AI VIETNAM
All-in-One Course
(TA Session)

End-to-end Question Answering

Project – P2

Dinh-Thang Duong - TA

Year 2024

Outline

- Introduction
- Module: Reader
- Module: Retriever
- Question

Introduction

Introduction

❖ Getting Started

Most famous search engines

Introduction

❖ Getting Started

Google

which is the highest mountain in solar system

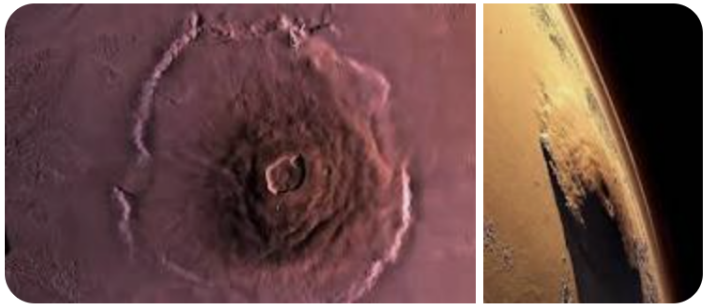
Images | Compared to | Videos | News | Shopping | Books | Flights | Finance | All filters | Tools

Olympus Mons

The highest mountain and volcano in the Solar System is on the planet Mars. It is called Olympus Mons and is 16 miles (24 kilometers) high which makes it about three times higher than Mt. Everest.

Cool Cosmos
https://coolcosmos.ipac.caltech.edu › ask › 199-Where-is...

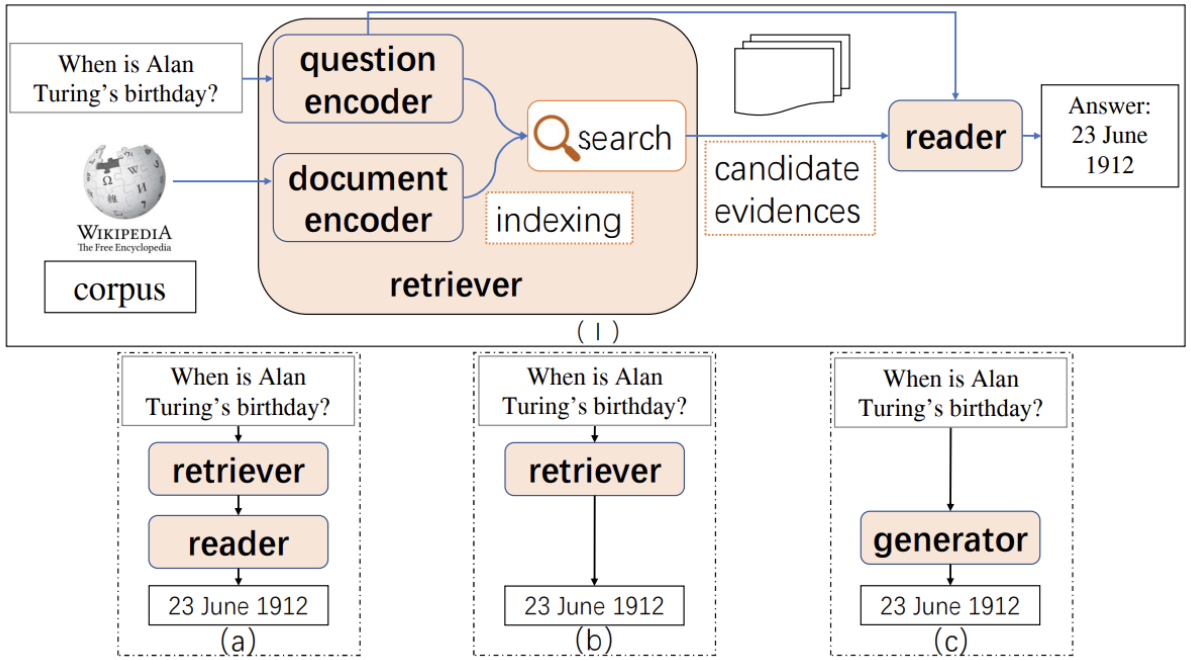
Where is the highest mountain in our Solar System?



Answer from Google for the query question: Open-domain Question Answering

Introduction

❖ What is Open-domain Question Answering?

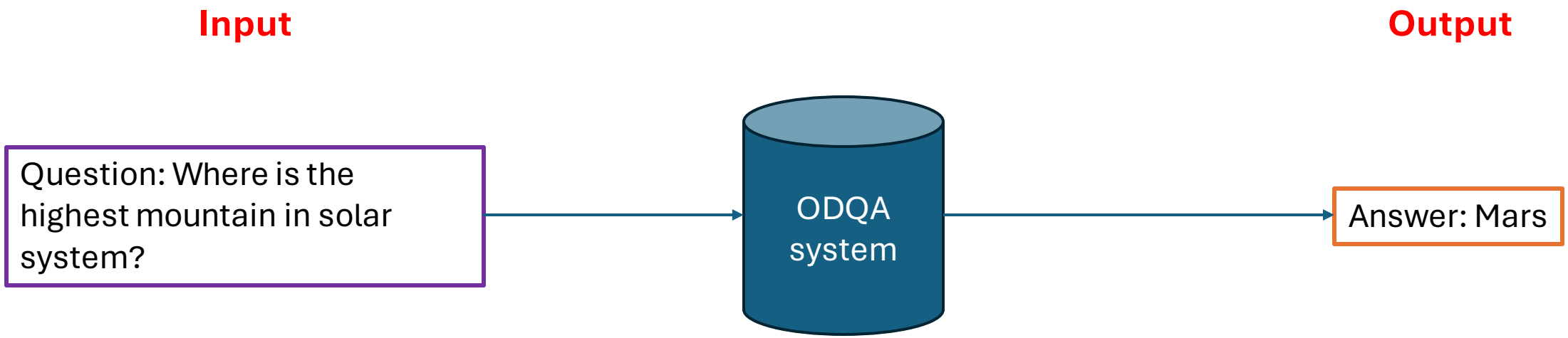


Open-domain Question Answering (ODQA): Refers to the task of providing answers to questions over a broad range of topics, leveraging vast corpora of unstructured text without being restricted to a specific domain.

Image source: https://arxiv.org/pdf/2211.07886.pdf

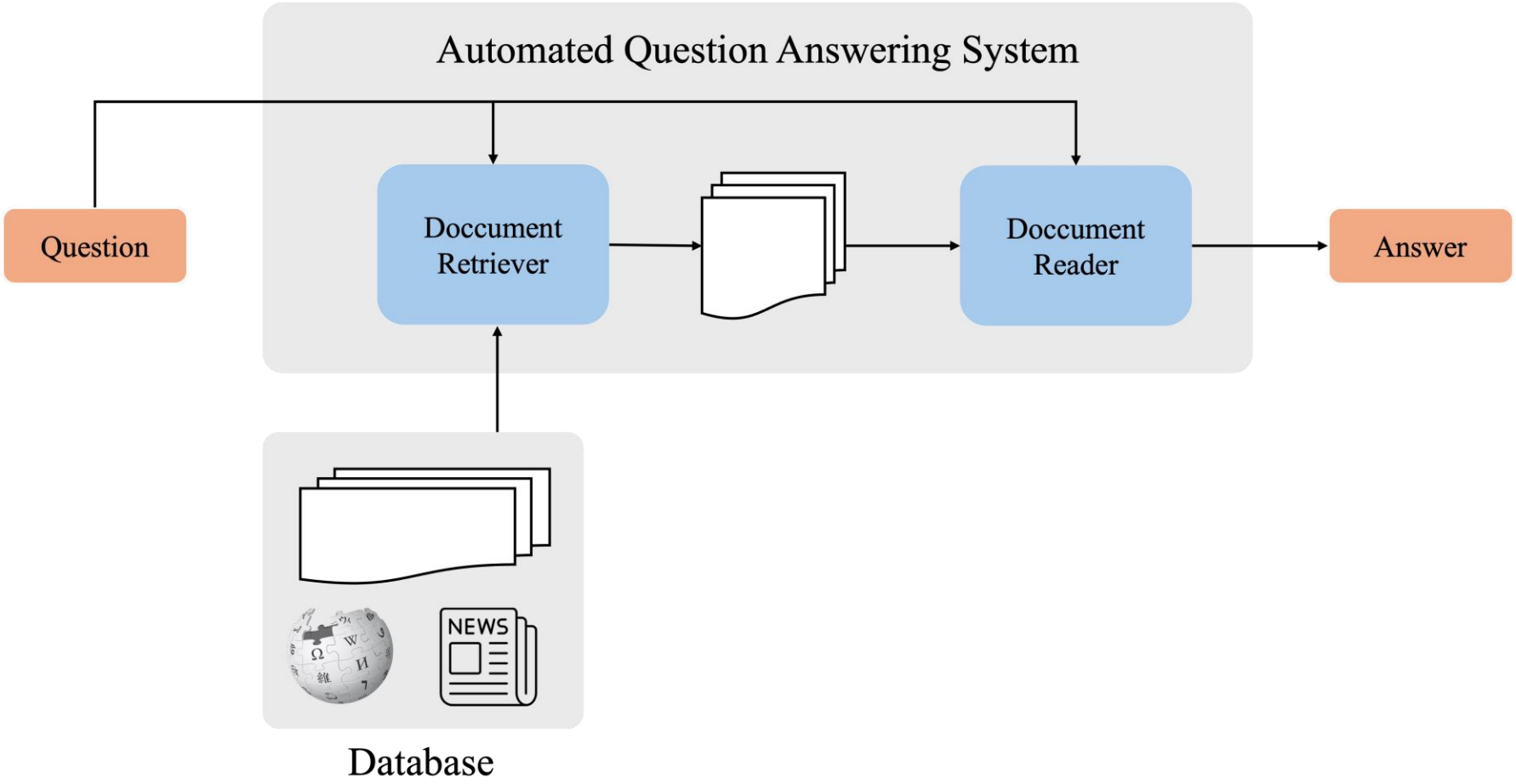
Introduction

❖ Open-domain Question Answering I/O



Introduction

❖ End-to-end Question Answering System I/O



Introduction

❖ Open-domain Question Answering I/O

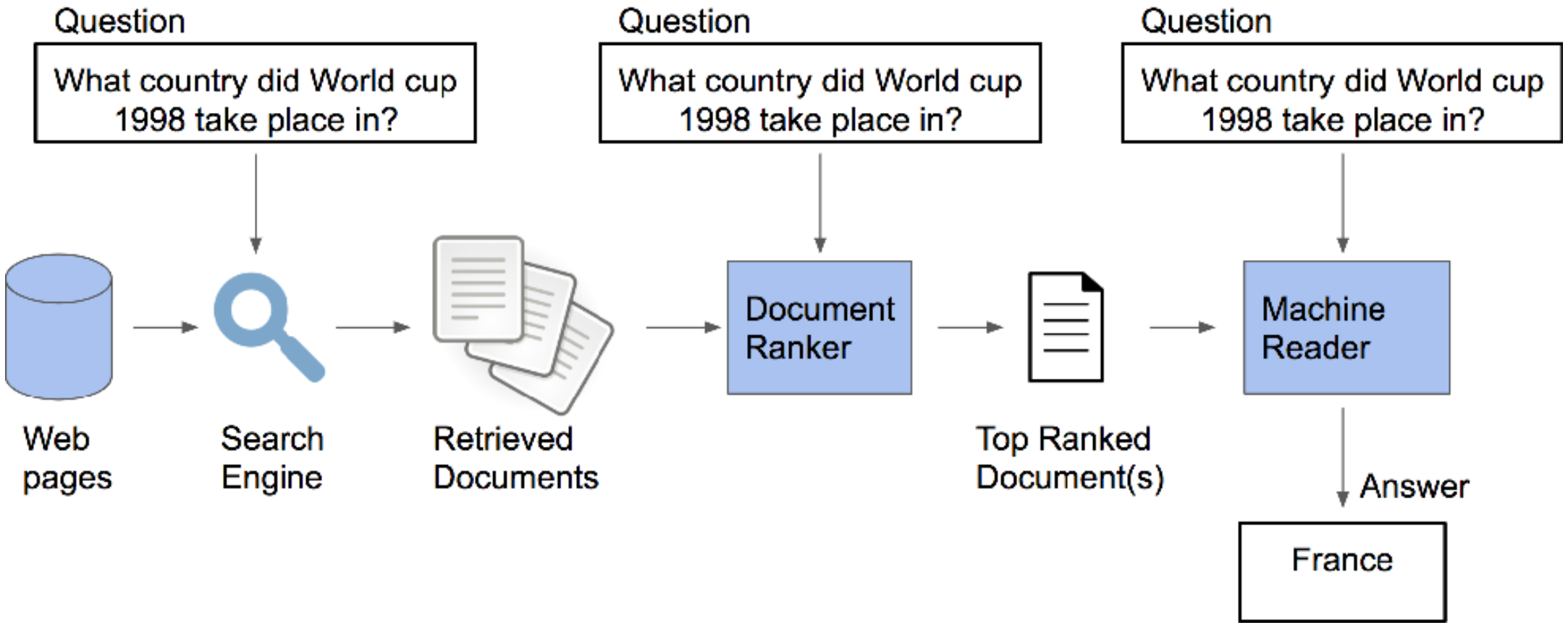
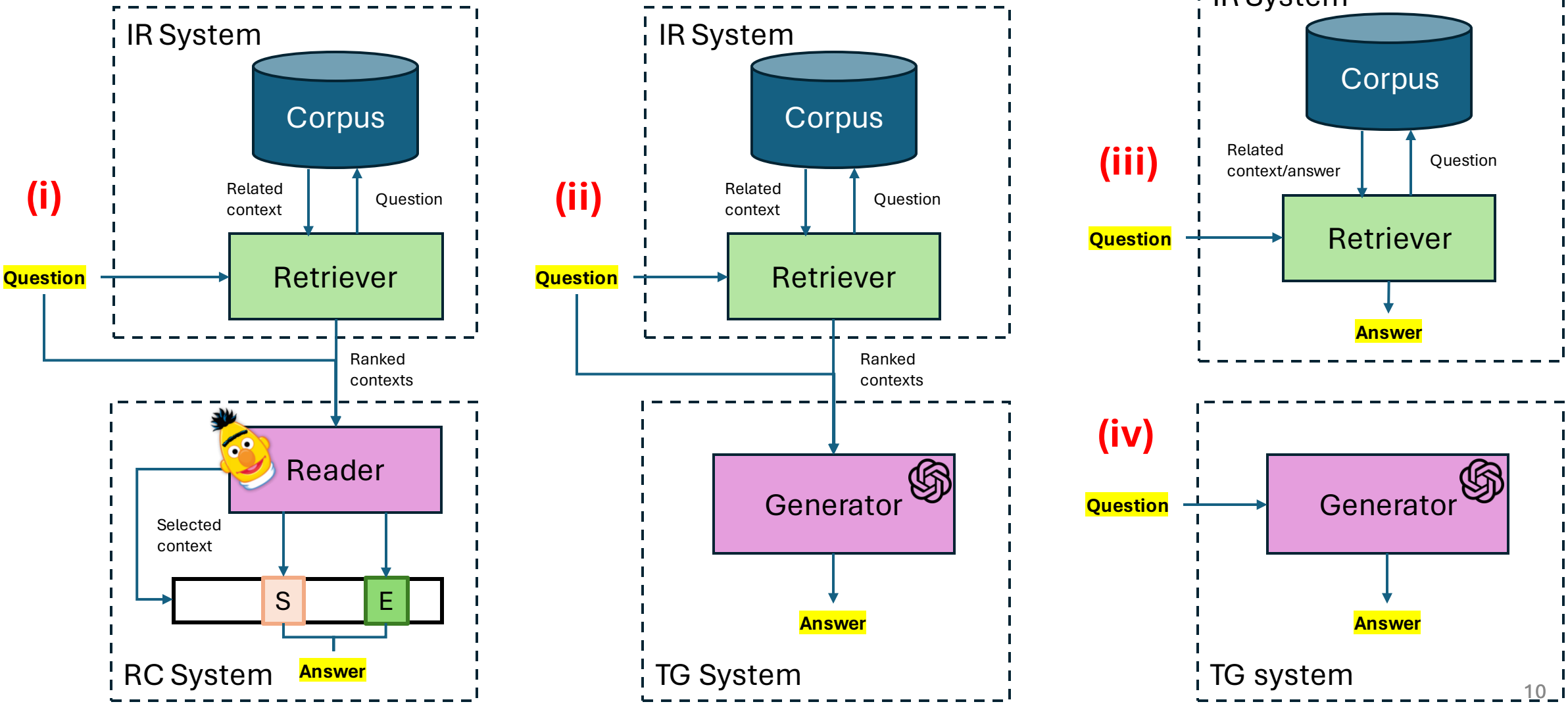


