Assignment 2 - Retrieval-based QA

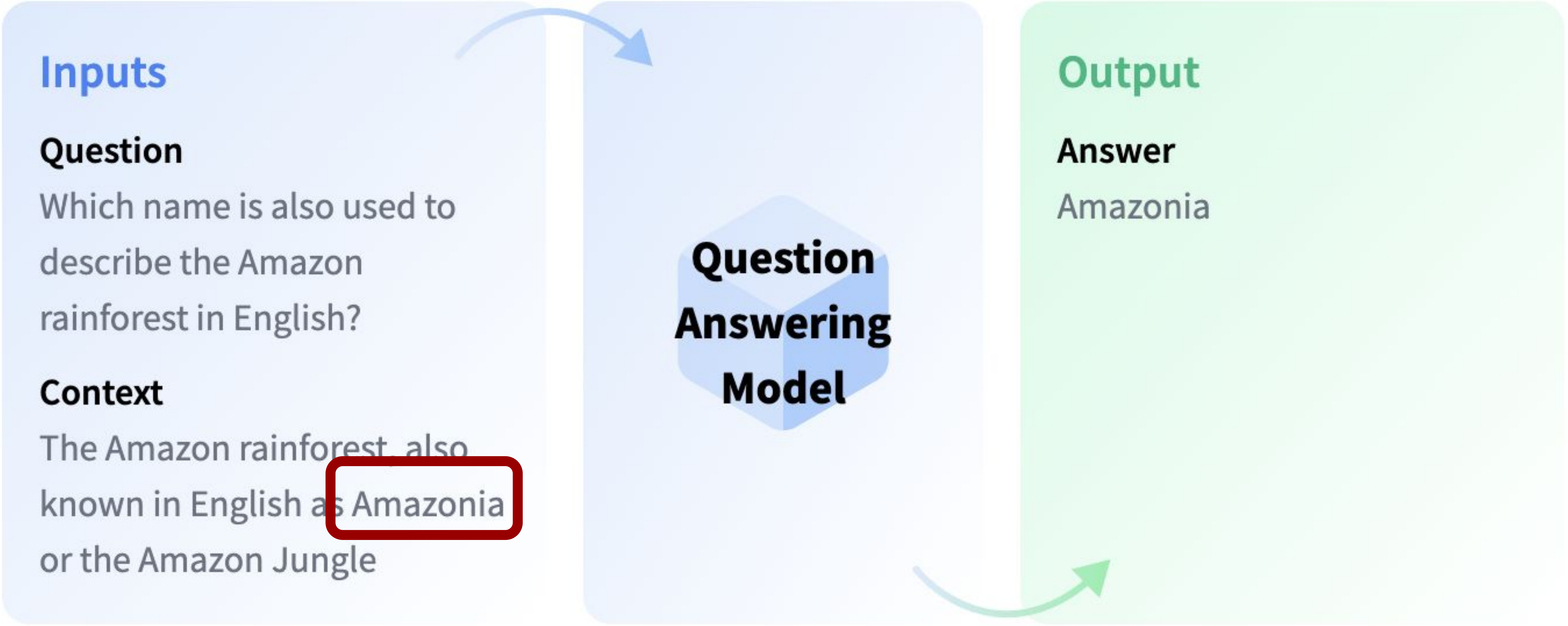
Outline

- ✓ Task Description
- ✓ Model Architecture
- ✓ What you need to do?

Task Description

1

Extractive QA



Inputs

Question

Which name is also used to describe the Amazon rainforest in English?

