Transformer based Models

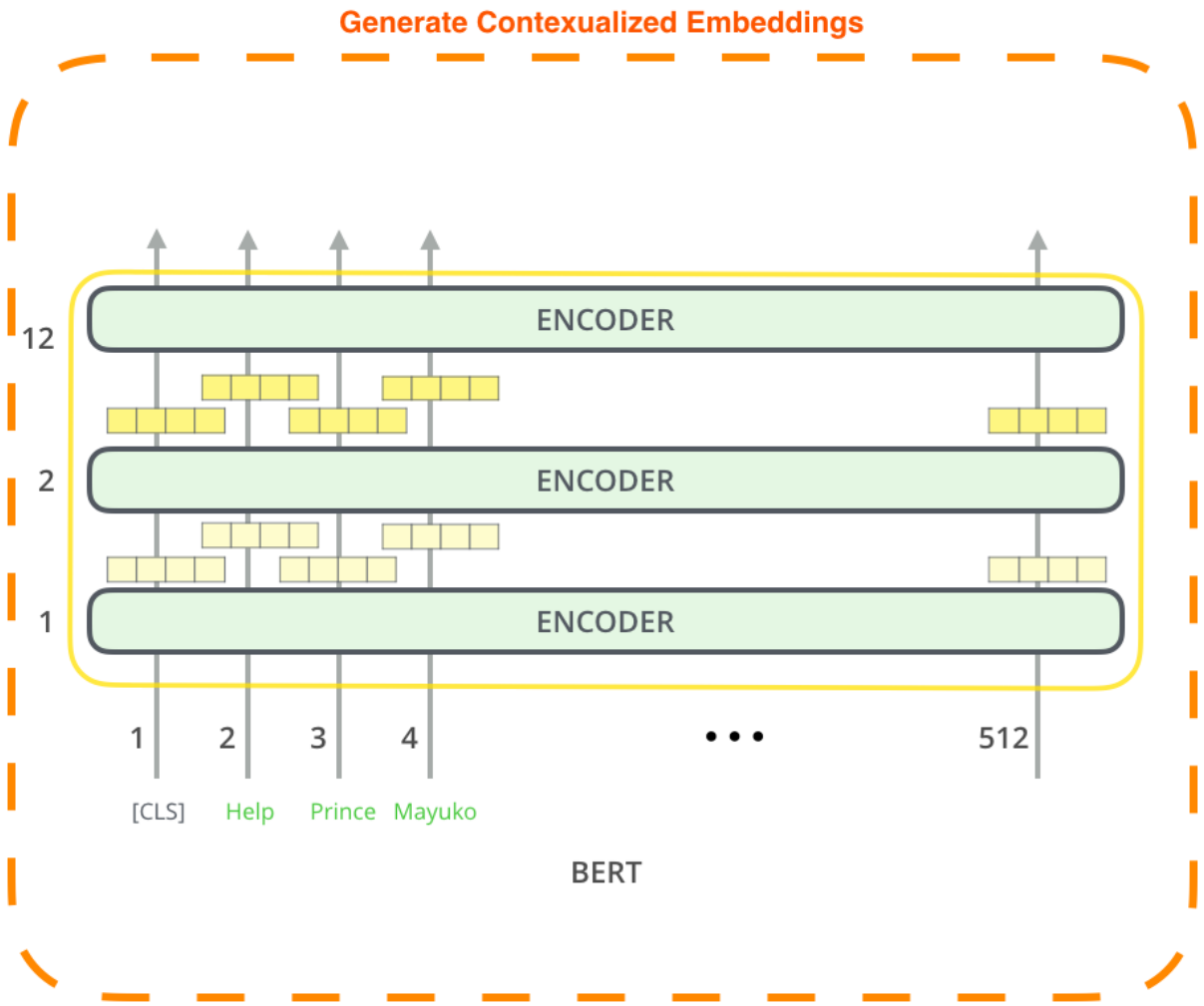
Lê Anh Cường

TDTU 2024

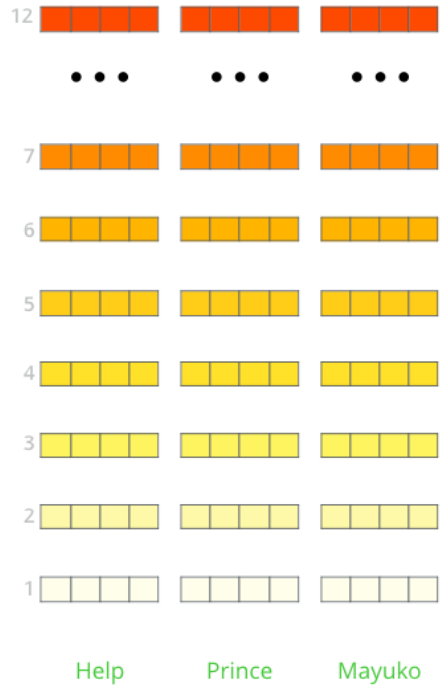
BERT (Bidirectional Encoder Representations from Transformers)

1. What is BERT? Why is BERT Important?
2. How to train BERT?
3. How to use BERT?

BERT



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?

1. What is BERT?

Full Form: Bidirectional Encoder Representations from Transformers.

Type: A pre-trained language model for Natural Language Processing (NLP).

Architecture: Based on the Transformer encoder, which uses self-attention mechanisms.

Bidirectional: Processes language contextually in both directions (left-to-right and right-to-left).

Pre-training: Trained on large text datasets (e.g., Wikipedia, BooksCorpus).

