Deep Learning for NLP, part II

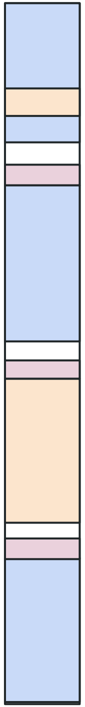
Stanford ICME Summer workshop 2021

Instructor: **Afshine Amidi**

18-20 August 2021

Schedule for today

8:00



Transformers (cont'd)

BREAK

Questions

9:00

BERT deep dive

BREAK

Questions

10:00

Hands on

BREAK

Questions

Conclusion

11:00

All times are in Pacific Time (UTC-7)

Schedule for today

8:00

Transformers (cont'd)

BREAK
Questions

9:00

BERT deep dive

BREAK
Questions

10:00

Hands on

BREAK
Questions

Conclusion

11:00

All times are in Pacific Time (UTC-7)

Schedule for today

8:00

Transformers (cont'd)

BREAK

Questions

9:00

BERT deep dive

BREAK

Questions

10:00

Hands on

BREAK

Questions

Conclusion

11:00

All times are in Pacific Time (UTC-7)

Schedule for today

8:00

Transformers (cont'd)

BREAK

Questions

9:00

BERT deep dive

BREAK

Questions

10:00

Hands on

BREAK

Questions

Conclusion

11:00

All times are in Pacific Time (UTC-7)

Schedule for today

8:00

Transformers (cont'd)

BREAK

Questions

9:00

BERT deep dive

BREAK

Questions

10:00

Hands on

BREAK

Questions

Conclusion

11:00

All times are in Pacific Time (UTC-7)

Schedule for today

8:00

Transformers (cont'd)

BREAK
Questions

9:00

BERT deep dive

BREAK
Questions

10:00

Hands on

BREAK
Questions

Conclusion

11:00

All times are in Pacific Time (UTC-7)

Deep Learning for NLP, part II

Stanford ICME Summer workshop 2021

Motivation and setup

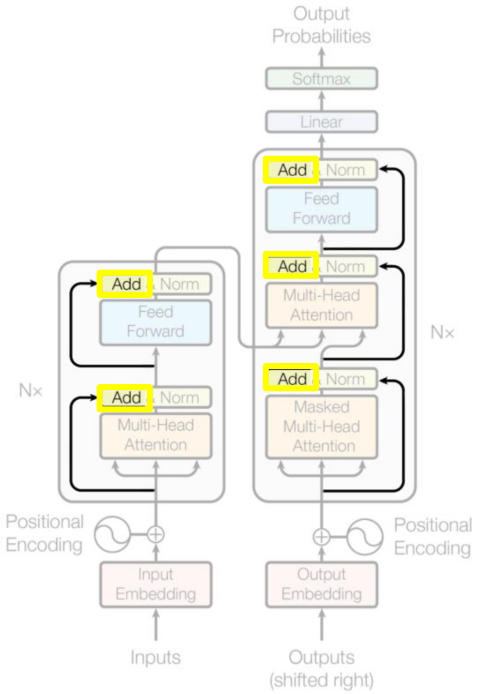
Background

Transformers

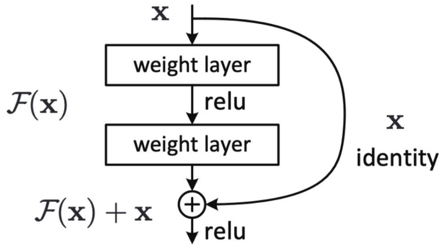
BERT

Conclusion

Computational tricks



Residual connections



Idea:

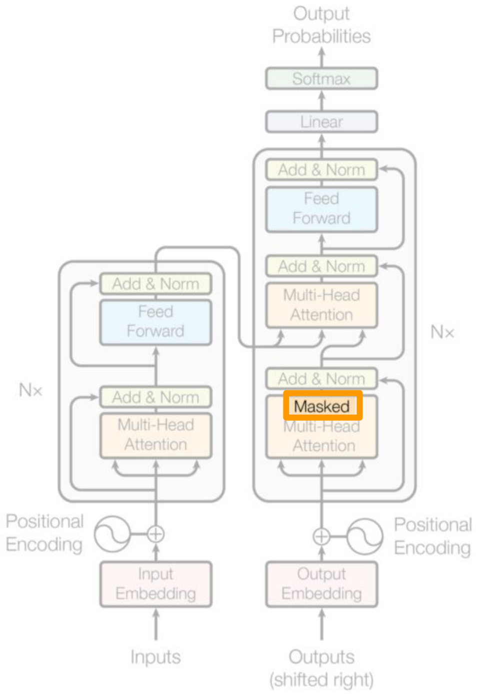
- Taken from the ResNet paper
- Element-wise addition of input and output features at different stages of the network

Benefits:

- Diversifies features at each level of the network
- Helps with gradient propagation

Left figure adapted from "Attention Is All You Need", Vaswani et al., 2017. Right figure from "Deep Residual Learning for Image Recognition", He et al., 2015.

Computational tricks



Masking

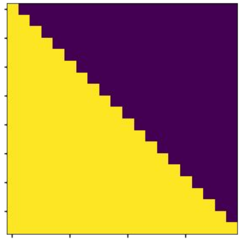
"Masked" also known as "causal" self-attention

Idea:

- Future tokens to be decoded need to be hidden at training time
- Create matrix of masked tokens at each decoding step

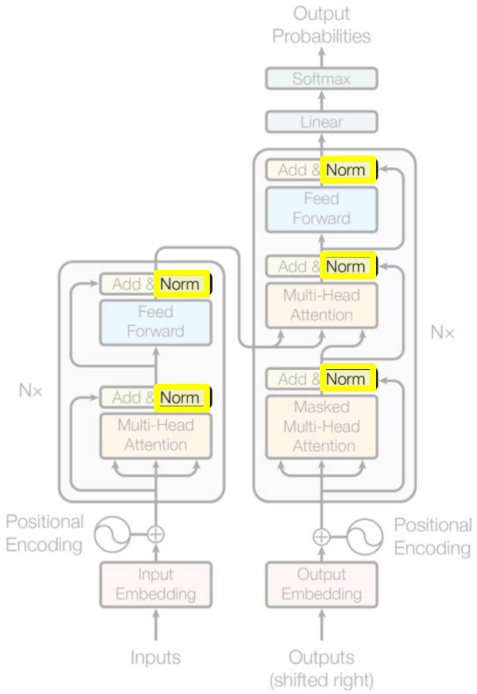
Benefit:

- All operations are vectorized

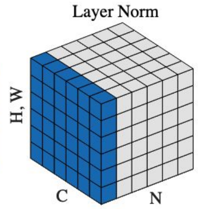


Left figure adapted from "Attention Is All You Need", Vaswani et al., 2017. Right figure from "The Annotated Transformer", Harvard NLP, 2018.