Image source: https://www.semanticscholar.org/paper/Training-a-Ranking-Function-for-Open-Domain-Htut-Bowman/2af0040b92c2ec3d07d09ed9a8eda6047fe3fed2/figure/0

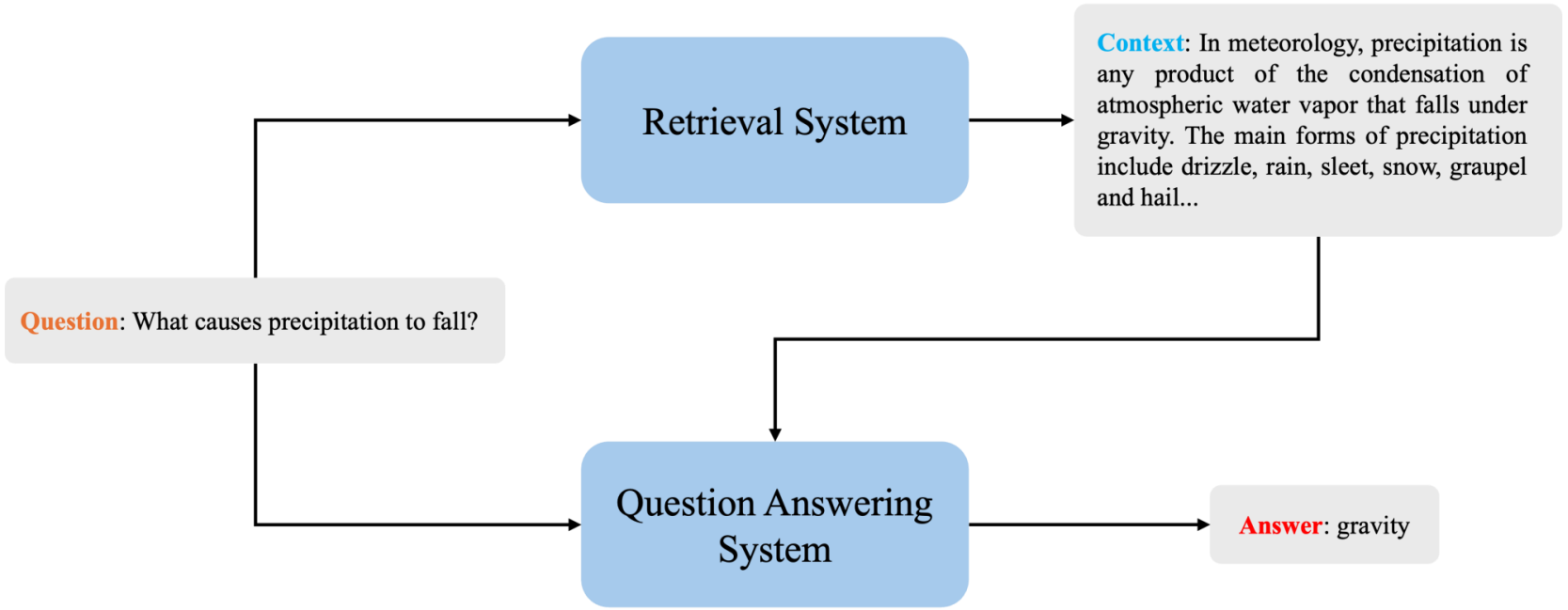
Introduction

❖ Open-domain Question Answering Categories



Introduction

❖ End-to-end Question Answering System



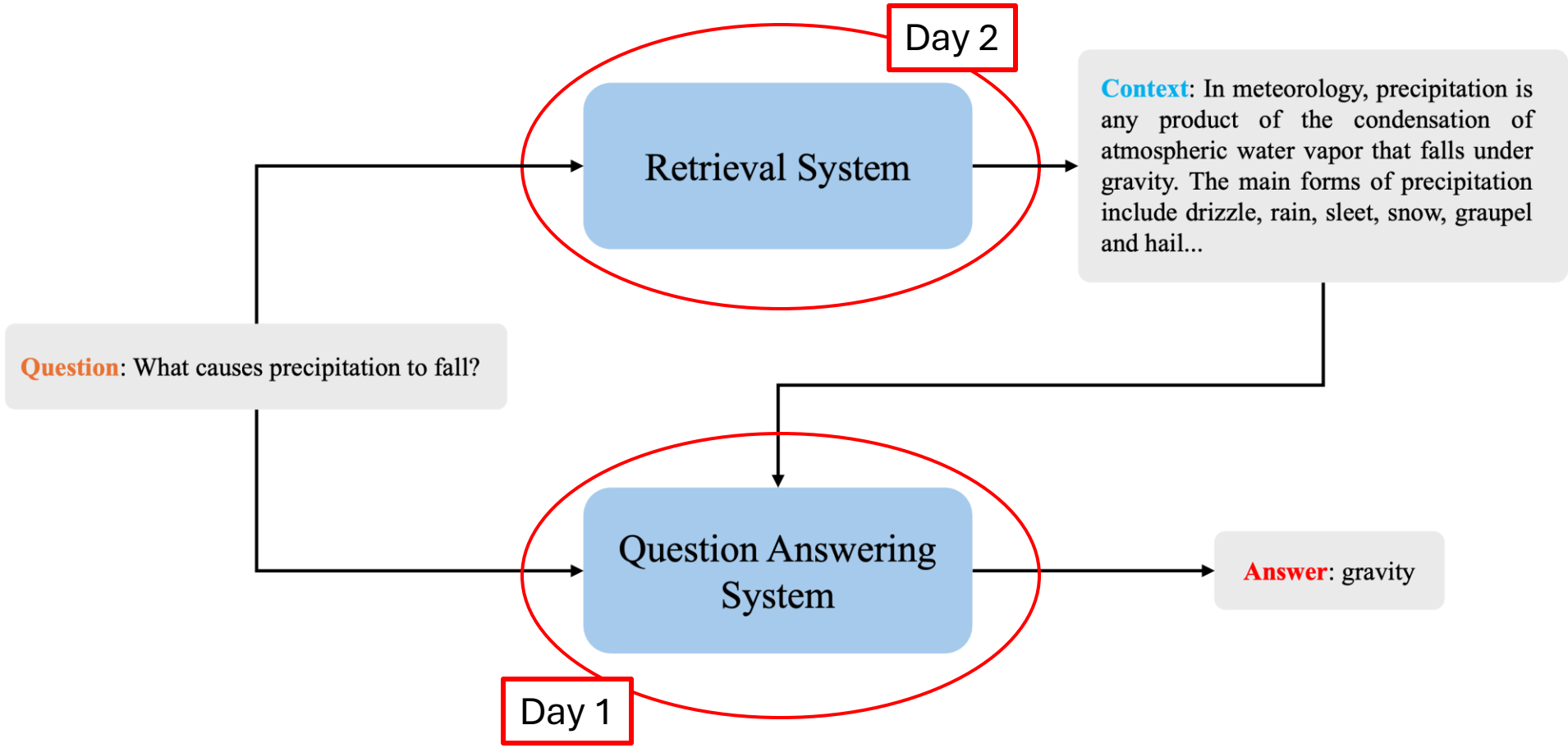
End-to-end QA system consists of:

1. Retriever
2. Reader

Introduction

❖ Project Description

Description: In this project, we will build an end-to-end QA system that can retrieve relevant contexts, extract best answer for a given question. The domain of the question can be varied in different topics.

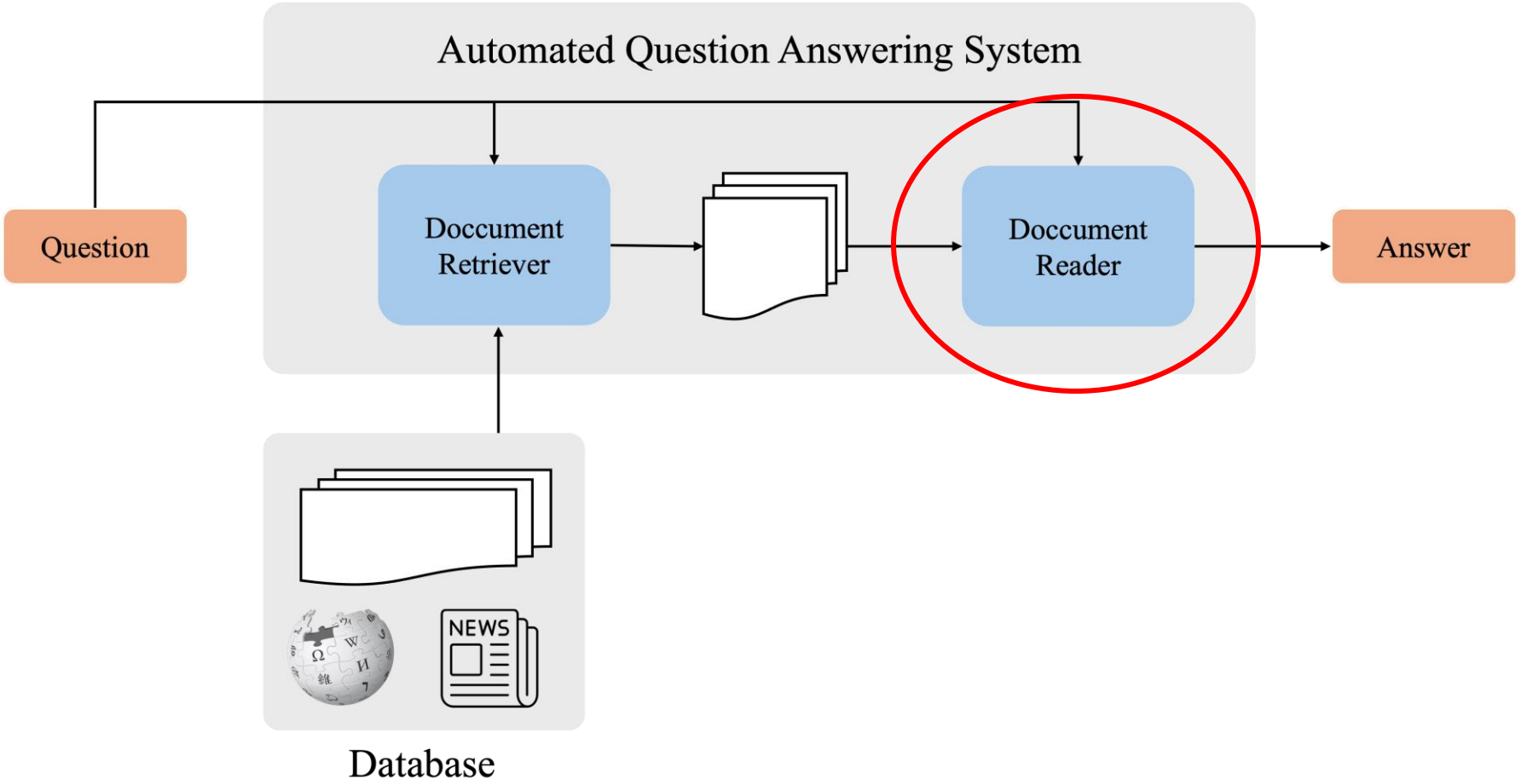


AI VIETNAM
All-in-One Course
(TA Session)

