Applied Deep Learning

BERT

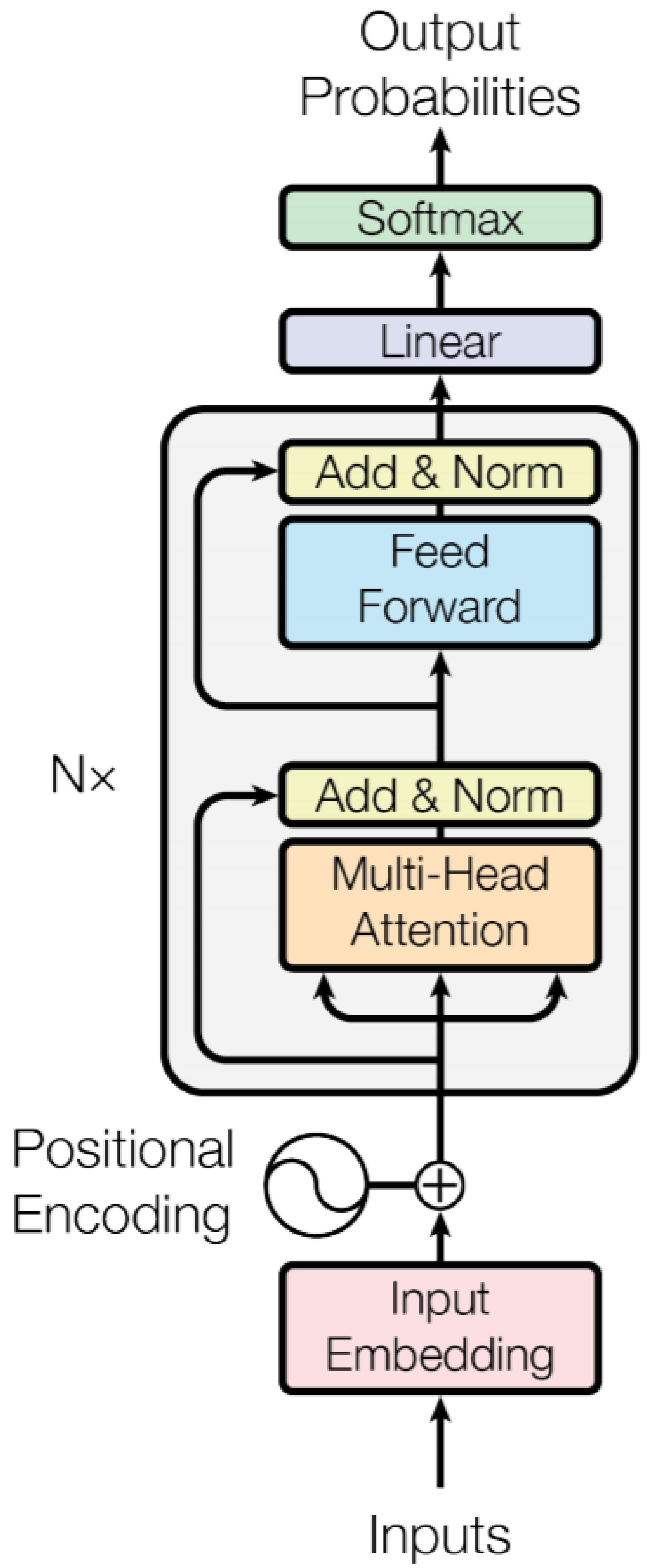
Bidirectional Encoder Representations from Transformers

April 7th, 2020 http://adl.miulab.tw

國立臺灣大學 National Taiwan University

BERT: Bidirectional Encoder Representations from Transformers

- Idea: contextualized word representations
  - Learn word vectors using long contexts using Transformer instead of LSTM

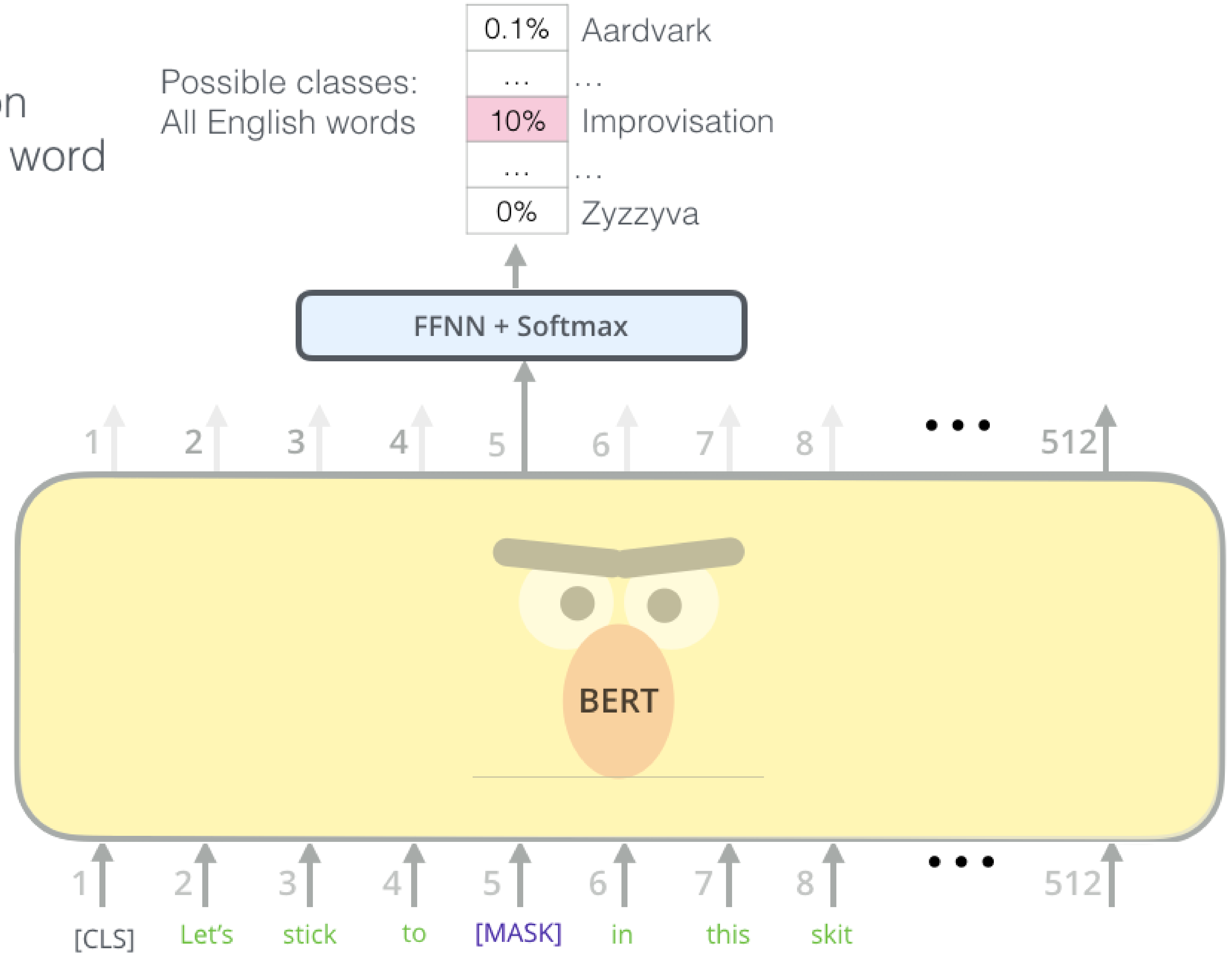


Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", in *NAACL-HLT*, 2019.