Computational tricks



Layer normalization



Idea:

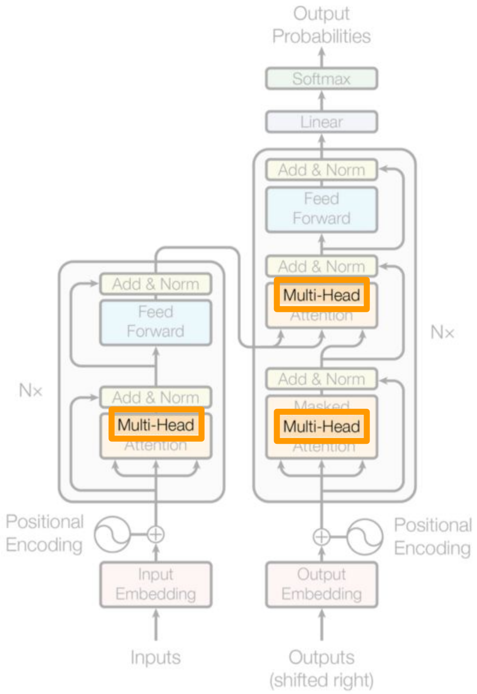
- Introduced in a 2016 UToronto paper, widely used today
- Normalize activation outputs over neurons of a hidden layer to reduce "covariate shift" across layers

Benefits:

- Interesting invariance properties in theory (e.g. independent of the batch)
- Faster, more stable convergence in practice

Left figure adapted from "Attention Is All You Need", Vaswani et al., 2017. Right figure adapted from "Group normalization", Wu et al., 2018.

Computational tricks



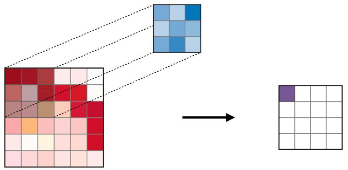
Multi-head attention

Idea:

- run **multiple** self-attention layers in parallel

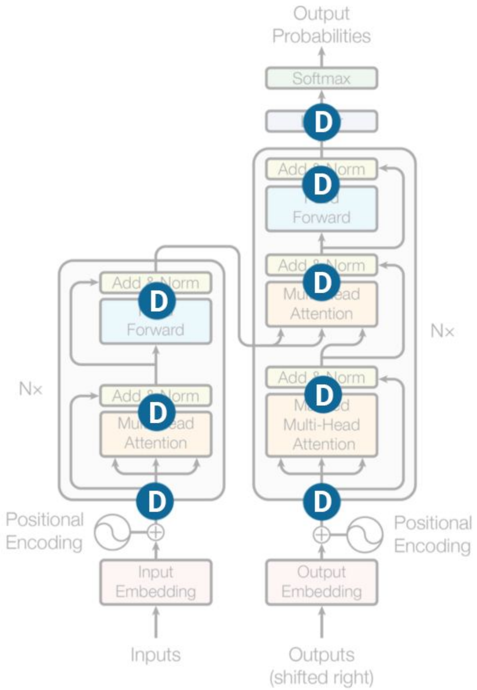
Benefits:

- Enables the model to capture different attention features in parallel
- Comparison: **multiple** filters of a convolutional layer in computer vision



Left figure adapted from "Attention Is All You Need", Vaswani et al., 2017. Right figure from "VIP cheatsheets for Stanford's CS 230", Amidi, 2019.

Computational tricks

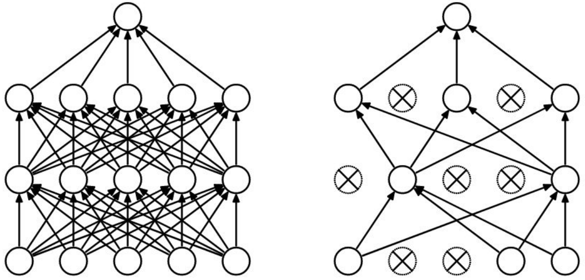


Dropout

Idea:

- From 2014, now ubiquitous in DL architectures
- Randomly drop neural net connections/units with a small probability

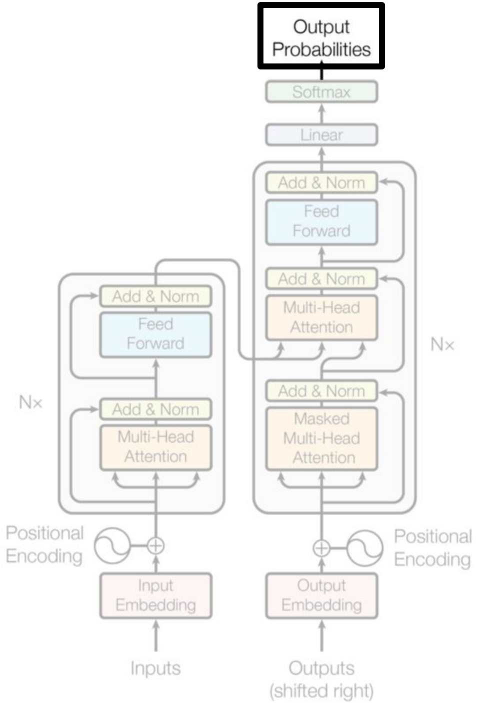
Better **generalization**



(a) Standard Neural Net (b) After applying dropout.

Left figure adapted from "Attention Is All You Need", Vaswani et al., 2017. Right figure from "Dropout: A Simple Way to Prevent Neural Networks from Overfitting", Srivastava et al., 2014.

Computational tricks



Label smoothing

Idea:

- 2015 vision paper: overconfidence is bad
- Introduce **noise** in true labels

$$q(k|x) = \delta_{k,y} \longrightarrow q'(k|x) = (1-\epsilon)\delta_{k,y} + \epsilon u(k)$$

Benefits:

- General technique that prevents overfitting
- Improves accuracy and BLEU score

Figure adapted from "Attention Is All You Need", Vaswani et al., 2017.

Let's compute the number of parameters!

Notebook 1: **Transformer**

Recommended post-workshop reading

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

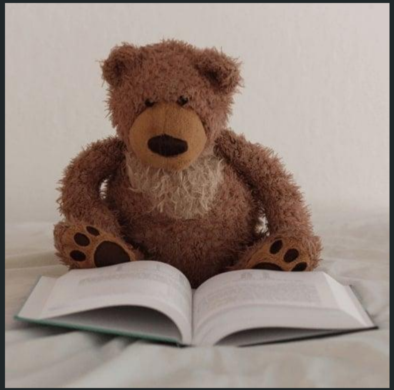
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best

“Attention Is All You Need”, Vaswani et al., 2017.

Break + questions



Deep Learning for NLP, part II

Stanford ICME Summer workshop 2021

Motivation and setup

Background

Transformers

BERT

Conclusion

Tokenization

```
A cute teddy bear is reading.
```

Tokenization

A cute teddy bear is reading.

arbitrary	A	cute	teddy bear	is	reading	.

Tokenization

A cute teddy bear is reading.

A	cute	teddy bear	is	reading	.

word	A	cute	teddy	bear	is	reading	.

Tokenization

A cute teddy bear is reading.

A	cute	teddy bear	is	reading	.

A	cute	teddy	bear	is	reading	.

sub-word	A	cute	ted	##dy	bear	is	read	##ing	.

Tokenization

A cute teddy bear is reading.

A	cute	teddy bear	is	reading	.																							
A	cute	teddy	bear	is	reading	.																						
A	cute	ted	##dy	bear	is	read	##ing	.																				
A	_	c	u	t	e	_	t	e	d	d	y	_	b	e	a	r	_	i	s	_	r	e	a	d	i	n	g	.