Context

The Amazon rainforest, also known in English as Amazonia or the Amazon Jungle

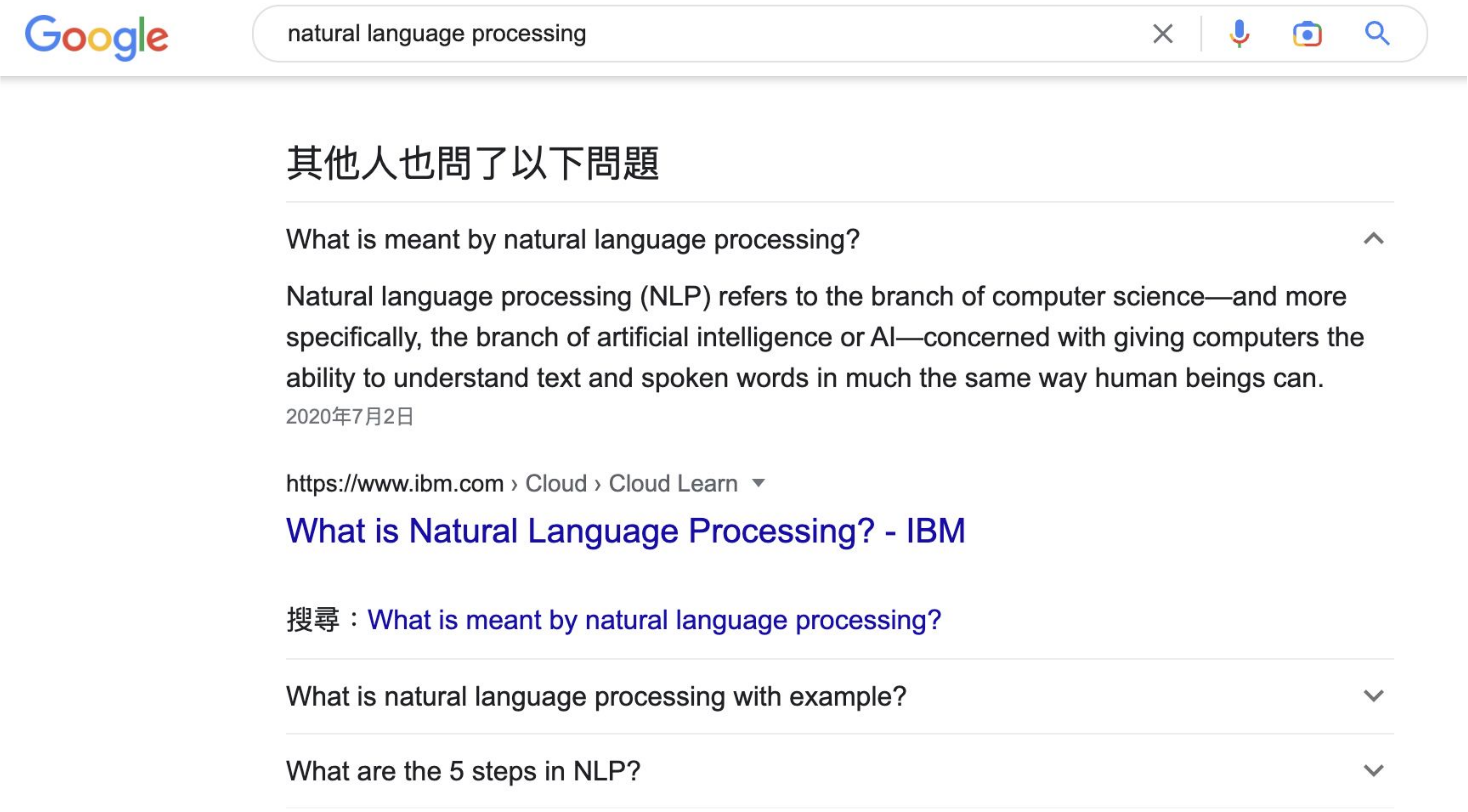
Output

Answer

Amazonia

Source: https://huggingface.co/tasks/question-answering

Retrieval-based QA



natural language processing

其他人也問了以下問題

What is meant by natural language processing?

Natural language processing (NLP) refers to the branch of computer science—and more specifically, the branch of artificial intelligence or AI—concerned with giving computers the ability to understand text and spoken words in much the same way human beings can.