3 BERT #1 – Masked Language Model

- Idea: language understanding is **bidirectional** while LM only uses *left* or *right* context

Use the output of the masked word's position to predict the masked word



Randomly mask 15% of tokens
- Too little: expensive to train
- Too much: not enough context

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

BERT #1 – Masked Language Model

Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", in *NAACL-HLT*, 2019.

BERT #2 – Next Sentence Prediction

- Idea: modeling ***relationship*** between sentences
  - QA, NLI etc. are based on understanding inter-sentence relationship

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]`

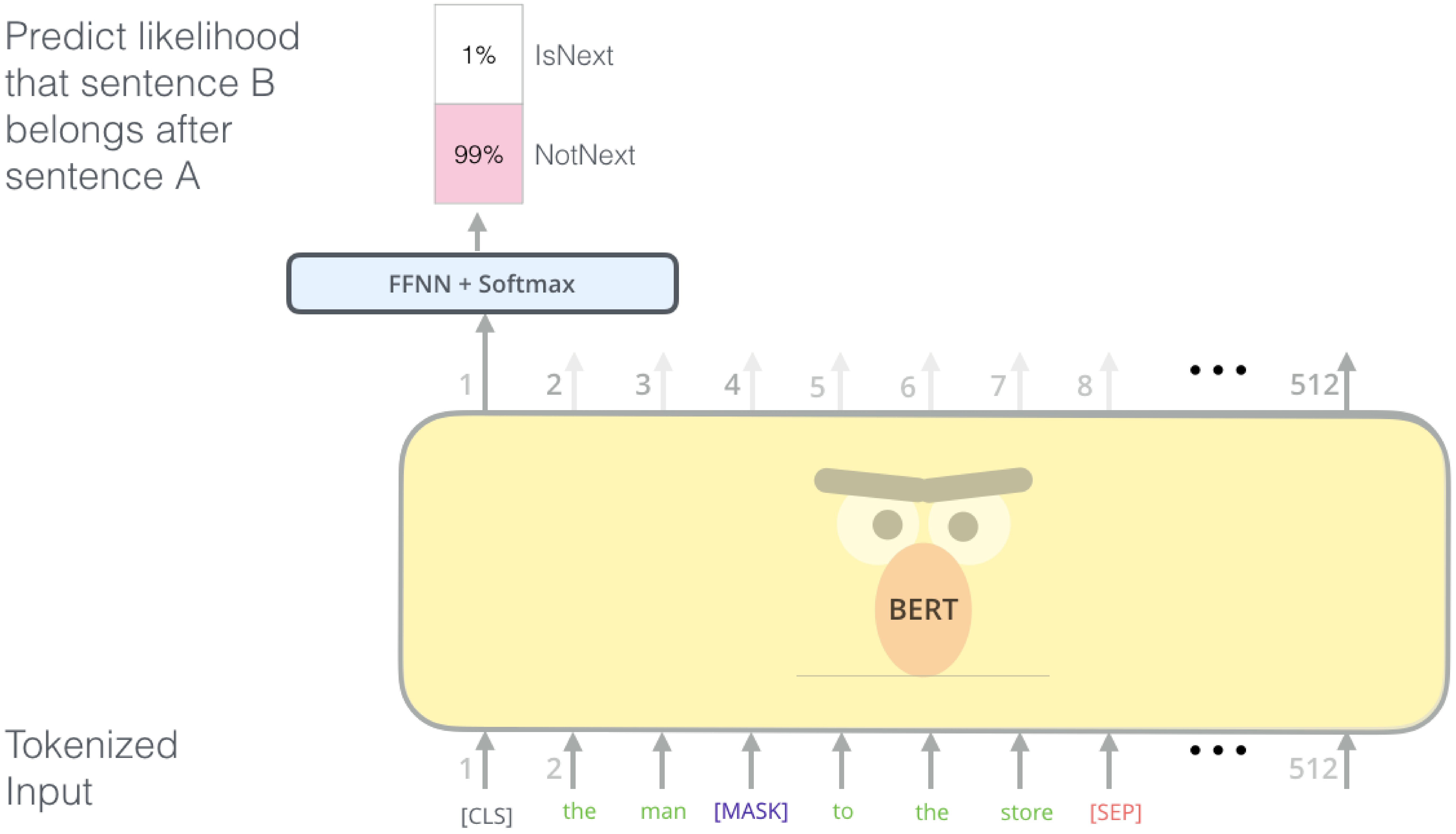
`penguin [MASK] are flight ##less birds [SEP]`

Label = `NotNext`

Devlin et al., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", in *NAACL-HLT*, 2019.

BERT #2 – Next Sentence Prediction

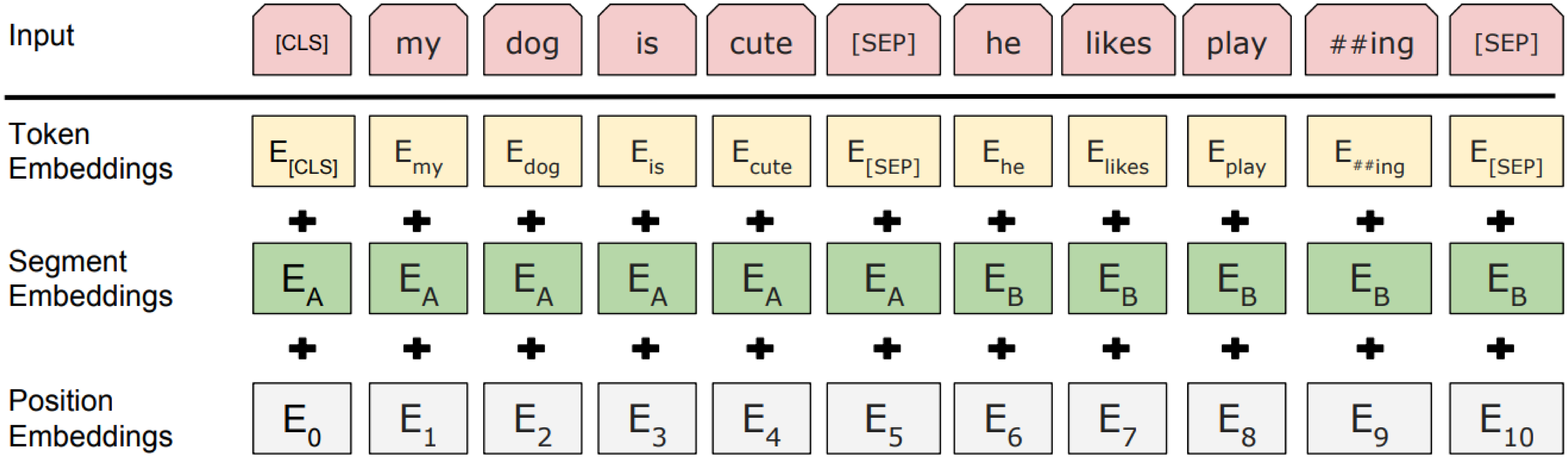
- Idea: modeling relationship between sentences



