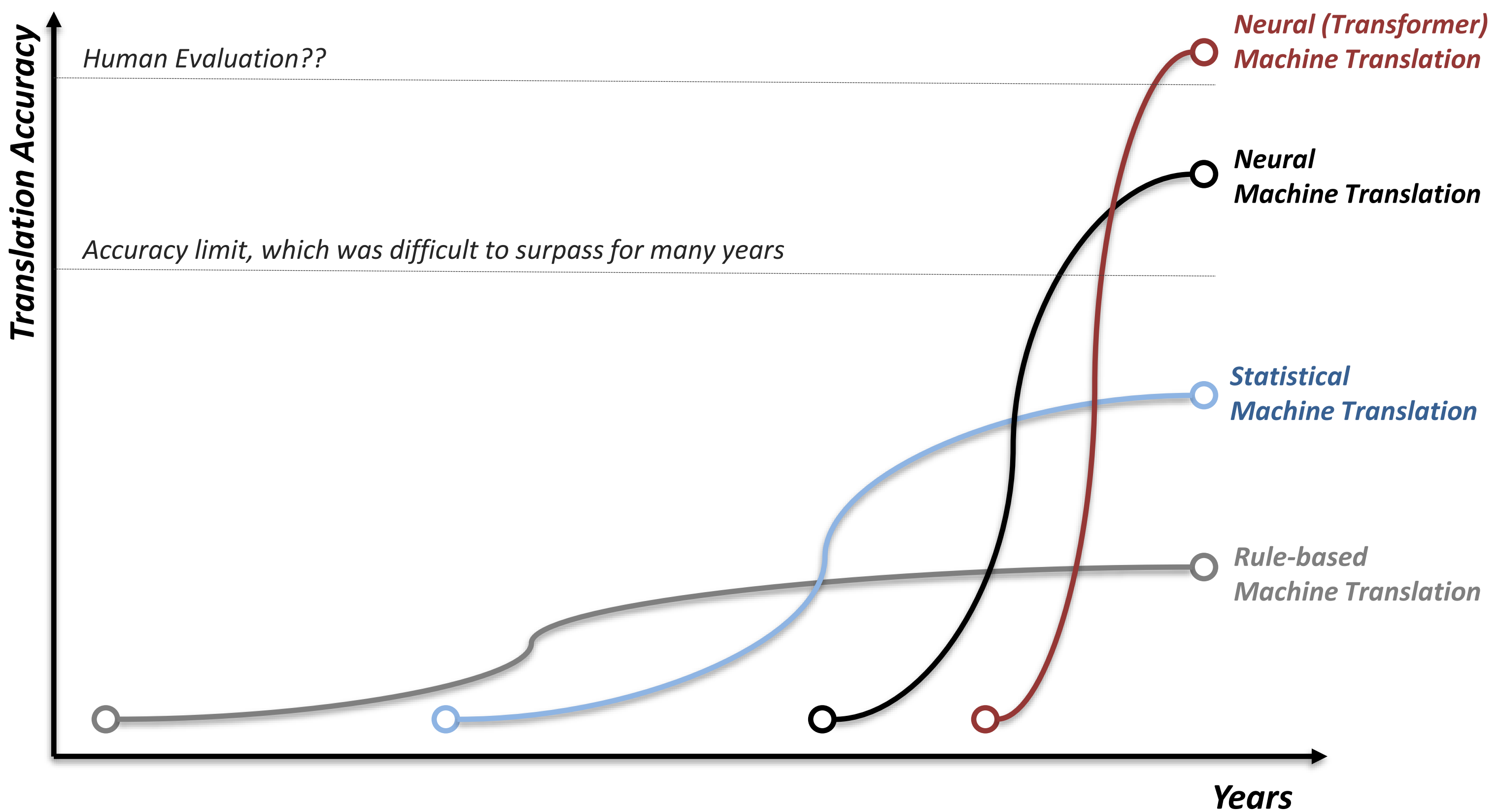


4 Attention and Transformer for MT

Early 2018 ~

4 Attention and Transformer for MT

Machine Translation progress over time



4 Attention and Transformer for MT

Attention is All You Need (Vaswani et al., 2017)


Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

Encoder-Decoder with only Attention

Core Task: Machine Translation with Parallel Corpus

- *Use self-attention in the encoder, instead of RNN or CNNs*
- *Predict each translated word*
- *Final cost/error function*

→ *standard cross-entropy error on top of a softmax classifier*

Attention is All You Need!

'The Transformer'!!

Output (Target Language)

Hello World

↑ ↑

The Transformer!

↑ ↑

こんにちは世界

Input (Source Language)

4 Attention and Transformer for MT

Attention is All You Need (Vaswani et al., 2017)

Encoder-Decoder with only Attention

Core Task: Machine Translation with Parallel Corpus

- *Use self-attention in the encoder, instead of RNN or CNNs*
- *Predict each translated word*
- *Final cost/error function*

→ *standard cross-entropy error on top of a softmax classifier*

Attention Is All You Need

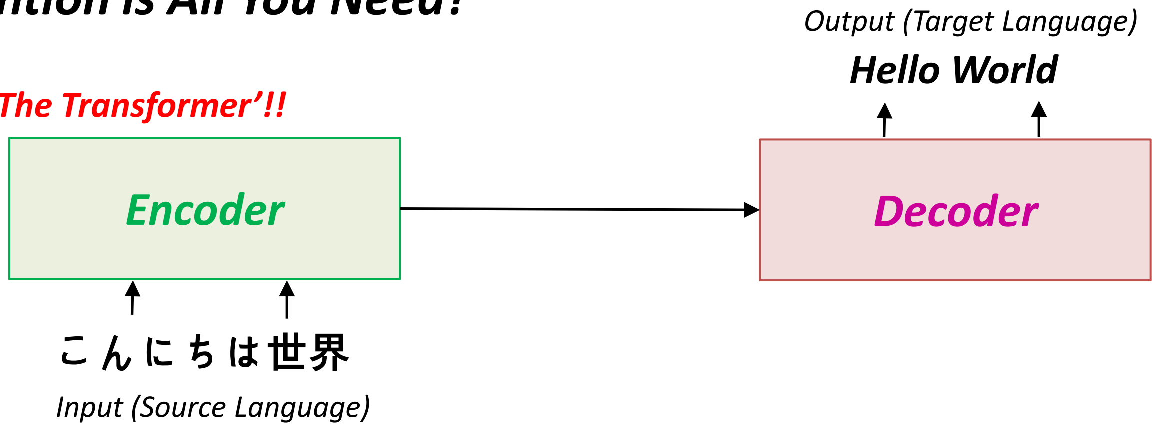
Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

Attention is All You Need?

'The Transformer'!!



Attention and Transformer for MT

The Transformer 

Encoder – Decoder Architecture

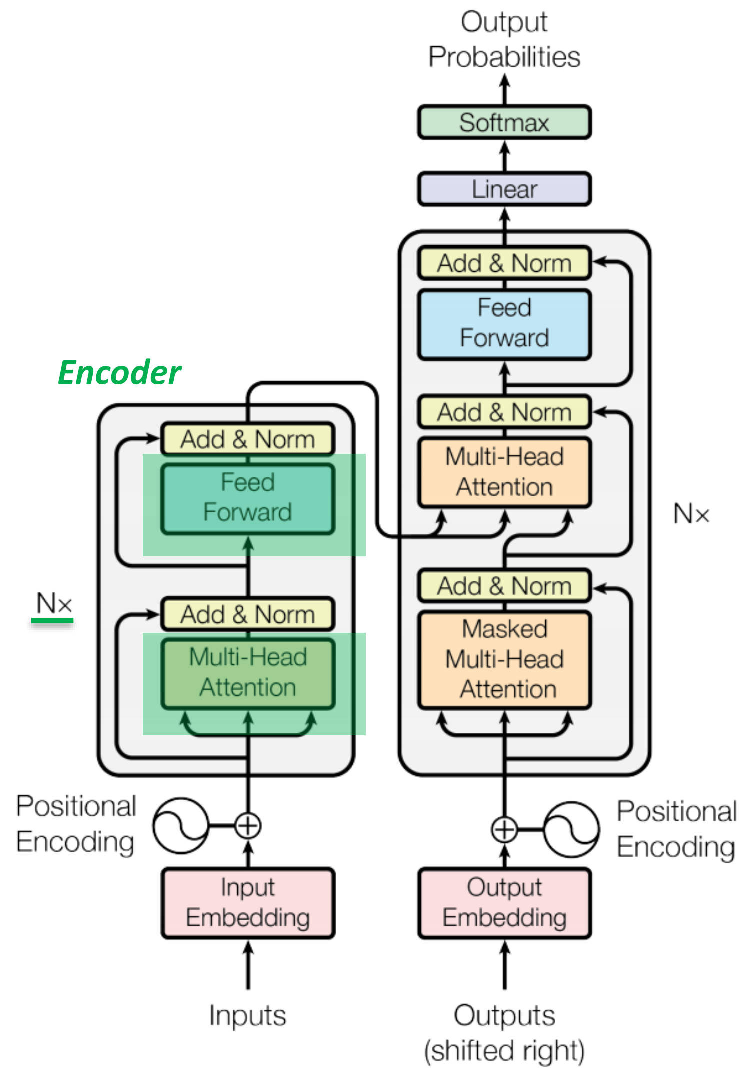
1. Encoder

A stack of N=6 identical layers.

Each layer with <u>two sub-layers:</u>

1. *Multi-head self-attention mechanism*
2. *Position-wise fully connected feed-forward network*

** Residual connection around each of the two sub-layers, followed by layer normalisation*



The transformer – model architecture

Attention and Transformer for MT

The Transformer 

Encoder – Decoder Architecture

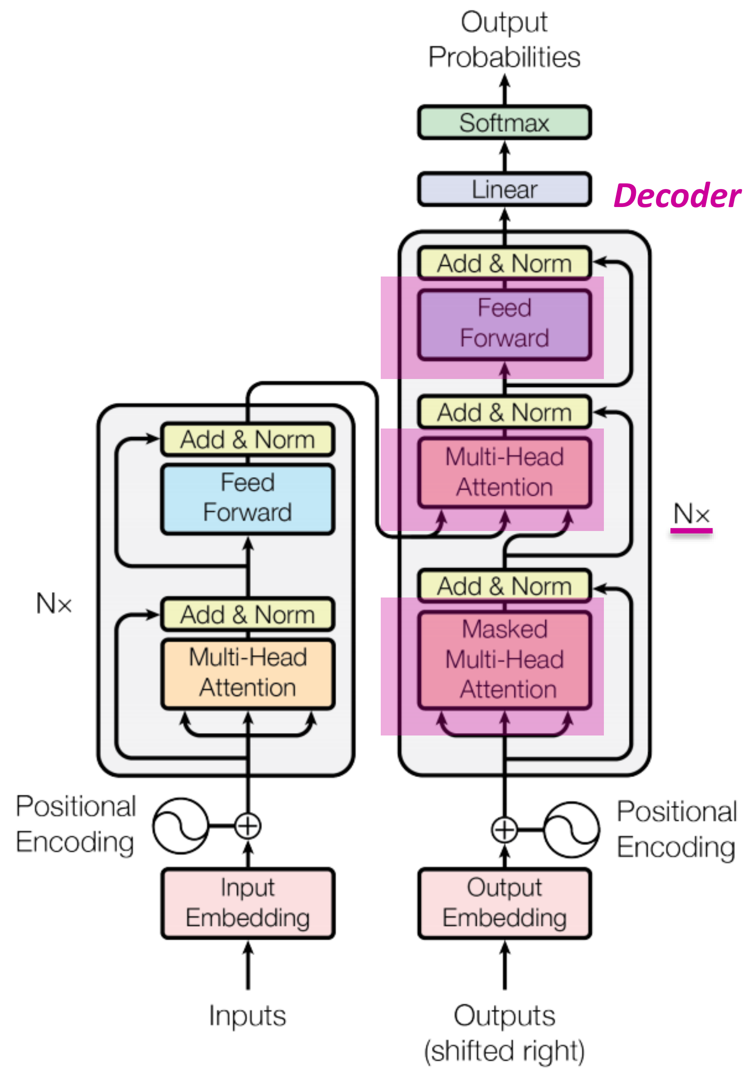
2. Decoder

A stack of N=6 identical layers.

Each layer with three sub-layers:

1. *Multi-head self-attention mechanism*
2. *Position-wise fully connected feed-forward network*
3. *Masked Multi-head self-attention*

** Residual connection around each of the two sub-layers, followed by layer normalisation*



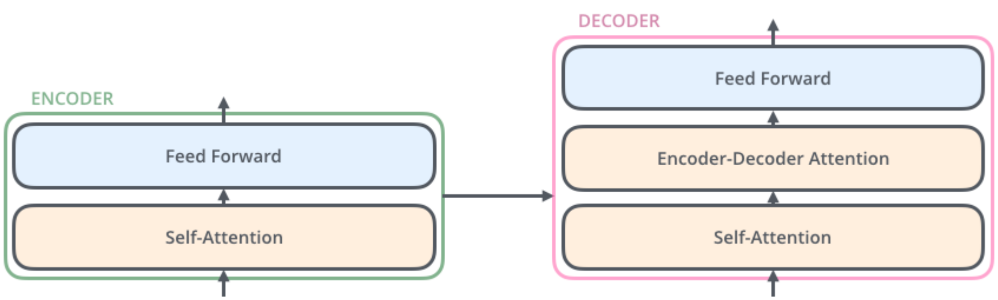
The transformer – model architecture

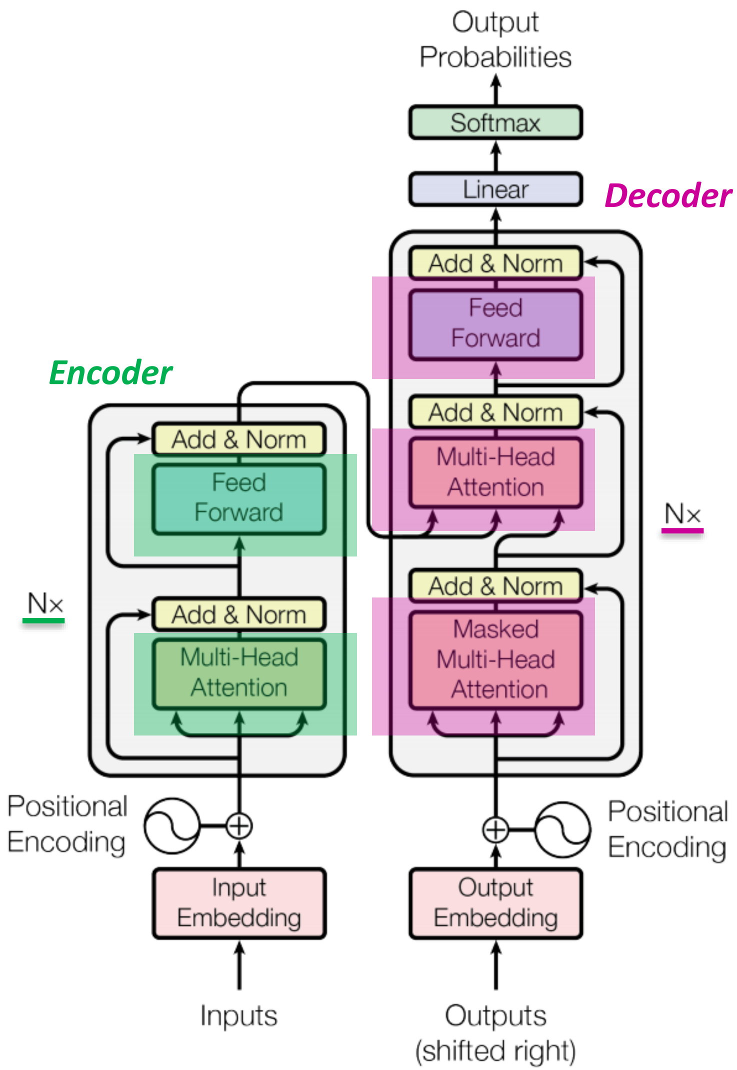
4 Attention and Transformer for MT

The Transformer 

Encoder – Decoder Architecture

Brief Summary





The transformer – model architecture

4 Attention and Transformer for MT

The Transformer – Encoder  (Stage1)