2020年7月2日

https://www.ibm.com › Cloud › Cloud Learn

What is Natural Language Processing? - IBM

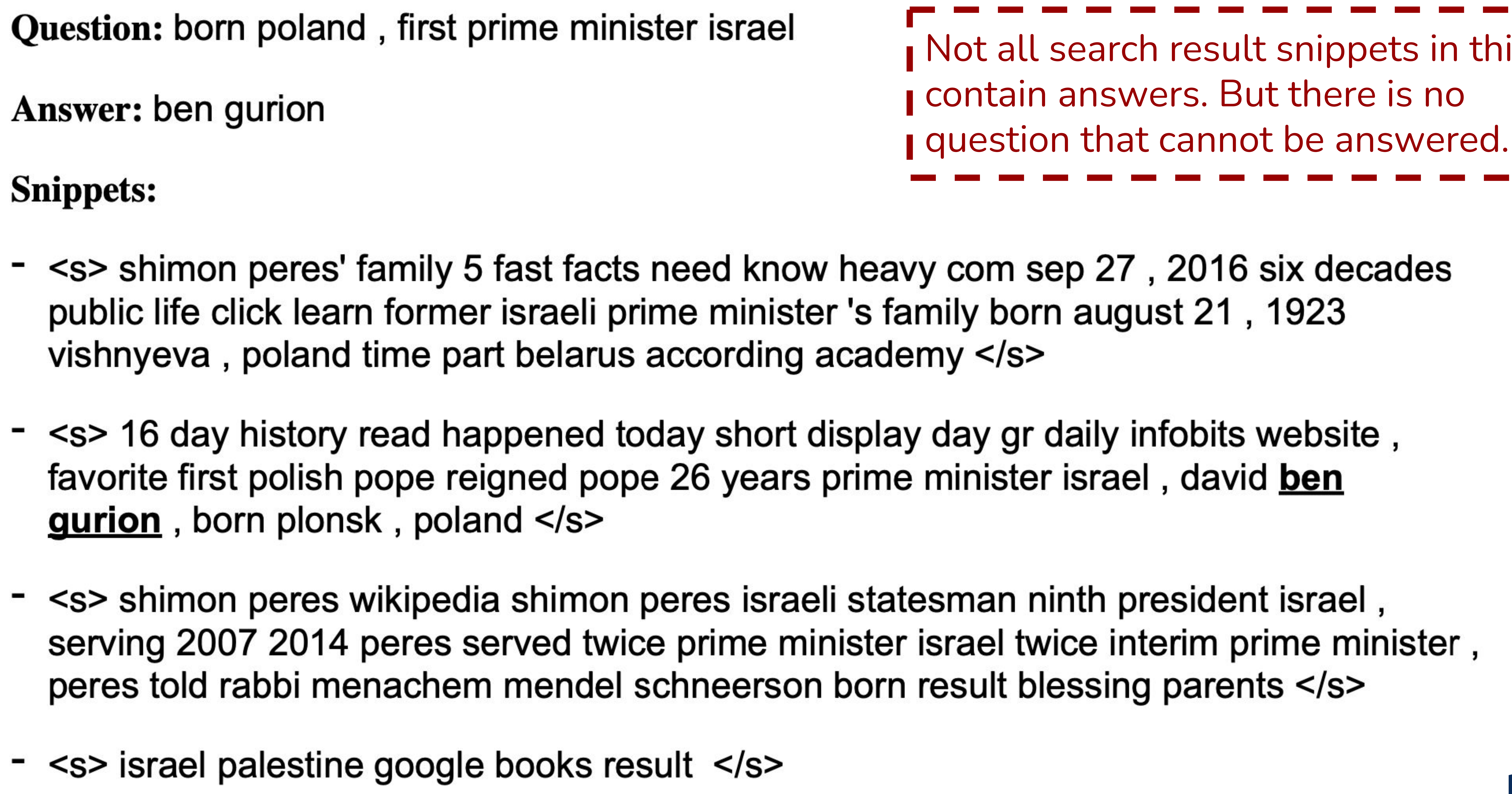
搜尋：What is meant by natural language processing?

What is natural language processing with example?

What are the 5 steps in NLP?

SearchQA

Not all search result snippets in this task contain answers. But there is no question that cannot be answered.



Question: born poland , first prime minister israel

Answer: ben gurion

Snippets:

- <s> shimon peres' family 5 fast facts need know heavy com sep 27 , 2016 six decades public life click learn former israeli prime minister 's family born august 21 , 1923 vishnyeva , poland time part belarus according academy </s>

- <s> 16 day history read happened today short display day gr daily infobits website , favorite first polish pope reigned pope 26 years prime minister israel , david **ben gurion** , born plonsk , poland </s>

- <s> shimon peres wikipedia shimon peres israeli statesman ninth president israel , serving 2007 2014 peres served twice prime minister israel twice interim prime minister , peres told rabbi menachem mendel schneerson born result blessing parents </s>

- <s> israel palestine google books result </s>

Dataset Overview

- train & val

  <s> snippet 1 </s> <s> snippet 2 </s> … … ||| Question ||| Answer

- test

  <s> snippet 1 </s> <s> snippet 2 </s> … … ||| Question ||| “answer”

- test-submit.txt

  Question ||| “answer”