Tokenization summary

Method	Pros	Cons
Word-level	• Simple • Interpretable	• Risk of OOV • Does not leverage knowledge of root
Subword-level e.g. WordPiece, BPE	• Leverages prefix suffixes • Learned from the data	• Risk of OOV, though less than word-level
Character-level	• Small chance of OOV • RoBUsT tO CASinG anD MIspeliNGs	• Makes computations slower

Deep Learning for NLP, part II

Stanford ICME Summer workshop 2021

Motivation and setup

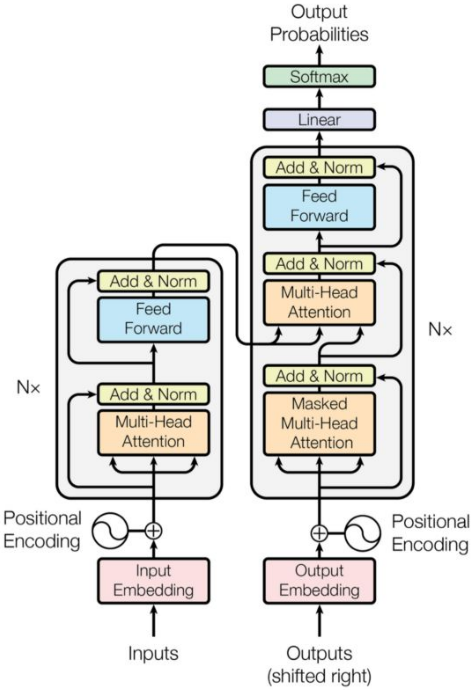
Background

Transformers

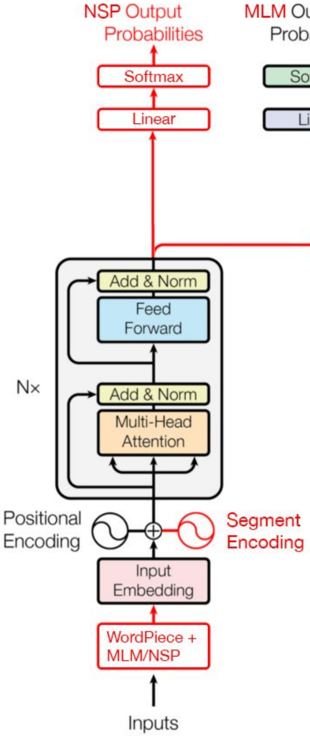
BERT (for real)

Conclusion

BERT: overview of the changes



Original transformer (2017)



BERT (2018)

Overview

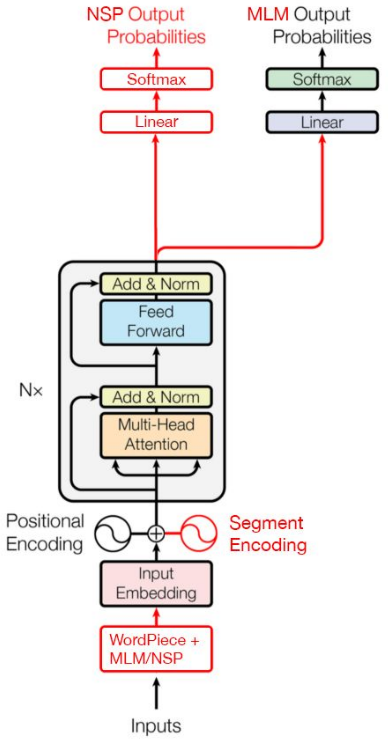


Figure adapted from "Attention Is All You Need", Vaswani et al., 2017.

Goal: leverage general language representation for NLP tasks

Pretraining

- Data: enormous unlabeled corpus of Books (800M words) and Wikipedia (2.5B words)
- MLM task: predict 15% of input tokens
- NSP task: predict whether sentences follow each other or not

Fine-tuning

- Dataset: task-specific
- Objective: tailored to end goal

Input processing

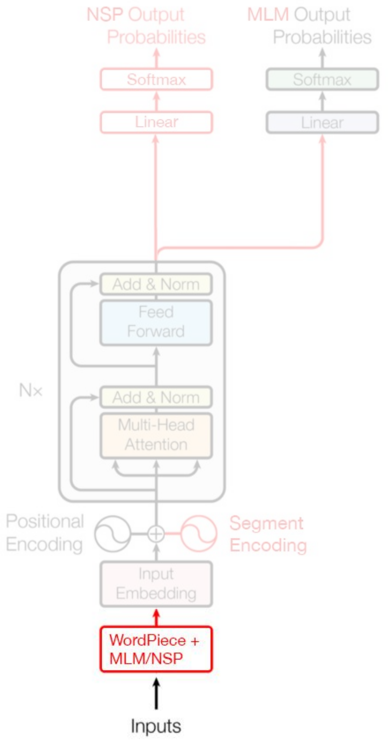


Figure adapted from "Attention Is All You Need", Vaswani et al., 2017.

WordPiece algorithm

- Tokenizer trained on a training set beforehand
- Vocabulary size: ~30,000
- Great at detecting common particles

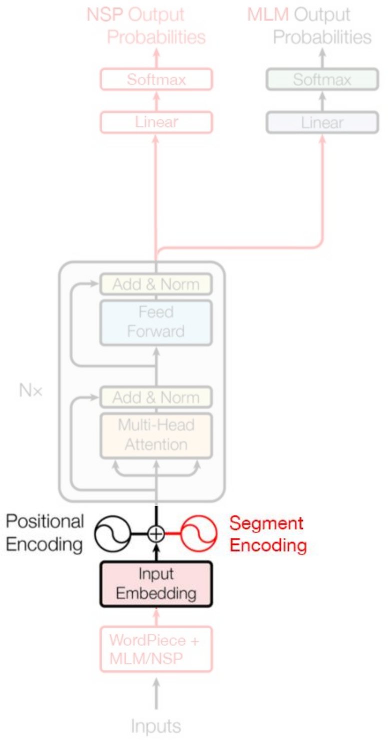
NSP task processing

- Add `[CLS]` token at the beginning of the input

MLM task processing

- Separate consecutive segments with the `[SEP]` token and put another one at the end

Input embedding



Input embeddings

- Gigantic lookup table
- Learns an embedding for each word of the vocabulary

Positional encoding

- Helps the network associate tokens with a position
- Encoding either learnt or fixed with cosines and sines

(new!) Segment encoding

- Shared embedding for a segment

Figure adapted from "Attention Is All You Need", Vaswani et al., 2017.

Encoder-only model

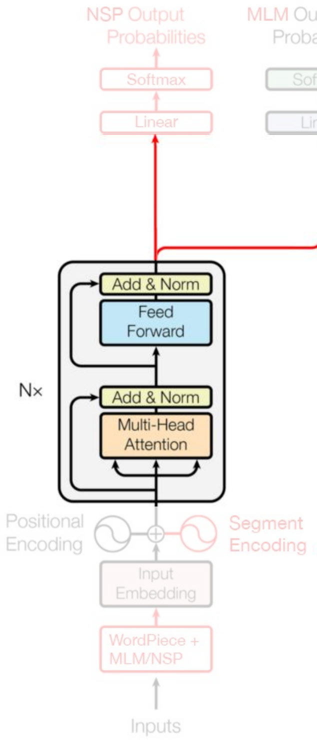


Figure adapted from "Attention Is All You Need", Vaswani et al., 2017.

Model

Encoder part of the original transformers paper

Goal

- Represent input data with features (hopefully) needed for NLP tasks
- Leverage the Transformer's self-attention mechanism
- Use learned embedding towards classification-oriented tasks

Proxy tasks

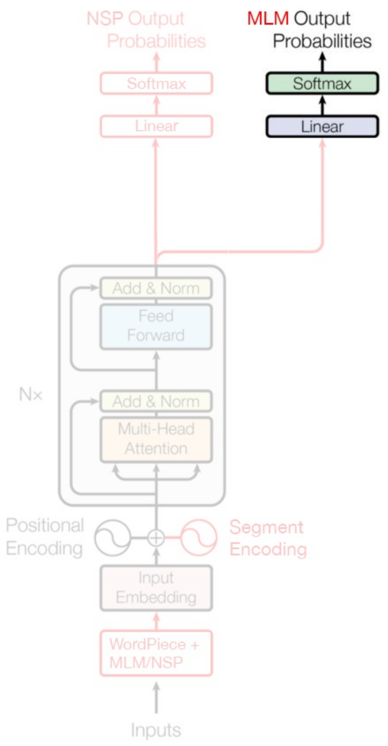


Figure adapted from "Attention Is All You Need", Vaswani et al., 2017.