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

BERT – Input Representation

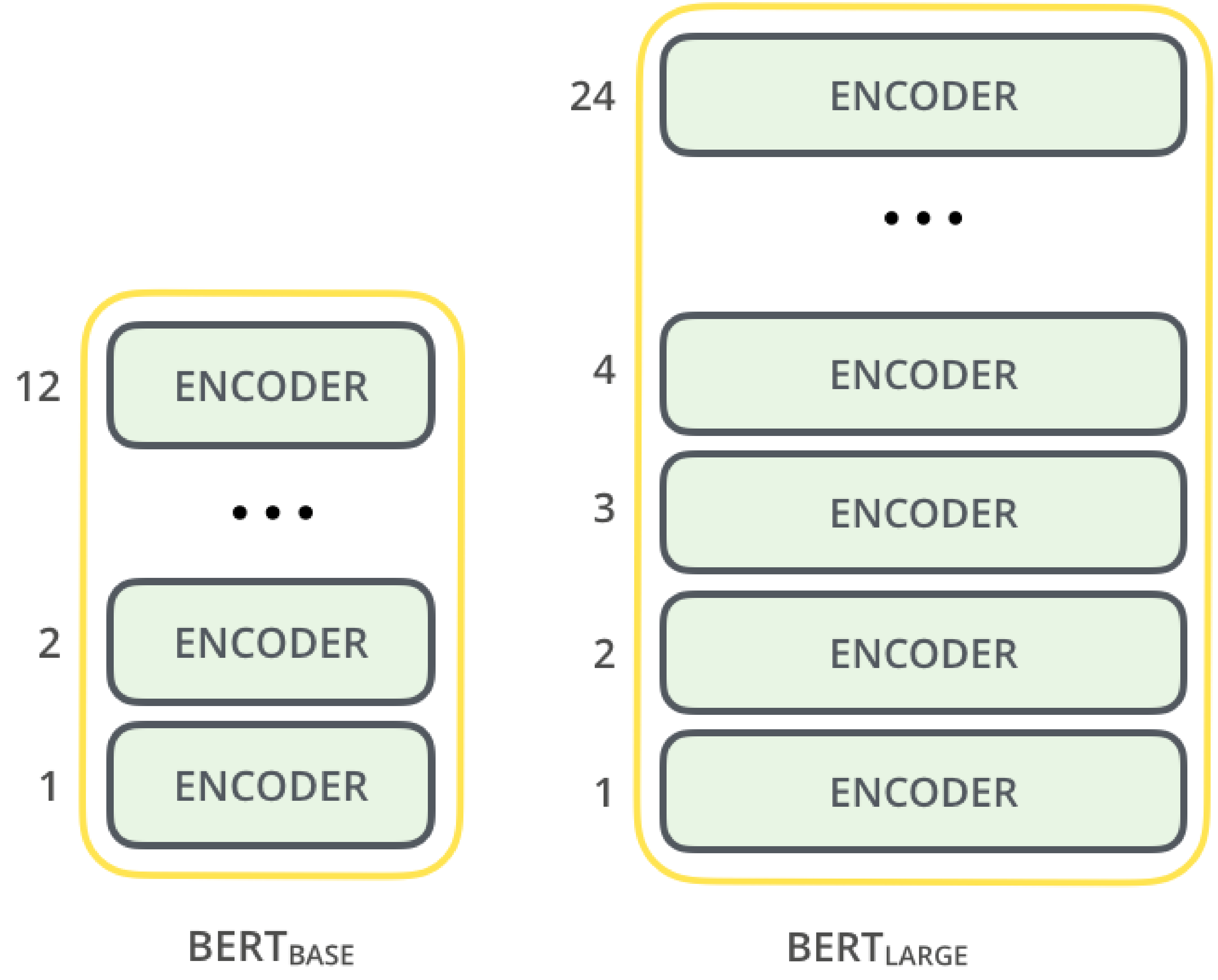
- Input embeddings contain
  - Word-level token embeddings
  - Sentence-level segment embeddings
  - Position embeddings