Module: Reader

Module: Reader

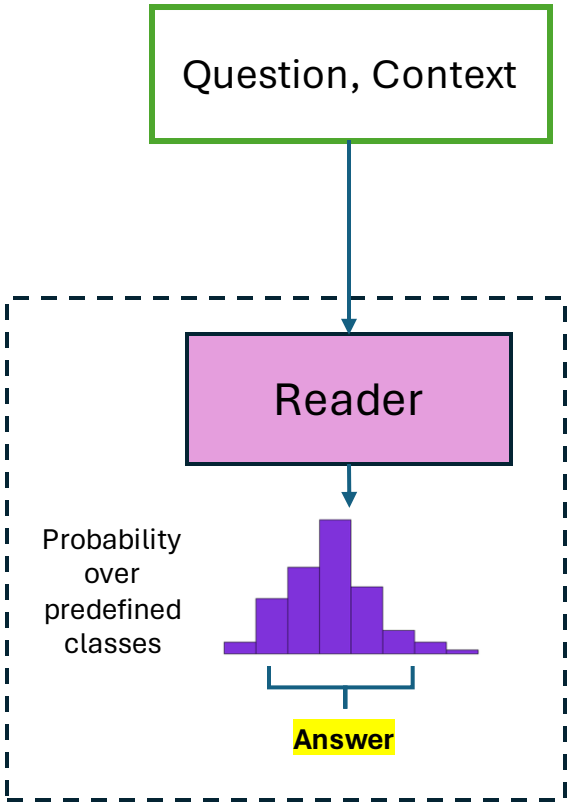
❖ E2E QA Pipeline



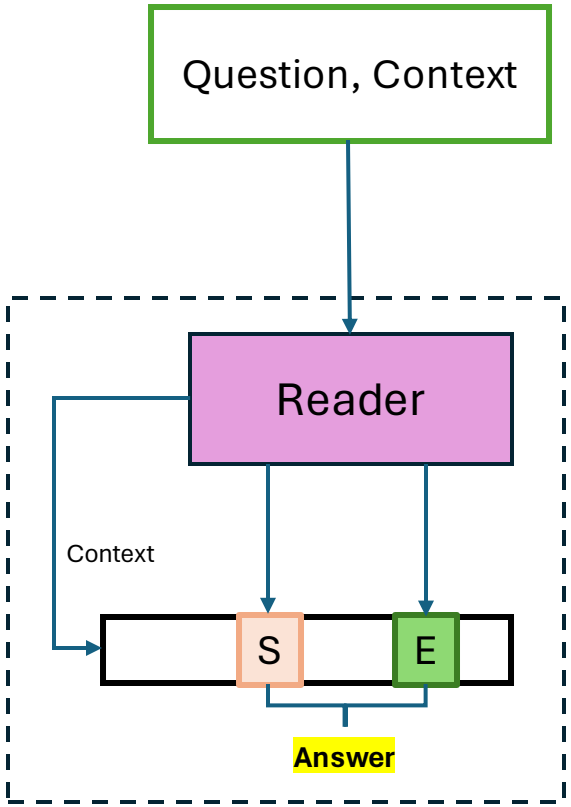
Module: Reader

❖ Type of Reader

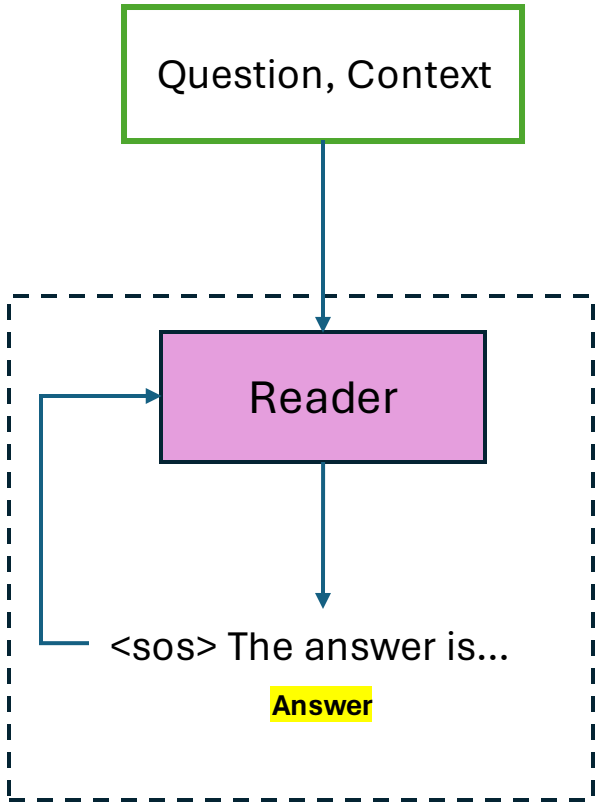
I. Classification Approach



II. Extractive Approach

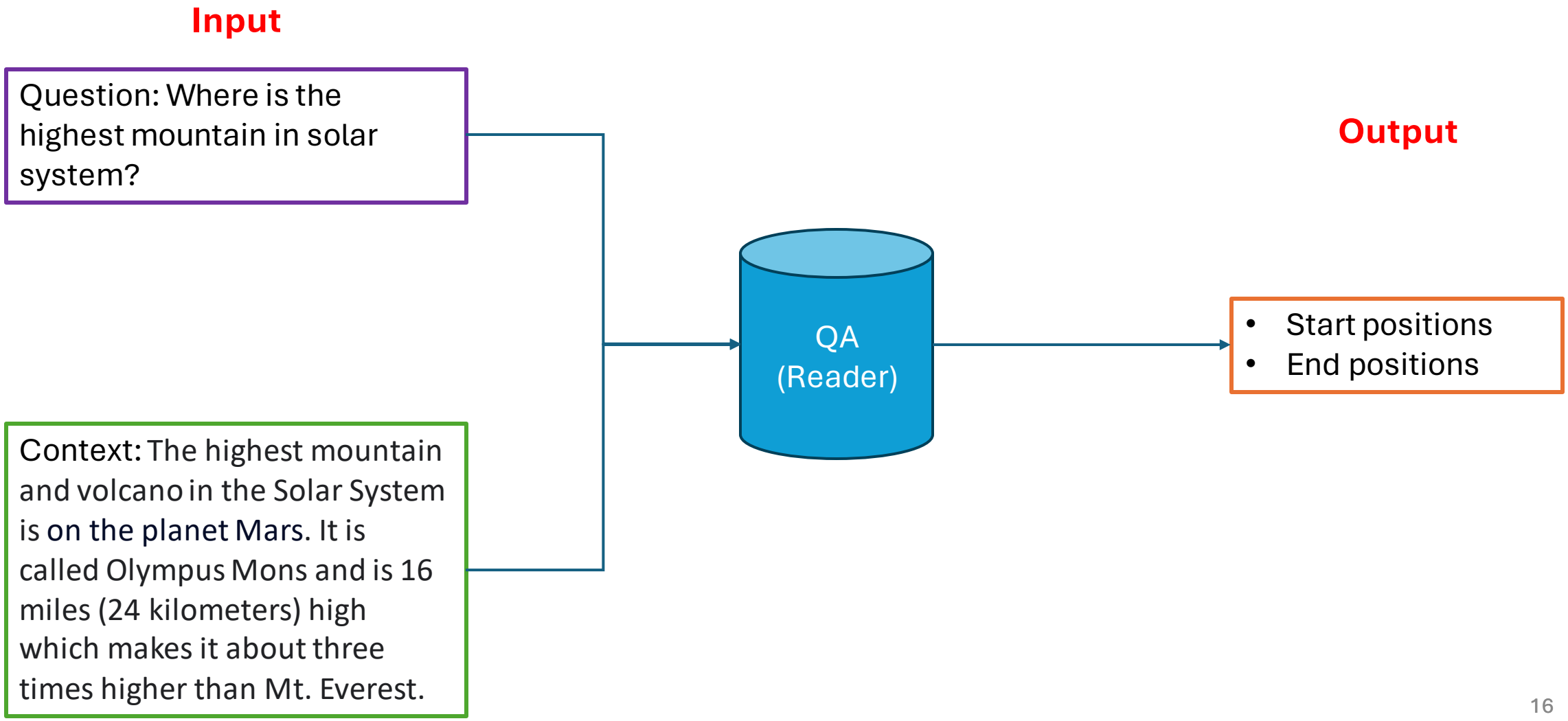


III. Generative Approach



Module: Reader

❖ Reader: Extractive Extraction



Module: Reader

❖ Model Design

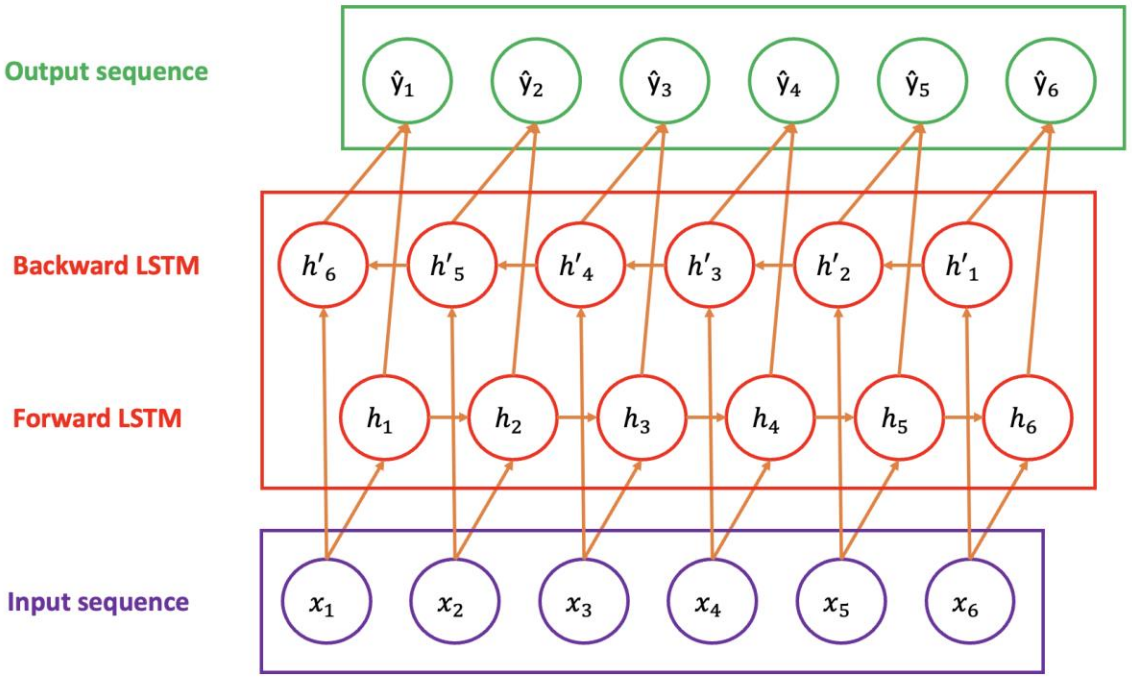
```
1 INPUT_QUESTION = 'What is my name?'
2 INPUT_CONTEXT = 'My name is AI Vietnam and I live in Vietnam.'
3 pipe(question=INPUT_QUESTION, context=INPUT_CONTEXT)
```

```
{'score': 0.97179114818573, 'start': 11, 'end': 21, 'answer': 'AI Vietnam'}
```

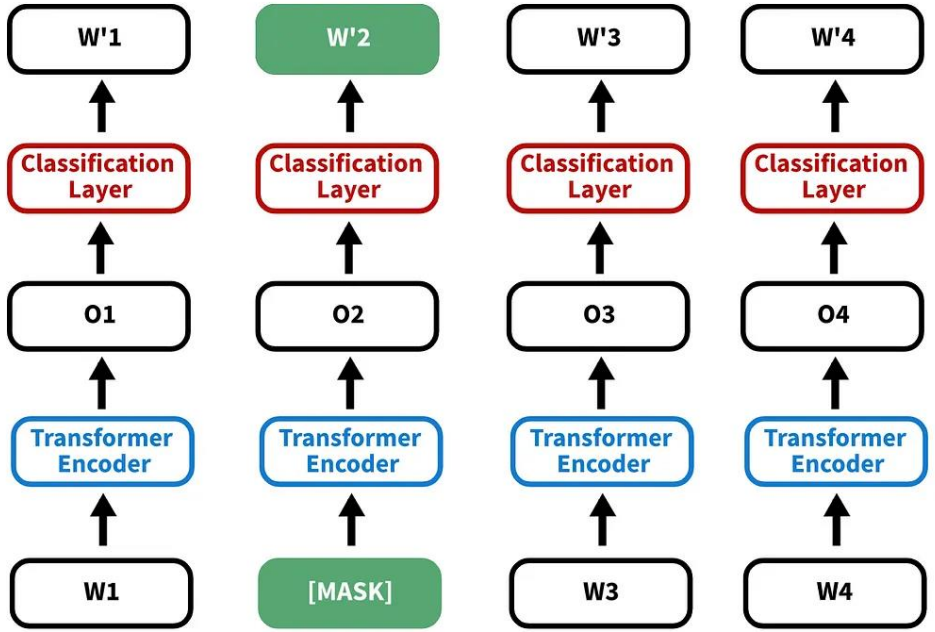
The answer span starts from index 11 to 21 in context text

Module: Reader

❖ Model Design: LSTM and Transformer Encoder



LSTM-based

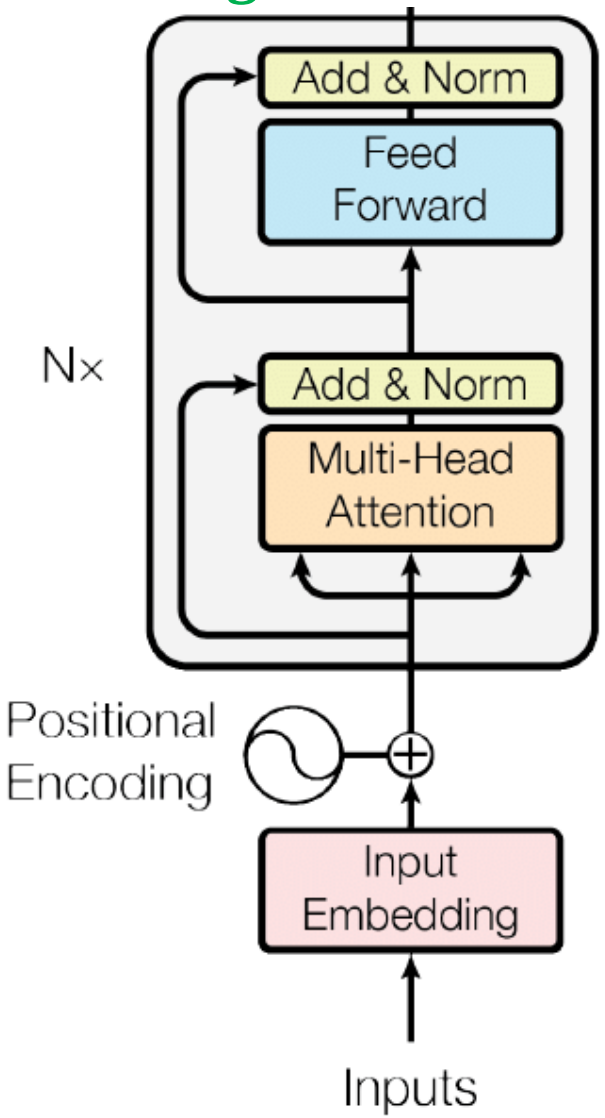


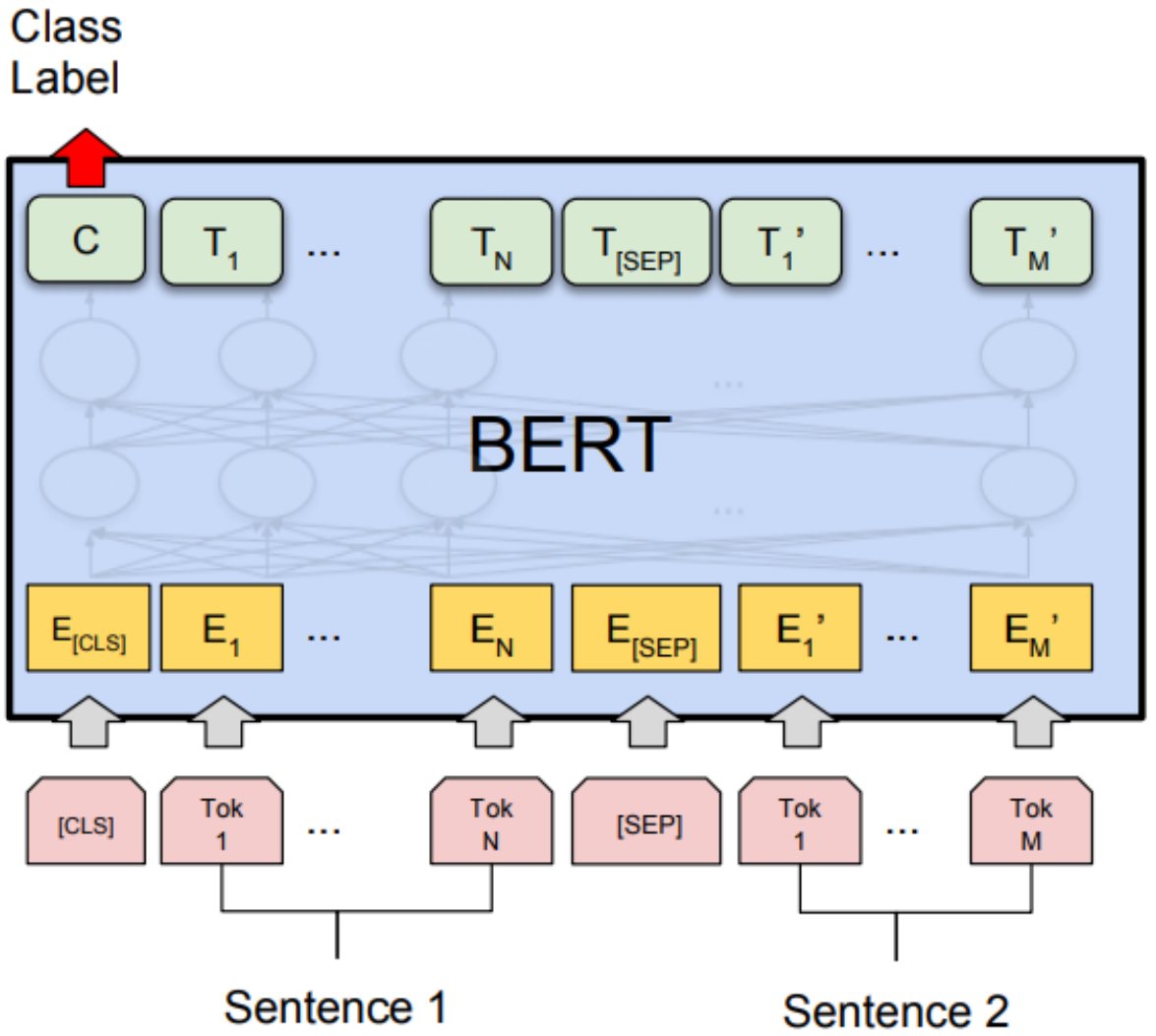
Transformer Encoder-based

https://towardsdatascience.com/how-to-answer-questions-with-machine-learning-6c21357a44fc