same order

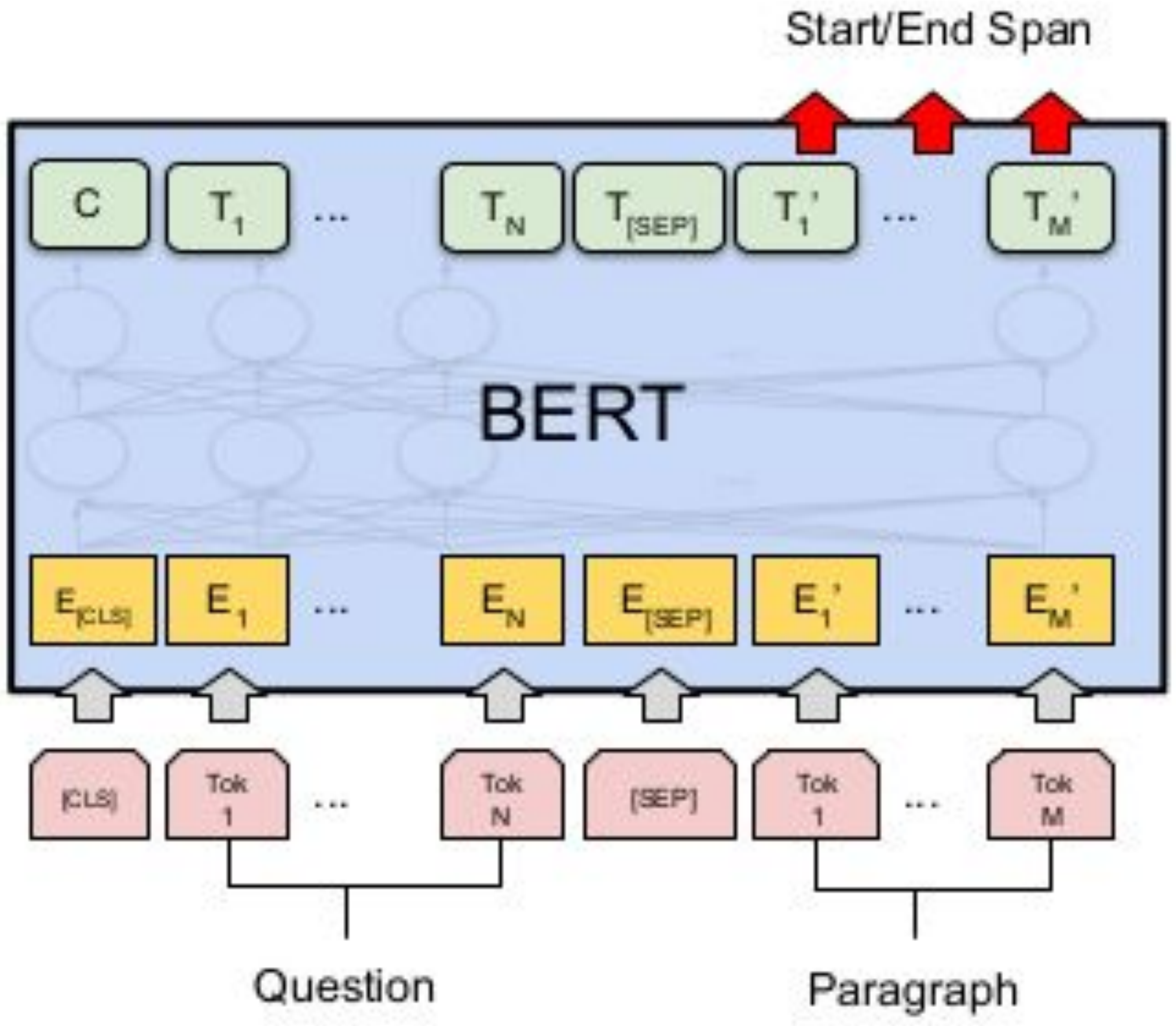
change to the answer your model predicts