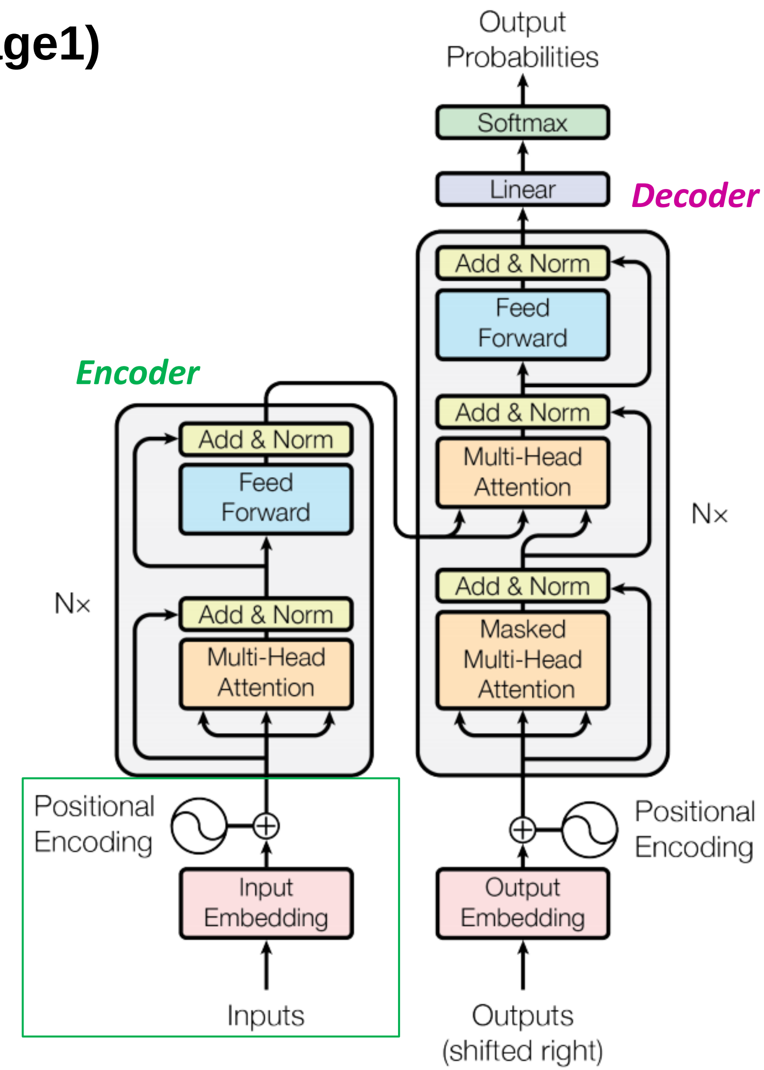
We are not using RNN anymore… No time step concept!

> *To make use of* ***the order of the sequence****,* ***inject*** *information about* ***the position of the tokens*** *in the sequence.*

Positional Encoding
(use sin and cos for position/dimension)

Input embedding
(a vector of size 512)

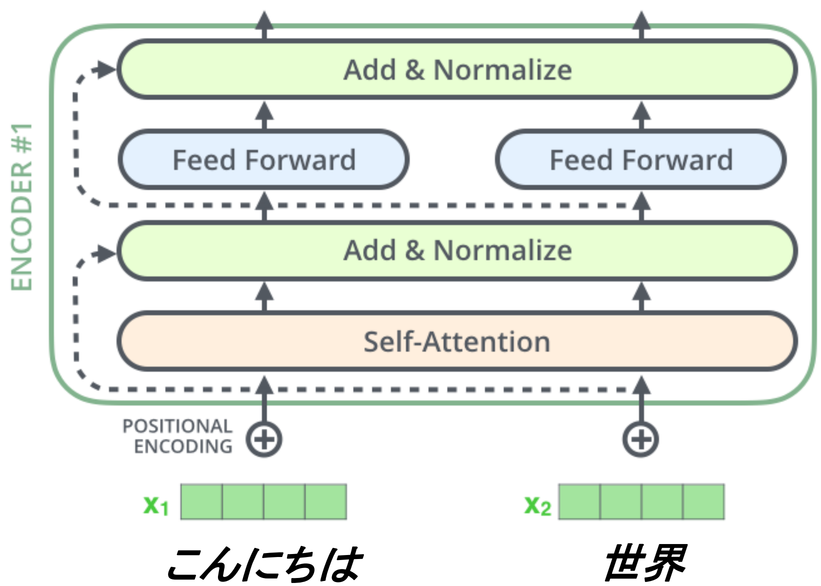
こんにちは 世界

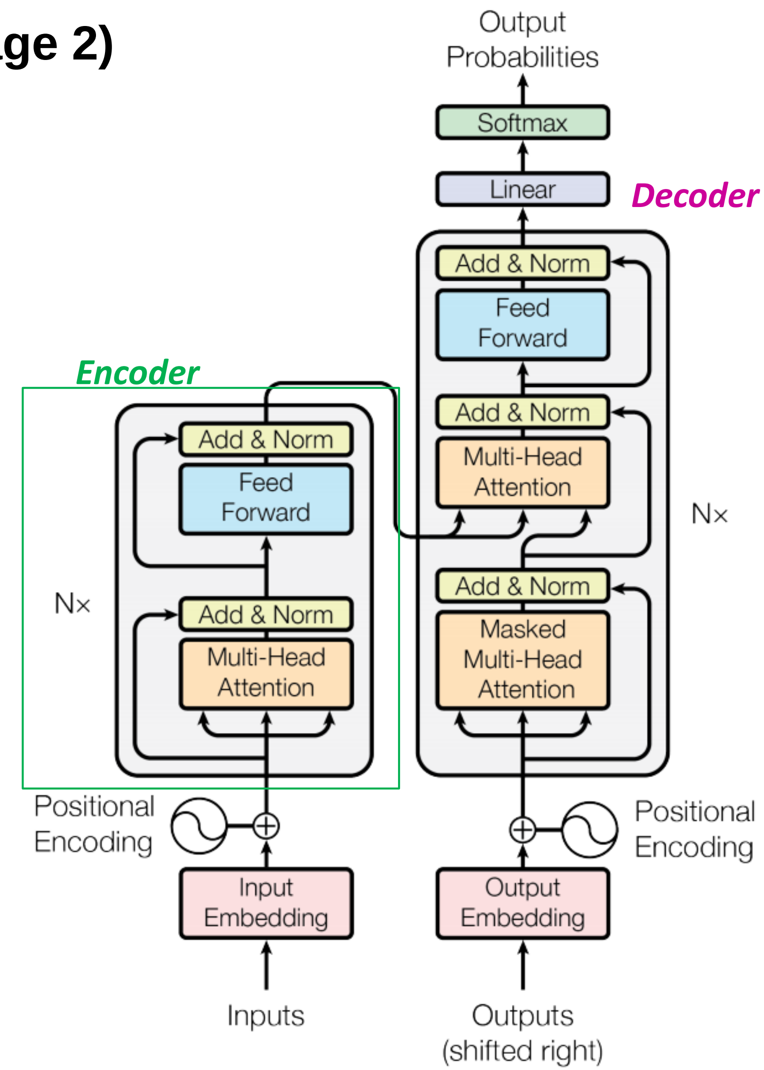


The transformer – model architecture

Attention and Transformer for MT

The Transformer – Encoder  (Stage 2)

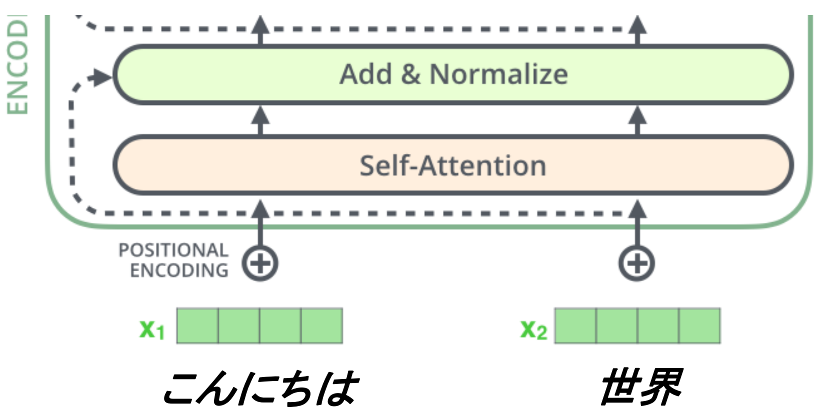


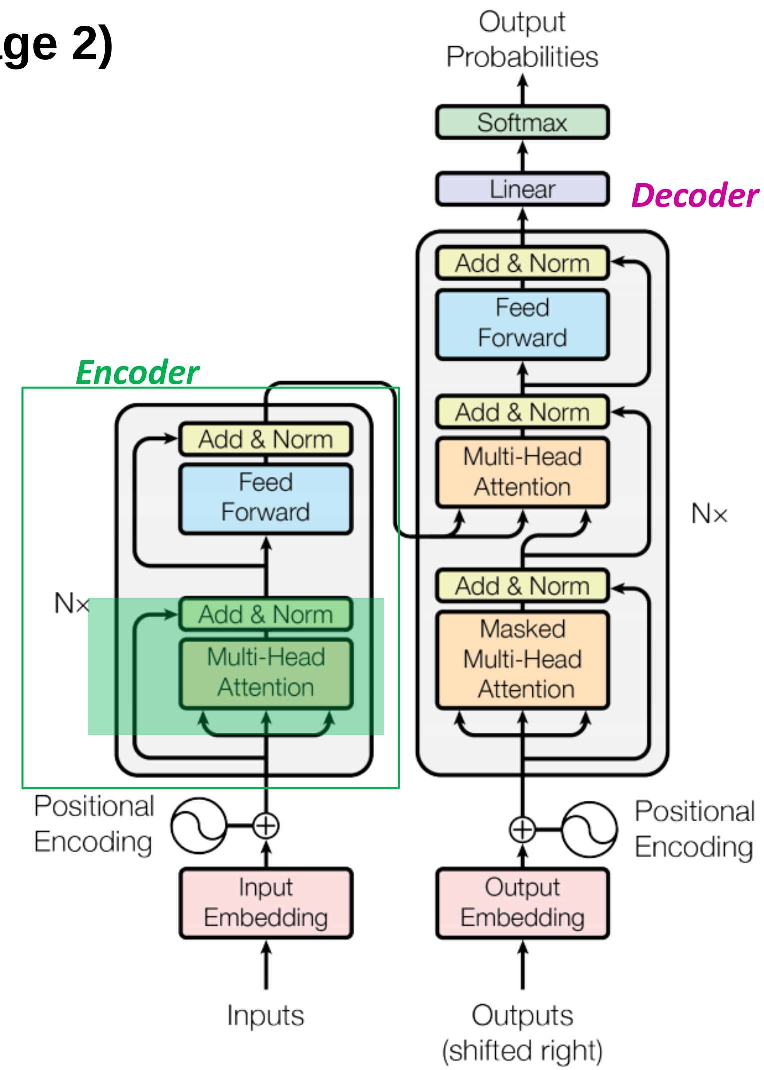


The transformer – model architecture

Attention and Transformer for MT

The Transformer – Encoder  (Stage 2)



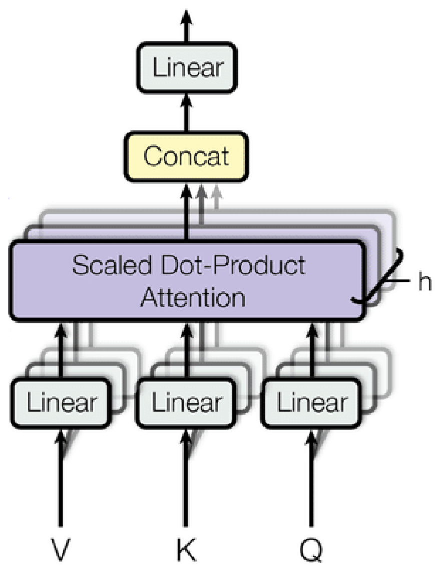


The transformer – model architecture

Attention and Transformer for MT

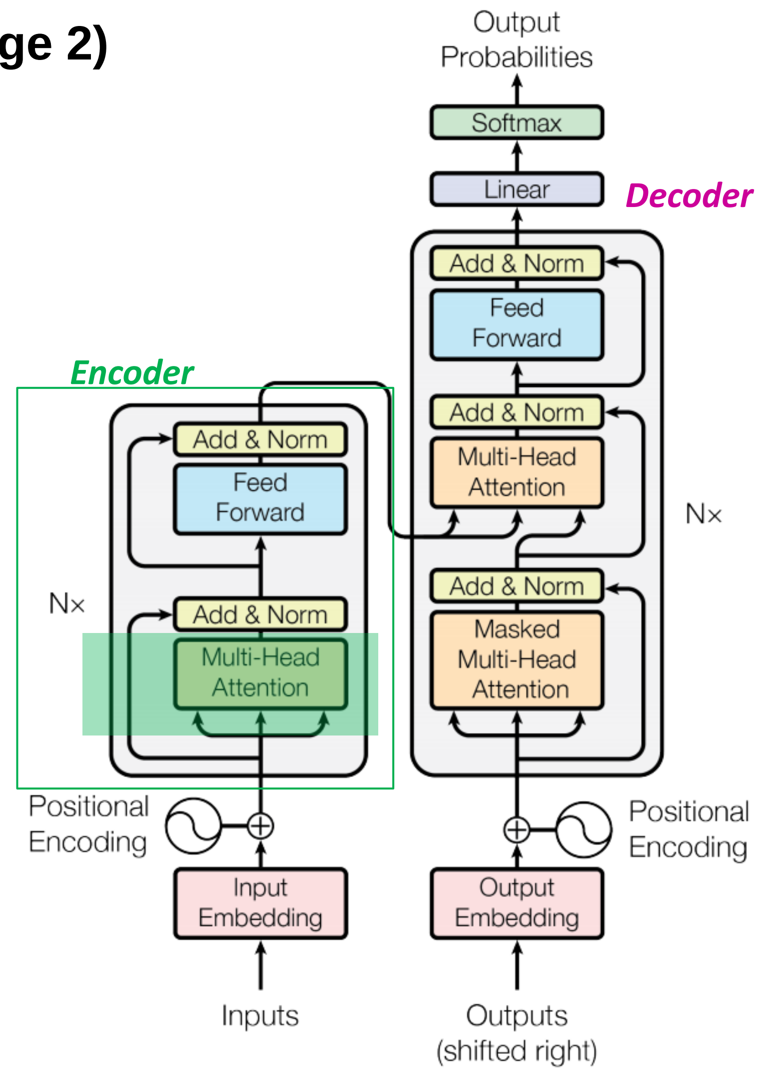
The Transformer – Encoder  (Stage 2)