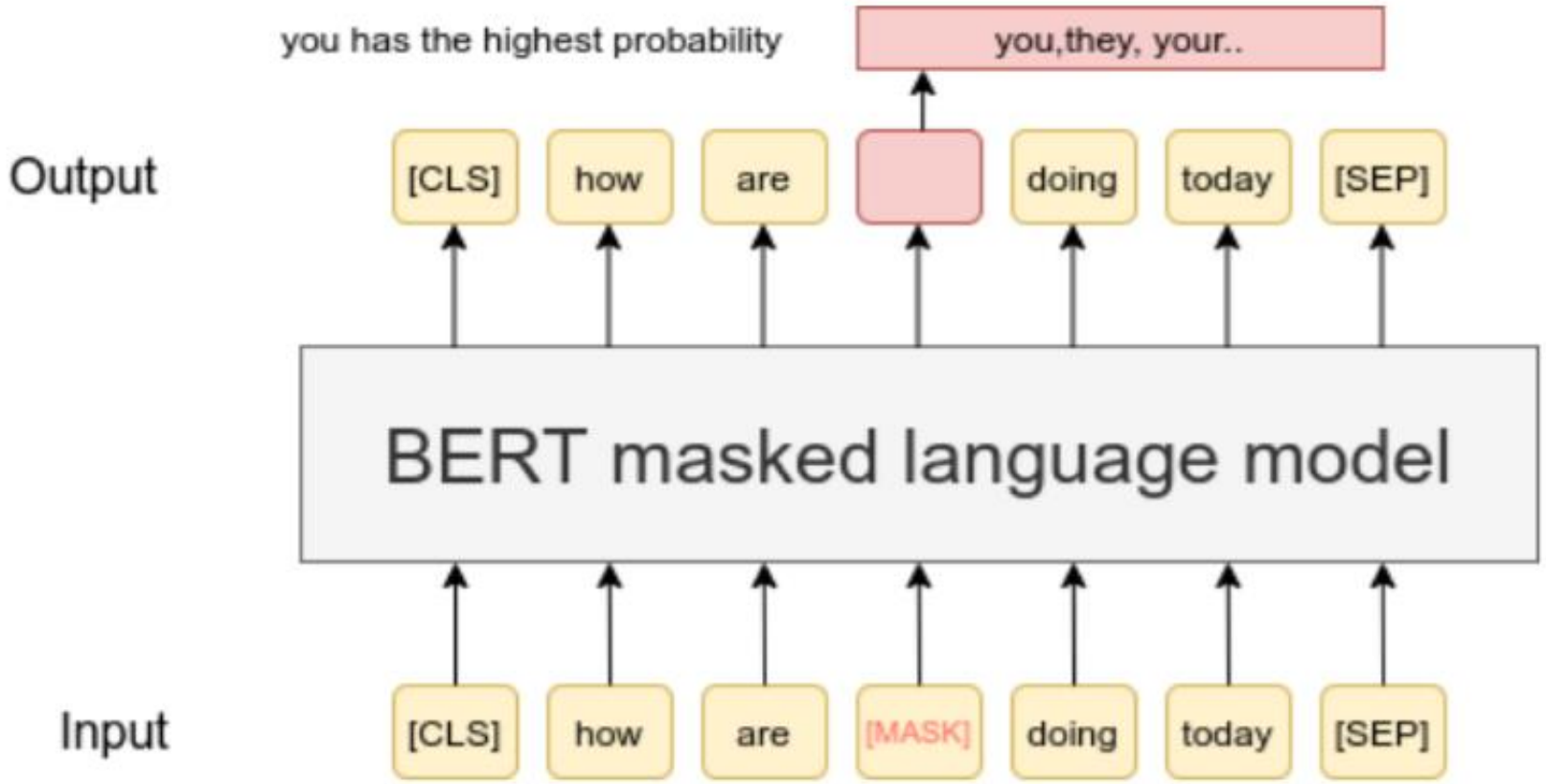
Tokenization: Uses WordPiece tokenization to handle rare and unknown words.

Fine-tuning: Adaptable for various NLP tasks like sentiment analysis, question answering, and text classification.

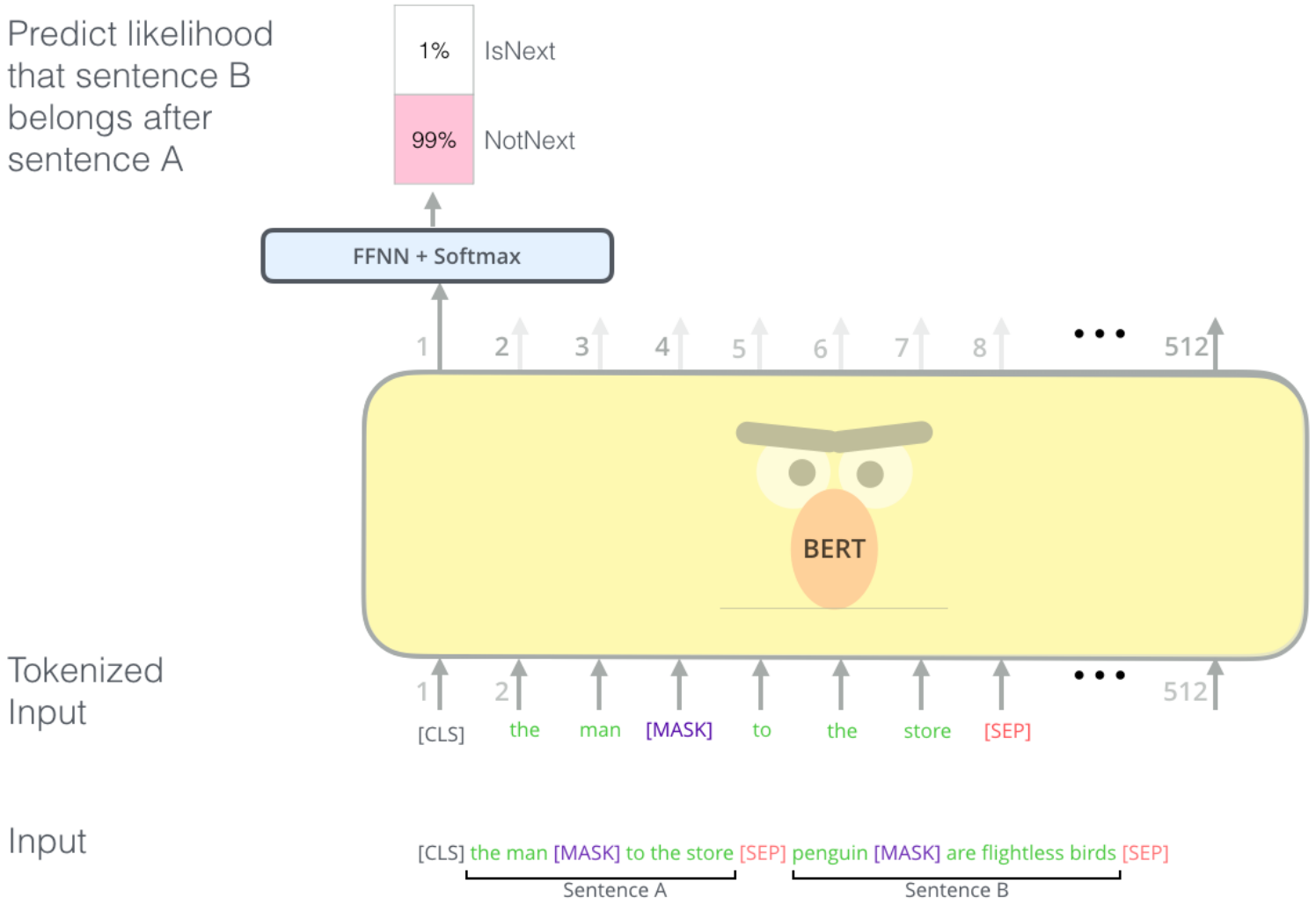
- **Impact**: Achieved state-of-the-art performance in numerous NLP benchmarks.
- **Key Training Objectives**:
  - **Masked Language Modeling (MLM)**: Predicts masked words in a sentence.
  - **Next Sentence Prediction (NSP)**: Determines sentence relationships.