Module: Reader

❖ Module Design: BERT

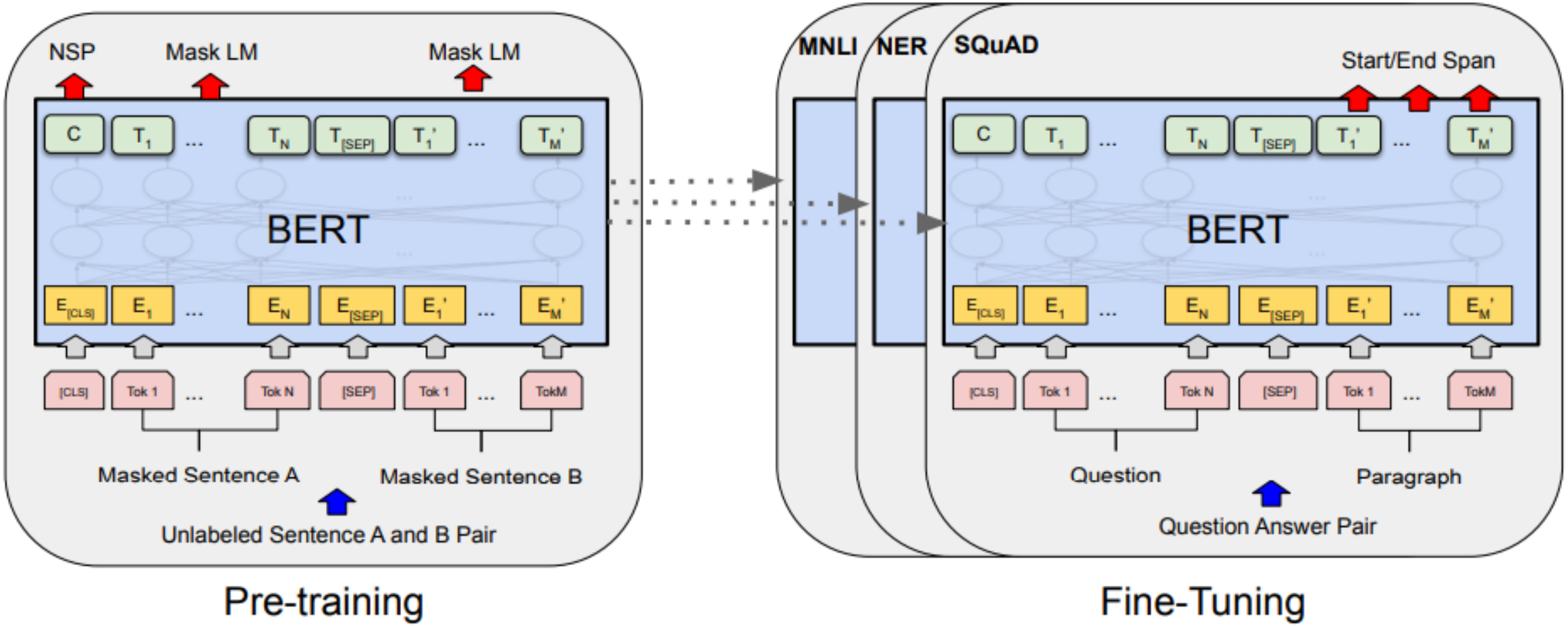




https://arxiv.org/abs/1810.04805

Module: Reader

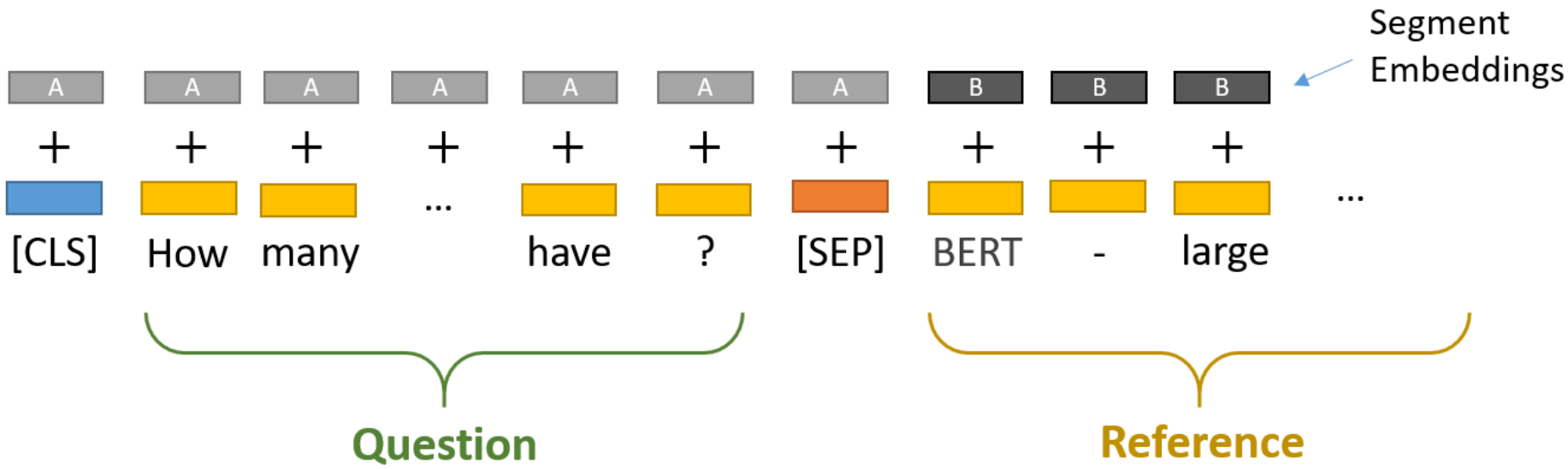
❖ From original BERT to BERT for QA



https://blog.paperspace.com/how-to-train-question-answering-machine-learning-models/

Module: Reader

❖ BERT Input for QA



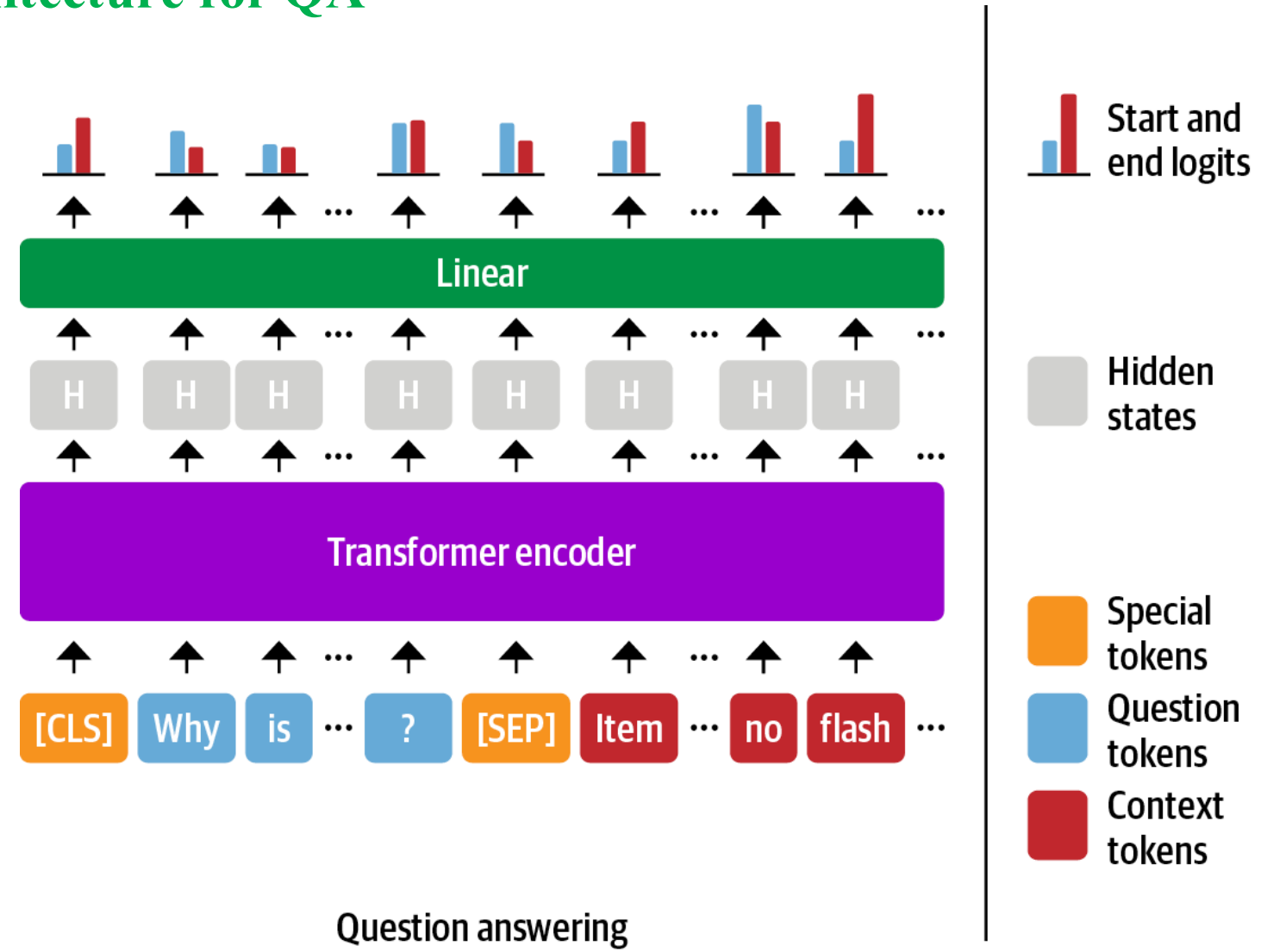
Question: How many parameters does BERT-large have?

Reference Text: BERT-large is really big... it has 24 layers and an embedding size of 1,024, for a total of 340M parameters! Altogether it is 1.34GB, so expect it to take a couple minutes to download to your Colab instance.

https://mccormickml.com/2020/03/10/question-answering-with-a-fine-tuned-BERT/

Module: Reader

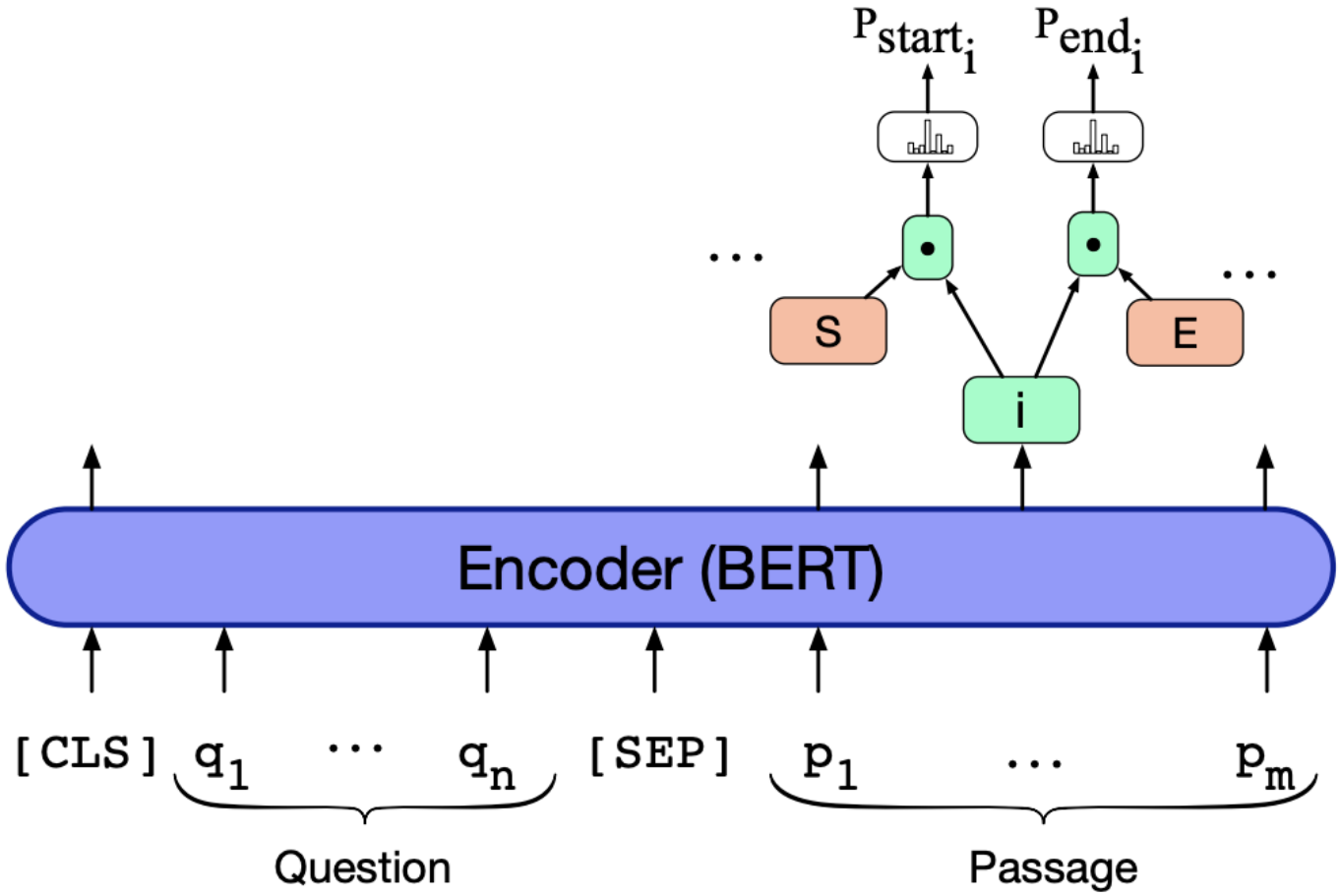
❖ BERT Architecture for QA



https://pub.towardsai.net/ask-and-you-shall-receive-using-bert-for-question-answering-with-the-world-happiness-report-3fa8bf4df536