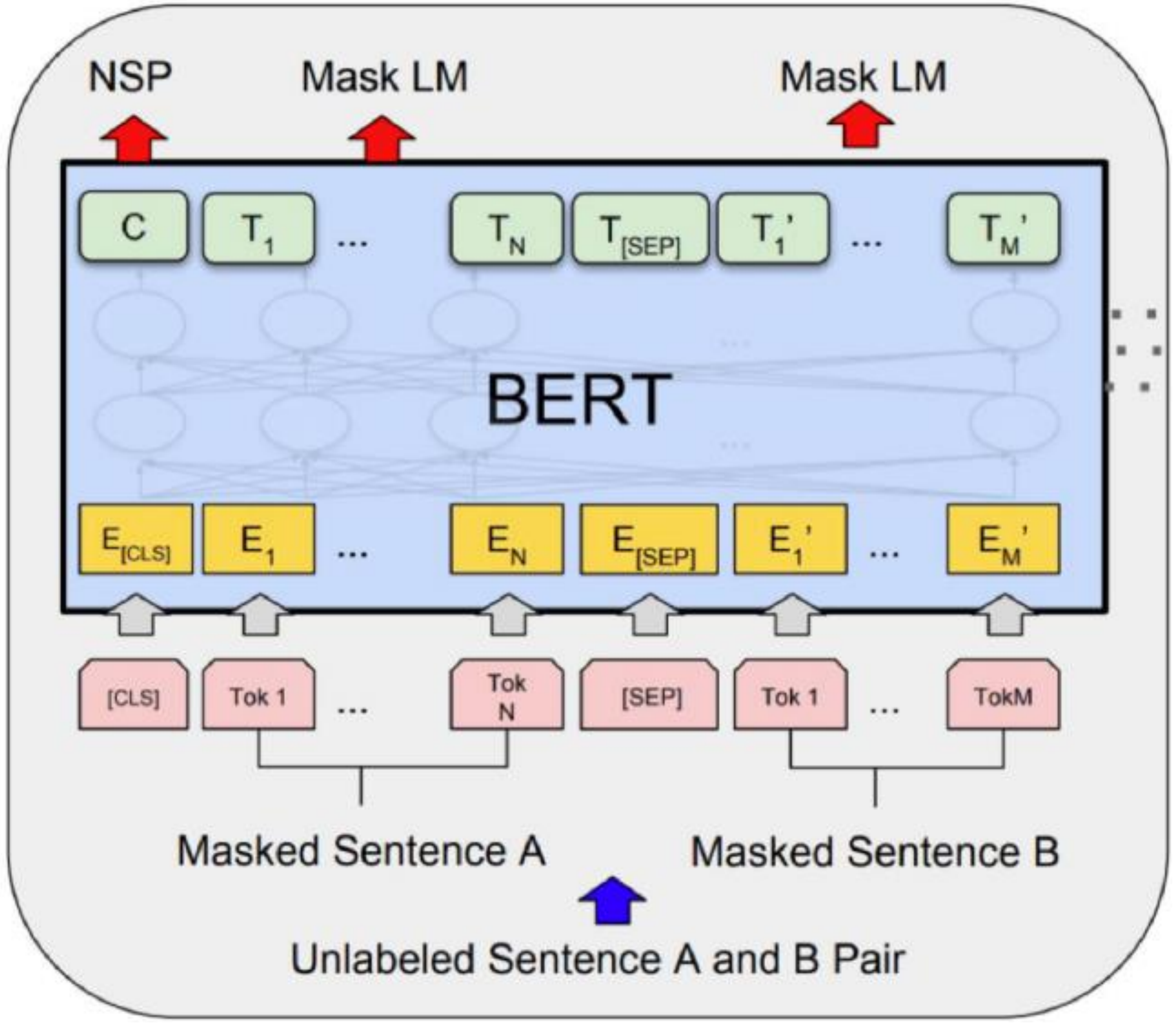


Devlin et al., “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”, in *NAACL-HLT*, 2019.

BERT Training

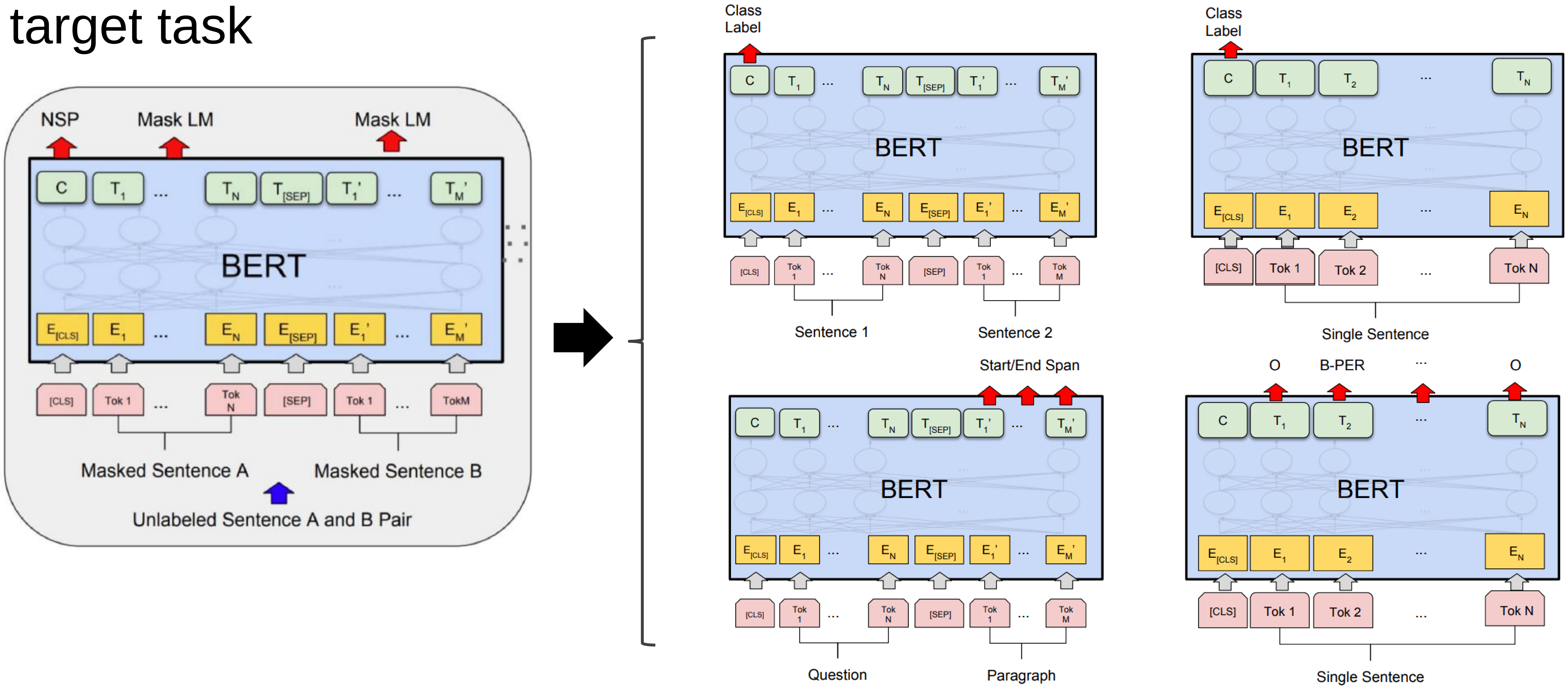
- Training data: Wikipedia + BookCorpus
- 2 BERT models
  - BERT-Base: 12-layer, 768-hidden, 12-head
  - BERT-Large: 24-layer, 1024-hidden, 16-head





BERT Fine-Tuning for Understanding Tasks

- Idea: simply learn a classifier/tagger built on the top layer for each target task