Multi-head attention** allows the model to jointly attend to **information from different representation subspaces at different positions



Multi-Head Attention
(With self attention)

Q=Query, K=Key, V=Value (64 dimension)

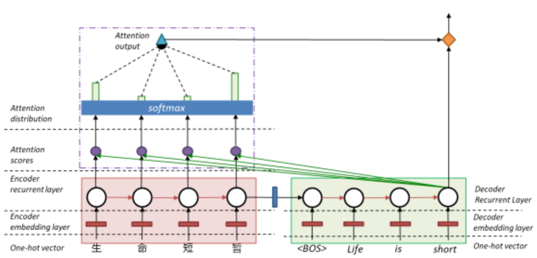


The transformer – model architecture

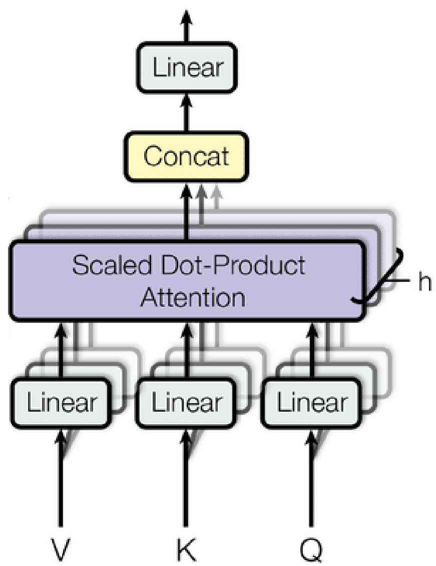
Attention and Transformer for MT

The Transformer – Encoder  (Stage 2)

Multi-head attention** allows the model to jointly attend to **information from different representation subspaces at different positions

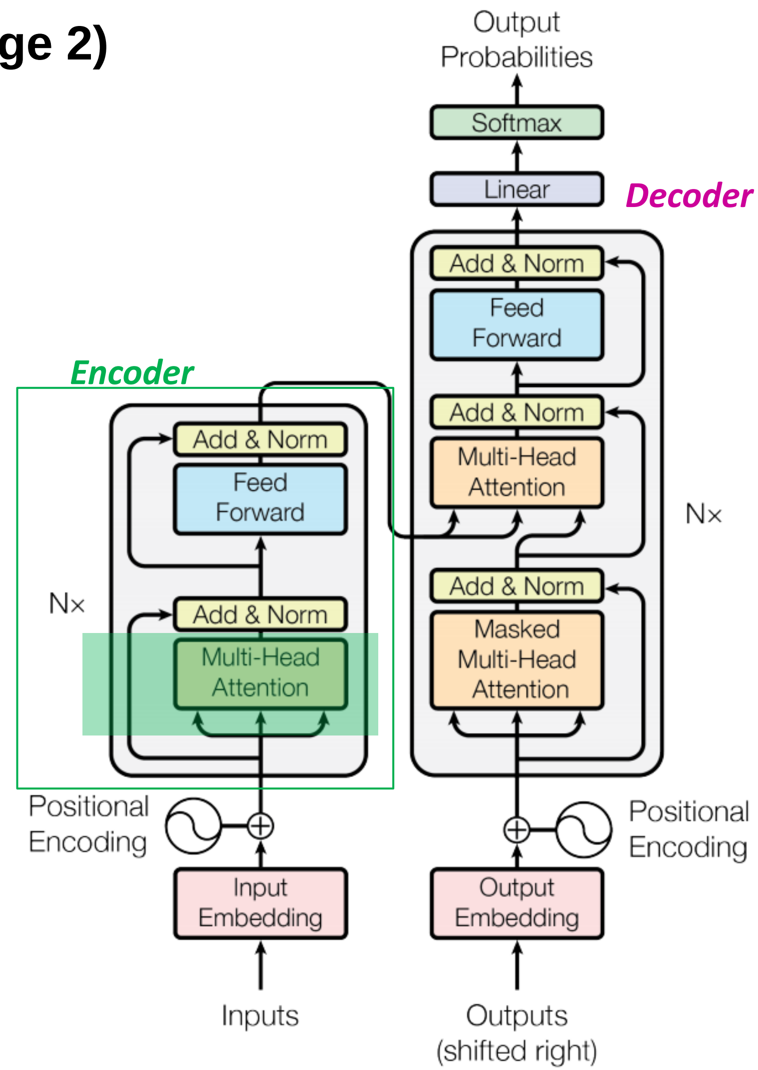


Source and Target attention



Multi-Head Attention
(With self attention)

Q=Query, K=Key, V=Value (64 dimension)

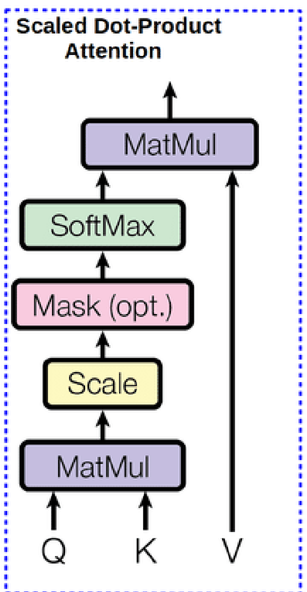


The transformer – model architecture

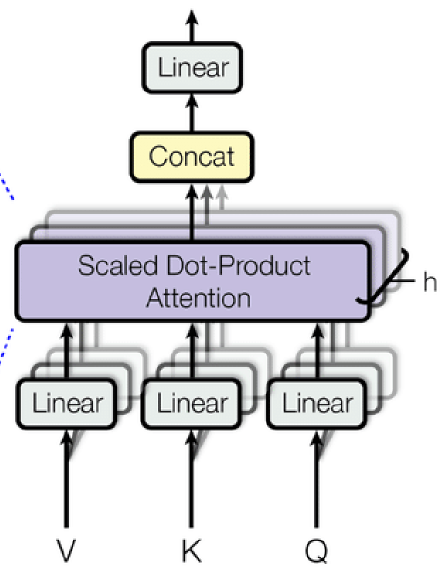
Attention and Transformer for MT

The Transformer – Encoder  (Stage 2)

Multi-head attention *allows the model to jointly attend to* ***information from different representation subspaces at different positions***

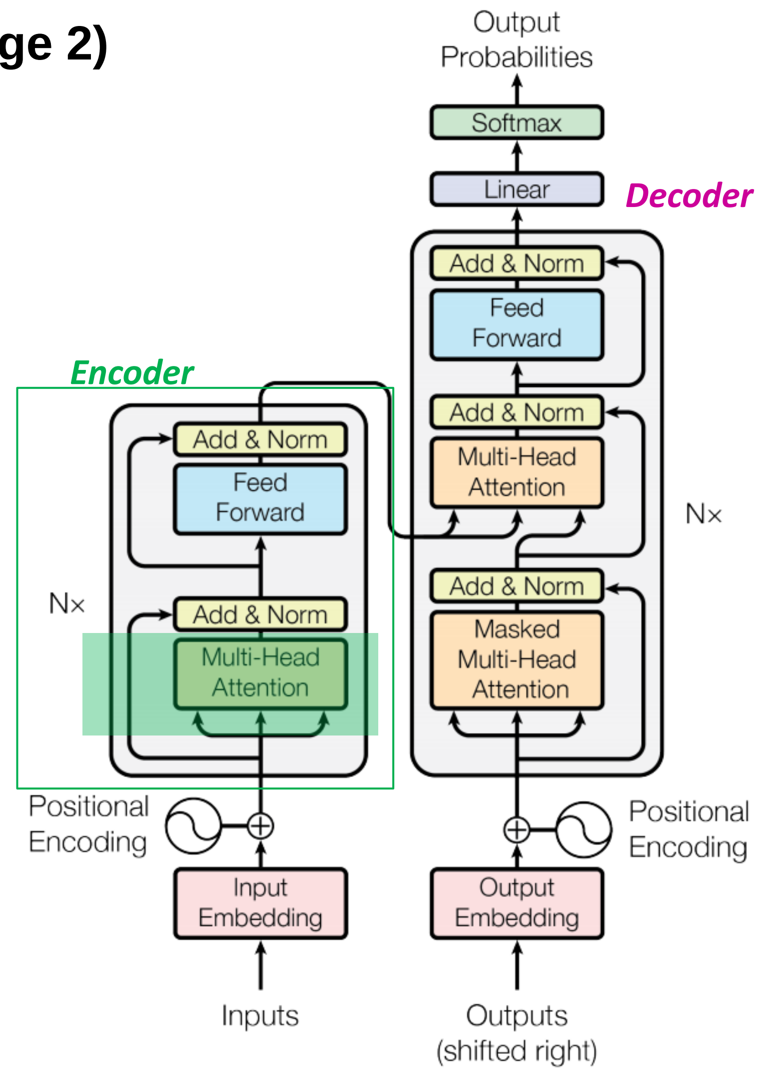


Self-attention



Multi-Head Attention
(With self attention)

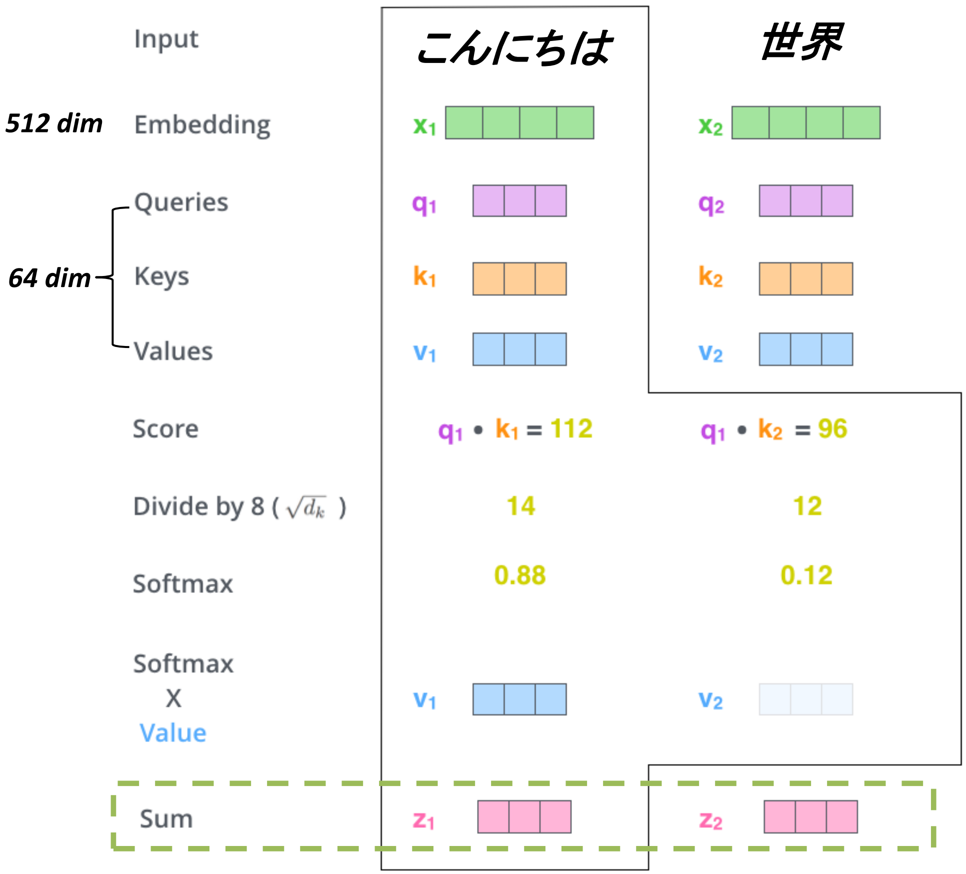
Q=Query, K=Key, V=Value
(64 dimension)

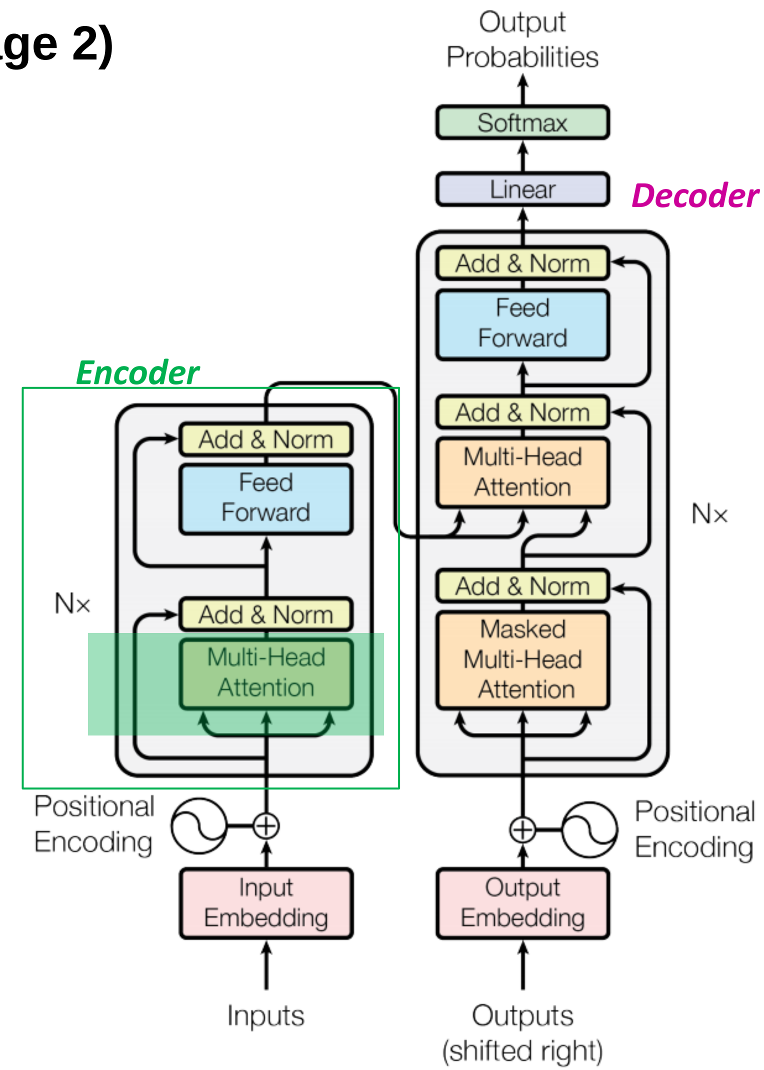


The transformer – model architecture

4 Attention and Transformer for MT

The Transformer – Encoder  (Stage 2)