Release: Developed by Google and open-sourced in 2018.

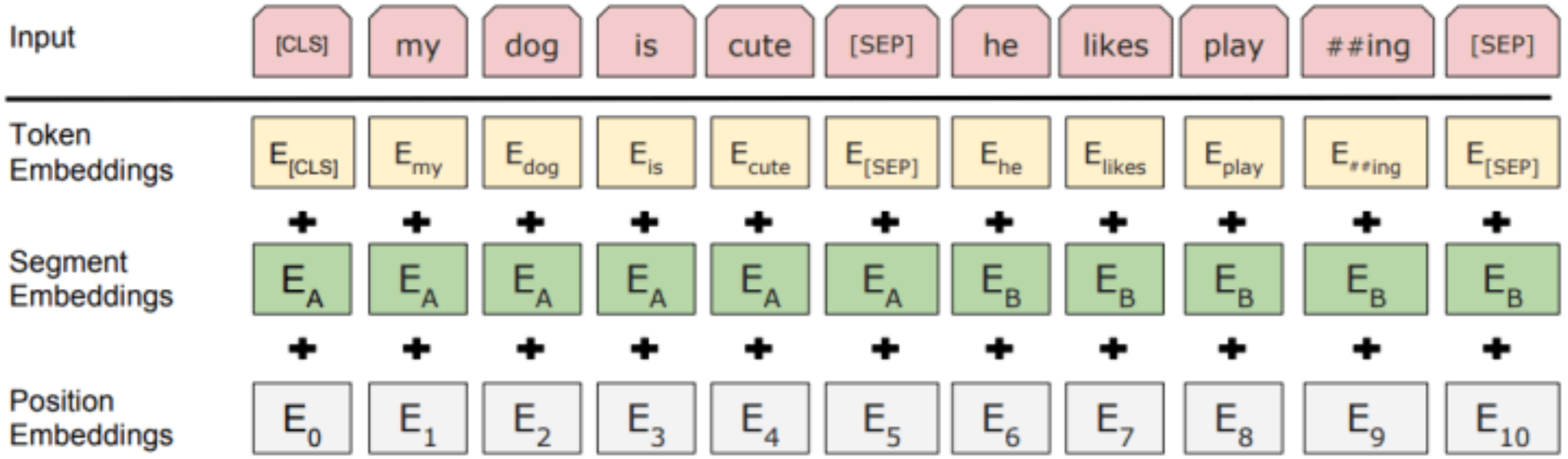
2. How BERT is trained?



Next Sentence Prediction (NSP): Determines sentence relationships.



Input


Input
[CLS] my dog is cute [SEP] he likes play ##ing [SEP]
Token Embeddings
E[CLS] Emy Edog Eis Ecute E[SEP] Ehe Elikes Eplay E##ing E[SEP]
Segment Embeddings
EA EA EA EA EA EA EB EB EB EB EB
Position Embeddings
E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 E10

1. Masked Language Modeling (MLM)

- **Objective**: Predict randomly masked words in a sentence using their context.
- **Process**:
  - Randomly mask **15% of the tokens** in each sentence.
  - Replace these tokens with:
    - `[MASK]` token (80% of the time).
    - A random word (10% of the time).
    - The original token (10% of the time).
  - The model predicts the original word for the masked positions.
- **Benefit**: Helps the model learn deep bidirectional representations of language, understanding both preceding and succeeding context.

2. **Next Sentence Prediction (NSP)**

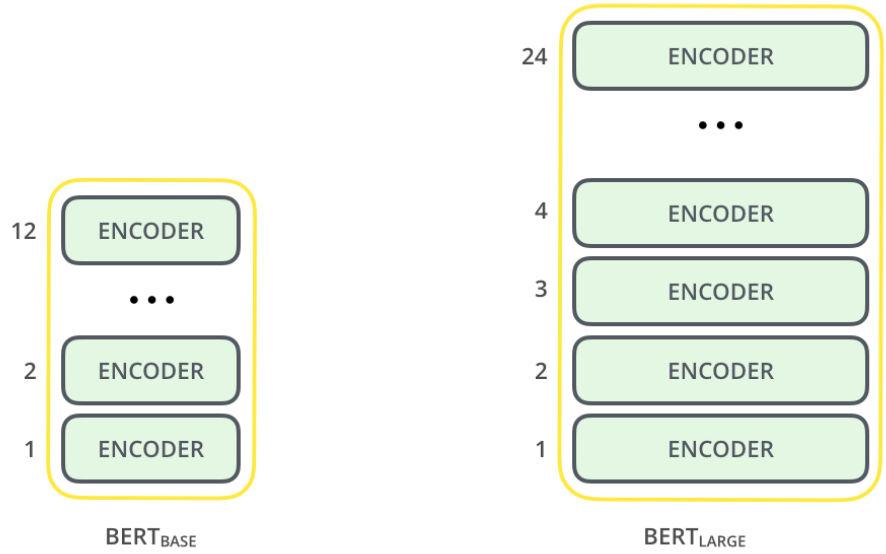
- **Objective**: Determine whether one sentence logically follows another.
- **Process**:
    - Sentences are paired as:
        - **Positive pairs (50%)**: The second sentence follows the first sentence in the original text.
        - **Negative pairs (50%)**: A random sentence is paired with the first sentence.
    - The model predicts whether the second sentence is a continuation of the first.
- **Benefit**: Enables BERT to understand relationships between sentences, which is critical for tasks like question answering and natural language inference.