Module: Reader

❖ BERT Architecture for QA



i : Hidden state of *i'th* token

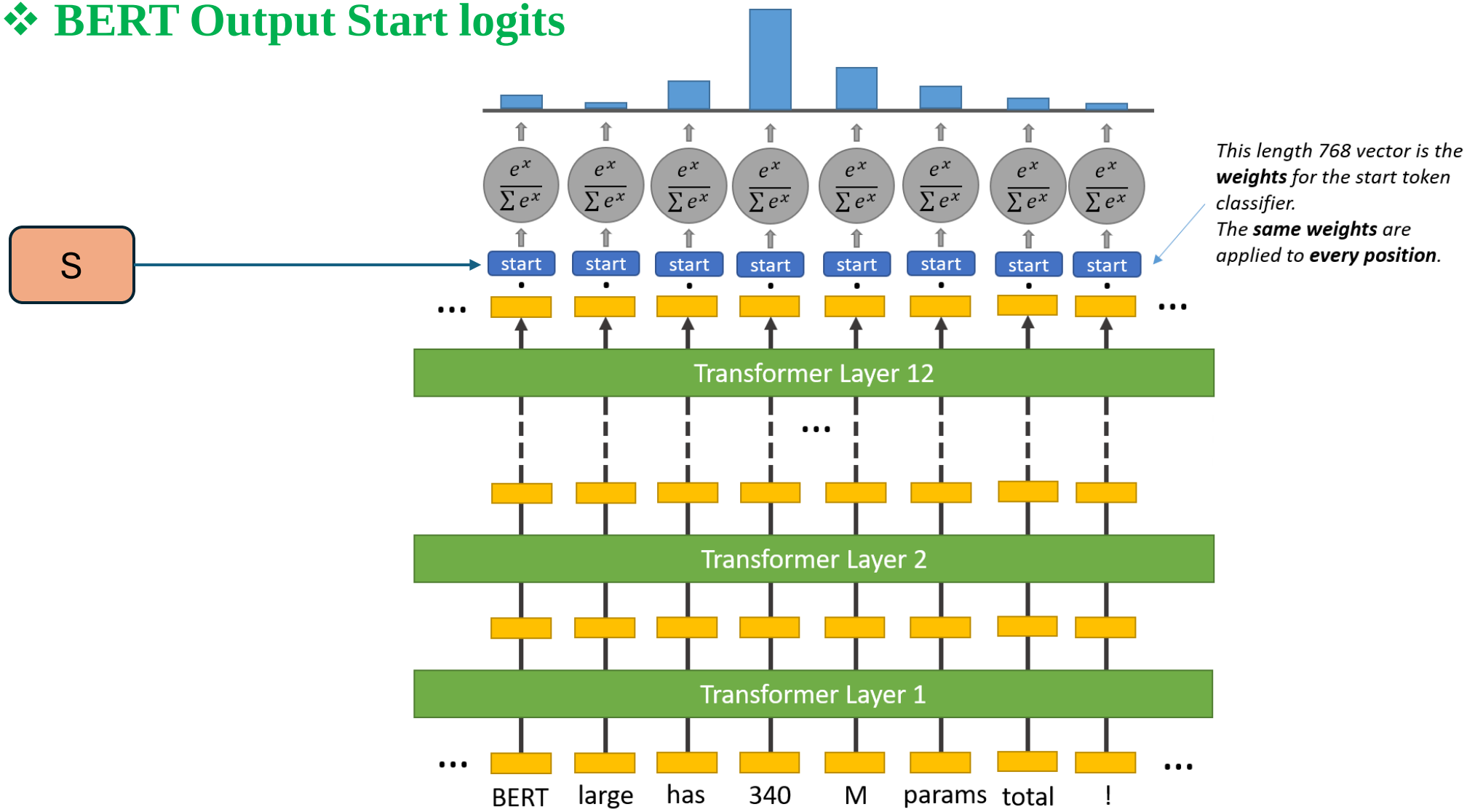
S : Start token embedding

E : End token embedding

• : Dot product

Module: Reader

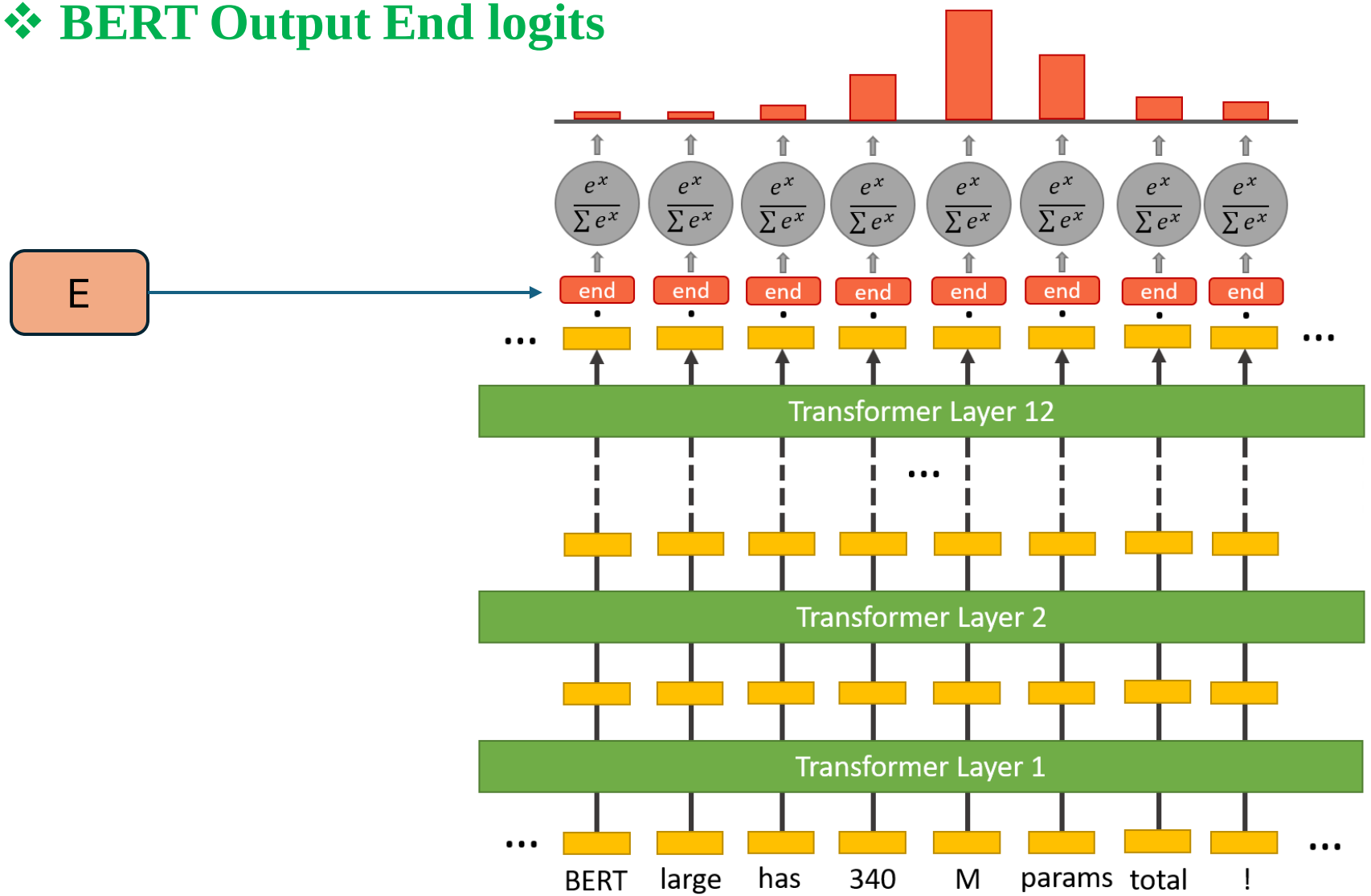
❖ BERT Output Start logits



https://mccormickml.com/2020/03/10/question-answering-with-a-fine-tuned-BERT/

Module: Reader

❖ BERT Output End logits



https://mccormickml.com/2020/03/10/question-answering-with-a-fine-tuned-BERT/

Module: Reader

❖ Step 1: Install and import libraries

```
!pip install -qq datasets==2.16.1 evaluate==0.4.1 transformers[sentencepiece]==4.35.2
!pip install -qq accelerate==0.26.1
!apt install git-lfs
```

```
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 507.1/507.1 kB 4.3 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 84.1/84.1 kB 13.4 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 115.3/115.3 kB 18.8 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 134.8/134.8 kB 20.9 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 134.8/134.8 kB 22.2 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 270.9/270.9 kB 7.1 MB/s eta 0:00:00
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git-lfs is already the newest version (3.0.2-1ubuntu0.2).
0 upgraded, 0 newly installed, 0 to remove and 33 not upgraded.
```

Module: Reader

❖ Step 1: Install and import libraries

```python
import numpy as np
from tqdm.auto import tqdm
import collections

import torch

from datasets import load_dataset
from transformers import AutoTokenizer
from transformers import AutoModelForQuestionAnswering
from transformers import TrainingArguments
from transformers import Trainer
import evaluate

device = torch.device("cuda") if torch.cuda.is_available() else torch.device("cpu")
```

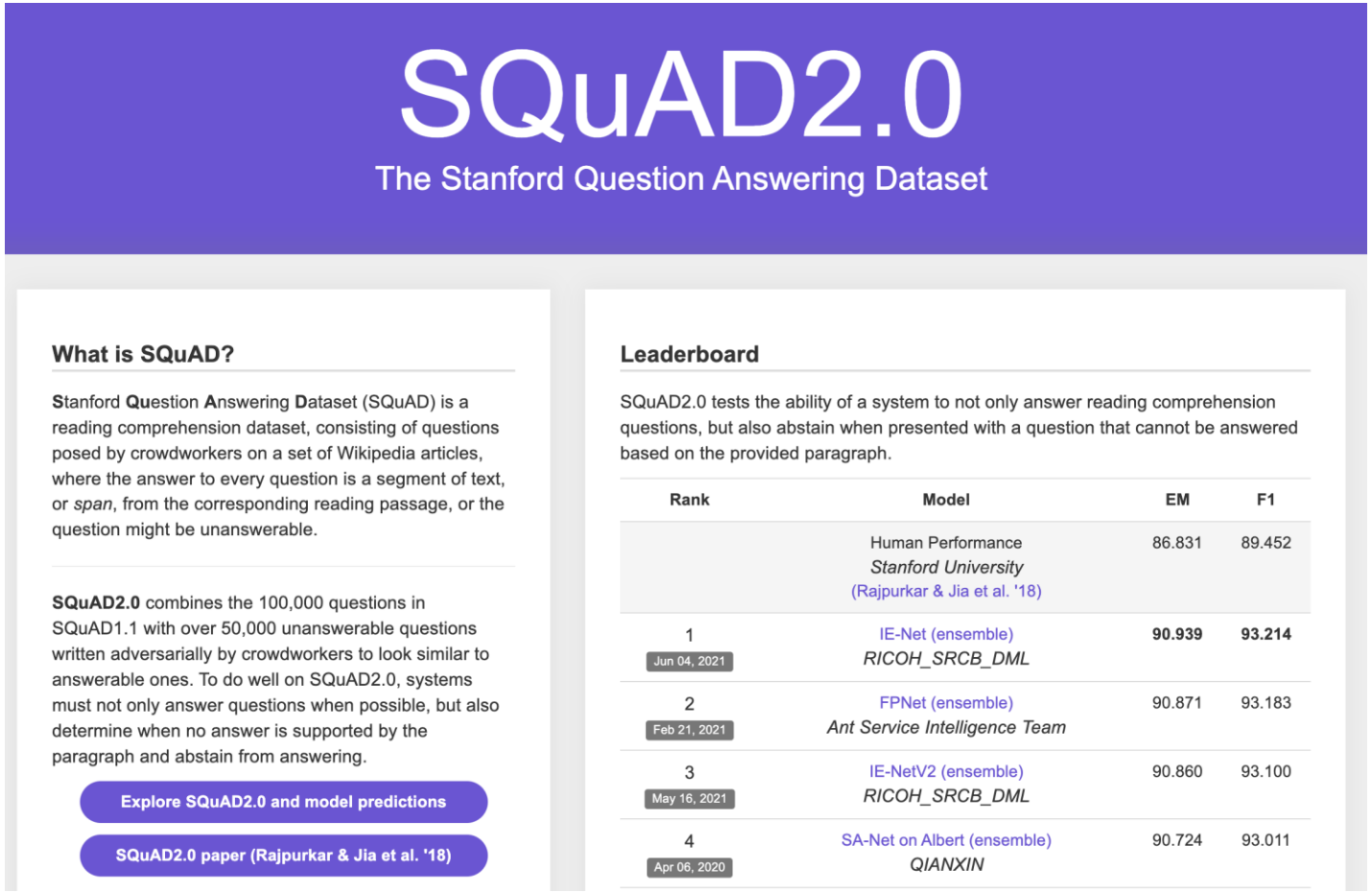
Module: Reader

❖ Step 2: Setup config

```
# Sử dụng mô hình "distilbert-base-uncased"
# làm mô hình checkpoint
MODEL_NAME = "distilbert-base-uncased"

# Độ dài tối đa cho mỗi đoạn văn bản
# sau khi được xử lý
MAX_LENGTH = 384

# Khoảng cách giữa các điểm bắt đầu
# của các đoạn văn bản liên tiếp
STRIDE = 128
```

Module: Reader

❖ Step 3: Download dataset



SQuAD2.0

The Stanford Question Answering Dataset

What is SQuAD?

Stanford **Qu**estion **A**nswering **D**ataset (SQuAD) is a reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or *span*, from the corresponding reading passage, or the question might be unanswerable.

SQuAD2.0 combines the 100,000 questions in SQuAD1.1 with over 50,000 unanswerable questions written adversarially by crowdworkers to look similar to answerable ones. To do well on SQuAD2.0, systems must not only answer questions when possible, but also determine when no answer is supported by the paragraph and abstain from answering.

Explore SQuAD2.0 and model predictions

SQuAD2.0 paper (Rajpurkar & Jia et al. '18)

Leaderboard

SQuAD2.0 tests the ability of a system to not only answer reading comprehension questions, but also abstain when presented with a question that cannot be answered based on the provided paragraph.

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar & Jia et al. '18)	86.831	89.452
1 Jun 04, 2021	IE-Net (ensemble) *RICOH_SRCB_DML*	**90.939**	**93.214**
2 Feb 21, 2021	FPNet (ensemble) *Ant Service Intelligence Team*	90.871	93.183
3 May 16, 2021	IE-NetV2 (ensemble) *RICOH_SRCB_DML*	90.860	93.100
4 Apr 06, 2020	SA-Net on Albert (ensemble) *QIANXIN*	90.724	93.011

SQuAD v2 dataset

Module: Reader

❖ Step 3: Download dataset

	SQuAD 2.0
Train	
Total examples	130,319
Negative examples	43,498
Total articles	442
Articles with negatives	285
Development	
Total examples	11,873
Negative examples	5,945
Total articles	35
Articles with negatives	35
Test	
Total examples	8,862
Negative examples	4,332
Total articles	28
Articles with negatives	28

Answer type	Percentage	Example
Date	8.9%	19 October 2023
Other Numeric	10.9%	12
Person	12.9%	Thomas Coke
Location	4.4%	Germany
Other Entity	15.3%	ABC Sports
Common Noun Phrase	31.8%	property damage
Adjective Phrase	3.9%	second-largest
Verb Phrase	5.5%	returned to Earth
Clause	3.7%	to avoid trivialization
Other	2.7%	quietly

Module: Reader

❖ Step 3: Download dataset

Context: The English name "Normans" comes from the French words Normans/Normanz, plural of Normant, modern French normand, which is itself borrowed from (Old Low Franconian Nortmann "Northman" or directly from Old Norse Nordmaðr, Latinized variously as Nortmannus, Normannus, or Nordmannus (recorded in Medieval Latin, **9th century**) to mean "Norseman, **Viking**".