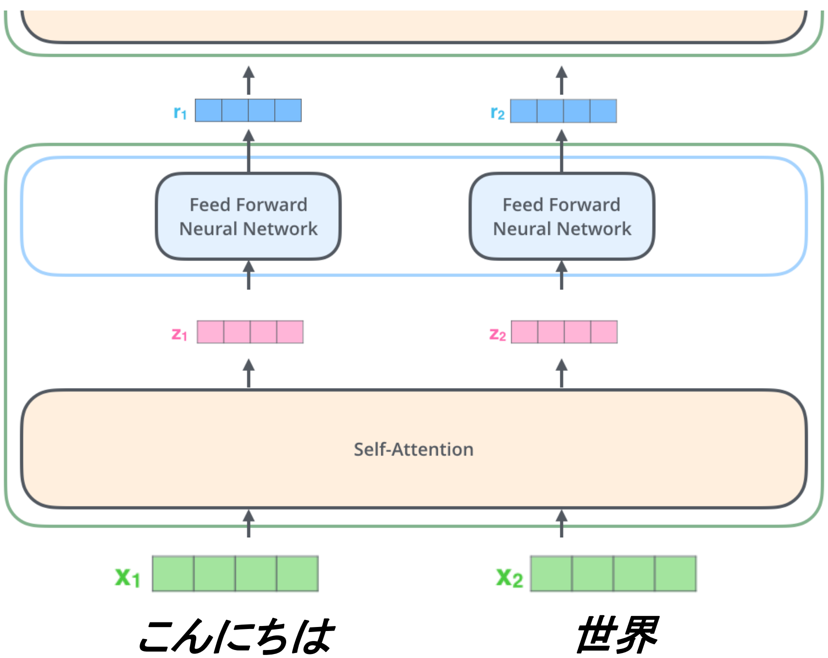


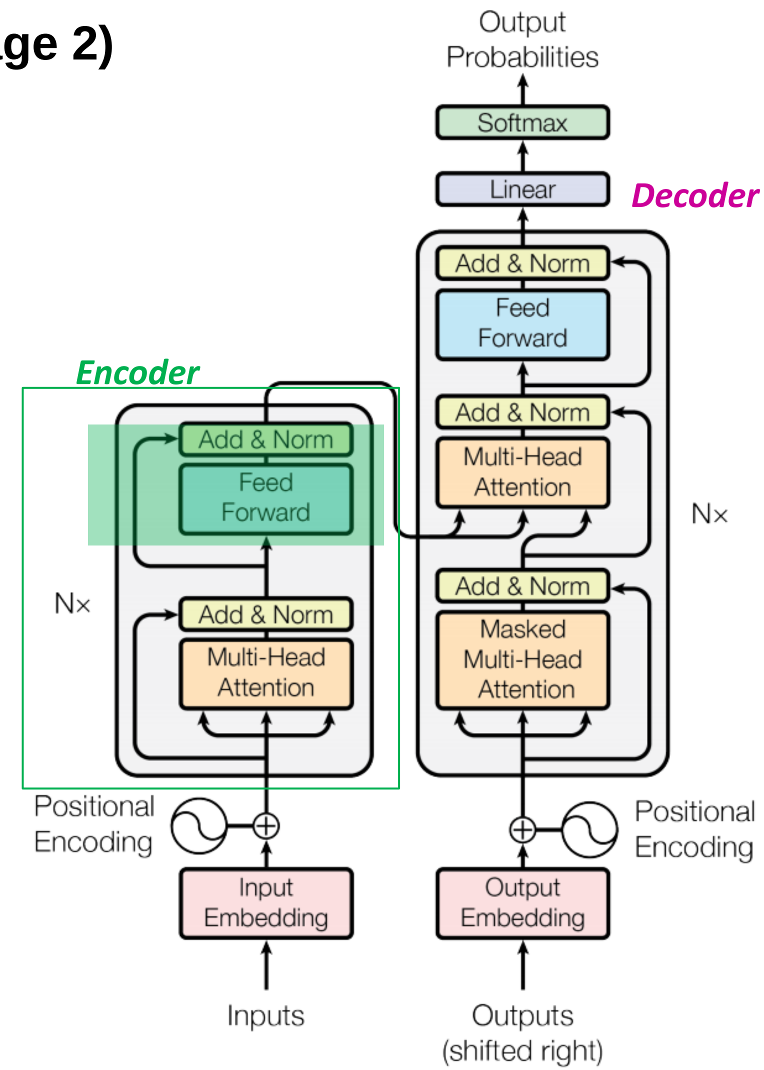


The transformer – model architecture

4 Attention and Transformer for MT

The Transformer – Encoder  (Stage 2)

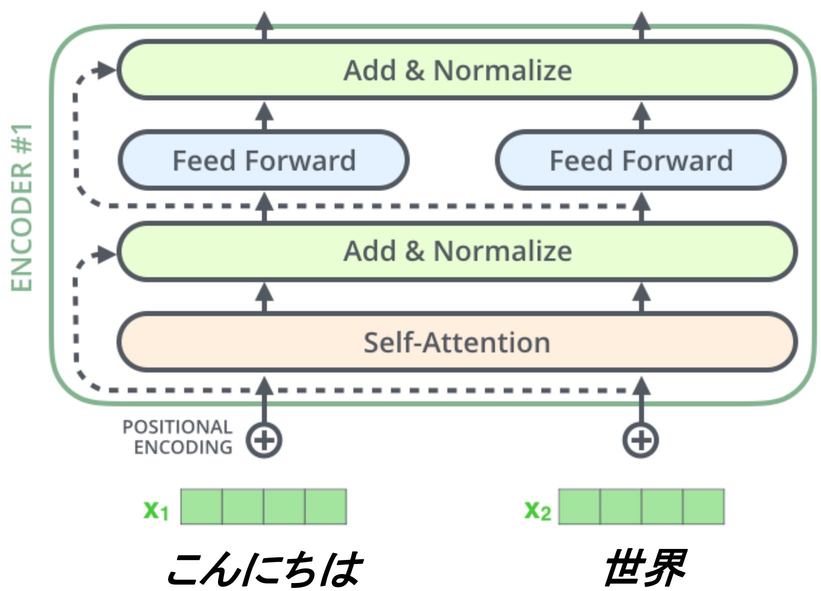


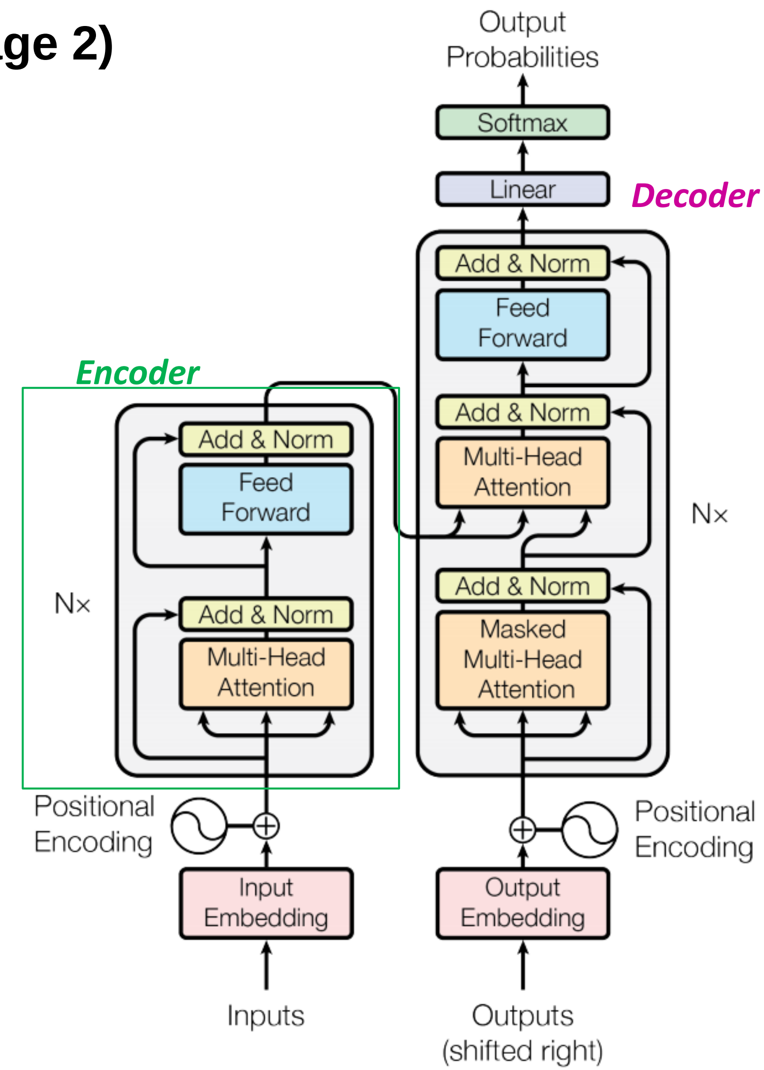


The transformer – model architecture

Attention and Transformer for MT

The Transformer – Encoder  (Stage 2)

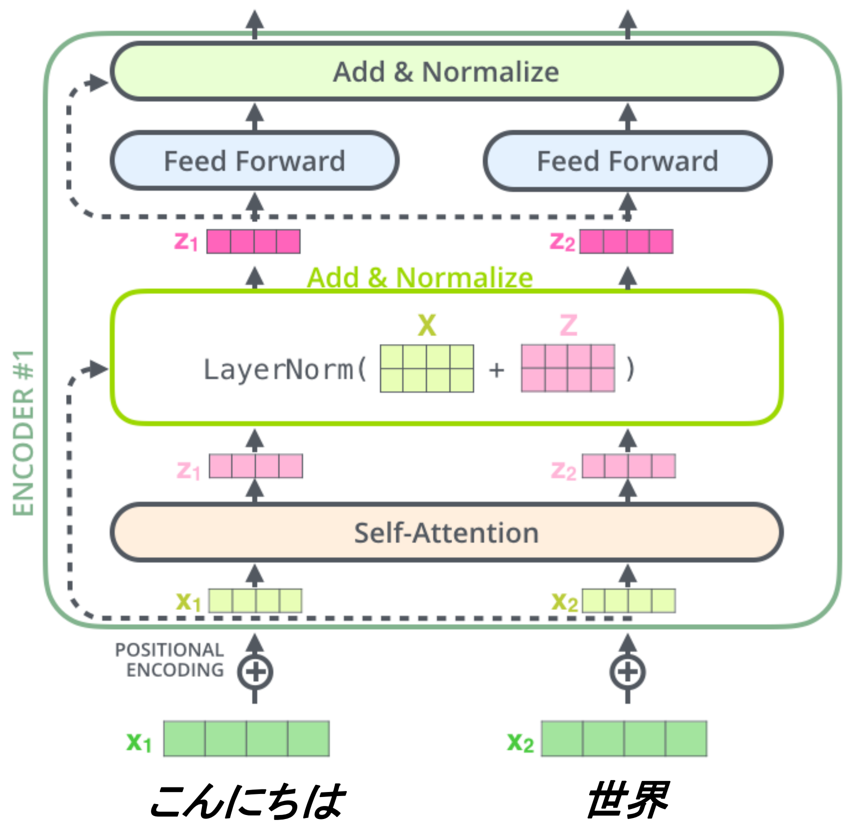


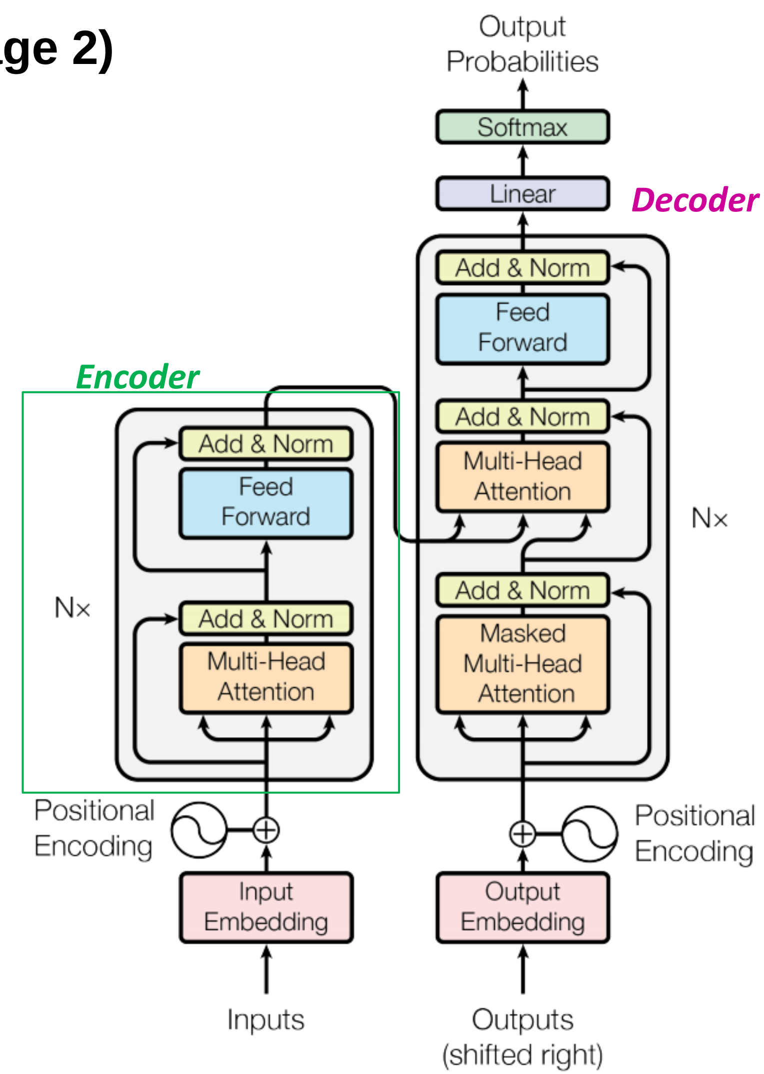


The transformer – model architecture

4 Attention and Transformer for MT

The Transformer – Encoder  (Stage 2)