Training data

3. Training Data

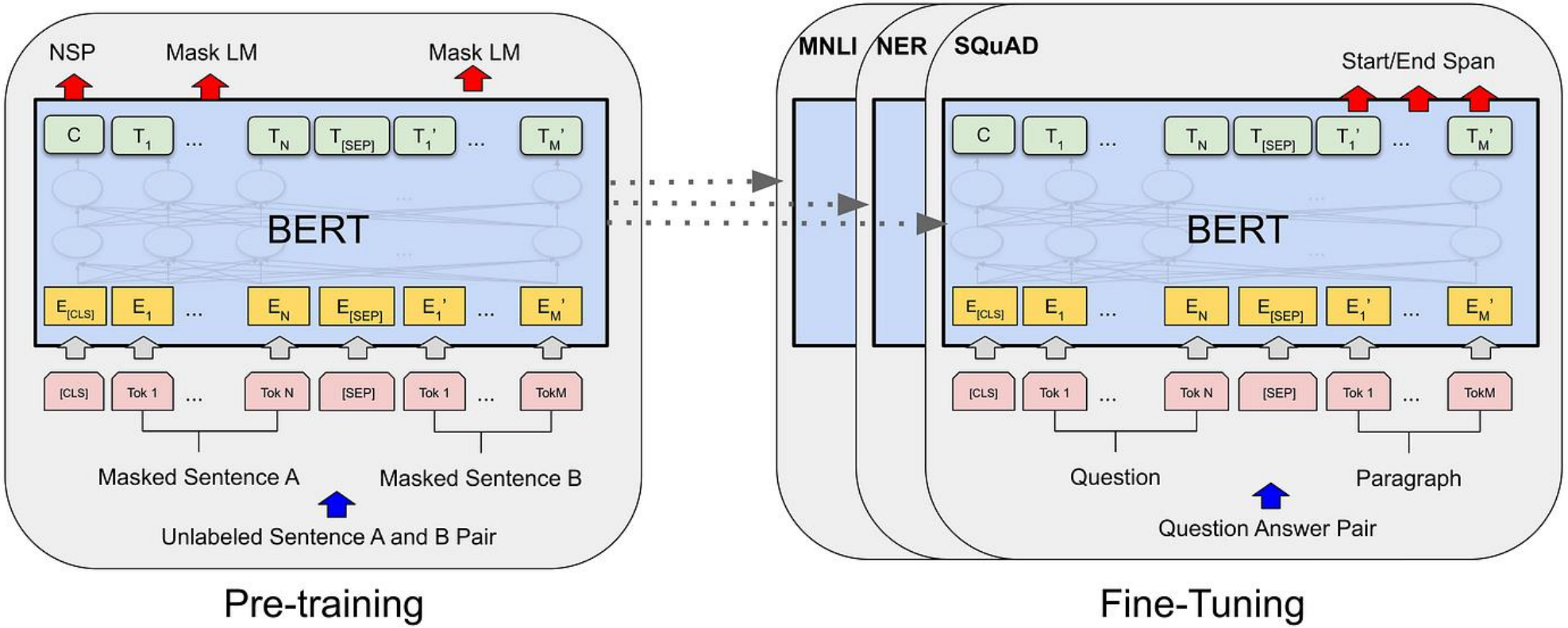
- BERT is pre-trained on large-scale datasets:
  - **Wikipedia (2.5 billion words)**.
  - **BooksCorpus (800 million words)**.
- It uses these diverse and rich datasets to learn general language understanding.

Architecture of BERTs

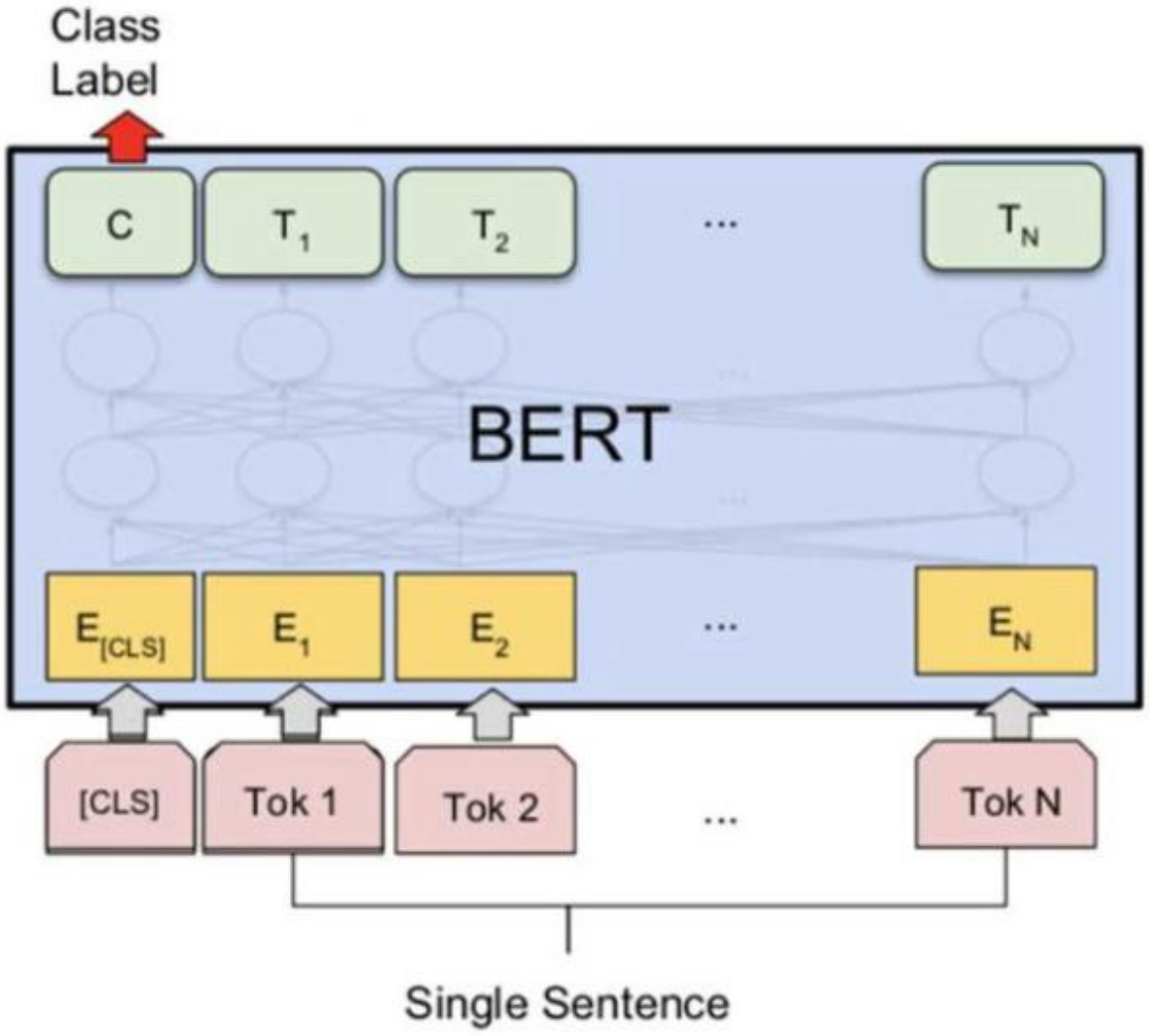


- Both BERT model sizes have a large number of encoder layers (which the paper calls Transformer Blocks) – twelve for the Base version, and twenty four for the Large version. These also have larger feedforward-networks (768 and 1024 hidden units respectively), and more attention heads (12 and 16 respectively) than the default configuration in the reference implementation of the Transformer in the initial paper (6 encoder layers, 512 hidden units, and 8 attention heads).