Model Architecture

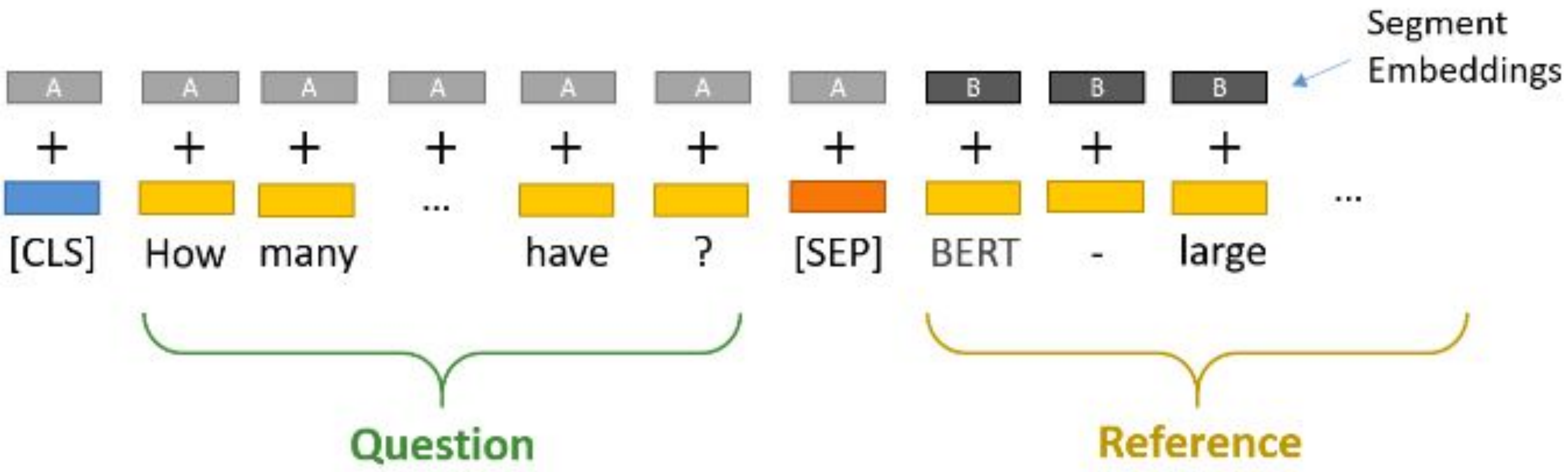
2

Most of the materials in this chapter are from
https://medium.com/saarthi-ai/build-a-smart-question-answering-system-with-fine-tuned-bert-b586e4cfa5f5

Fine-tune PLM in QA task



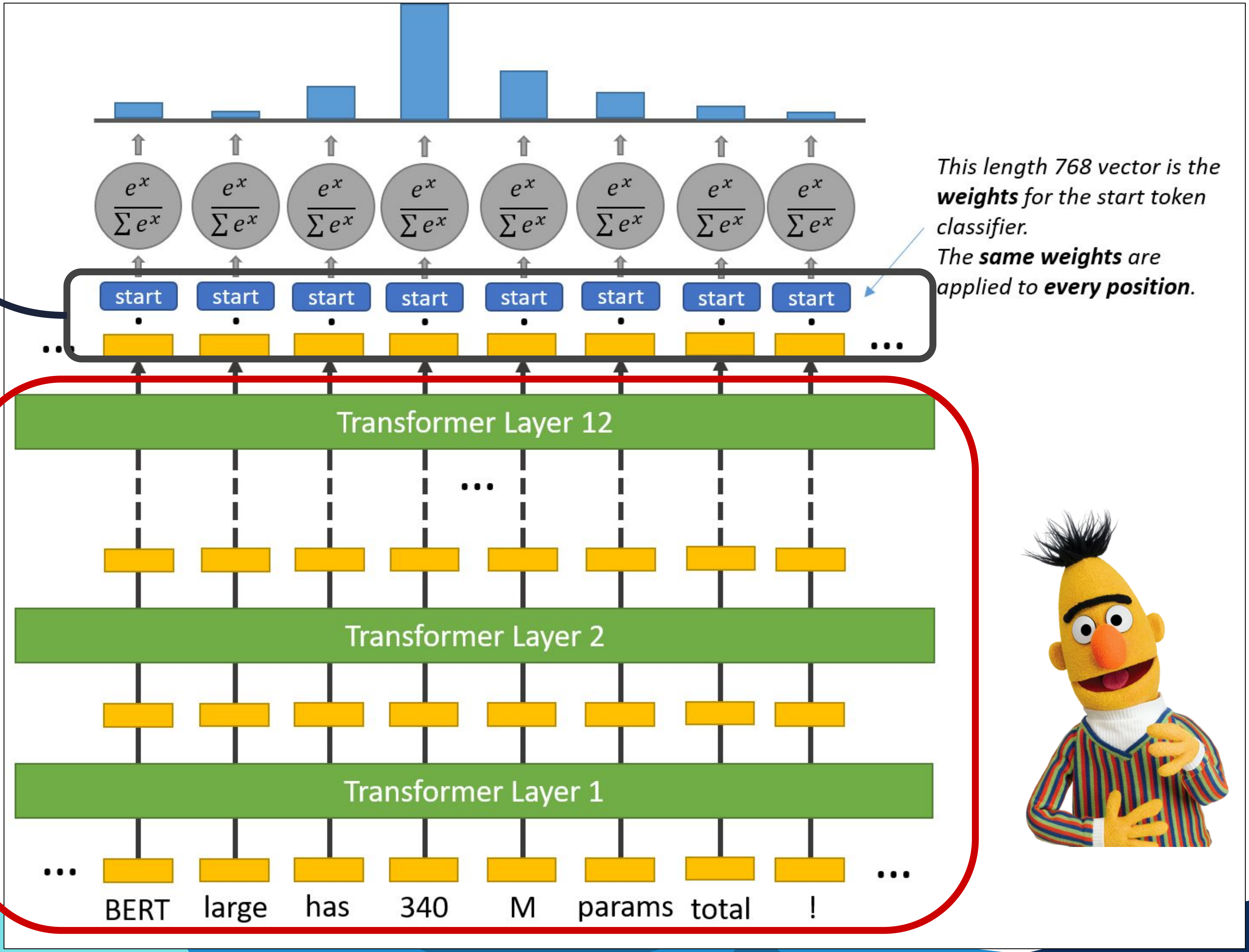
Fine-tune PLM in QA task



Fine-tune PLM in QA task

Can be a simple model, like a linear layer.