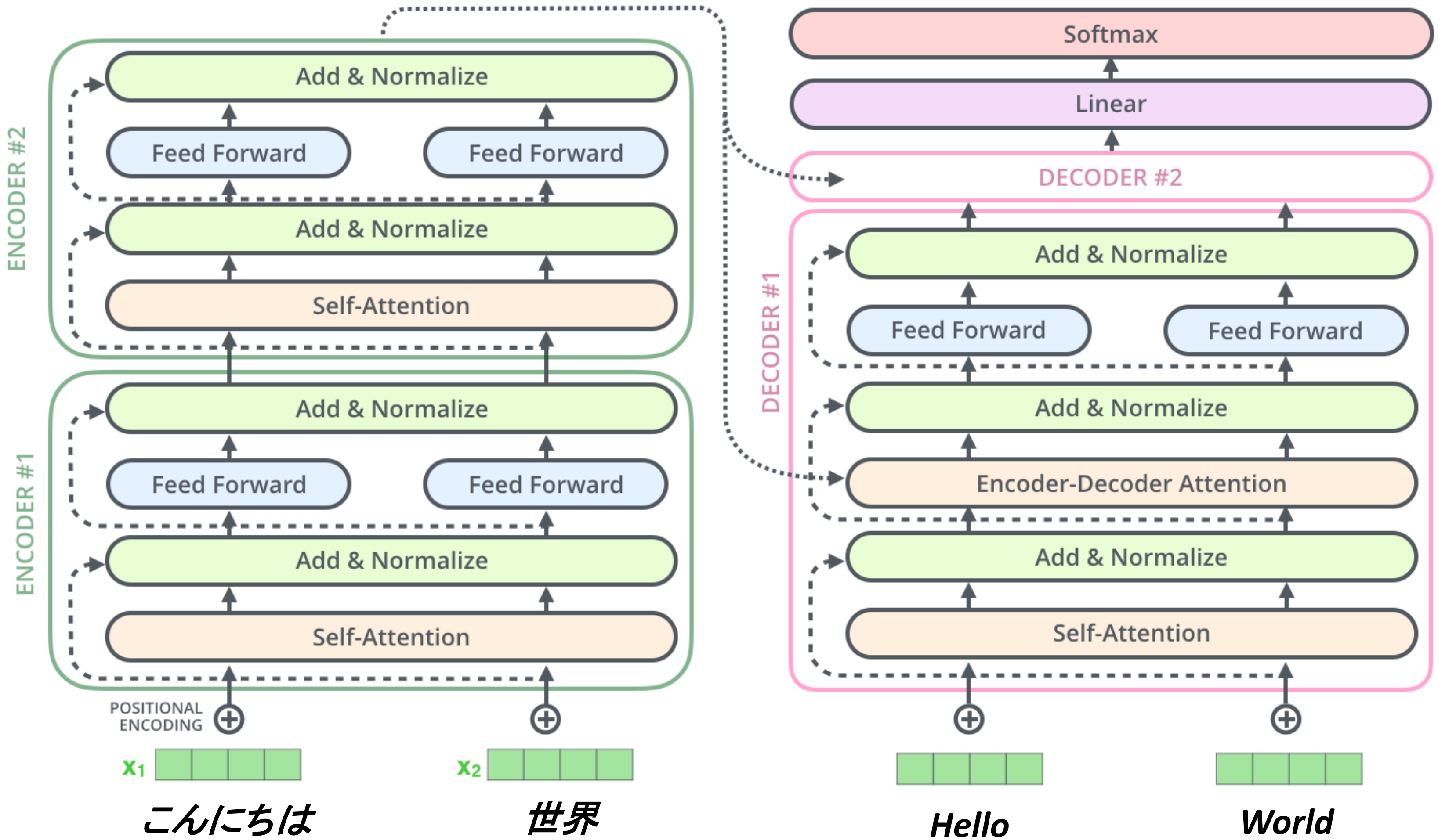




The transformer – model architecture

4 Attention and Transformer for MT

The Transformer – Encoder to Decoder



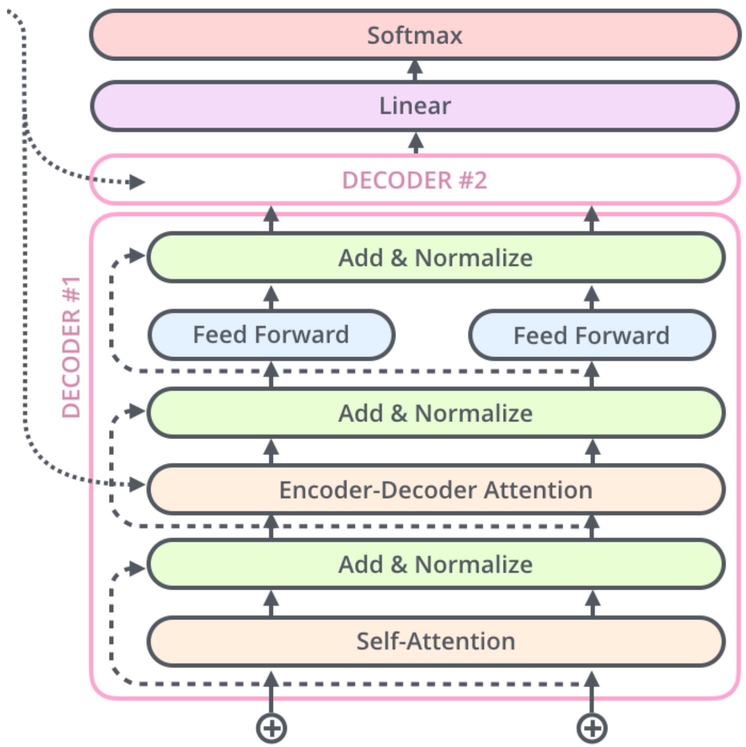
4 Attention and Transformer for MT

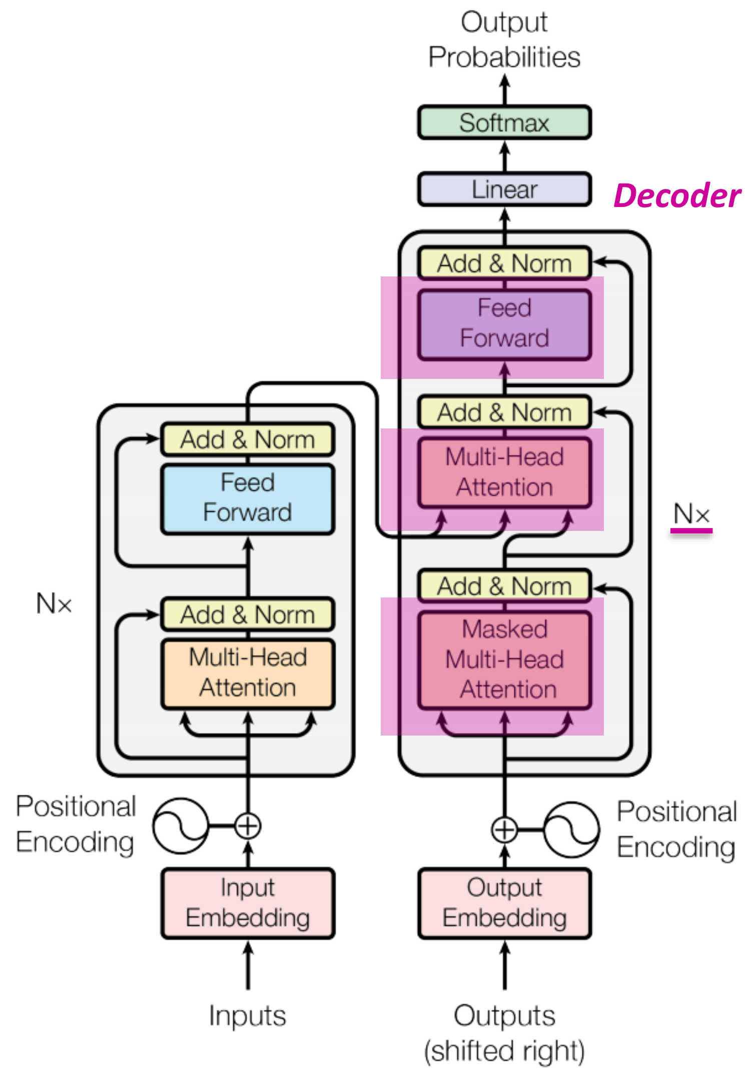
The Transformer - Decoder

Hello

*Label [0, 0, ..., **1,** 0]*

Output[0.1, 0.01,........................., 0.8, 0.1]

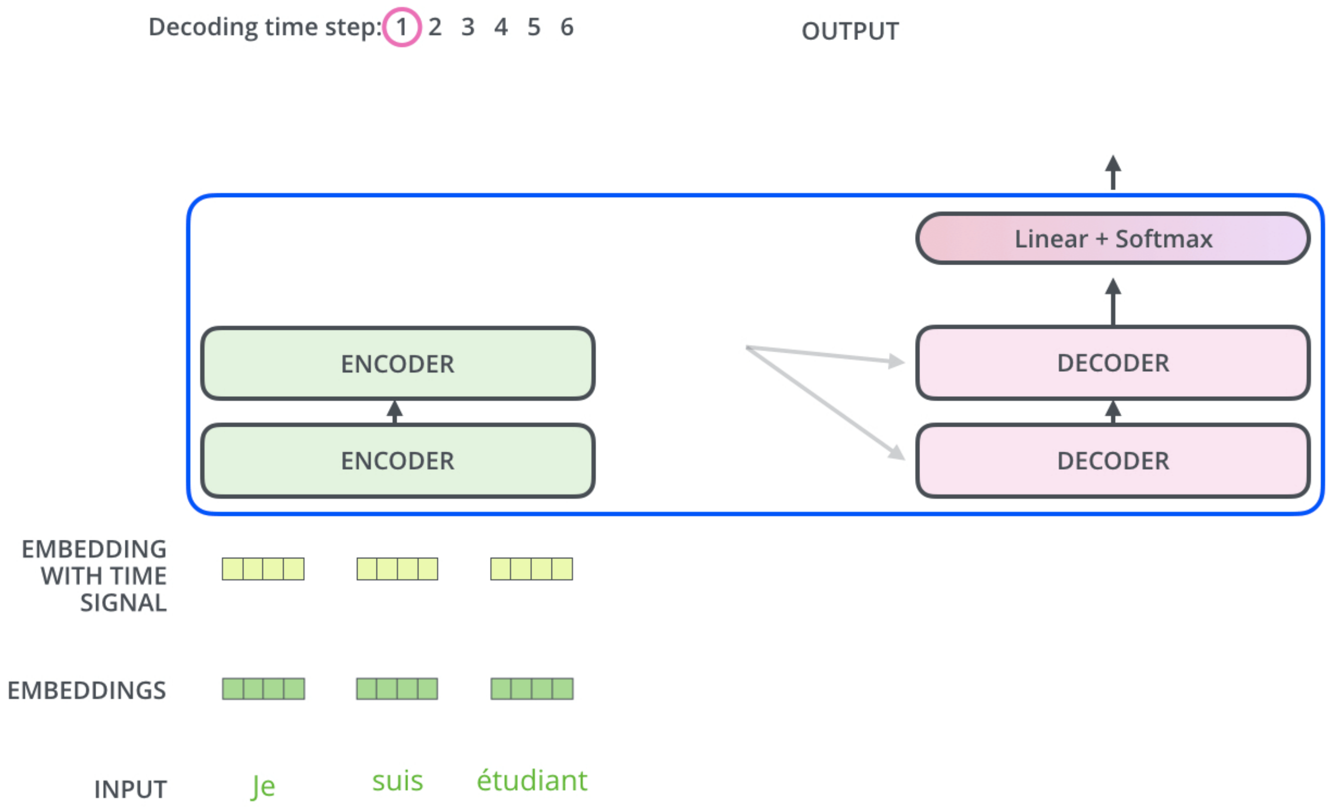




The transformer – model architecture

Attention and Transformer for MT

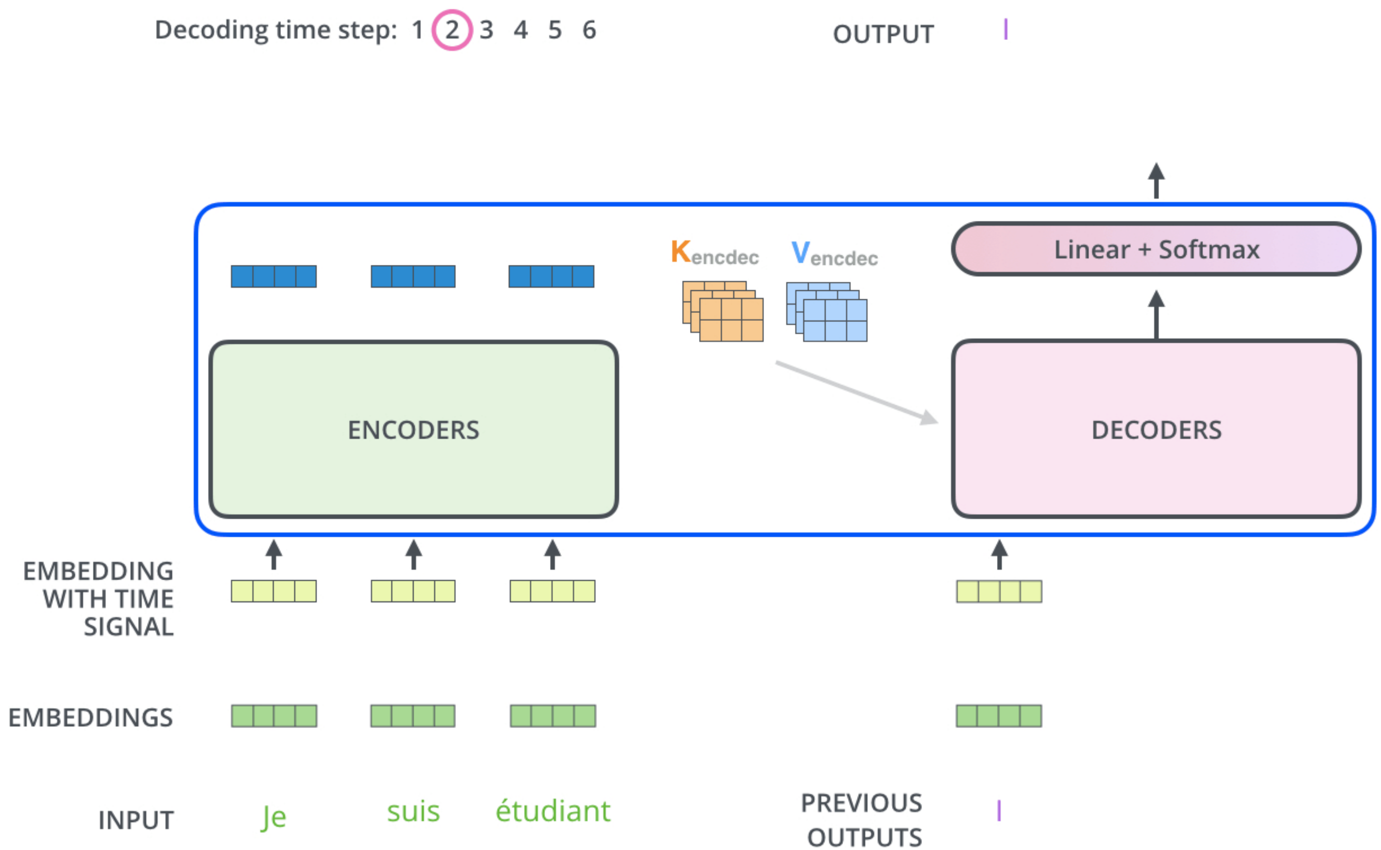
The Transformer  with example – Encoder to Decoder