Fine-Tuning BERT for Specific Tasks



Fine tune BERT for text classification



Other pretrained models

- Distil BERT
- RoBERTa
- XLNet

	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

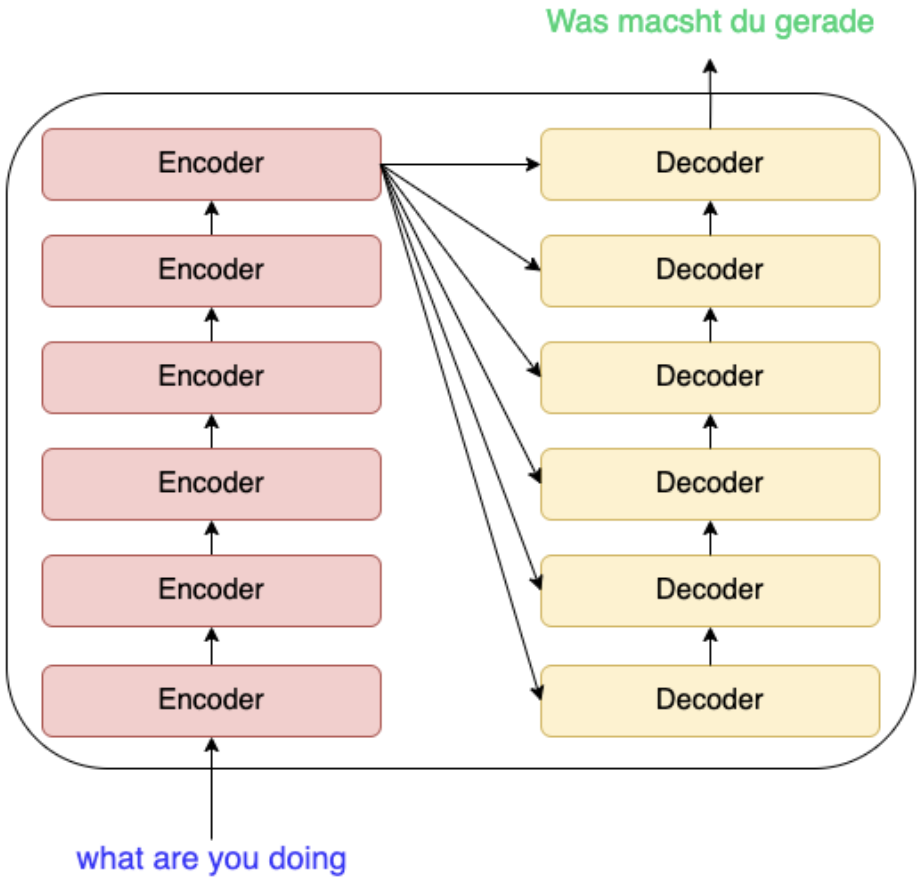
Comparison	BERT October 11, 2018	RoBERTa July 26, 2019	DistilBERT October 2, 2019	ALBERT September 26, 2019
Parameters	**Base:** 110M **Large:** 340M	**Base:** 125 **Large:** 355	**Base:** 66	**Base:** 12M **Large:** 18M
Layers / Hidden Dimensions / Self-Attention Heads	**Base:** 12 / 768 / 12 **Large:** 24 / 1024 / 16	**Base:** 12 / 768 / 12 **Large:** 24 / 1024 / 16	**Base:** 6 / 768 / 12	**Base:** 12 / 768 / 12 **Large:** 24 / 1024 / 16
Training Time	**Base:** 8 x V100 x 12d **Large:** 280 x V100 x 1d	1024 x V100 x 1 day (4-5x more than BERT)	**Base:** 8 x V100 x 3.5d (4 times less than BERT)	[not given] **Large:** 1.7x faster
Performance	Outperforming SOTA in Oct 2018	88.5 on GLUE	97% of BERT-base's performance on GLUE	89.4 on GLUE
Pre-Training Data	BooksCorpus + English Wikipedia = 16 GB	BERT + CCNews + OpenWebText + Stories = 160 GB	BooksCorpus + English Wikipedia = 16 GB	BooksCorpus + English Wikipedia = 16 GB
Method	Bidirectional Trans-former, MLM & NSP	BERT without NSP, Using Dynamic Masking	BERT Distillation	BERT with reduced para-meters & SOP (not NSP)

Practice

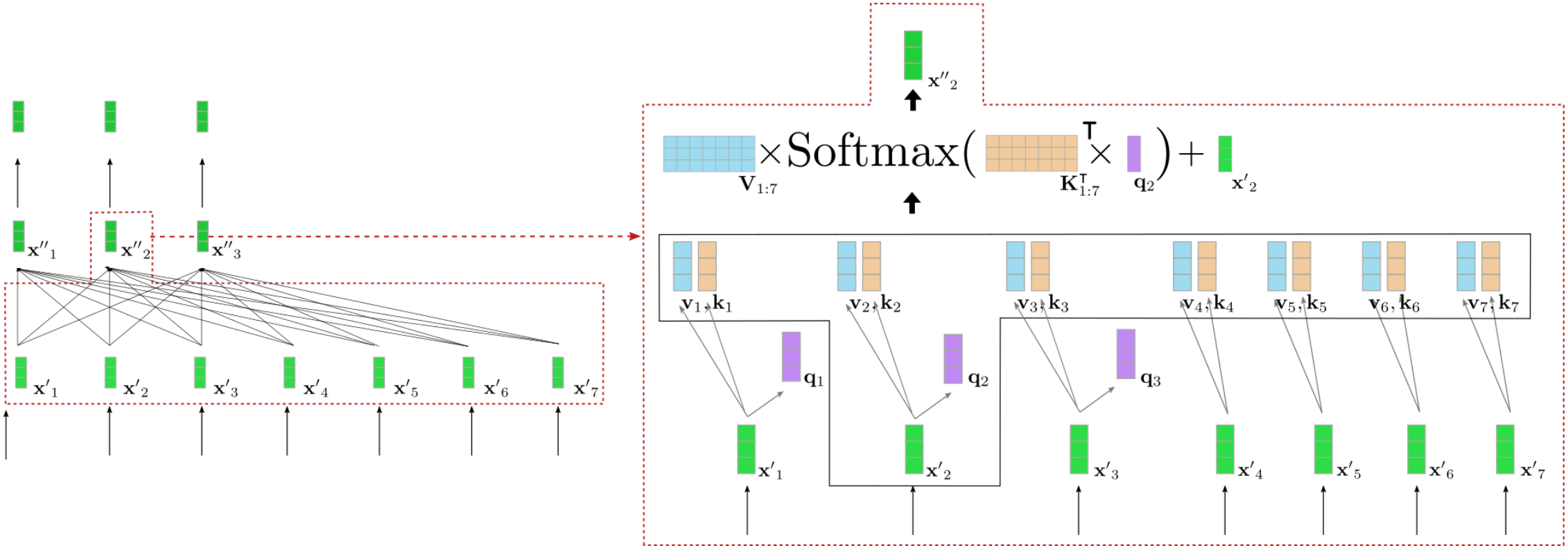
- https://colab.research.google.com/drive/1JvQvc6RZzlwVeWTAv7p2HMqHCzyesCLk

Encoder-Decoder Transformer

- T5,
- Bart,
- Pegasus,
- ProphetNet,
- Marge,
- *etc*...