Masked Language Modeling

Idea:

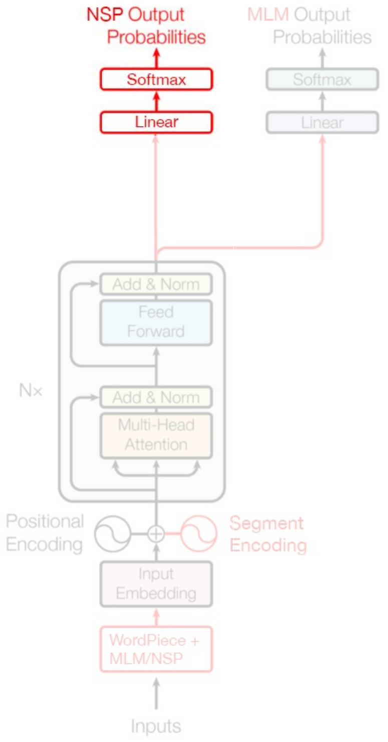
15% of input tokens are set up for prediction where

- 80% are masked
- 10% are changed to a random word
- 10% are unchanged

Benefits:

- Network learns language modeling based on contextual information
- Regularization reflects probabilistic nature of language

Proxy tasks



Next Sentence Prediction

Idea: pick two sentences from the corpus, where

- 50% of the time, they follow each other
- 50% of the time, they **do not** follow each other

Task: predict if they actually follow each other

Benefits:

- Network implicitly learns to detect useful contextual information
- Easy classification task that does not require any labels

Figure adapted from "Attention Is All You Need", Vaswani et al., 2017.

General parameters

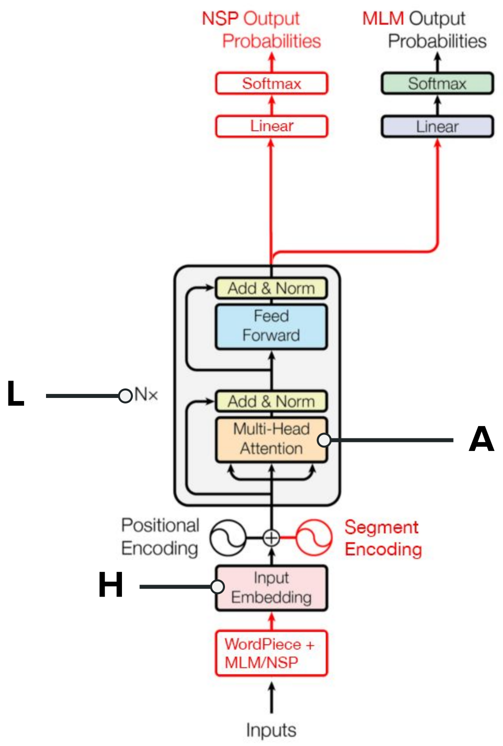


Figure adapted from "Attention Is All You Need", Vaswani et al., 2017.

Layers (L). Number of layers. Corresponds to the "N" parameter in the original Transformers paper.

Hidden (H). Hidden layer size. Dimension of embeddings.

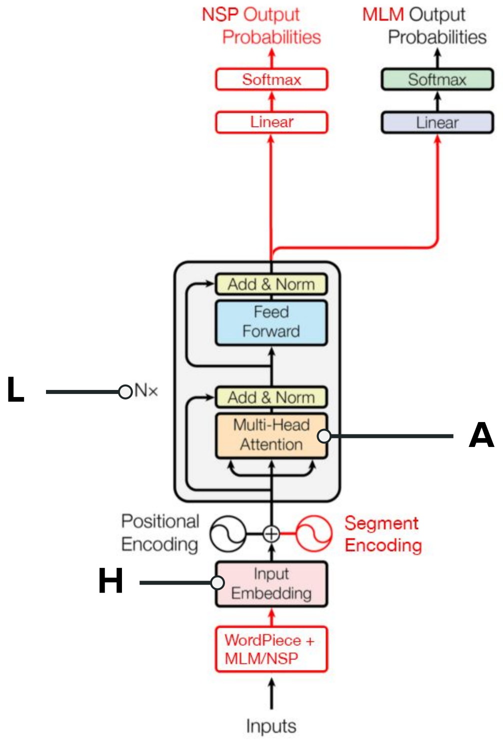
Attention heads (A). Number of attention heads operating in parallel.

Original / whole word masking. Whether separate tokens or entire words were masked during pre-training.

Language-specific / multilingual. Languages on which the model has been trained on.

Cased / uncased. Whether inputs are converted to lowercase or not.

Some numbers



	L	H	A	Parameters
BERT-Tiny	2	128	2	4M
BERT-Mini	4	256	4	11M
BERT-Small	4	512	8	30M
BERT-Medium	8	512	8	42M
BERT-Base	12	768	12	110M
BERT-Large	24	1024	16	330M

Left figure adapted from "Attention Is All You Need", Vaswani et al., 2017. Parameters in the right table were computed in this paper.

Pretraining

Pick a pair of sentences that follow each other 50% of the time, and not the other 50%.

```
A cute teddy bear is reading. The book is about Persian poetry.
```

Pretraining

(optional) Apply casing constraints.

`a cute teddy bear is reading. the book is about persian poetry.`

Pretraining

Apply WordPiece tokenization.

a	cute	teddy	bear	is	rea	##ing	.	the	book	is	about	pers	##ian	poetry	.

Pretraining

Insert `[CLS]` and `[SEP]` tokens at the right position.

[CLS]	a	cute	teddy	bear	is	rea	##ing	[SEP]	the	book	is	about	pers	##ian	poetry	[SEP]

Pretraining

Note: if the number of tokens is smaller than the expected input size, add padding with `[PAD]` tokens to the right.

`[CLS]` `a` `cute` `teddy` `bear` `is` `rea` `##ing` `[SEP]` `the` `book` `is` `about` `pers` `##ian` `poetry` `[SEP]`

Pretraining

Randomly choose 15% of the tokens for the prediction task…

[CLS] a cute **teddy** bear is rea **##ing** [SEP] the **book** is about pers ##ian **poetry** [SEP]

Pretraining

...out of which 80% are replaced with `[MASK]` tokens

`[CLS]` `a` `cute` `[MASK]` `bear` `is` `rea` `##ing` `[SEP]` `the` `[MASK]` `is` `about` `pers` `##ian` `poetry` `[SEP]`

Pretraining

…10% others changed to a random token,

[CLS]	a	cute	[MASK]	bear	is	rea	##ing	[SEP]	the	[MASK]	is	about	pers	##ian	swim	[SEP]