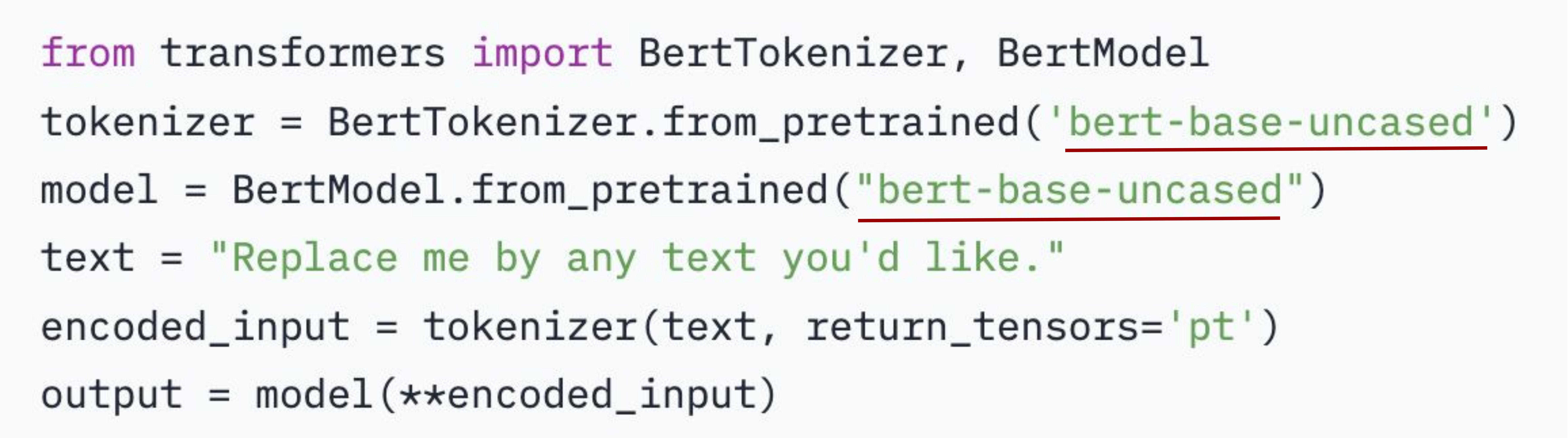
Pretrained Language Model such as BERT or RoBERTa



This length 768 vector is the ***weights*** *for the start token classifier.*
The ***same weights*** *are applied to* ***every position****.*

How to implement a BERT?

- Hugging Face 🤗



```
from transformers import BertTokenizer, BertModel
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
model = BertModel.from_pretrained("bert-base-uncased")
text = "Replace me by any text you'd like."
encoded_input = tokenizer(text, return_tensors='pt')
output = model(**encoded_input)
```

Source: https://huggingface.co/bert-base-uncased?text=The+goal+of+life+is+%5BMASK%5D.

What you need to do?