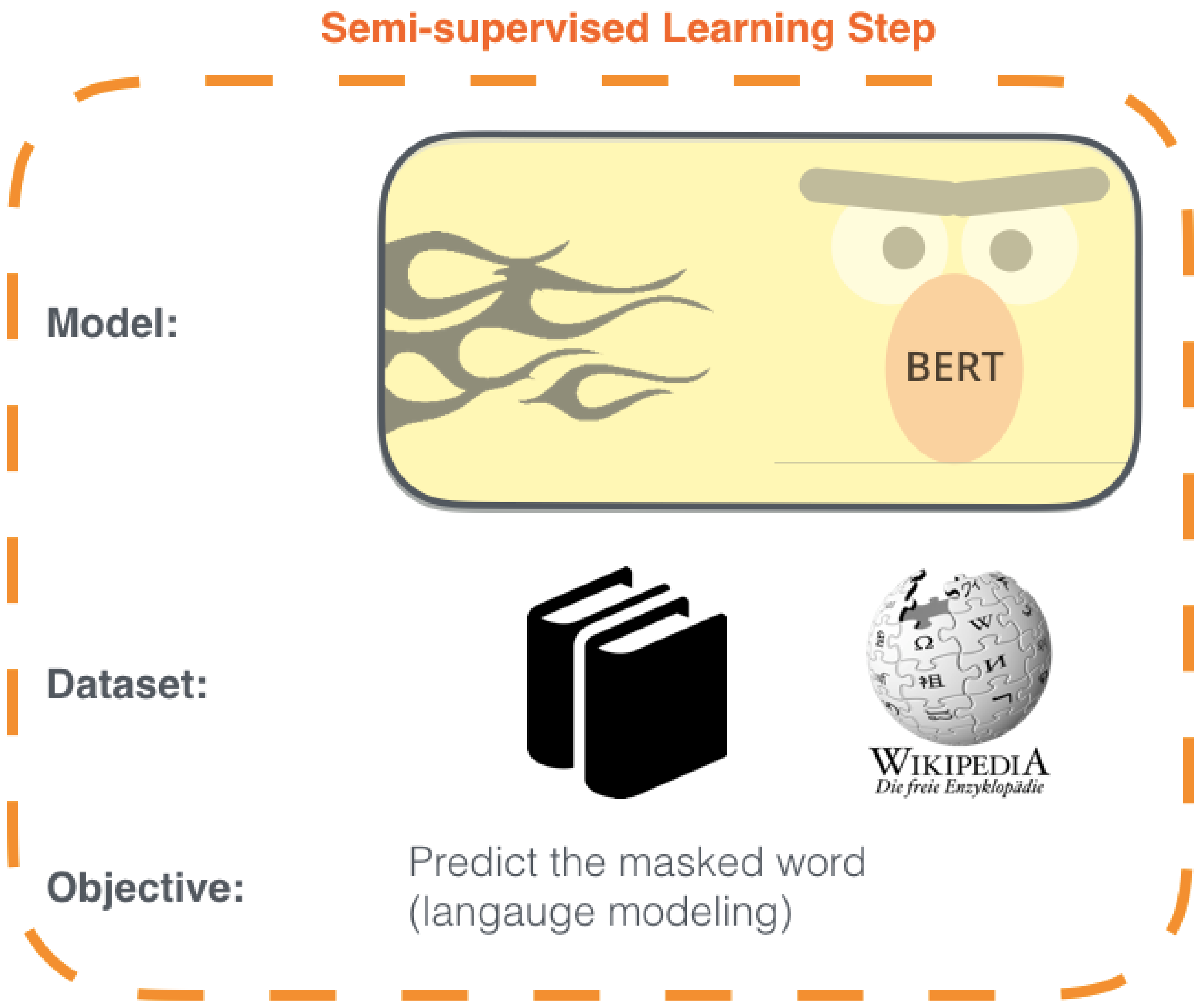


Devlin et al., “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”, in *NAACL-HLT*, 2019.

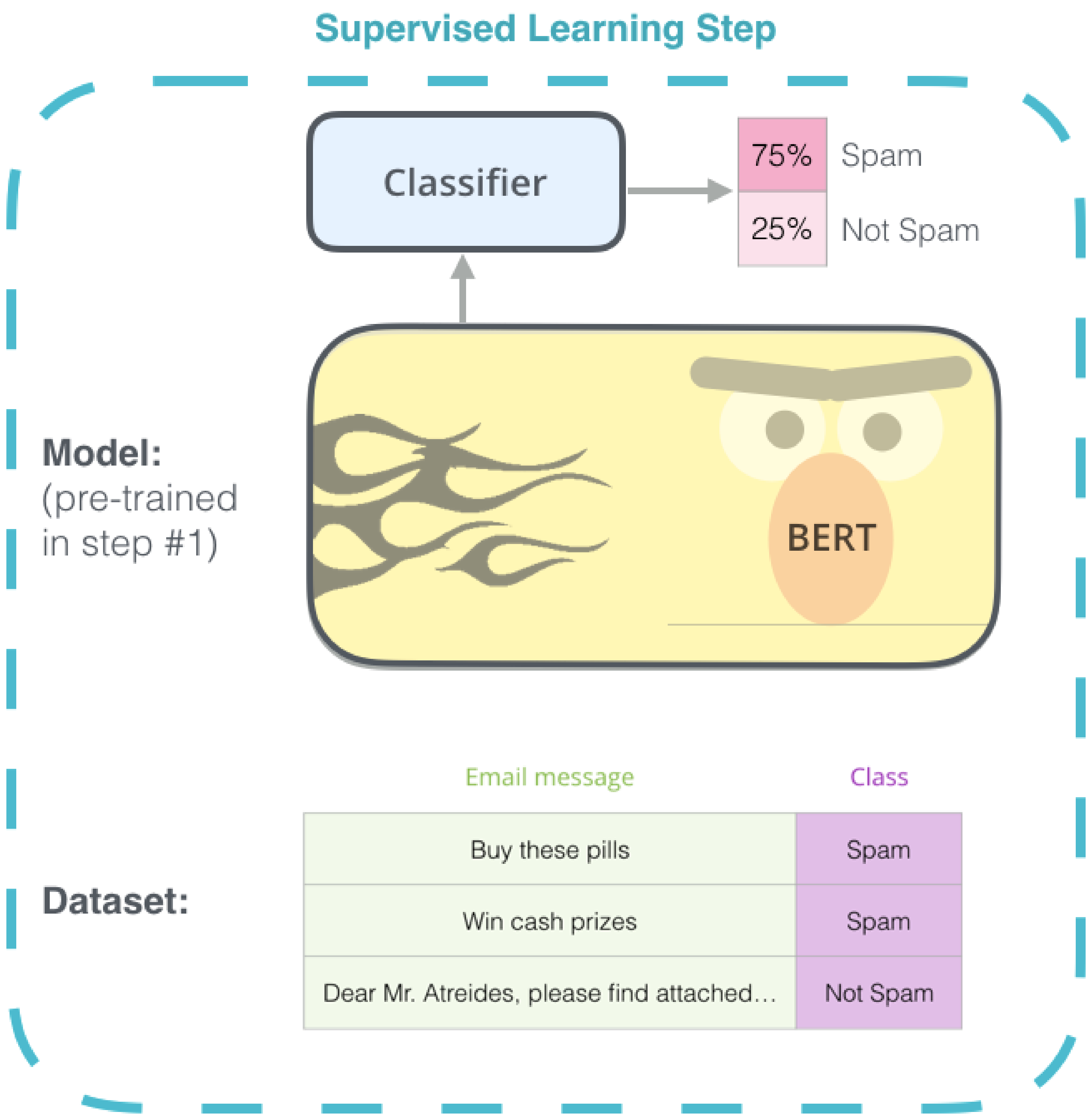
BERT Overview

1 - Semi-supervised training on large amounts of text (books, wikipedia..etc).

The model is trained on a certain task that enables it to grasp patterns in language. By the end of the training process, BERT has language-processing abilities capable of empowering many models we later need to build and train in a supervised way.