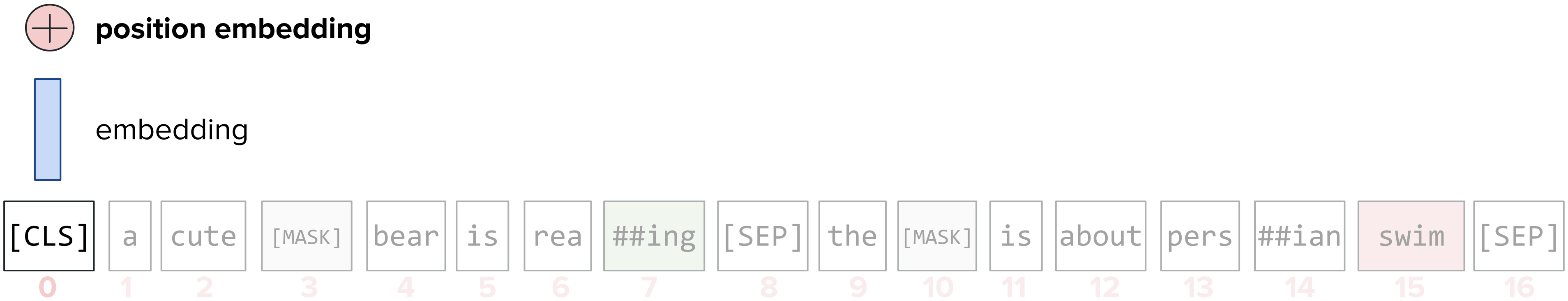
Pretraining

...and the last 10% remain unchanged.

[CLS]	a	cute	[MASK]	bear	is	rea	##ing	[SEP]	the	[MASK]	is	about	pers	##ian	swim	[SEP]

Pretraining

Pretraining



Pretraining

segment embedding

position embedding

embedding

[CLS] a cute [MASK] bear is rea ##ing [SEP] the [MASK] is about pers ##ian swim [SEP]

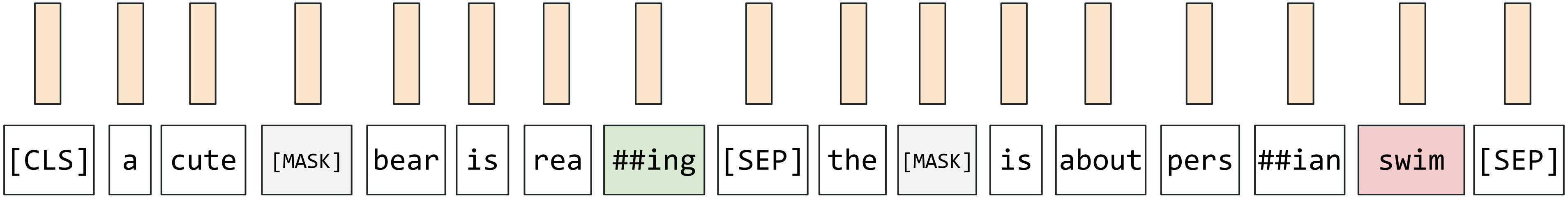
A A A A A A A A A B B B B B B B B

Pretraining

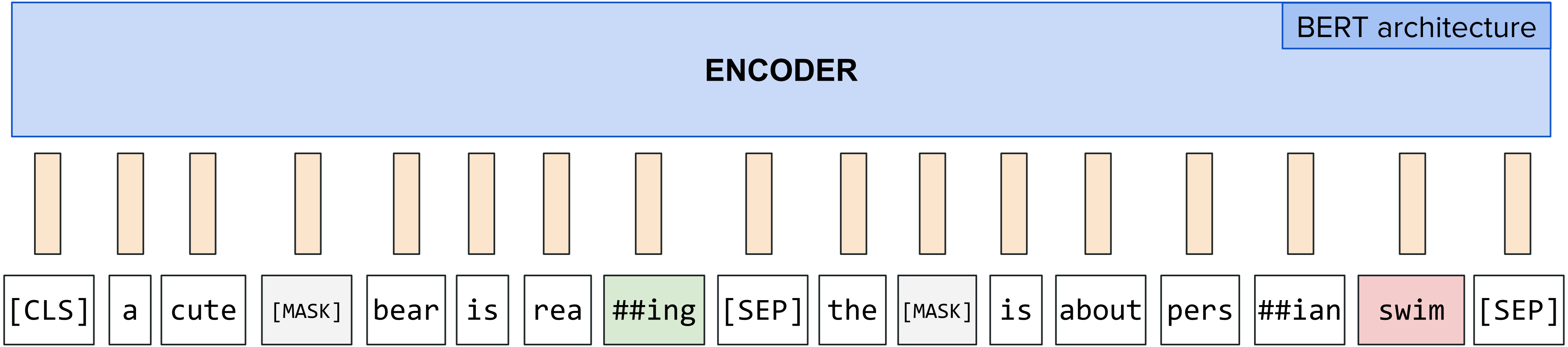
position- and segment-aware embedding

[CLS] a cute [MASK] bear is rea ##ing [SEP] the [MASK] is about pers ##ian swim [SEP]