3

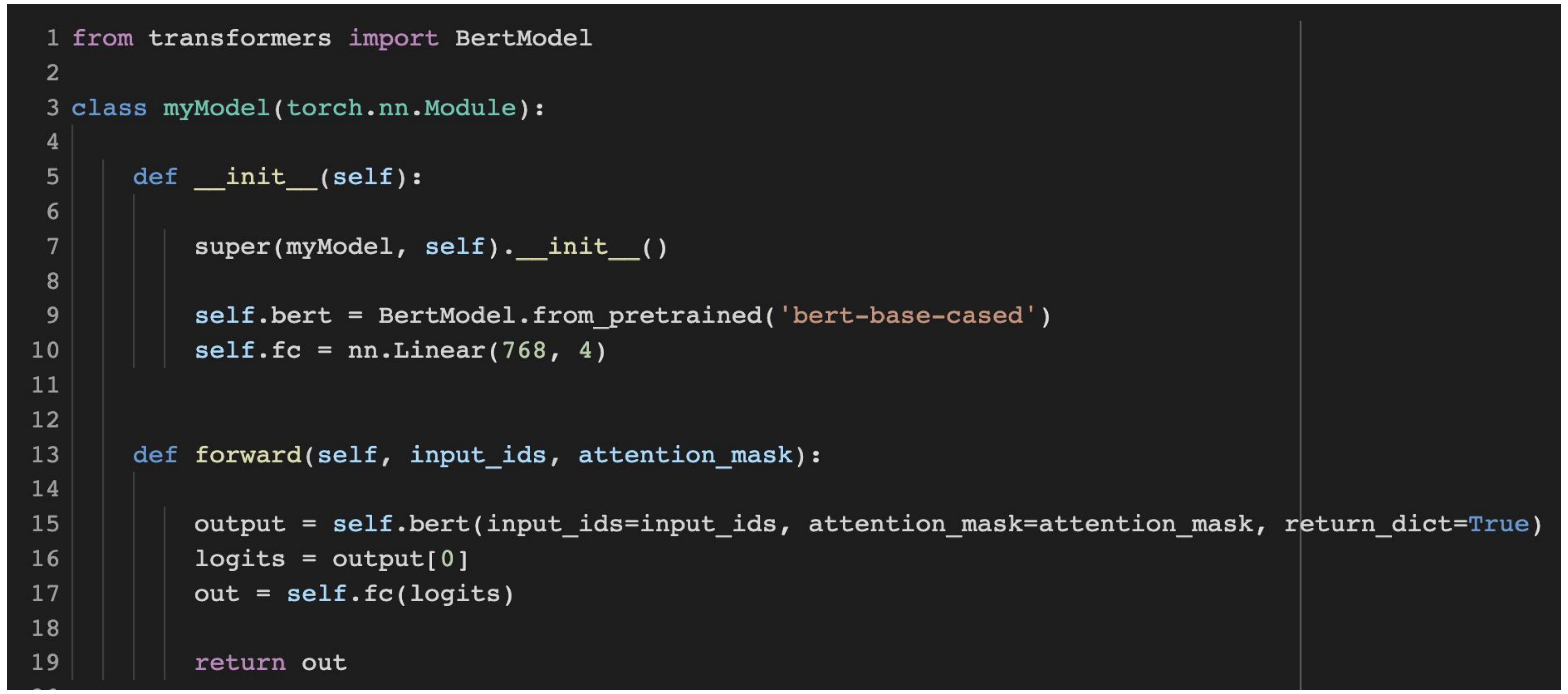
1. Document retrieval (e.g. BM25 or TF-IDF)
2. Build your dataset and doing tokenization.

   e.g. { Question, Reference, Ans_start_index, Ans_end_index...}

   - Tip: If there are entries with no answer, you can set both the start and end indice of the answer to the [CLS] token (usually index=0).
3. Implement the training process and then make inferences on the test data.

Note that you need to implement the output layer yourself, so you can't use any library that already assembles all the pieces for you.