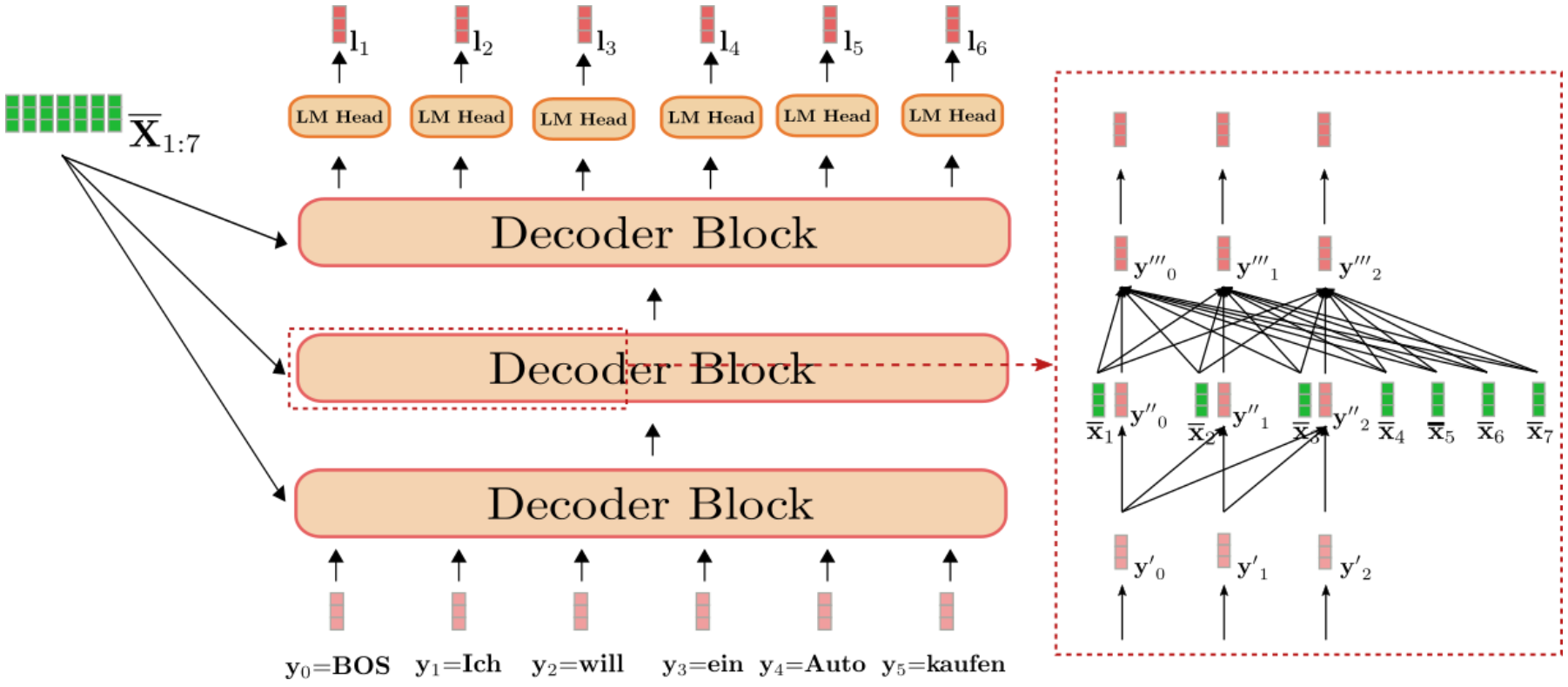
How to Encode



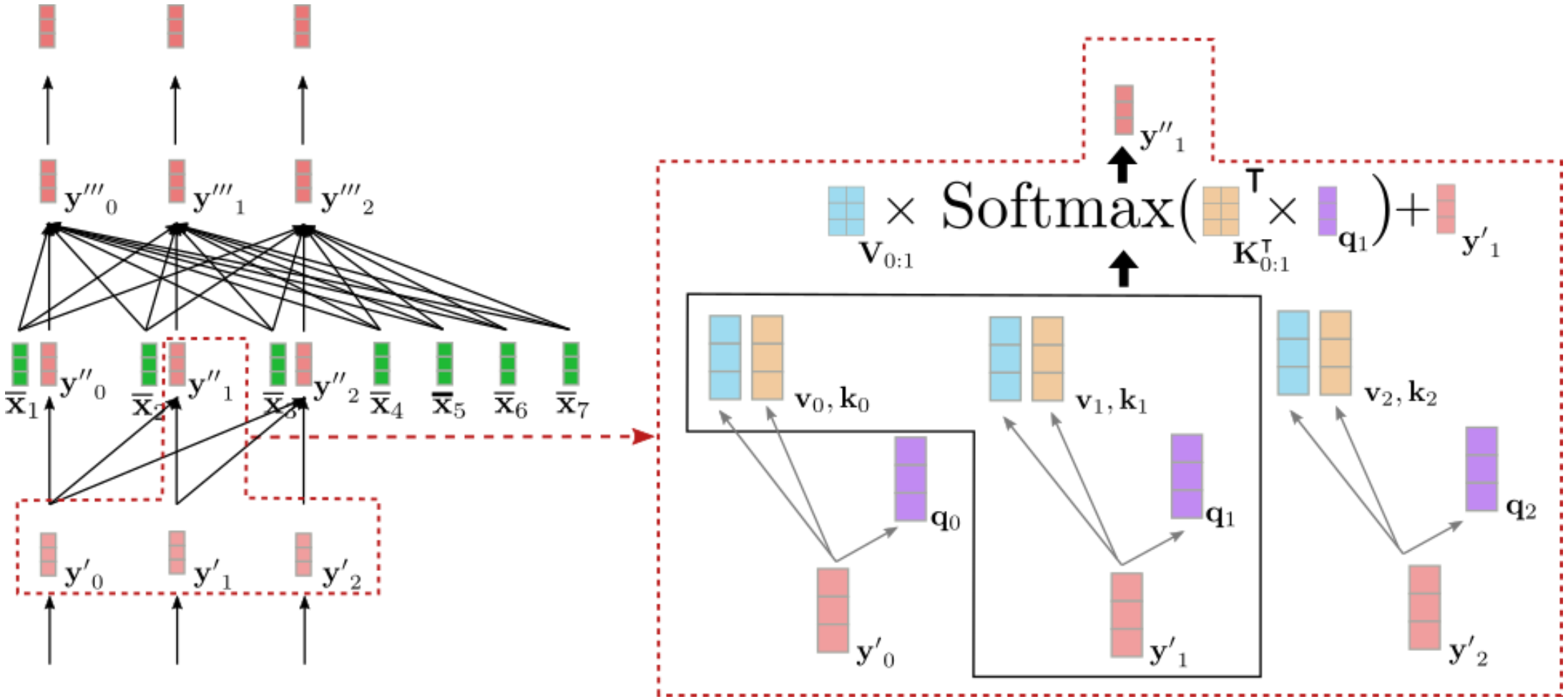
https://huggingface.co/blog/encoder-decoder