Q1: What causes precipitation to fall?
A1: Viking

Q2: When was the Latin version of the word Norman first recorded?
A2: 9th century

Q3: When was the French version of the word Norman first recorded?
A3: No Answers

Context: In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under **gravity**. The main forms of precipitation include drizzle, rain, sleet, snow, **graupel** and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals **within a cloud**. Short, intense periods of rain in scattered locations are called "showers".

Q1: What causes precipitation to fall?
A1: Gravity

Q2: What is another main form of precipitation be- sides drizzle, rain, snow, sleet and hail?
A2: Graupel

Q3: Where do water droplets collide with ice crystals to form precipitation?
A3: Within a cloud

SQuAD v2 dataset samples

Module: Reader

❖ Step 3: Download dataset

id string · *lengths* 24 24	title string · *lengths* 3 59	context string · *lengths* 151 3.71k	question string · *lengths* 1 25.7k	answers sequence
56be85543aeaaa14008c9063	Beyoncé	Beyoncé Giselle Knowles-Carter (/bi…	When did Beyonce start becoming…	{ "text": ["in the late 1990s"]…
56be85543aeaaa14008c9065	Beyoncé	Beyoncé Giselle Knowles-Carter (/bi…	What areas did Beyonce compete in…	{ "text": ["singing and…
56be85543aeaaa14008c9066	Beyoncé	Beyoncé Giselle Knowles-Carter (/bi…	When did Beyonce leave Destiny's…	{ "text": ["2003"], "answer_start"…
56bf6b0f3aeaaa14008c9601	Beyoncé	Beyoncé Giselle Knowles-Carter (/bi…	In what city and state did Beyonce…	{ "text": ["Houston, Texas"…
56bf6b0f3aeaaa14008c9602	Beyoncé	Beyoncé Giselle Knowles-Carter (/bi…	In which decade did Beyonce become…	{ "text": ["late 1990s"],…
56bf6b0f3aeaaa14008c9603	Beyoncé	Beyoncé Giselle Knowles-Carter (/bi…	In what R&B group was she the lead…	{ "text": ["Destiny's Child"…

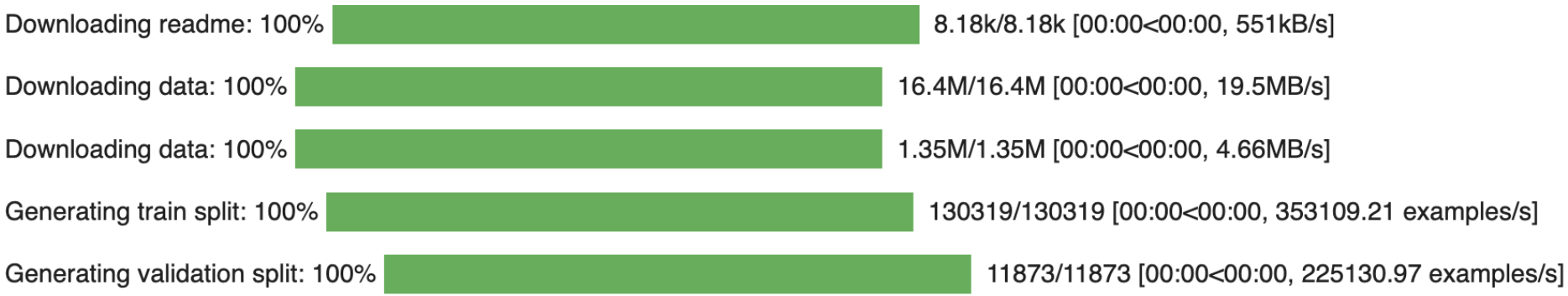
Download SQuAD v2 dataset in here

Module: Reader

❖ Step 3: Download dataset

```
1 # Download squad dataset từ HuggingFace
2 DATASET_NAME = 'squad_v2'
3 raw_datasets = load_dataset(DATASET_NAME)
```

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:88: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens)
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
```



Module: Reader

❖ Step 3: Download dataset

```
1 raw_datasets
```

```
DatasetDict({
    train: Dataset({
        features: ['id', 'title', 'context', 'question', 'answers'],
        num_rows: 130319
    })
    validation: Dataset({
        features: ['id', 'title', 'context', 'question', 'answers'],
        num_rows: 11873
    })
})
```

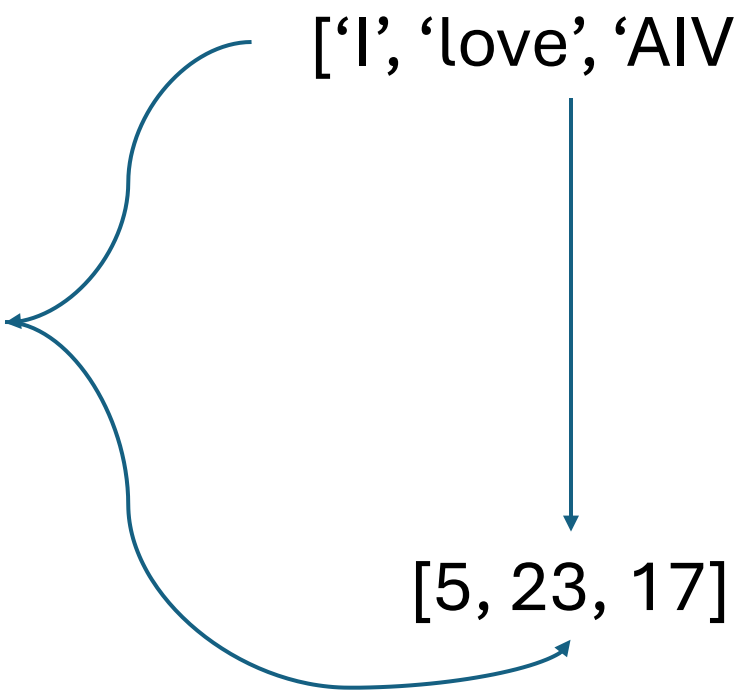
```
1 # Print các thông tin Context, Question, vaf Answer trong dataset
2 print("Context: ", raw_datasets["train"][0]["context"])
3 print("Question: ", raw_datasets["train"][0]["question"])
4 print("Answer: ", raw_datasets["train"][0]["answers"])
```

```
Context:  Beyoncé Giselle Knowles-Carter (/biːˈjɒnseɪ/ bee-YON-say) (born September 4, 1981)
Question:  When did Beyonce start becoming popular?
Answer:  {'text': ['in the late 1990s'], 'answer_start': [269]}
```

Module: Reader

❖ Step 4: Create tokenizer

```
{'vincent': 32,
 'vietnam': 31,
 'van': 30,
 'university': 29,
 'what': 11,
 '<sep>': 4,
 '<bos>': 2,
 'of': 24,
 'am': 10,
 'my': 7,
 'is': 6,
 'at': 19,
 '.': 8,
 'gogh': 22,
 '<eos>': 3,
 '<pad>': 1,
 'computer': 20,
 'painting': 25,
 'and': 12,
 '<unk>': 0,
 'artist': 18,
 'favorite': 13,
 'studying': 15,
 'i': 5,
```




```
1 # Load tokenizer để run một số example
2 tokenizer = AutoTokenizer.from_pretrained(MODEL_NAME)
```

tokenizer_config.json: 100% 28.0/28.0 [00:00<00:00, 810B/s]

config.json: 100% 483/483 [00:00<00:00, 5.10kB/s]

vocab.txt: 100% 232k/232k [00:00<00:00, 973kB/s]

tokenizer.json: 100% 466k/466k [00:00<00:00, 1.91MB/s]

Module: Reader

❖ Step 4: Create tokenizer

[CLS]	This	is	the	question	[SEP]	This	is	the	context
with	lots	of	info	##rma	##tion	.	Some	use	##less
.	The	answer	is	here	some	more	words	.	[SEP]

BERT Input format for QA

https://huggingface.co/learn/nlp-course/chapter6/3b#using-the-question-answering-pipeline

Module: Reader

❖ Step 4: Create tokenizer

```python
# Lấy ra 1 example từ tập train
context = raw_datasets["train"][0]["context"]
question = raw_datasets["train"][0]["question"]
```

```python
# Sử dụng tokenizer để mã hóa dữ liệu đầu vào
inputs = tokenizer(
    question,  # Danh sách các câu hỏi
    context,  # Nội dung liên quan đến câu hỏi
    max_length=MAX_LENGTH,  # Độ dài tối đa cho đầu ra mã hóa
    truncation="only_second",  # Cắt bớt dữ liệu chỉ cho phần thứ hai (context)
    stride=STRIDE,  # Độ dài bước nhảy trong trường hợp dữ liệu dài hơn max_length
    return_overflowing_tokens=True,  # Trả về các tokens vượt quá độ dài tối đa
    return_offsets_mapping=True,  # Trả về bản đồ vị trí của các tokens trong văn bản gốc
    padding="max_length" # Điền vào dữ liệu để có cùng độ dài max_length
)
```

Module: Reader

❖ Step 4: Create tokenizer

```
1 print(f'Question: {question}')
2 print()
3 print(f'Context: {context}')
```

```
Question: When did Beyonce start becoming popular?

Context: Beyoncé Giselle Knowles-Carter (/biːˈjɒnseɪ/ bee-YON-say) (born September 4, 1981)
```

```
1 inputs.keys()
```

```
dict_keys(['input_ids', 'attention_mask', 'offset_mapping', 'overflow_to_sample_mapping'])
```

```
1 inputs
```

```
{'input_ids': [[101, 2043, 2106, 20773, 2707, 3352, 2759, 1029, 102, 20773, 21025, 19358, 22815,
29685, 1013, 10506, 1011, 10930, 2078, 1011, 2360, 1007, 1006, 2141, 2244, 1018, 1010, 3261, 100
1012, 2141, 1998, 2992, 1999, 5395, 1010, 3146, 1010, 2016, 2864, 1999, 2536, 4823, 1998, 5613,
2397, 4134, 2004, 2599, 3220, 1997, 1054, 1004, 1038, 2611, 1011, 2177, 10461, 1005, 1055, 2775,
2150, 2028, 1997, 1996, 2088, 1005, 1055, 2190, 1011, 4855, 2611, 2967, 1997, 2035, 2051, 1012,
1010, 20754, 1999, 2293, 1006, 2494, 1007, 1010, 2029, 2511, 2014, 2004, 1037, 3948, 3063, 4969,
2193, 1011, 2028, 3895, 1000, 4689, 1999, 2293, 1000, 1998, 1000, 3336, 2879, 1000, 1012, 102, 0
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]], 'attention_mask': [[1, 1, 1,
```

Module: Reader

❖ Step 4: Create tokenizer

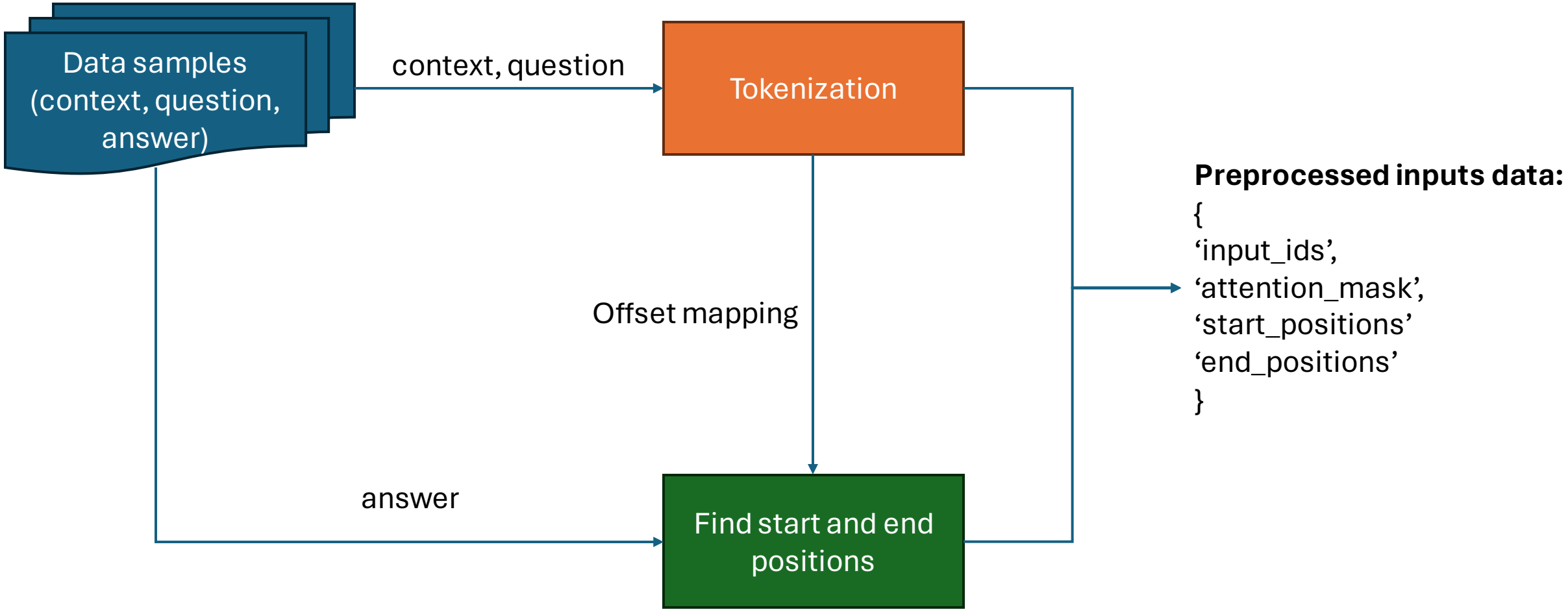
[CLS] question [SEP] context [SEP]

```
1 tokenizer.decode(inputs['input_ids'][0])
```

```
'[CLS] when did beyonce start becoming popular? [SEP] beyonce giselle knowles - carter ( / biːˈjɒnseɪ / bee - yon - say )
singer, songwriter, record producer and actress. born and raised in houston, texas, she performed in various singing and d
o fame in the late 1990s as lead singer of r & b girl - group destiny\'s child. managed by her father, mathew knowles, the
elling girl groups of all time. their hiatus saw the release of beyonce\'s debut album, dangerously in love ( 2003 ), whic
de, earned five grammy awards and featured the billboard hot 100 number - one singles " crazy in love " and " baby boy ".
[PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [PAD] [P
AD] [PAD] [PAD] [...]'
```

Module: Reader

❖ Step 5: Create preprocessing function



Module: Reader

❖ Step 5: Create preprocessing function

```python
# Định nghĩa hàm preprocess_training_examples và nhận đối số examples là dữ liệu đào tạo
def preprocess_training_examples(examples):
    # Trích xuất danh sách câu hỏi từ examples và loại bỏ các khoảng trắng dư thừa
    questions = [q.strip() for q in examples["question"]]