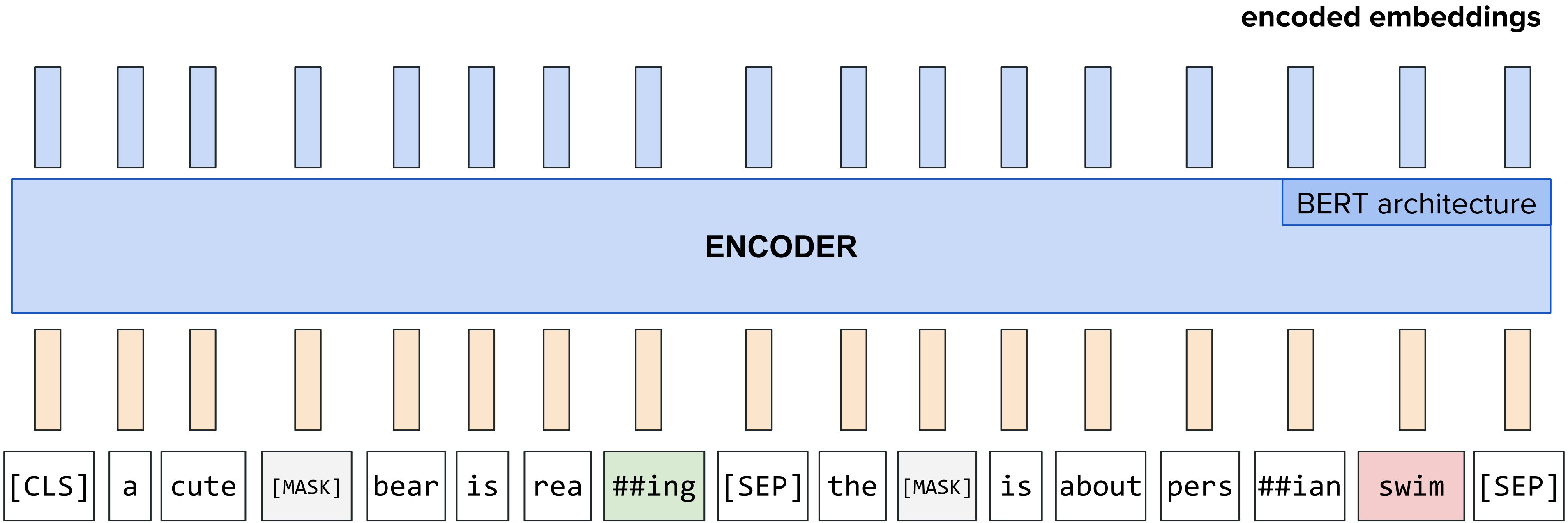
Pretraining



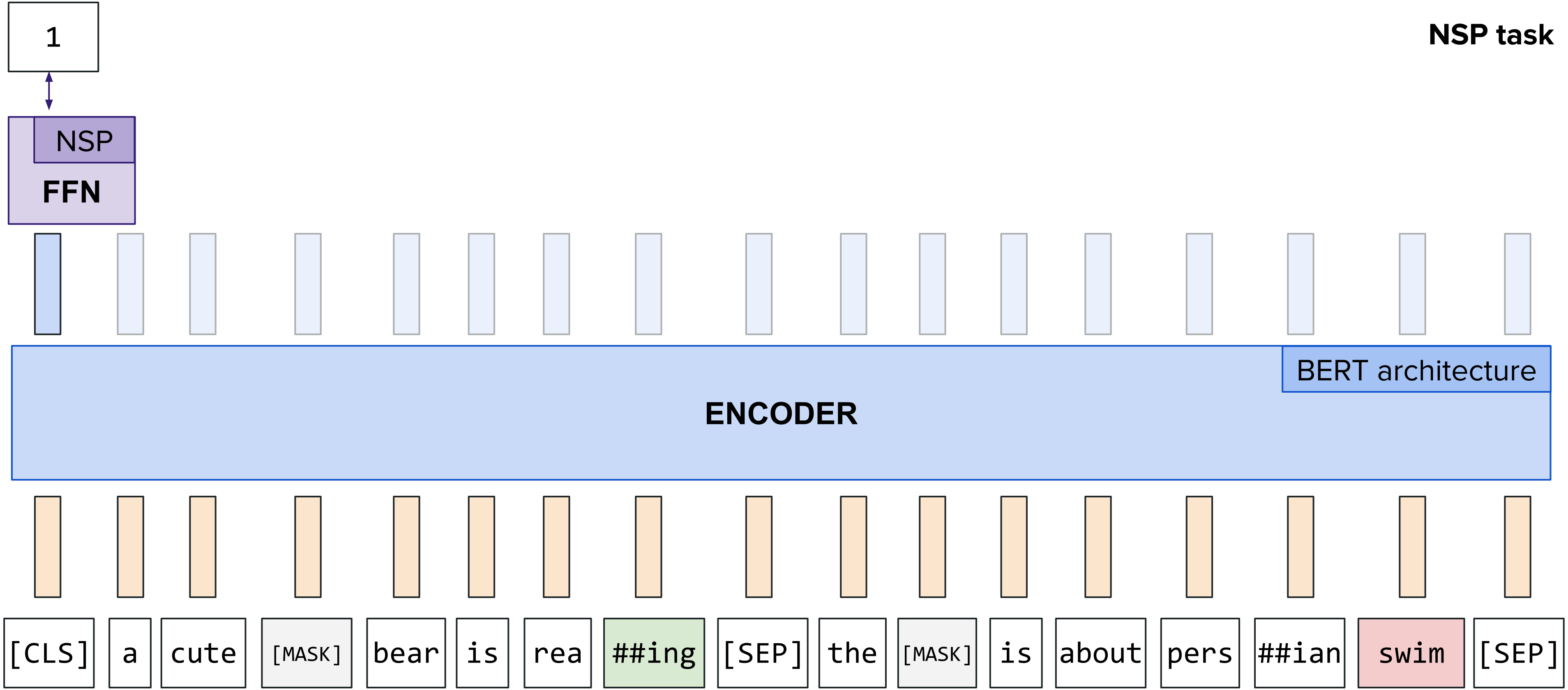
Pretraining



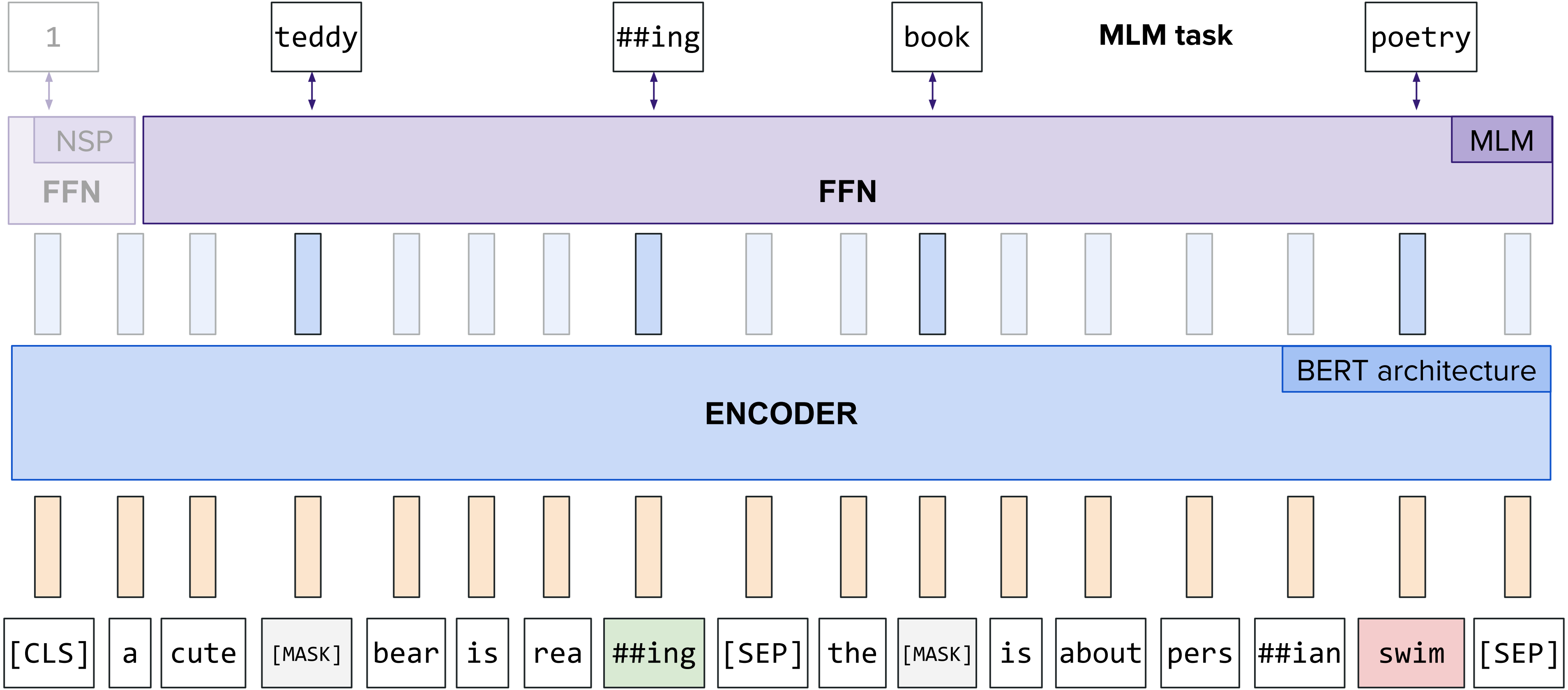
Pretraining



Pretraining



Pretraining



Tips on choosing the right resources

Datasets

- Multi-framework: HuggingFace, paperswithcode
- Tensorflow: TFDS
- PyTorch: Torchtext

Pre-trained models

- Multi-framework: HuggingFace, ModelZoo
- TensorFlow: Hub, Model Garden
- PyTorch: PyTorch Models

Fine-tuning

Goal: Leverage embeddings learned by BERT for a “sister” task

Tricks

- Use weights from **already massively pre-trained** model
- **Freezing** early layers: sometimes better trade-off complexity/performance
- Great results possible with **minimal labeled data** (depending on complexity/proximity to pre-trained data distribution + objectives)

Use cases

- Sequence classification: e.g. sentiment extraction
- Token classification: e.g. question answering

Fine-tuning: sentiment extraction

```
This teddy bear is SO CUTE!
```

Fine-tuning: sentiment extraction

```
this teddy bear is so cute!
```

Fine-tuning: sentiment extraction

this	teddy	bear	is	so	cute	!