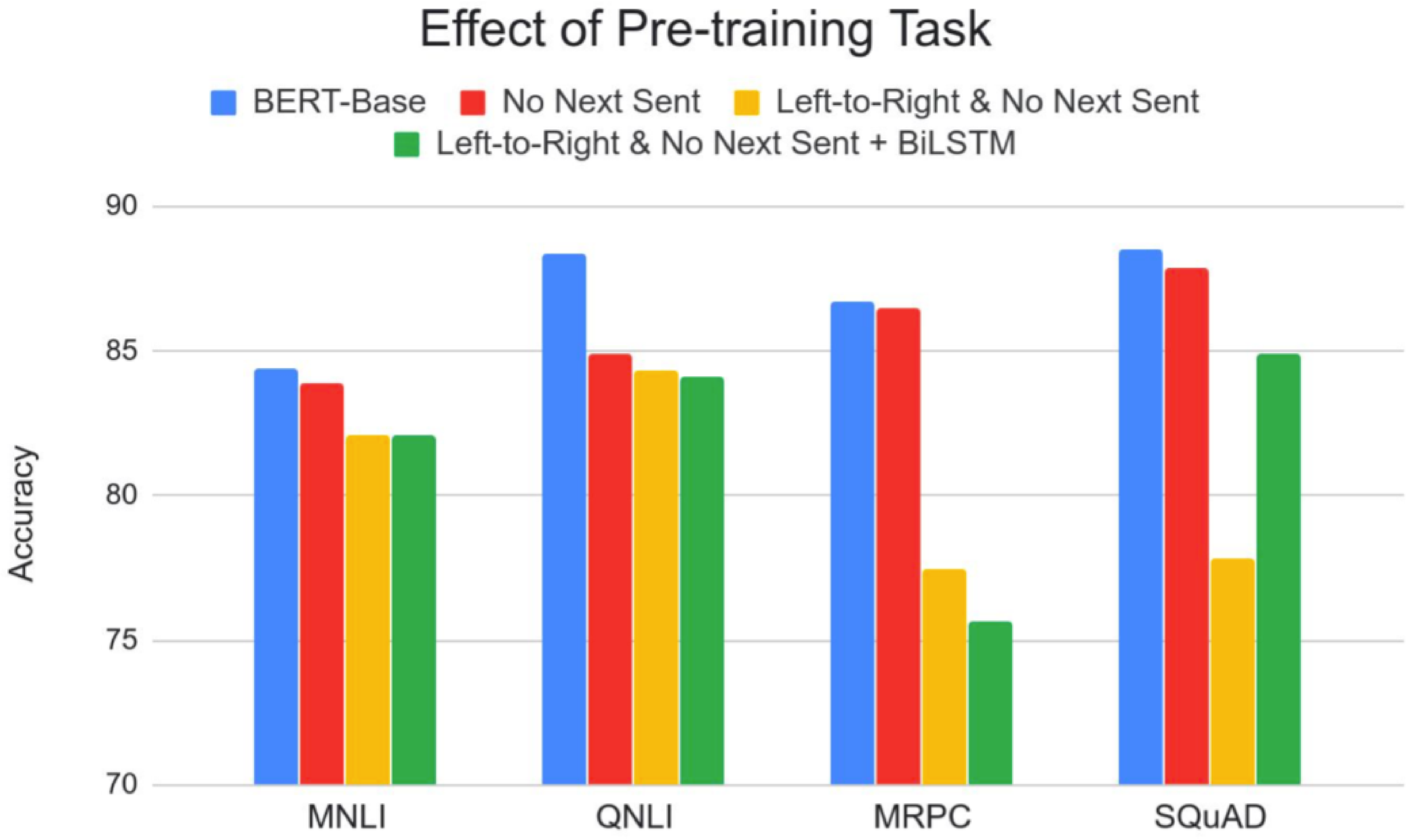


2 - Supervised training on a specific task with a labeled dataset.



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

BERT Fine-Tuning Results



Devlin et al., “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”, in *NAACL-HLT*, 2019.

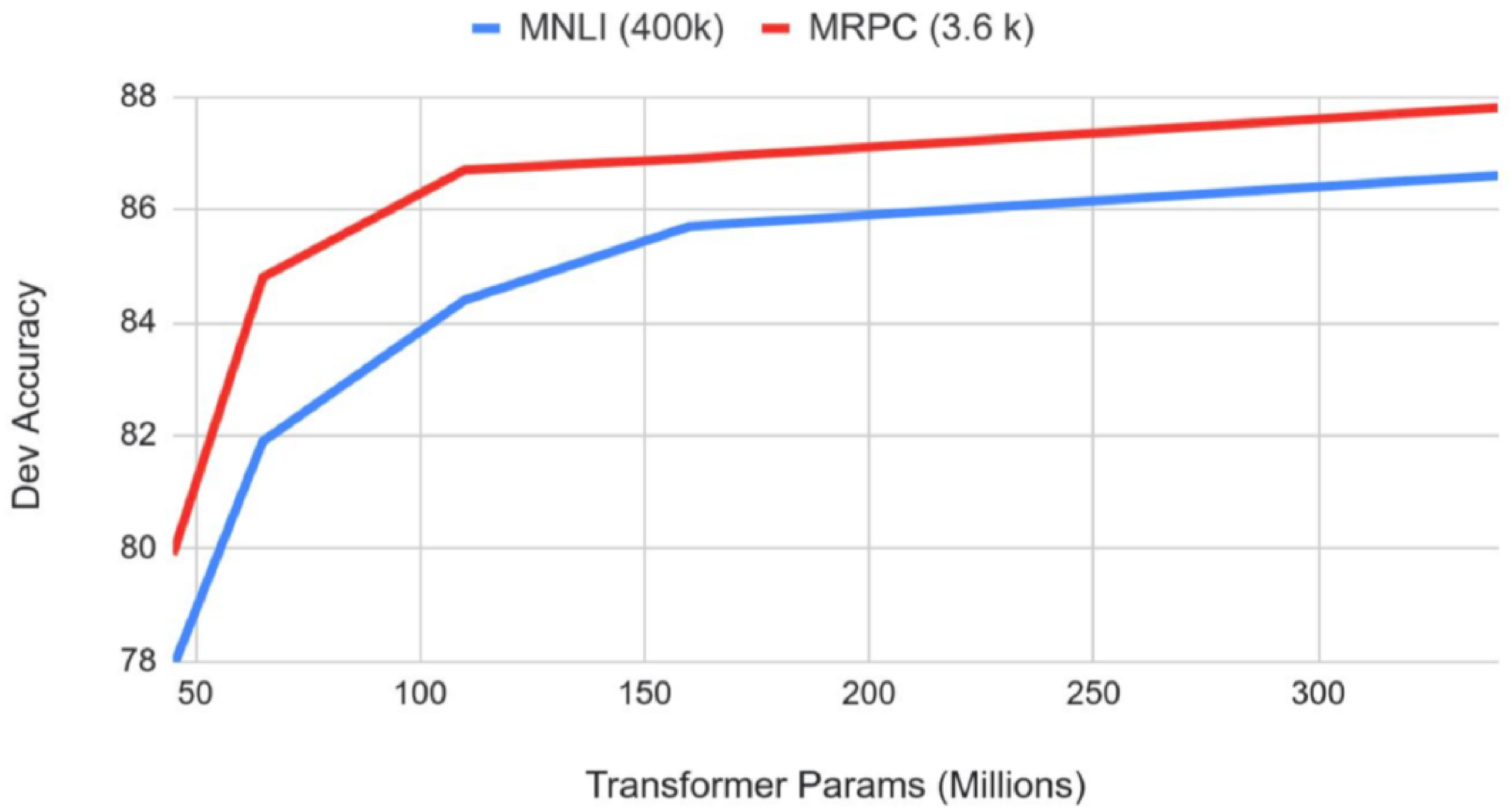
BERT Results on NER

Model	Description	CONLL 2003 F1
TagLM (Peters+, 2017)	LSTM BiLM in BLSTM Tagger	91.93
ELMo (Peters+, 2018)	ELMo in BLSTM	92.22
BERT-Base (Devlin+, 2019)	Transformer LM + fine-tune	<u>92.4</u>
CVT Clark	Cross-view training + multitask learn	92.61
BERT-Large (Devlin+, 2019)	Transformer LM + fine-tune	<u>92.8</u>
Flair	Character-level language model	93.09

Devlin et al., “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”, in *NAACL-HLT*, 2019.