    # Tiến hành mã hóa thông tin đầu vào sử dụng tokenizer
    inputs = tokenizer(
        questions,
        examples["context"],
        max_length=MAX_LENGTH,
        truncation="only_second",
        stride=STRIDE,
        return_overflowing_tokens=True,
        return_offsets_mapping=True,
        padding="max_length",
    )

    # Trích xuất offset_mapping từ inputs và loại bỏ nó ra khỏi inputs
    offset_mapping = inputs.pop("offset_mapping")

    # Trích xuất sample_map từ inputs và loại bỏ nó ra khỏi inputs
    sample_map = inputs.pop("overflow_to_sample_mapping")

    # Trích xuất thông tin về câu trả lời (answers) từ examples
    answers = examples["answers"]
```

Module: Reader

❖ Step 5: Create preprocessing function

```
# Trích xuất thông tin về câu trả lời (answers) từ examples
answers = examples["answers"]

# Khởi tạo danh sách các vị trí bắt đầu và kết thúc câu trả lời
start_positions = []
end_positions = []

# Duyệt qua danh sách offset_mapping
for i, offset in enumerate(offset_mapping):
    # Xác định index của mẫu (sample) liên quan đến offset hiện tại
    sample_idx = sample_map[i]

    # Trích xuất sequence_ids từ inputs
    sequence_ids = inputs.sequence_ids(i)

    # Xác định vị trí bắt đầu và kết thúc của ngữ cảnh
    idx = 0
    while sequence_ids[idx] != 1:
        idx += 1
    context_start = idx
    while sequence_ids[idx] == 1:
        idx += 1
    context_end = idx - 1

    # Trích xuất thông tin về câu trả lời cho mẫu này
    answer = answers[sample_idx]
```

Finding start and end positions for each sample

Module: Reader

❖ Step 5: Create preprocessing function

```
    # Trích xuất thông tin về câu trả lời cho mẫu này
    answer = answers[sample_idx]

    if len(answer['text']) == 0:
        start_positions.append(0)
        end_positions.append(0)
    else:
        # Xác định vị trí ký tự bắt đầu và kết thúc của câu trả lời trong ngữ cảnh
        start_char = answer["answer_start"][0]
        end_char = answer["answer_start"][0] + len(answer["text"][0])

        # Nếu câu trả lời không nằm hoàn toàn trong ngữ cảnh, gán nhãn là (0, 0)
        if offset[context_start][0] > start_char or offset[context_end][1] < end_char:
            start_positions.append(0)
            end_positions.append(0)
        else:
            # Nếu không, gán vị trí bắt đầu và kết thúc dựa trên vị trí của các mã thông tin
            idx = context_start
            while idx <= context_end and offset[idx][0] <= start_char:
                idx += 1
            start_positions.append(idx - 1)

            idx = context_end
            while idx >= context_start and offset[idx][1] >= end_char:
                idx -= 1
            end_positions.append(idx + 1)

# Thêm thông tin vị trí bắt đầu và kết thúc vào inputs
inputs["start_positions"] = start_positions
inputs["end_positions"] = end_positions
```

Finding start and end positions for each sample

Module: Reader

❖ Step 5: Create preprocessing function

```
train_dataset = raw_datasets["train"].map(
    preprocess_training_examples,
    batched=True,
    remove_columns=raw_datasets["train"].column_names,
)

# In ra độ dài của tập dữ liệu "train" ban đầu và độ dài của tập dữ liệu đã được xử lý (train_dataset)
len(raw_datasets["train"]), len(train_dataset)
```

```
Map: 100% 130319/130319 [01:33<00:00, 1327.24 examples/s]
(130319, 131754)
```

```
print(train_dataset[0:10]['start_positions'])
print(train_dataset[0:10]['end_positions'])
```

```
[75, 68, 143, 58, 78, 94, 134, 101, 77, 84]
[78, 70, 143, 60, 79, 97, 136, 102, 78, 85]
```

Module: Reader

❖ Step 5: Create preprocessing function

```python
def preprocess_validation_examples(examples):
    # Chuẩn bị danh sách câu hỏi bằng cách loại bỏ khoảng tr
    questions = [q.strip() for q in examples["question"]]

    # Sử dụng tokenizer để mã hóa các câu hỏi và văn bản liê
    inputs = tokenizer(
        questions,
        examples["context"],
        max_length=MAX_LENGTH,
        truncation="only_second",
        stride=STRIDE,
        return_overflowing_tokens=True,
        return_offsets_mapping=True,
        padding="max_length",
    )
    # Lấy ánh xạ để ánh xạ lại ví dụ tham chiếu cho từng dòng trong inputs
    sample_map = inputs.pop("overflow_to_sample_mapping")
    example_ids = []

    # Xác định ví dụ tham chiếu cho mỗi dòng đầu vào và điều chỉnh ánh xạ offset
    for i in range(len(inputs["input_ids"])):
        sample_idx = sample_map[i]
        example_ids.append(examples["id"][sample_idx])

        sequence_ids = inputs.sequence_ids(i)
        offset = inputs["offset_mapping"][i]

        # Loại bỏ các offset không phù hợp với sequence_ids
        inputs["offset_mapping"][i] = [
            o if sequence_ids[k] == 1 else None for k, o in enumerate(offset)
        ]

    # Thêm thông tin ví dụ tham chiếu vào đầu vào
    inputs["example_id"] = example_ids

    return inputs
```

Module: Reader

❖ Step 5: Create preprocessing function

```
validation_dataset = raw_datasets["validation"].map(
    preprocess_validation_examples,  # Gọi hàm preprocess_validation_examples
    batched=True,  # Xử lý dữ liệu theo từng batch.
    remove_columns=raw_datasets["validation"].column_names,  # Loại bỏ các cột
)

# In ra độ dài của raw_datasets["validation"] và validation_dataset để so sánh
len(raw_datasets["validation"]), len(validation_dataset)
```

```
Map: 100% 11873/11873 [00:14<00:00, 938.54 examples/s]
(11873, 12134)
```

Module: Reader

❖ Step 6: Training

```
# Load model
model = AutoModelForQuestionAnswering.from_pretrained(MODEL_NAME)
```

```
model.safetensors: 100% 268M/268M [00:01<00:00, 170MB/s]
Some weights of DistilBertForQuestionAnswering were not initialized from the model checkpoint
You should probably TRAIN this model on a down-stream task to be able to use it for predictior
```

```
# Tạo đối tượng args là các tham số cho quá trình huấn luyện
args = TrainingArguments(
    output_dir="distilbert-finetuned-squadv2",  # Thư mục lưu trữ kết quả huấn luyện
    evaluation_strategy="no",  # Chế độ đánh giá không tự động sau mỗi epoch
    save_strategy="epoch",  # Lưu checkpoint sau mỗi epoch
    learning_rate=2e-5,  # Tốc độ học
    num_train_epochs=3,  # Số epoch huấn luyện
    weight_decay=0.01,  # Giảm trọng lượng mô hình để tránh overfitting
    fp16=True,  # Sử dụng kiểu dữ liệu half-precision (16-bit) để tối ưu hóa tài nguyên
    push_to_hub=True,  # Đẩy kết quả huấn luyện lên một nơi chia sẻ trực tuyến (Hub)
)
```

```
# Khởi tạo một đối tượng Trainer để huấn luyện mô hình
trainer = Trainer(
    model=model,  # Sử dụng mô hình đã tạo trước đó
    args=args,  # Các tham số và cấu hình huấn luyện
    train_dataset=train_dataset,  # Sử dụng tập dữ liệu huấn luyện
    eval_dataset=validation_dataset,  # Sử dụng tập dữ liệu đánh giá
    tokenizer=tokenizer,  # Sử dụng tokenizer để xử lý văn bản
)

# Bắt đầu quá trình huấn luyện
trainer.train()
```

```
You're using a DistilBertTokenizerFast tokenizer.
[49354/49410 1:3
```

Step	Training Loss
500	3.102900
1000	2.268800
1500	1.971300
2000	1.810200
2500	1.702800
3000	1.625000
3500	1.574800
4000	1.566400
4500	1.495000
5000	1.480800
5500	1.438300
6000	1.402200
6500	1.418000
7000	1.426900
7500	1.366800
8000	1.351200
8500	1.318100
9000	1.307100
9500	1.308100
10000	1.284500

Module: Reader

❖ Step 7: Test

```
predicted_answers = []
for example in tqdm(examples):
    example_id = example['id']
    context = example['context']
    answers = []

    # Lặp qua tất cả các đặc trưng liên quan đến ví dụ đó
    for feature_index in example_to_features[example_id]:
        start_logit = start_logits[feature_index]
        end_logit = end_logits[feature_index]
        offsets = features[feature_index]['offset_mapping']

        # Lấy các chỉ số có giá trị lớn nhất cho start và end logits
        start_indexes = np.argsort(start_logit)[-1 : -N_BEST - 1 : -1].tolist()
        end_indexes = np.argsort(end_logit)[-1 : -N_BEST - 1 : -1].tolist()
        for start_index in start_indexes:
            for end_index in end_indexes:
                # Bỏ qua các câu trả lời không hoàn toàn nằm trong ngữ cảnh
                if offsets[start_index] is None or offsets[end_index] is None:
                    continue
                # Bỏ qua các câu trả lời có độ dài > max_answer_length
                if end_index - start_index + 1 > MAX_ANS_LENGTH:
                    continue

                # Tạo một câu trả lời mới
                answer = {
                    'text': context[offsets[start_index][0] : offsets[end_index][1]],
                    'logit_score': start_logit[start_index] + end_logit[end_index],
                }
                answers.append(answer)

    # Chọn câu trả lời có điểm số tốt nhất
    if len(answers) > 0:
        best_answer = max(answers, key=lambda x: x['logit_score'])
        answer_dict = {
            'id': example_id,
            'prediction_text': best_answer['text'],
            'no_answer_probability': 1 - best_answer['logit_score']
        }
    else:
        answer_dict = {
            'id': example_id,
            'prediction_text': '',
            'no_answer_probability': 1.0
        }
    predicted_answers.append(answer_dict)
```

```
100%
{'exact': 47.452202476206516,
 'f1': 51.28265600122848,
 'total': 11873,
 'HasAns_exact': 74.78070175438596,
 'HasAns_f1': 82.45259357331055,
 'HasAns_total': 5928,
 'NoAns_exact': 20.201850294365013,
 'NoAns_f1': 20.201850294365013,
 'NoAns_total': 5945,
 'best_exact': 64.46559420533984,
 'best_exact_thresh': -11.35546875,
 'best_f1': 66.16170764530801,
 'best_f1_thresh': -9.61328125}
```

Module: Reader

❖ Step 7: Test

```
# Use a pipeline as a high-level helper
from transformers import pipeline

PIPELINE_NAME = 'question-answering'
MODEL_NAME = 'thangduong0509/distilbert-finetuned-squadv2'
pipe = pipeline(PIPELINE_NAME, model=MODEL_NAME)
```

```
INPUT_QUESTION = 'What is my name?'
INPUT_CONTEXT = 'My name is AI Vietnam and I live in Vietnam.'
pipe(question=INPUT_QUESTION, context=INPUT_CONTEXT)
```

```
{'score': 0.97179114818573, 'start': 11, 'end': 21, 'answer': 'AI Vietnam'}
```

Module: Reader

❖ Step 7: Test