How to decode

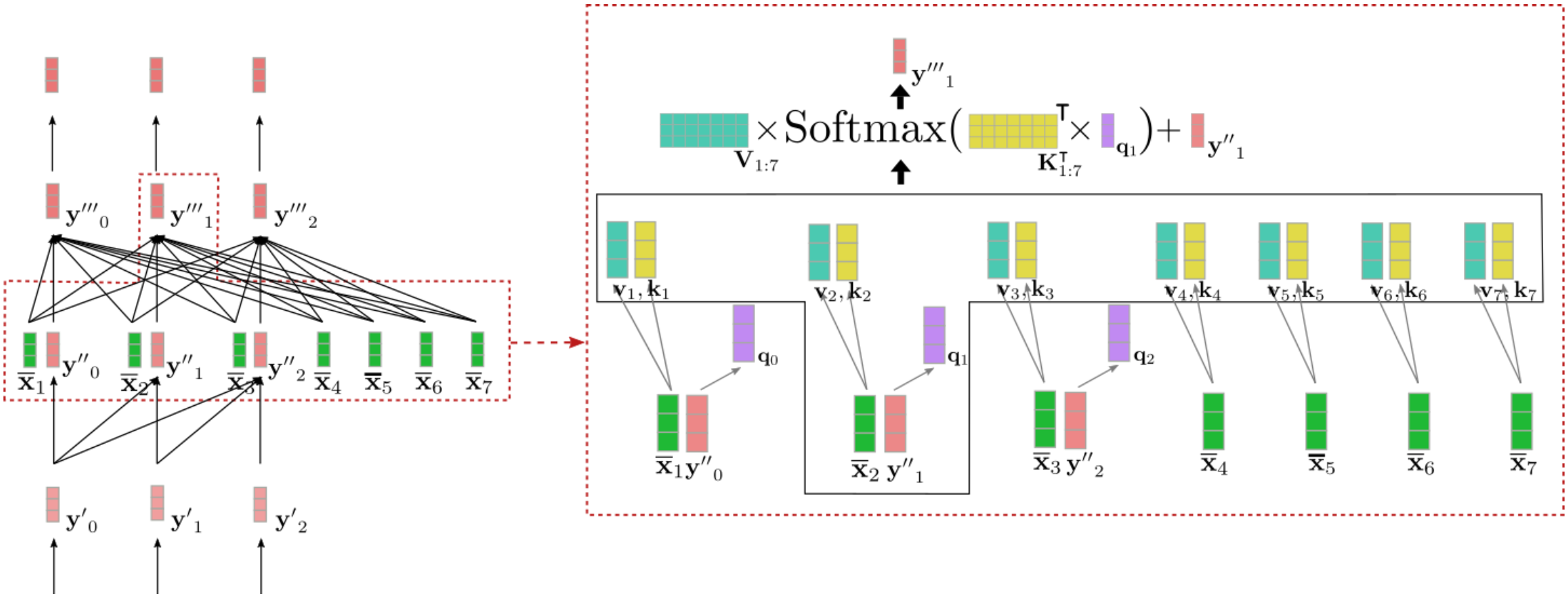
- Let's visualize the *transformer-based* decoder for our English to German translation example.



How to decode



How to decode



The model T5

- T5 (Text-To-Text Transfer Transformer) is a versatile language model that can be adapted to various NLP tasks by framing them as text-to-text problems. This flexibility is achieved through a unified architecture that can generate text given an input text prompt.
- T5 was developed by Google AI, introduced in 2019

T5

Dataset for Training

The T5 model was trained on the **C4 (Colossal Clean Crawled Corpus)** dataset:

- **C4 Dataset**: A large-scale dataset derived from web crawls, specifically designed to exclude low-quality and redundant data. It includes over 750 GB of text after preprocessing.
- The dataset is part of Google's effort to focus on high-quality and diverse textual data for better generalization.

T5: pretrained

Pretraining and Fine-tuning

1. **Pretraining**:
    - **Objective**: A span of tokens is masked, and the model is trained to predict the missing tokens.
    - Example: `The quick brown ___ jumps over the lazy ___` → `fox, dog`
    - This approach allows T5 to generalize across various text-based tasks.
2. **Fine-tuning**:
    - For specific tasks like sentiment analysis, summarization, or question answering, T5 can be fine-tuned by providing task-specific examples in the same text-to-text format.

T5 properties[note 1]

Name	Total parameters	Encoder parameters	Decoder parameters	n_{layer}	d_{model}	d_{ff}	d_{kv}	n_{head}
Small	76,956,160	35,330,816	41,625,344	6	512	2048	64	8
Base	247,577,856	109,628,544	137,949,312	12	768	3072	64	12
Large	770,567,168	334,939,648	435,627,520	24	1024	4096	64	16
3B	2,884,497,408	1,240,909,824	1,643,587,584	24	1024	16384	128	32
11B	11,340,220,416	4,864,791,552	6,475,428,864	24	1024	65536	128	128

*The encoder and the decoder have the same shape. So for example, the T5-small has 6 layers in the encoder and 6 layers in the decoder.