4

Attention and Transformer for MT

The Transformer  with example – Decoding Phrases



5 The Rise of the Pre-trained Model

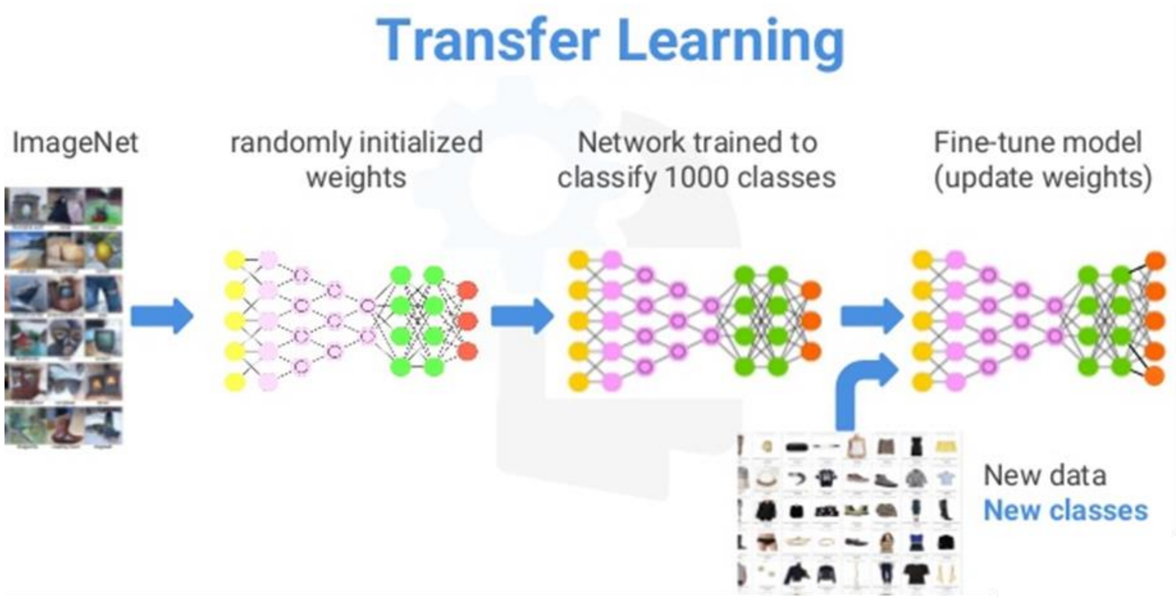
Early 2019 ~

5 The Rise of the Pre-trained Model

Pre-training and Transfer Learning

In computer vision, prove the value of transfer learning

- *pre-training a neural network on a known task (i.e. ImageNet)*
- *performing fine-tuning*
- *using the trained neural network as the basis of a new purpose-specific model.*



5 The Rise of the Pre-trained Model

Pre-training and Transfer Learning in NLP

Popular Pre-trained Model in NLP

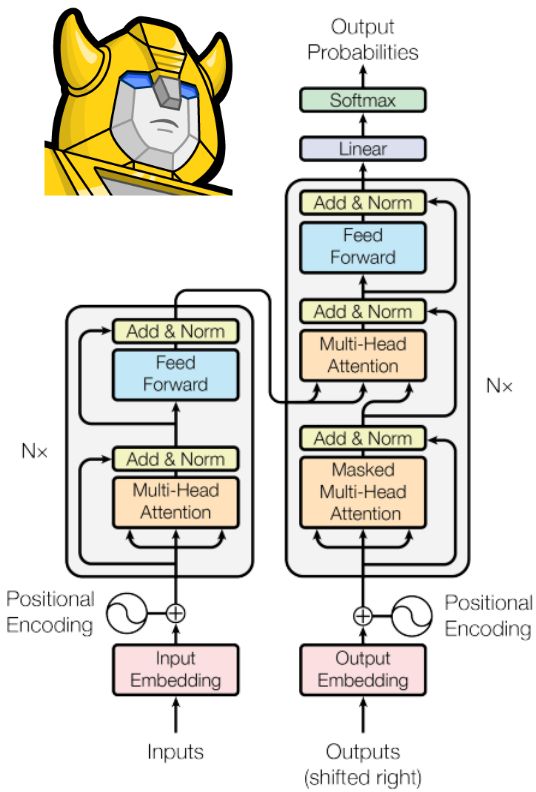
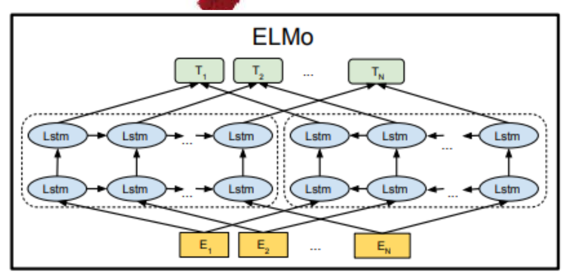
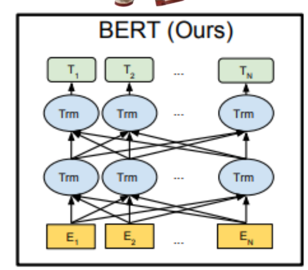


Figure 1: The Transformer - model architecture.



(Peters et al, 2018)



(Devlin et al, 2018)

Using Contextual word representations

5 The Rise of the Pre-trained Model

Pre-training and Transfer Learning in NLP

Popular Pre-trained Model: Contextual Representations

Word embeddings (i.e. word2vec, fastText, GloVe) are applied in a context free manner

*Step up to the **bat*** — ***bat** [0.7, 0.2, -0.5, 1.1, …]*

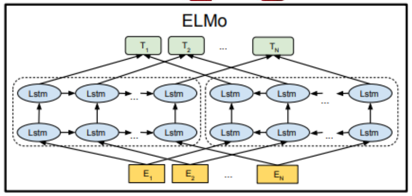
*A vampire **bat*** — ***bat** [0.7, 0.2, -0.5, 1.1, …]*

*Need to train **contextual representation** on text corpus*

*Step up to the **bat*** — ***bat** [1.1, -0.7, 0.8, 2.1, …]*

*A vampire **bat*** — ***bat** [0.3, 0.5, -0.9, 1.3, …]*

5 The Rise of the Pre-trained Model



Pre-training and Transfer Learning in NLP

ELMo: Deep Contextual Word Embeddings (2017)

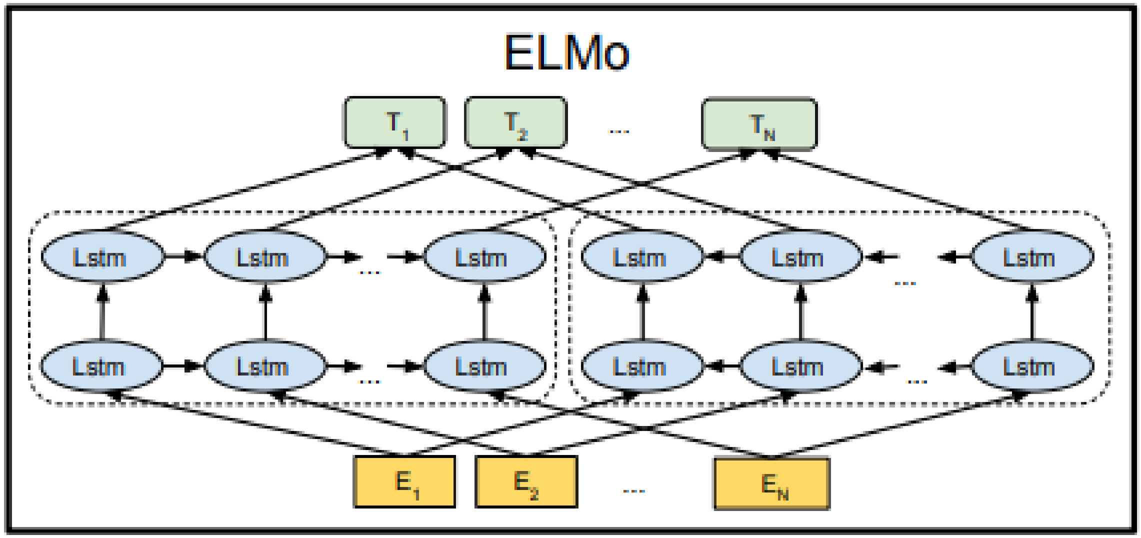
*ELMo provided a **significant step towards pre-training in the context of NLP**. Let's dig in what the ELMo's big secret is!*

The Rise of the Pre-trained Model

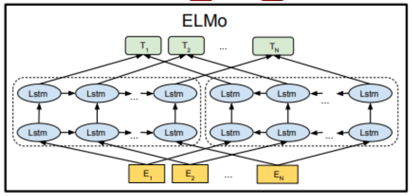
Pre-training and Transfer Learning in NLP

ELMo: Deep Contextual Word Embeddings (2017)

ELMo gained its language understanding from being trained to predict the next word in a sequence of words, Language Modeling Tasks. This is convenient because we have vast amounts of text data that such a model can learn from without needing labels.



5 The Rise of the Pre-trained Model

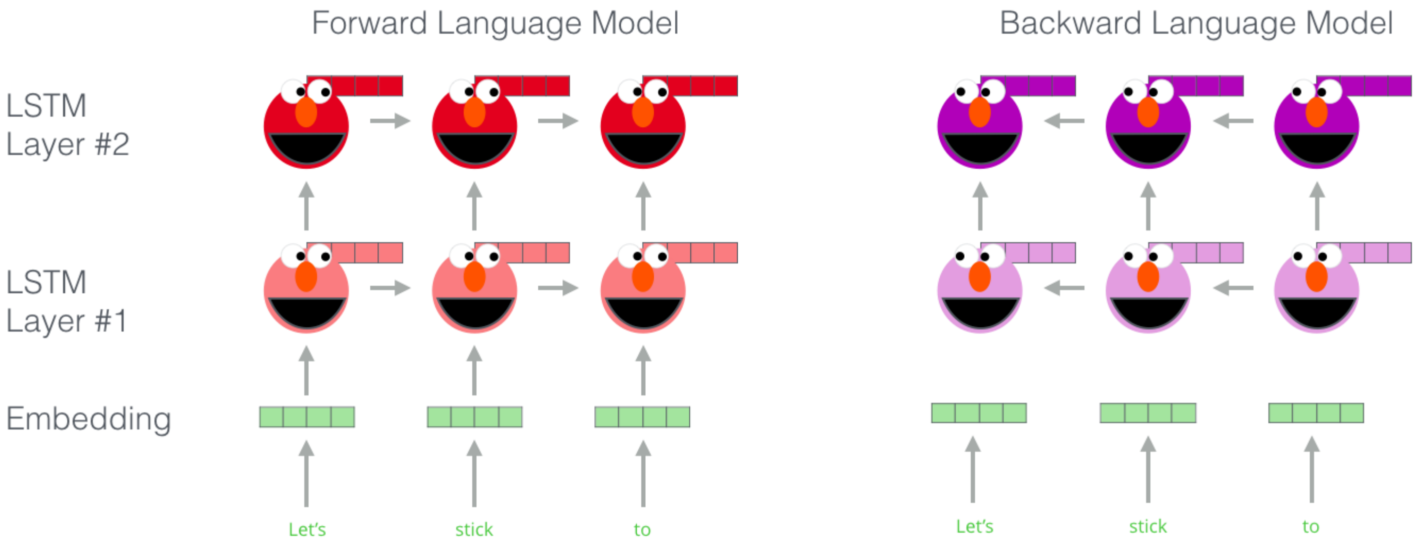


Pre-training and Transfer Learning in NLP

ELMo: Deep Contextual Word Embeddings (2017)

We can see the hidden state of each unrolled-LSTM step peaking out from behind ELMo’s head. Those come in handy in the embedding process after this pre-training is done.

ELMo goes a step further and trains a bi-directional LSTM – so that its language model doesn’t only have a sense of the next word, but also the previous word.

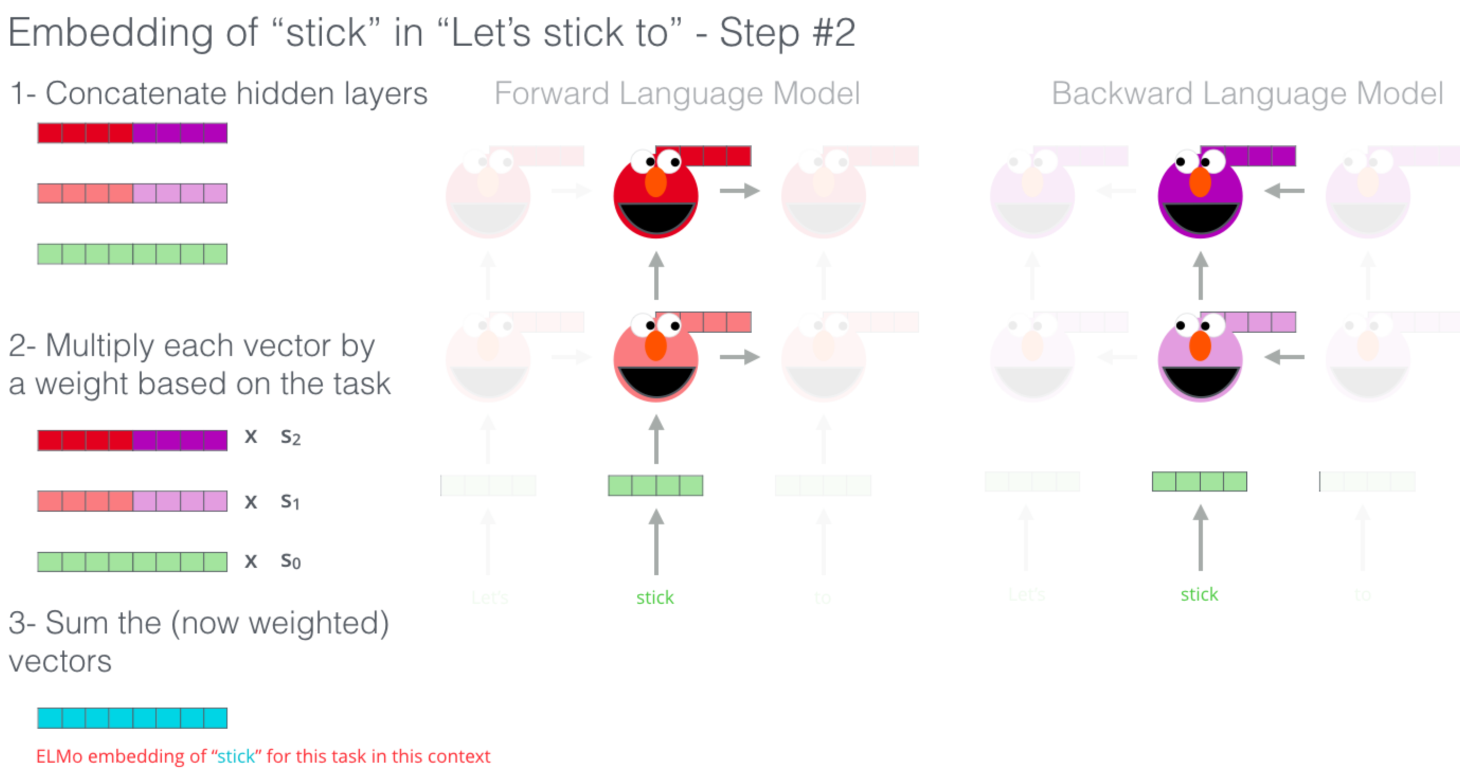


5 The Rise of the Pre-trained Model

Pre-training and Transfer Learning in NLP

ELMo: Deep Contextual Word Embeddings (2017)

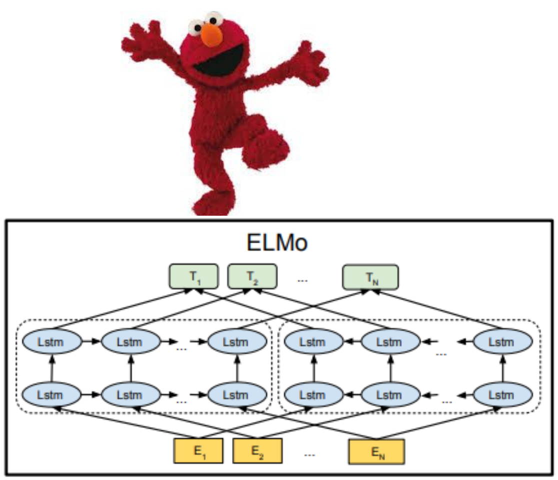
ELMo comes with the contextualized embedding through grouping together the hidden states (and initial embedding) in a certain way (concatenation followed by weighted summation).