Fine-tuning: sentiment extraction

Add `[CLS]` token as a placeholder for sentiment.

`[CLS]` `this` `teddy` `bear` `is` `so` `cute` `!`

Fine-tuning: sentiment extraction

[CLS]	this	teddy	bear	is	so	cute	!	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]	[PAD]

Fine-tuning: sentiment extraction

Fine-tuning: sentiment extraction

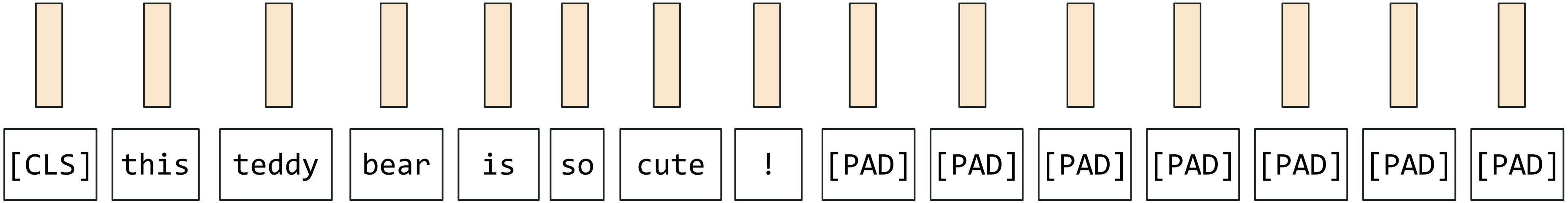
Fine-tuning: sentiment extraction

Finetuning: sentiment extraction

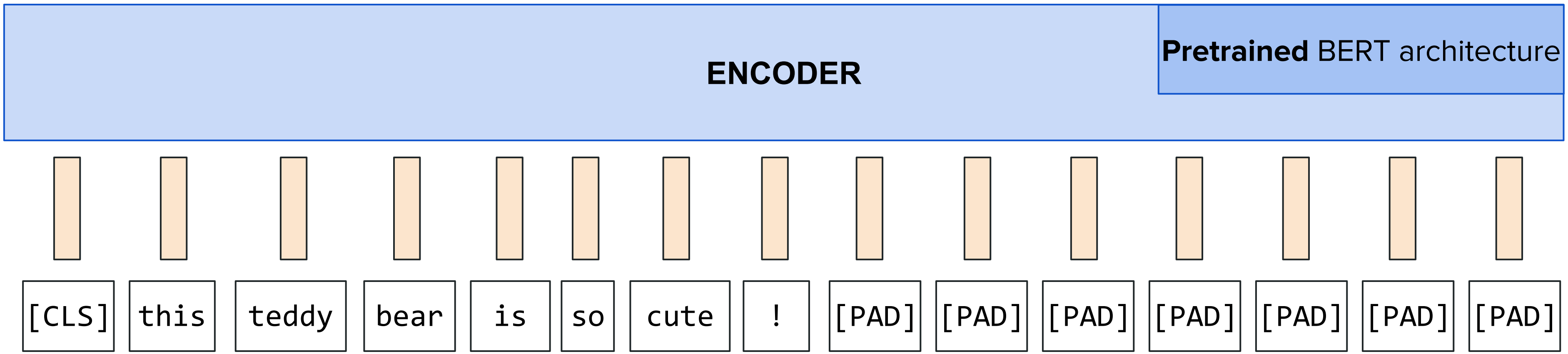
position- and segment-aware embedding

`[CLS]` `this` `teddy` `bear` `is` `so` `cute` `!` `[PAD]` `[PAD]` `[PAD]` `[PAD]` `[PAD]` `[PAD]` `[PAD]`