BERT Results with Different Model Sizes

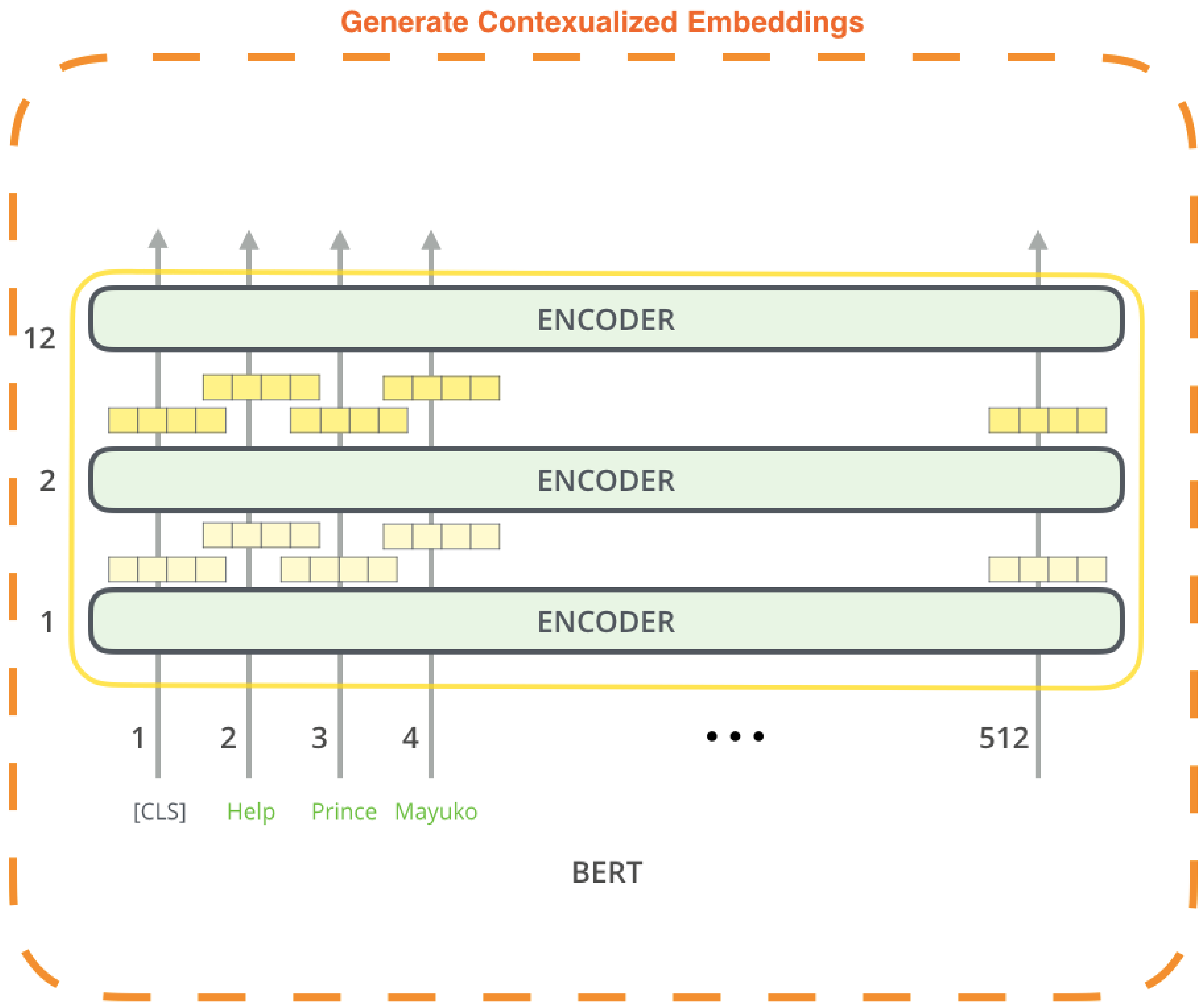
- Improving performance by increasing model size



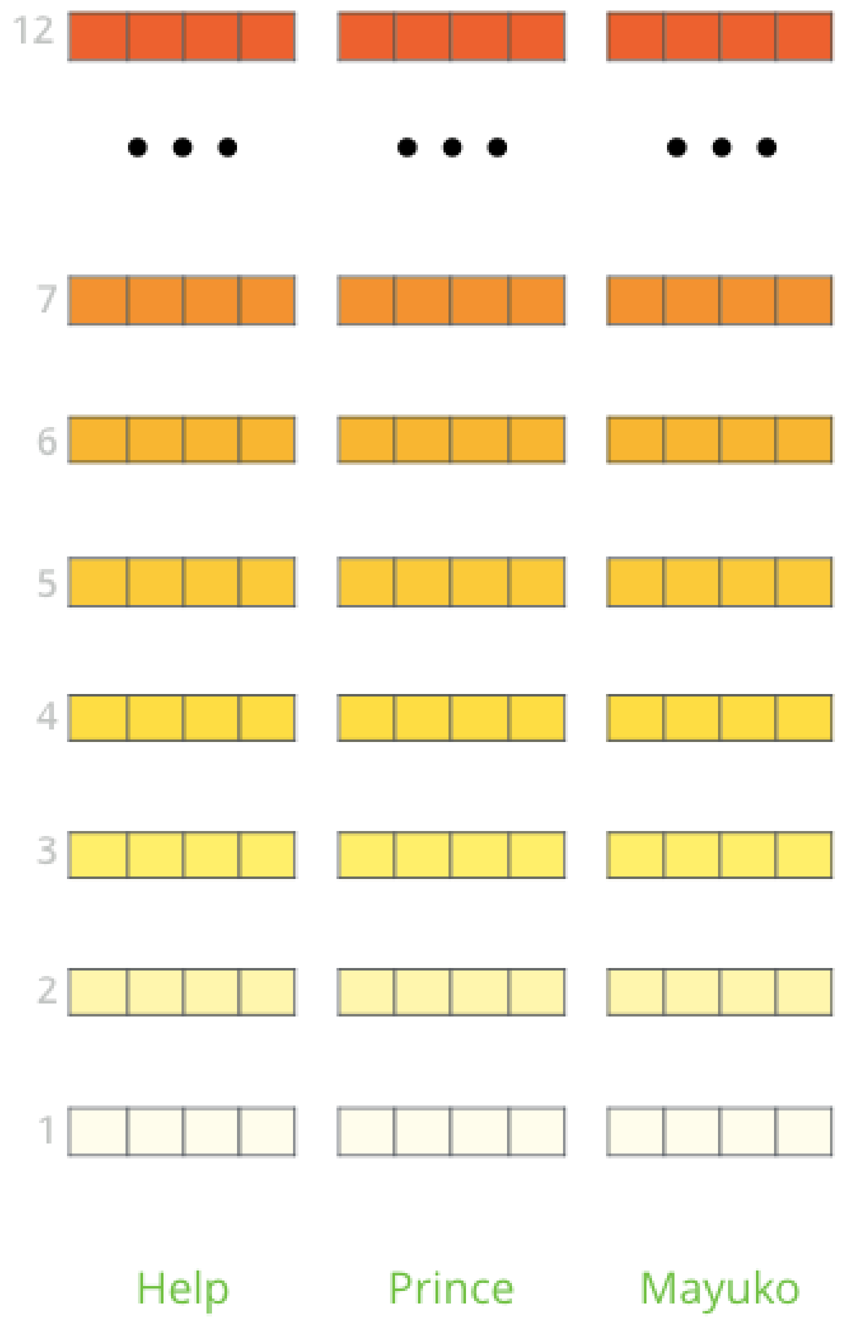
Devlin et al., “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”, in *NAACL-HLT*, 2019.