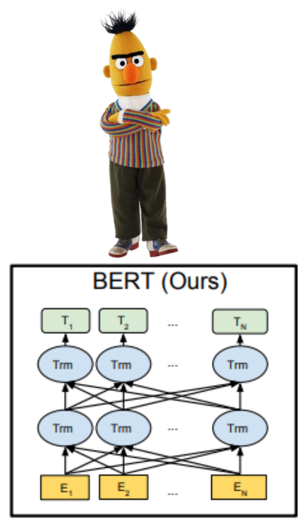
5 The Rise of the Pre-trained Model

Pre-training and Transfer Learning in NLP

ELMo and BERT



(Peters et al, 2018)

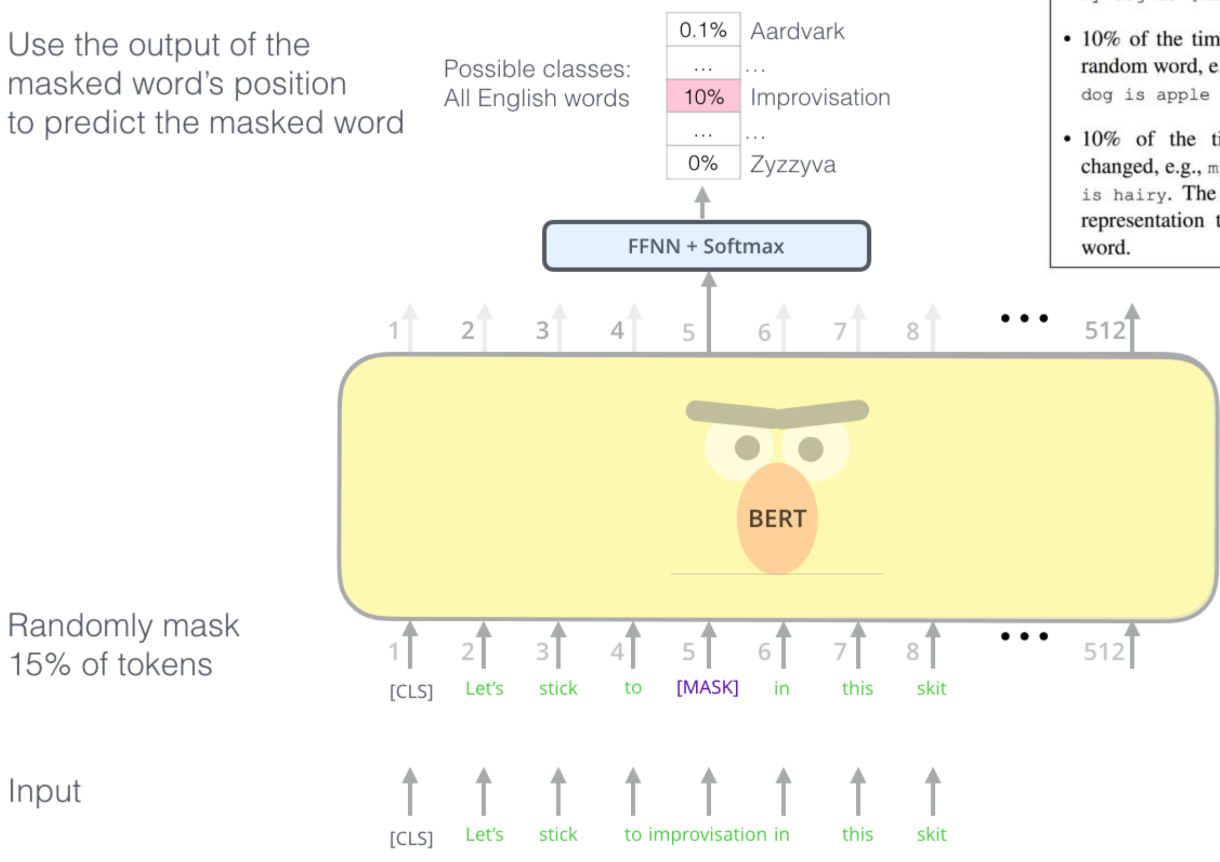


(Devlin et al, 2018)

5 The Rise of the Pre-trained Model

Pre-training and Transfer Learning in NLP

BERT: Masked language Model



- Rather than *always* replacing the chosen words with [MASK], the data generator will do the following:
- 80% of the time: Replace the word with the [MASK] token, e.g., `my dog is hairy` → `my dog is [MASK]`
- 10% of the time: Replace the word with a random word, e.g., `my dog is hairy` → `my dog is apple`
- 10% of the time: Keep the word unchanged, e.g., `my dog is hairy` → `my dog is hairy`. The purpose of this is to bias the representation towards the actual observed word.

Pre-training and Transfer Learning in NLP

BERT: Input Representation

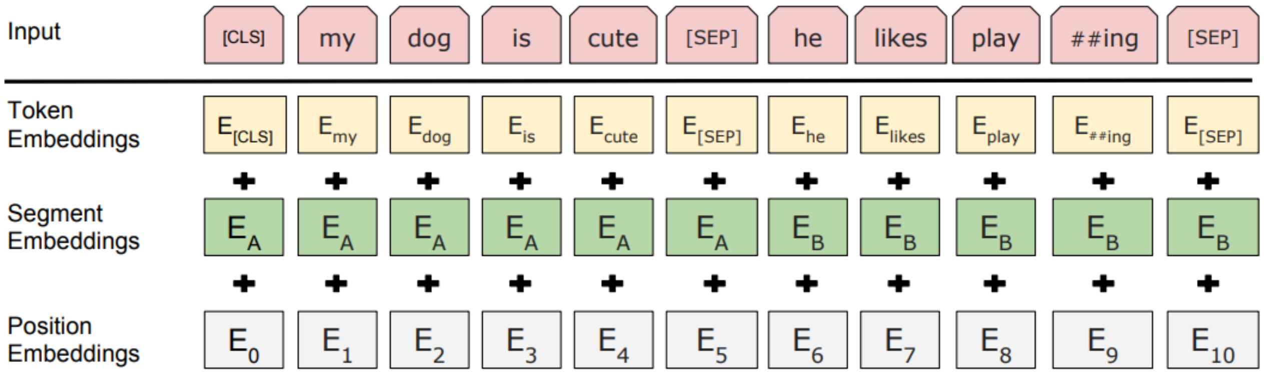
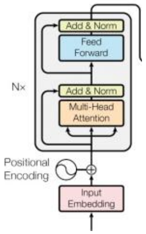


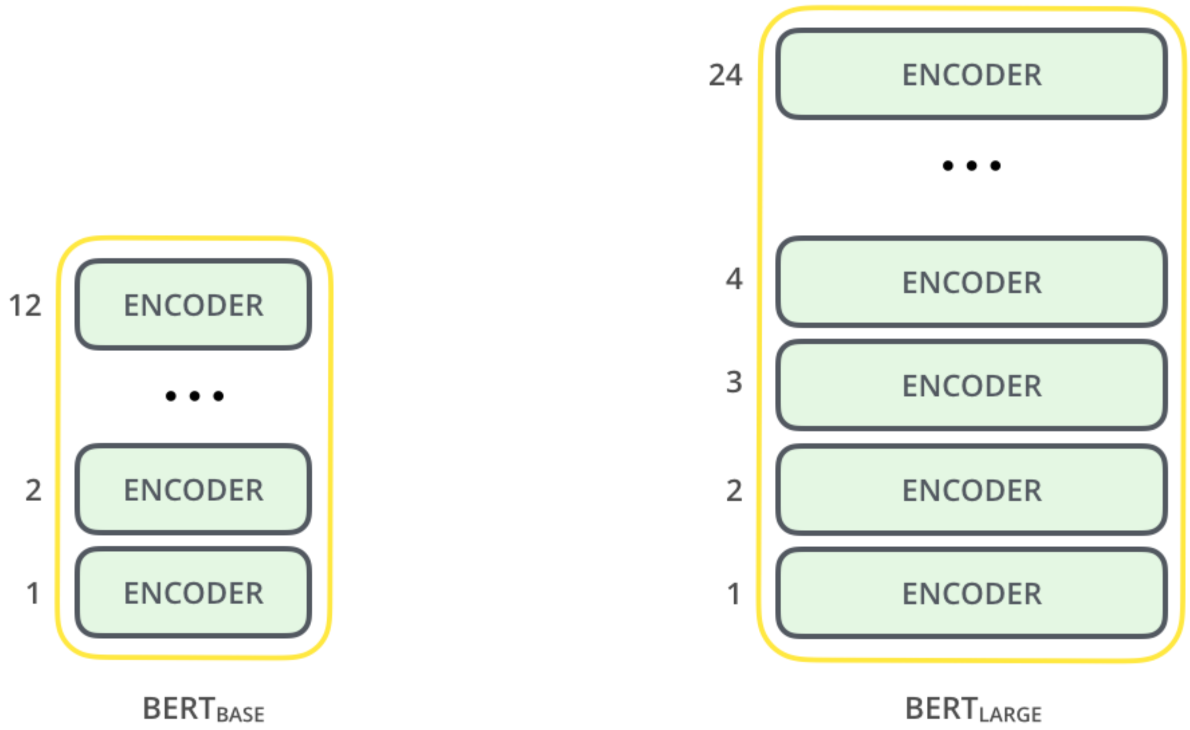
Figure 2: BERT input representation. The input embeddings is the sum of the token embeddings, the segmentation embeddings and the position embeddings.

5 The Rise of the Pre-trained Model



Pre-training and Transfer Learning in NLP

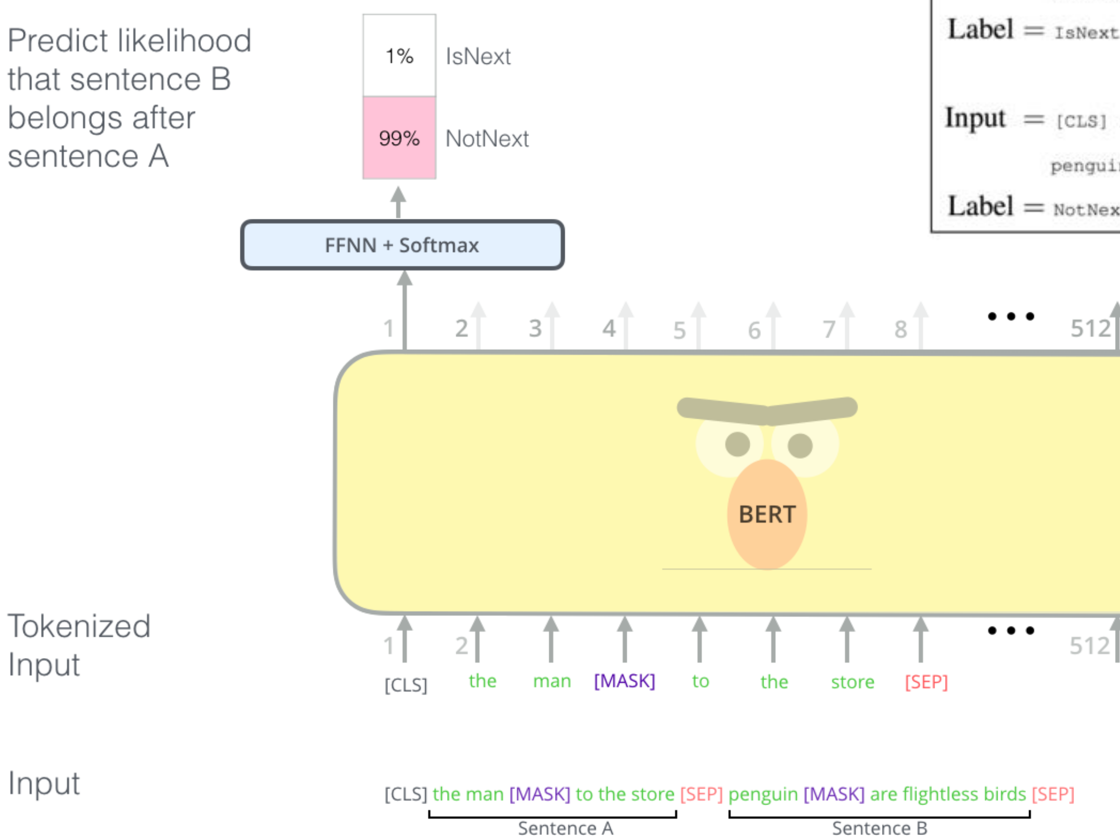
BERT: *Pre-training of Deep Bidirectional Transformers for Language Understanding*



5 The Rise of the Pre-trained Model

Pre-training and Transfer Learning in NLP

BERT: Next Sentence Prediction



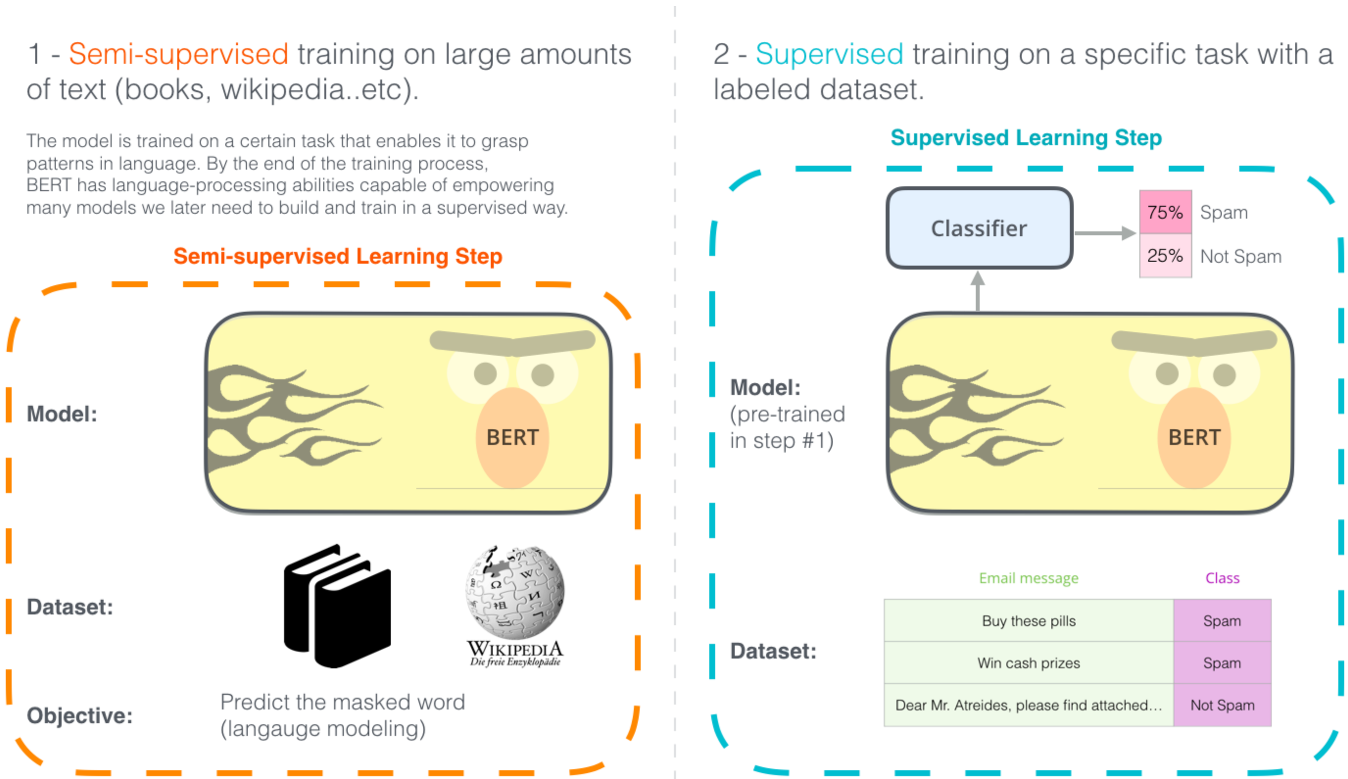
Input = [CLS] the man went to [MASK] store [SEP]
he bought a gallon [MASK] milk [SEP]
Label = IsNext