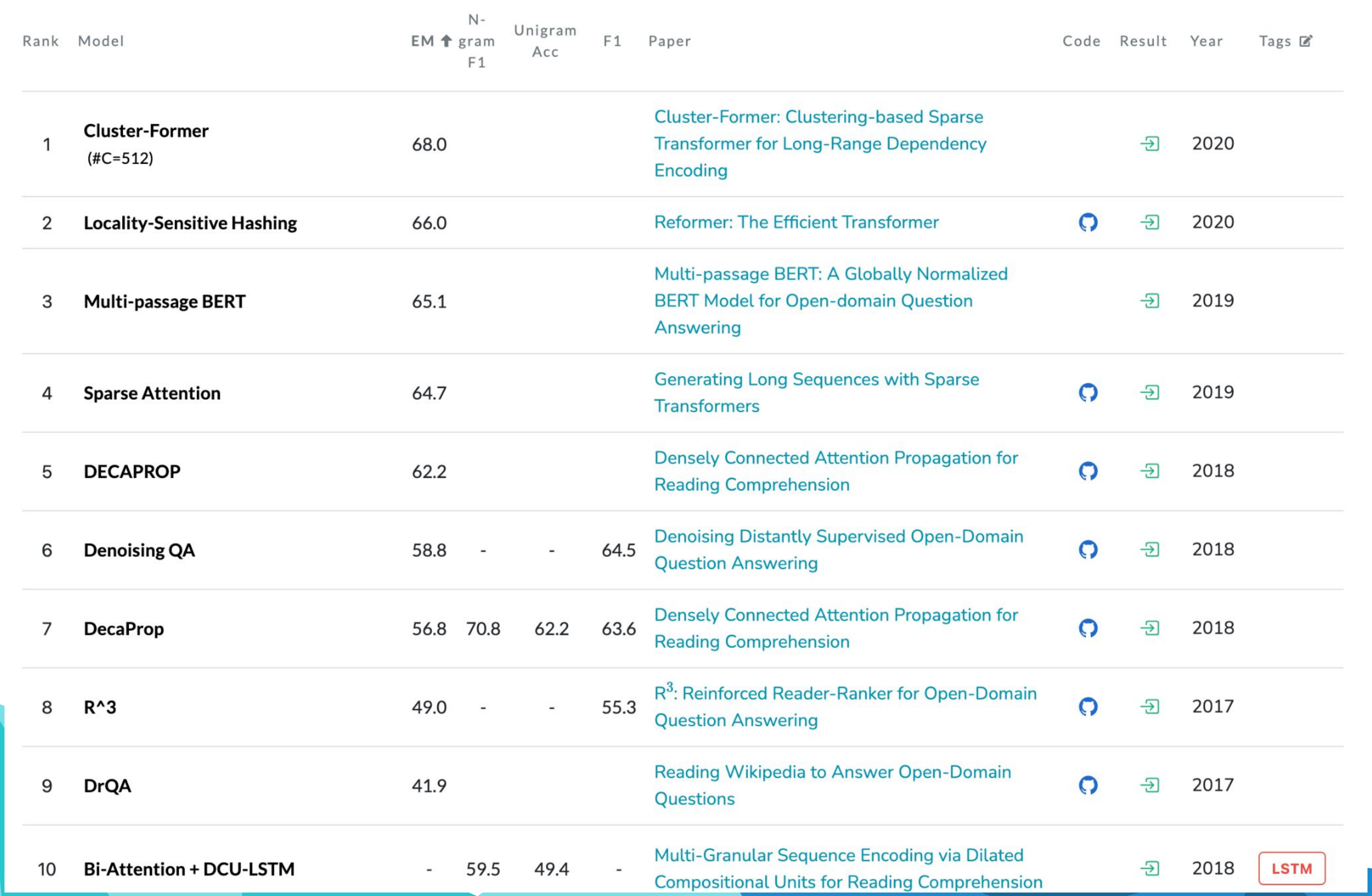
For example ~~AutoModelForQuestionAnswering.from_pretrained(model_name)~~



```python
from transformers import BertModel

class myModel(torch.nn.Module):

    def __init__(self):

        super(myModel, self).__init__()

        self.bert = BertModel.from_pretrained('bert-base-cased')
        self.fc = nn.Linear(768, 4)


    def forward(self, input_ids, attention_mask):

        output = self.bert(input_ids=input_ids, attention_mask=attention_mask, return_dict=True)
        logits = output[0]
        out = self.fc(logits)

        return out
```

In fact, you can refer to the sample code of aicup as we treat the aicup task as a QA-like task.

Leaderboard for this task



Rank	Model	EM ↑	N-gram F1	Unigram Acc	F1	Paper	Code	Result	Year	Tags
1	**Cluster-Former** (#C=512)	68.0				Cluster-Former: Clustering-based Sparse Transformer for Long-Range Dependency Encoding		✓	2020	
2	**Locality-Sensitive Hashing**	66.0				Reformer: The Efficient Transformer	✓	✓	2020	
3	**Multi-passage BERT**	65.1				Multi-passage BERT: A Globally Normalized BERT Model for Open-domain Question Answering		✓	2019	
4	**Sparse Attention**	64.7				Generating Long Sequences with Sparse Transformers	✓	✓	2019	
5	**DECAPROP**	62.2				Densely Connected Attention Propagation for Reading Comprehension	✓	✓	2018	
6	**Denoising QA**	58.8	-	-	64.5	Denoising Distantly Supervised Open-Domain Question Answering	✓	✓	2018	
7	**DecaProp**	56.8	70.8	62.2	63.6	Densely Connected Attention Propagation for Reading Comprehension	✓	✓	2018	
8	**R^3**	49.0	-	-	55.3	R^3: Reinforced Reader-Ranker for Open-Domain Question Answering	✓	✓	2017	
9	**DrQA**	41.9				Reading Wikipedia to Answer Open-Domain Questions	✓	✓	2017	
10	**Bi-Attention + DCU-LSTM**	-	59.5	49.4	-	Multi-Granular Sequence Encoding via Dilated Compositional Units for Reading Comprehension		✓	2018	LSTM

https://paperswithcode.com/sota/open-domain-question-answering-on-searchqa

Thanks!

Any questions?