BERT for Contextual Embeddings

- Idea: use pre-trained BERT to get contextualized word embeddings and feed them into the task-specific models



The output of each encoder layer along each token's path can be used as a feature representing that token.

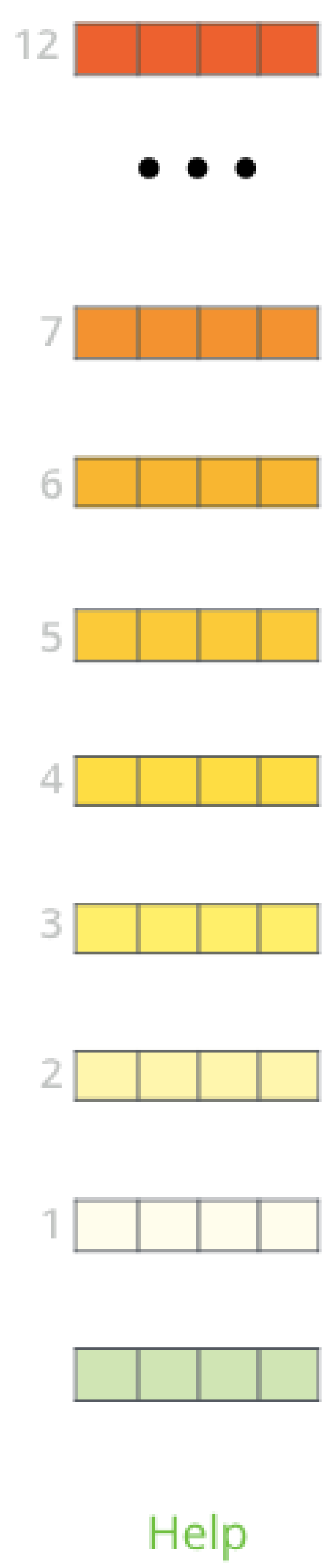


But which one should we use?

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

BERT Contextual Embeddings Results on NER

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



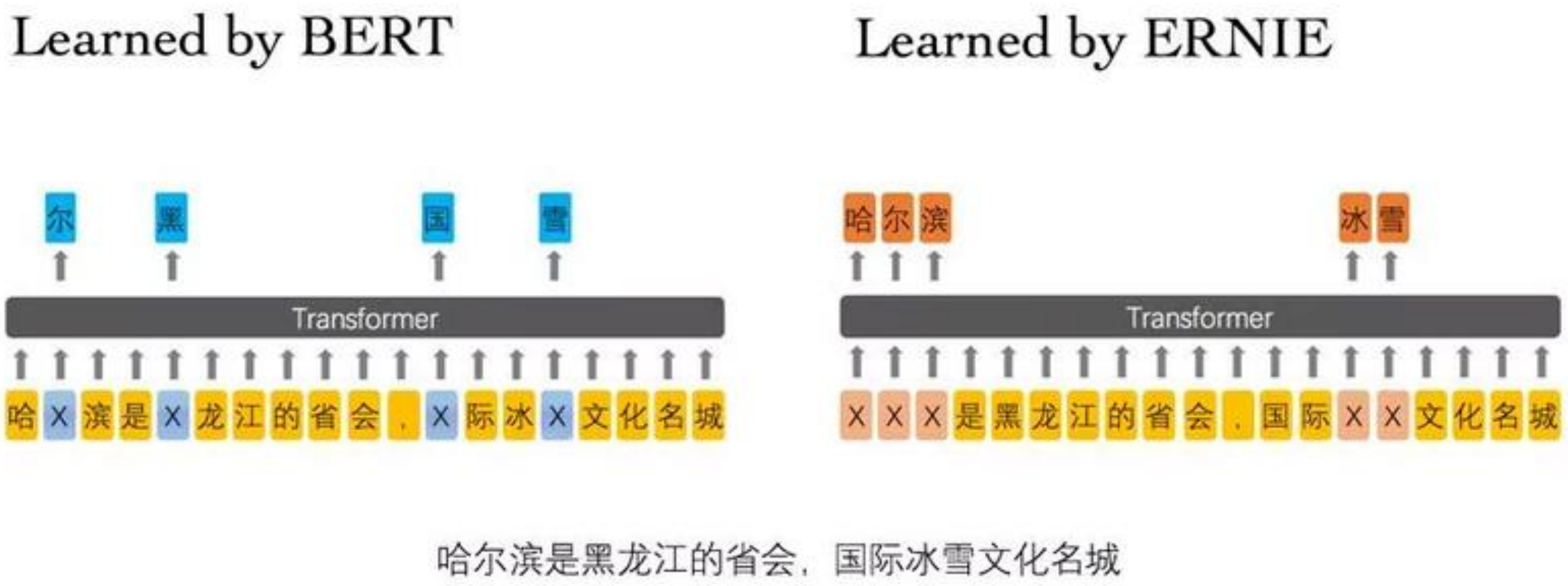
		Dev F1 Score
First Layer	Embedding	91.0
Last Hidden Layer	12	94.9
Sum All 12 Layers	12 + … + 2 + 1 =	95.5
Second-to-Last Hidden Layer	11	95.6
Sum Last Four Hidden	12 + 11 + 10 + 9 =	95.9
Concat Last Four Hidden	9 10 11 12	96.1

fine-tune = 96.4

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

ERNIE: Enhanced Representation through kNowledge IntEgration

- BERT models local cooccurrence between tokens, while characters are modeled independently
  - 哈(ha), 爾(er), 濱(bin) instead 哈爾濱(Harbin)
- ERNIE incorporates knowledge by masking semantic units/entities



Concluding Remarks

- Contextualized embeddings learned from masked LM via Transformers provide informative cues for **transfer learning**
- BERT – a general approach for learning contextual representations from Transformers and benefiting language understanding
  - ✓ Pre-trained BERT:
    https://github.com/google-research/bert
    https://github.com/huggingface/transformers