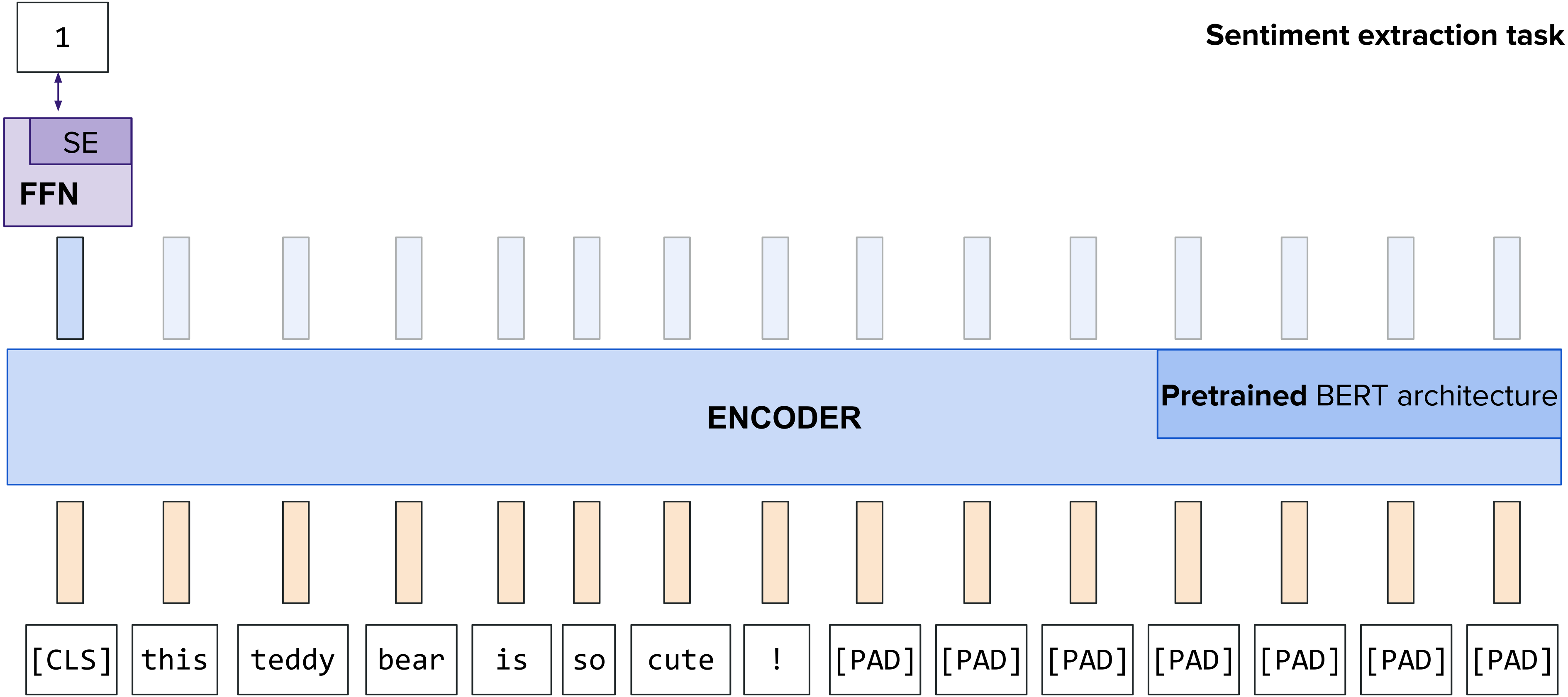
Fine-tuning: sentiment extraction



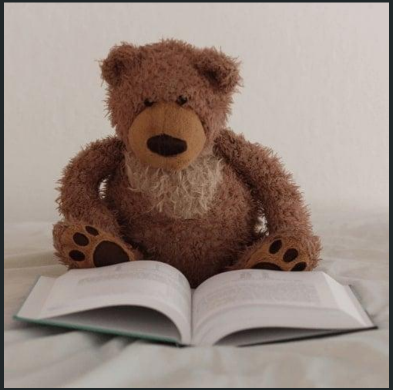
Fine-tuning: sentiment extraction



Fine-tuning: sentiment extraction



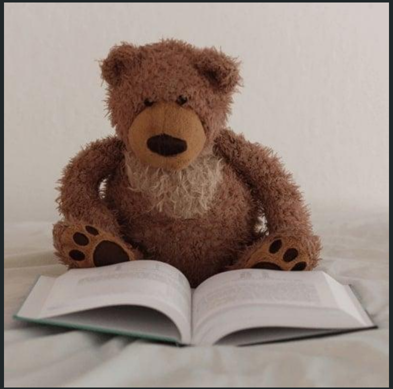
Break + questions



Hands-on

Notebook 2: **sentiment extraction with BERT**

Break + questions



Deep Learning for NLP, part II

Stanford ICME Summer workshop 2021

Main transformer-based models

Architecture	**Models**
Encoder	BERT, DistilBERT, RoBERTa
Decoder	GPT-2, 3
Encoder - Decoder	T5, mT5, ByT5

Latest trends

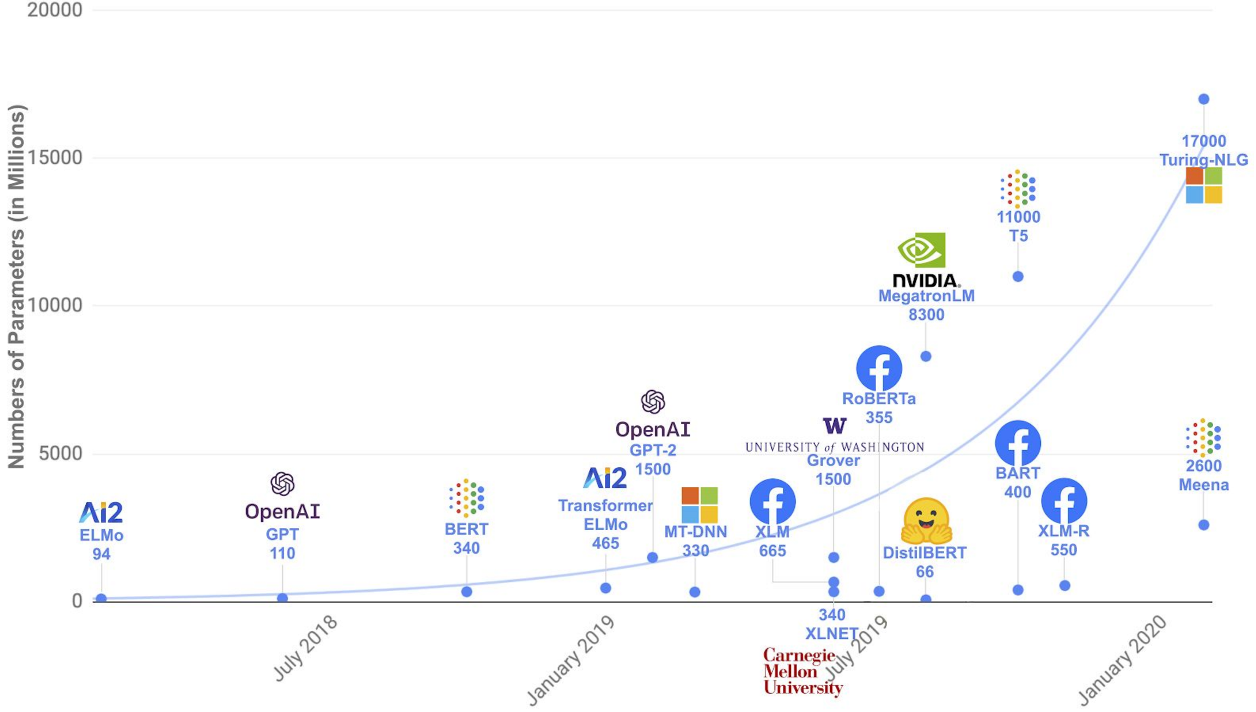


Figure adapted from this Tensorflow blog post, initially designed by the HuggingFace team.

Some final thoughts

- NLP is still an area with **many** unsolved problems
  - In-domain VS out-of-domain generalization

Some final thoughts

- NLP is still an area with **many** unsolved problems
  - In-domain VS out-of-domain generalization
  - Model size / Data requirement / Efficiency

Some final thoughts

- NLP is still an area with **many** unsolved problems
  - In-domain VS out-of-domain generalization
  - Model size / Data requirement / Efficiency
  - Hard to get common sense

Some final thoughts

- NLP is still an area with **many** unsolved problems
  - In-domain VS out-of-domain generalization
  - Model size / Data requirement / Efficiency
  - Hard to get common sense
  - Knowledge that changes with respect to time

Some final thoughts

- NLP is still an area with **many** unsolved problems
  - In-domain VS out-of-domain generalization
  - Model size / Data requirement / Efficiency
  - Hard to get common sense
  - Knowledge that changes with respect to time

- Performance can only be at most **as good as labels**
  - Challenges with NLG labeling
  - Interesting: "All That's Human is Not Gold: Evaluating Human Evaluation of Generated Text" by Clark et al., 2021

How to stay up-to-date with NLP advances

Papers

- arXiv > Computer Science > Computation and Language (curated: arxiv-sanity)
- General ML (NeurIPS, ICML, ICLR) and NLP (ACL, EMNLP) venues

How to stay up-to-date with NLP advances

Papers

- arXiv > Computer Science > Computation and Language (curated: arxiv-sanity)
- General ML (NeurIPS, ICML, ICLR) and NLP (ACL, EMNLP) venues

Code

- Authors' GitHub repositories linked in their papers
- paperswithcode.com: browse state-of-the-art datasets/methods for each task

How to stay up-to-date with NLP advances

Papers

- arXiv > Computer Science > Computation and Language (curated: arxiv-sanity)
- General ML (NeurIPS, ICML, ICLR) and NLP (ACL, EMNLP) venues

Code

- Authors' GitHub repositories linked in their papers
- paperswithcode.com: browse state-of-the-art datasets/methods for each task

Miscellaneous

- Stanford course websites (CS 224N, CS 224U and to some extent CS 230)
- Twitter (academics + industry leaders)
- YouTube theoretical (Two Minute Papers, Yannic Kilcher) and practical (Google Developers, HuggingFace)
- Company/academia technical papers + blogs (Amazon Science, Apple ML, Google AI, Google Brain, Microsoft Research, Stanford NLP)

Flashback from before first class

*"Some **context on evolution of NLP** will be super helpful"*

*"Start from **medium level** and then go upwards to **higher difficulty**"*

*"Hope to get the **summary of the materials** (including additional articles/books) and **links to them** to have better understanding. Hope to **try BERT in practice** (in Python notebooks). [...]"*

*I am very interested in **hands on** experience. I hope this session will help to start running my first deep learning model.*

*"[...] My hope is we **dig into the code** and the **details** of running a program"*

*"Among the given topics, I am more interested in the application areas of **sentiment extraction**"*

Thank you for your attention!