In the above table,

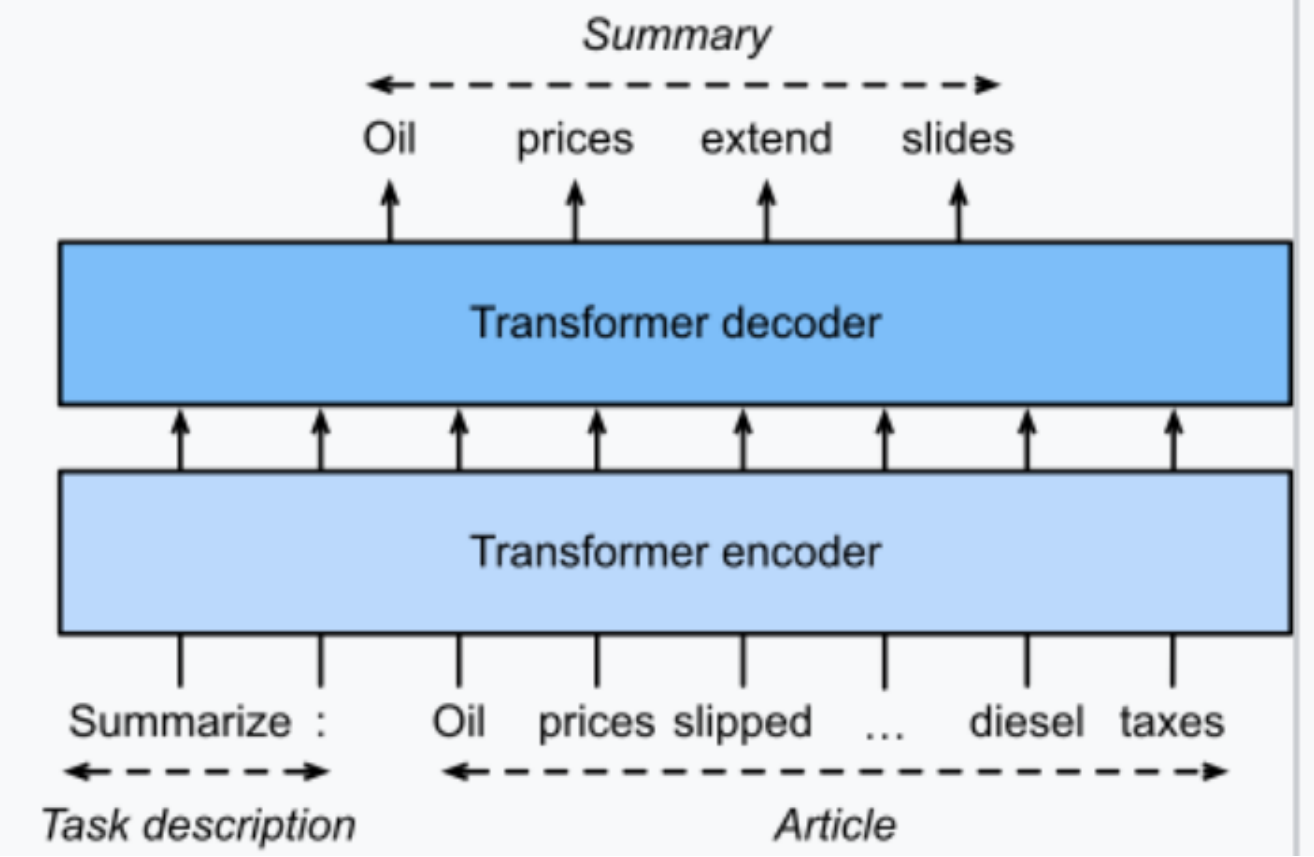
- n_{layer}: Number of layers in the encoder; also, number of layers in the decoder. They always have the same number of layers.
- n_{head}: Number of attention heads in each attention block.
- d_{model}: Dimension of the embedding vectors.
- d_{ff}: Dimension of the feedforward network within each encoder and decoder layer.
- d_{kv}: Dimension of the key and value vectors used in the self-attention mechanism.

T5 Applications

T5 is applicable in:

- Machine Translation
- Text Summarization
- Sentiment Analysis
- Question Answering
- Text Classification

Fine tune T5 for Text Summarization



ViT5 published in 2022

ViT5 (Vietnamese Text-to-Text Transfer Transformer) is a language model tailored for Vietnamese, inspired by Google's T5. It transforms all NLP tasks into a text-to-text format, enabling applications like machine translation, text summarization, and sentiment analysis.

- **Dataset**: Trained on diverse Vietnamese data (Wikipedia, online articles) with preprocessing for Vietnamese-specific characteristics.
- **Architecture**: Encoder-decoder transformer with positional encoding and multi-head attention. Available in multiple sizes.
- **Pretraining**: Uses "span corruption" for general context understanding.
- **Fine-tuning**: Adaptable to specific tasks like classification, summarization, and question answering.

Applications:

- Summarization, machine translation, Q&A, sentiment analysis, and semantic search.

ViT5 advances Vietnamese NLP by leveraging a robust and adaptable text-to-text framework.

ViT5

Pretraining Data

We use the CC100 Dataset (Monolingual Datasets from Web Crawl Data) [1] The total size for the Vietnamese Corpus is 138GB of raw text. We process and filter out 69GB of short paragraphs for 256-length model and 71GB of long paragraphs for 1024-length model.

ViT5

Model

ViT5 follows the encoder-decoder architecture and the T5 framework[3]. The original works of T5 proposed five different configs of model size: small, base, large, 3B, and 11B. For the purpose of practical study, we adapt the base (310M parameters) and large (866M parameters) models for ViT5 models and leave bigger models for future works.

We train ViT5 models with two different input and output lengths: 256-length and 1024-length. We thoroughly experimented with these two models to have an insight into the importance of pretraining data length for summarization tasks. For the self-supervised training learning objectives, we use the span-corruption objective with a corruption rate of 15%.

Results of using ViT5

Models	WikiLingua			Vietnews		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
Transformer (RND2RND)	46.25	16.57	29.82	57.56	24.25	35.53
PhoBERT2PhoBERT	50.4	19.88	32.49	60.37	29.12	39.44
mBERT2mBERT	52.82	20.57	31.55	59.67	27.36	36.73
mBART	55.21	25.69	37.33	59.81	28.28	38.71
mT5	55.27	27.63	38.30	58.05	26.76	37.38
BARTpho	57.16	31.18	40.89	61.14	30.31	40.15
$ViT5_{base\ 256\text{-}length}$	57.86	29.98	40.23	61.85	31.70	41.70
$ViT5_{base\ 1024\text{-}length}$	58.61	31.46	41.45	62.77	33.16	42.75
$ViT5_{large\ 1024\text{-}length}$	**60.22**	**33.12**	**43.08**	**63.37**	**34.24**	**43.55**

Test result on Wikilingua and Vietnews Abstractive Summarization.

Code and models

- https://huggingface.co/VietAI/vit5-large-vietnews-summarization
- https://github.com/vietai/ViT5

Fine tune T5 for text summarization

- https://www.kaggle.com/code/maverickss26/fine-tune-t5-transformer-for-text-summarization

Fine tune T5 for Legal question-answering

- https://www.kaggle.com/code/nvhuan/fine-tune-t5-for-vietnamese-legal-qa