Input = [CLS] the man [MASK] to the store [SEP]
penguin [MASK] are flight ##less birds [SEP]
Label = NotNext

5 The Rise of the Pre-trained Model

Pre-training and Transfer Learning in NLP

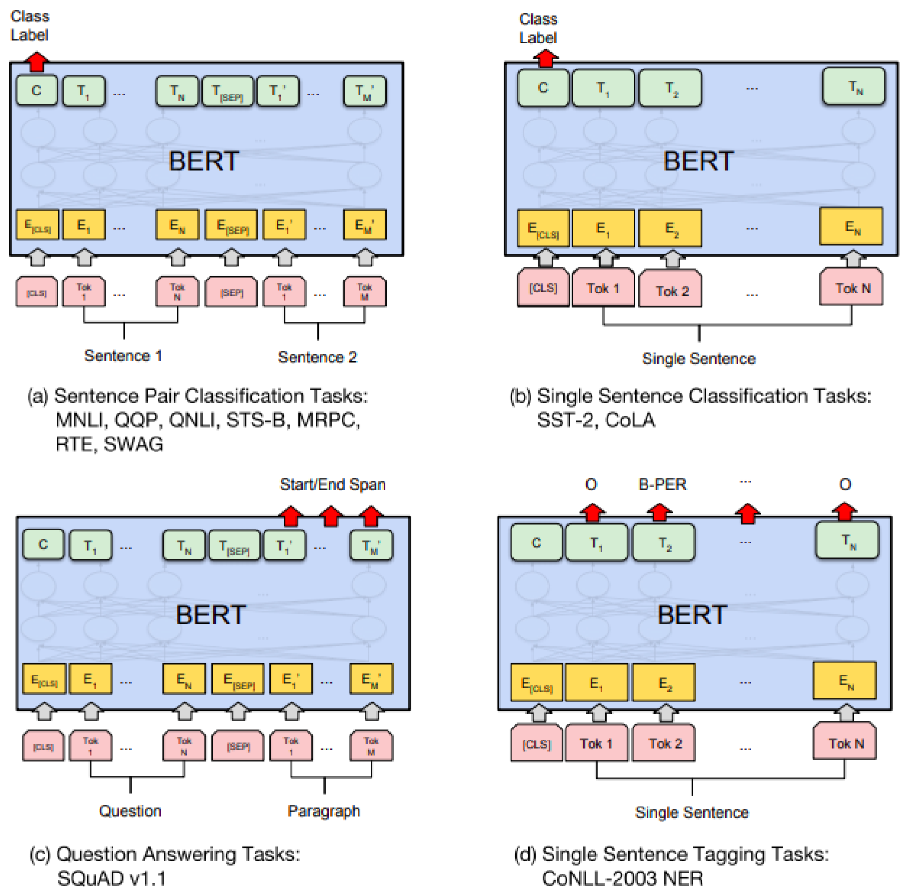
The two steps of how BERT is developed. Download the model pre-trained in step 1 (trained on un-annotated data), and only worry about fine-tuning it for step 2



5 # The Rise of the Pre-trained Model

Pre-training and Transfer Learning in NLP

The following shows a number of ways to use BERT for different tasks.



(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

(b) Single Sentence Classification Tasks: SST-2, CoLA

(c) Question Answering Tasks: SQuAD v1.1

(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

5 The Rise of the Pre-trained Model

Accuracy… Performance

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.9	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	88.1	91.3	45.4	80.0	82.3	56.0	75.2
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.1	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**91.1**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**81.9**

Table 1: GLUE Test results, scored by the GLUE evaluation server. The number below each task denotes the number of training examples. The "Average" column is slightly different than the official GLUE score, since we exclude the problematic WNLI set. OpenAI GPT = (L=12, H=768, A=12); $\text{BERT}_{\text{BASE}}$ = (L=12, H=768, A=12); $\text{BERT}_{\text{LARGE}}$ = (L=24, H=1024, A=16). BERT and OpenAI GPT are single-model, single task. All results obtained from https://gluebenchmark.com/leaderboard and https://blog.openai.com/language-unsupervised/.

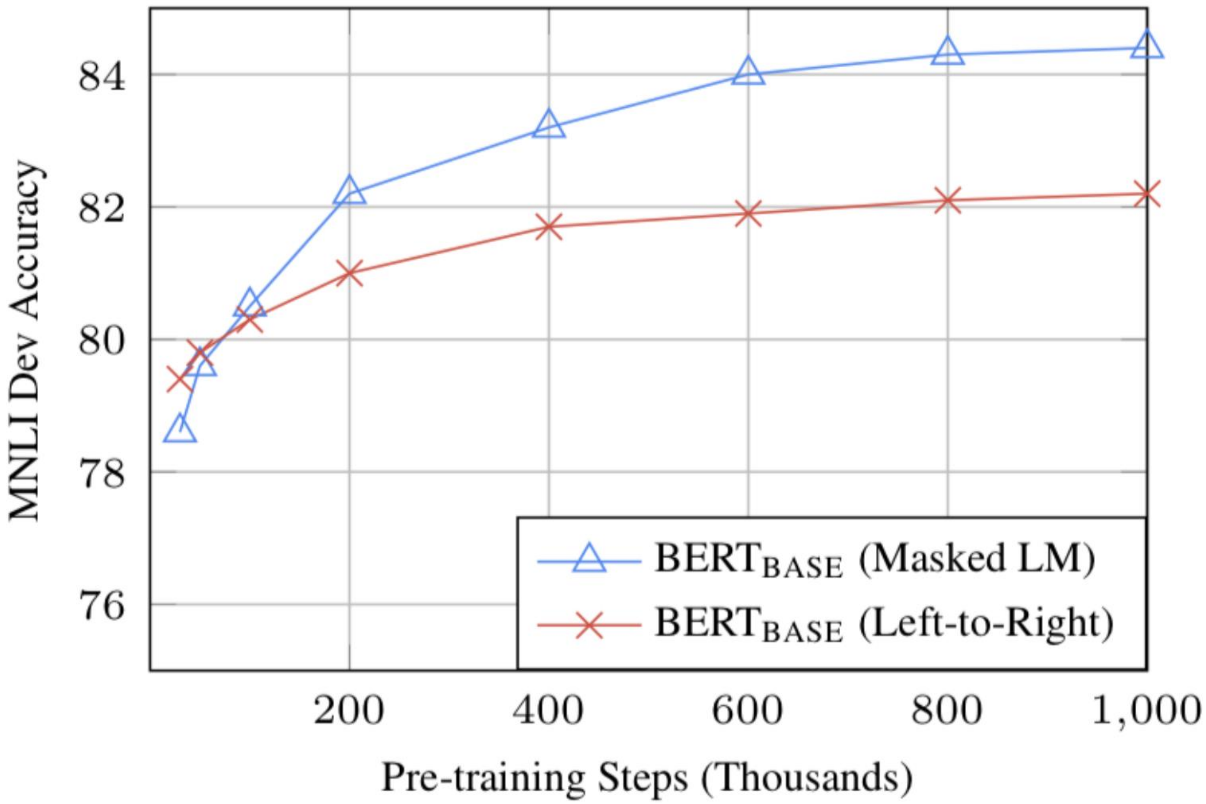
https://rajpurkar.github.io/SQuAD-explorer/

5 The Rise of the Pre-trained Model

Resource! Resource!

TPUS… and Resources..

BERT-Base*: 4 Cloud TPUs (16 TPU chips total) in 4 days*
BERT-Large*: 16 Cloud TPUs (64 TPU chips total)*



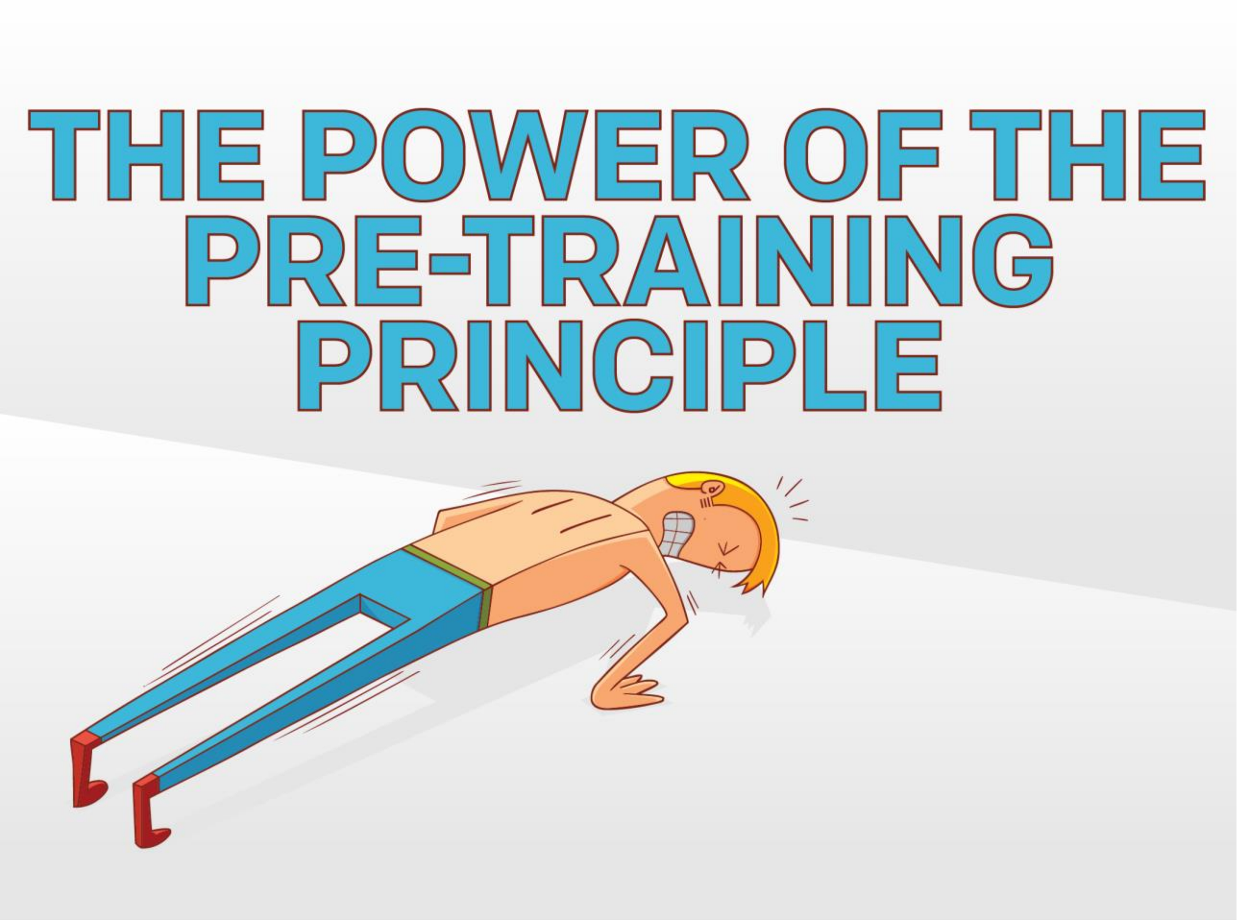
The Rise of the Pre-trained Model

More advanced pre-trained model

	BERT	RoBERTa	DistilBERT	XLNet
Size (millions)	**Base**: 110 **Large**: 340	**Base**: 110 **Large**: 340	**Base:** 66	**Base**: ~110 **Large**: ~340
Training Time	**Base**: 8 x V100 x 12 days* **Large:** 64 TPU Chips x 4 days (or 280 x V100 x 1 days*)	**Large:** 1024 x V100 x 1 day; 4-5 times more than BERT.	**Base**: 8 x V100 x 3.5 days; 4 times less than BERT.	**Large:** 512 TPU Chips x 2.5 days; 5 times more than BERT.
Performance	Outperforms state-of-the-art in Oct 2018	2-20% improvement over BERT	3% degradation from BERT	2-15% improvement over BERT
Data	16 GB BERT data (Books Corpus + Wikipedia). 3.3 Billion words.	160 GB (16 GB BERT data + 144 GB additional)	16 GB BERT data. 3.3 Billion words.	**Base**: 16 GB BERT data **Large**: 113 GB (16 GB BERT data + 97 GB additional). 33 Billion words.
Method	BERT (Bidirectional Transformer with MLM and NSP)	BERT without NSP**	BERT Distillation	Bidirectional Transformer with Permutation based modeling

The Rise of the Pre-trained Model

The future of NLP…



What we learned in this course!

- Week 1: Introduction to Natural Language Processing (NLP)
- Week 2: Word Embeddings (Word Vector for Meaning)
- Week 3: Word Classification with Machine Learning I
- Week 4: Word Classification with Machine Learning II

NLP and Machine Learning

- Week 5: Language Fundamental
- Week 6: Part of Speech Tagging
- Week 7: Dependency Parsing
- Week 8: Language Model

NLP Techniques

- Week 9: Information Extraction: Named Entity Recognition
- Week 10: Advanced NLP: Attention and Reading Comprehension
- Week 11: Advanced NLP: Transformer and Machine Translation
- Week 12: Advanced NLP: Graph Neural Network with NLP

Advanced Topic

- Week 13: Future of NLP and Exam Review

/ Reference

Reference for this lecture

- Deng, L., & Liu, Y. (Eds.). (2018). Deep Learning in Natural Language Processing. Springer.
- Rao, D., & McMahan, B. (2019). Natural Language Processing with PyTorch: Build Intelligent Language Applications Using Deep Learning. " O'Reilly Media, Inc.".
- Manning, C. D., Manning, C. D., & Schütze, H. (1999). Foundations of statistical natural language processing. MIT press.
- Manning, C 2018, Natural Language Processing with Deep Learning, lecture notes, Stanford University

- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
- Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. arXiv preprint arXiv:1802.05365.
- Miller, A., Fisch, A., Dodge, J., Karimi, A. H., Bordes, A., & Weston, J. (2016). Key-value memory networks for directly reading documents. arXiv preprint arXiv:1606.03126.

- Drawings
- http://jalammar.github.io/illustrated-bert/
- http://jalammar.github.io/illustrated-transformer/