```
1 INPUT_QUESTION = 'where is the highest mountain in solar system located'
2 INPUT_CONTEXT = 'The highest mountain and volcano in the Solar System is on the planet Mars.
3 pipe(question=INPUT_QUESTION, context=INPUT_CONTEXT)
```

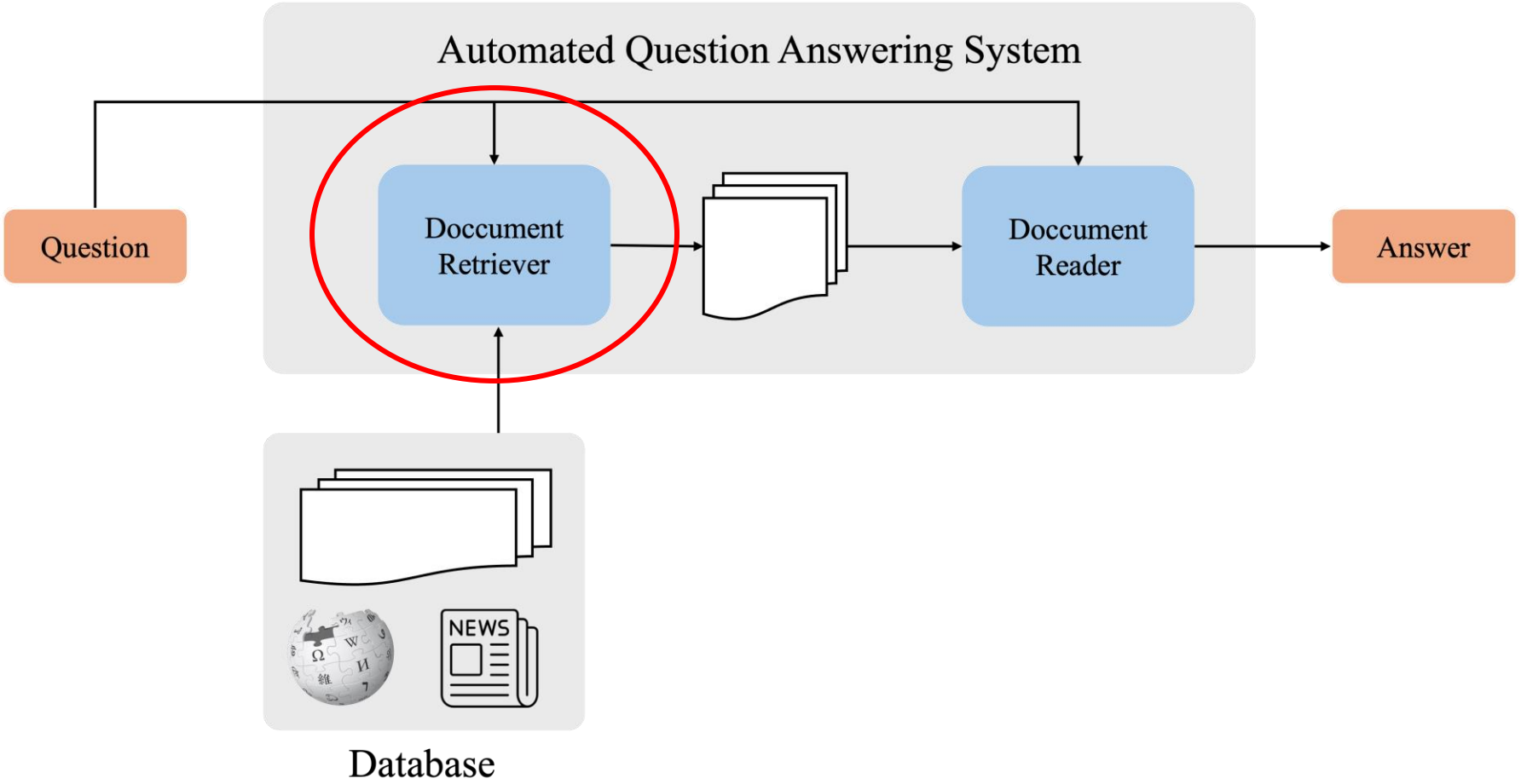
```
{'score': 0.852756679058075, 'start': 70, 'end': 74, 'answer': 'Mars'}
```

AI VIETNAM
All-in-One Course
(TA Session)

Module: Retriever

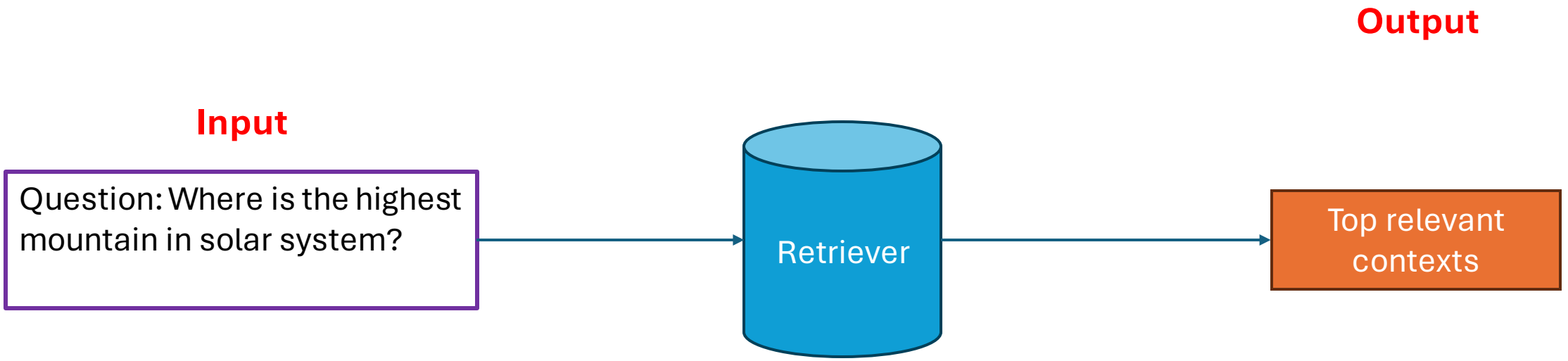
Module: Retriever

❖ E2E QA Pipeline



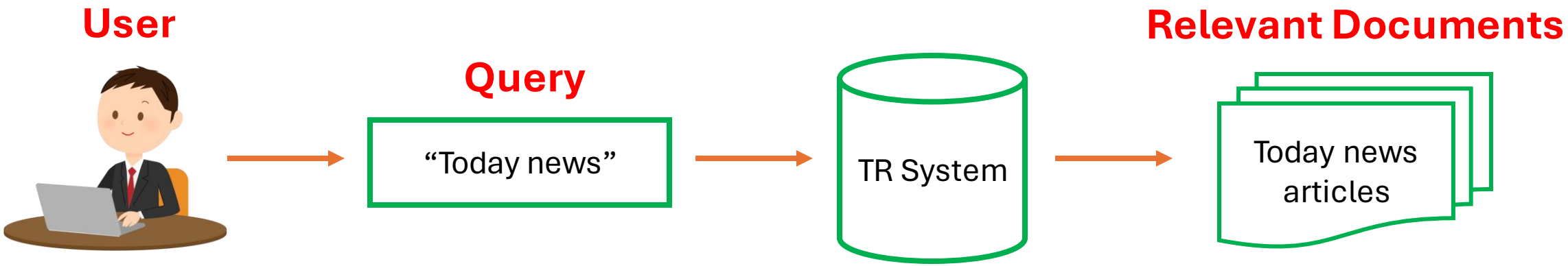
Module: Retriever

❖ Retriever I/O



Module: Retriever

❖ Text Retrieval

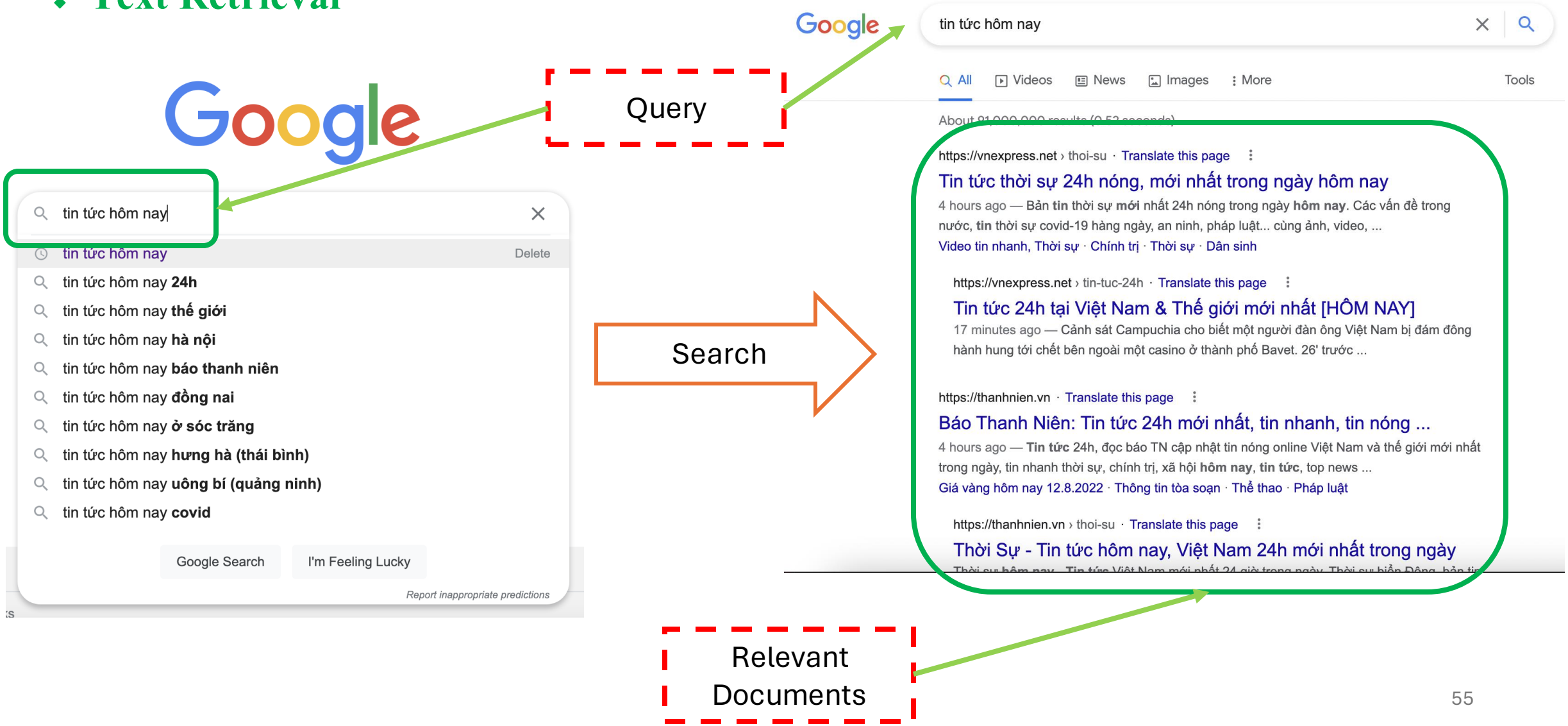


Text Retrieval (TR) (also called as Document Retrieval)[1]: A branch of Information Retrieval (IR) where the system matching of some stated user search query against a set of texts.

1: https://en.wikipedia.org/wiki/Document_retrieval
2: https://nlp.stanford.edu/IR-book/html/htmledition/an-example-information-retrieval-problem-1.html

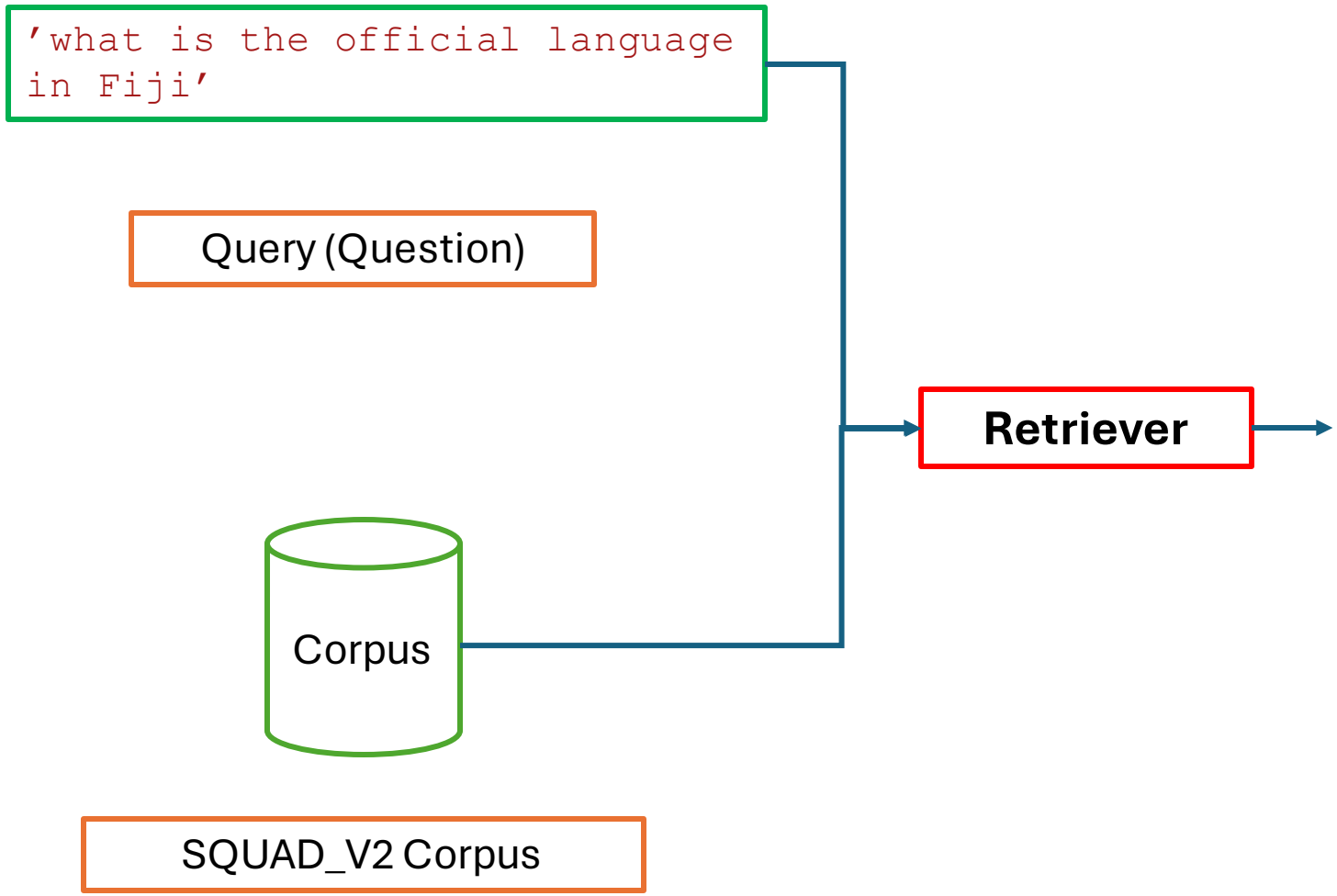
Module: Retriever

❖ Text Retrieval



Module: Retriever

❖ Text Retriever Pipeline

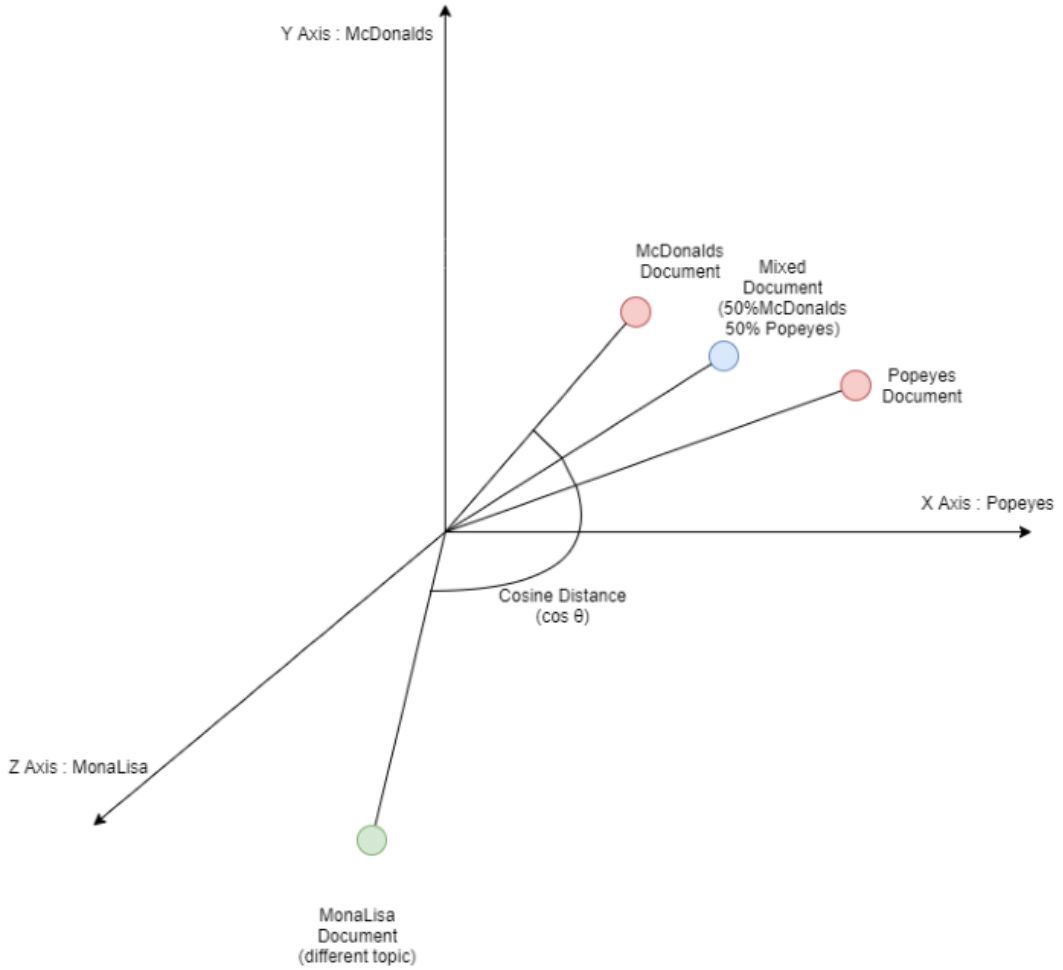



```
Query: what is the official language in Fiji
=== Relevant docs ===
Top 1; Score: 0.6556
The official languages in Fiji are Fijian and English. A dialect of Hindustani is als

Top 2; Score: 0.6556
The official languages in Fiji are Fijian and English. A dialect of Hindustani is als

Top 3; Score: 0.5715
The official languages. Fiji's 1997 Constitution established Fijian as one of the off

Top 4; Score: 0.5604
Of all the languages of Russia, Russian is the only official language. There are 35 d

Top 5; Score: 0.5592
Finnish is one of two official languages of Finland (the other being Swedish, spoken

Top 6; Score: 0.5307
Liberia is a multilingual country where more than thirty languages are spoken. Englis

Top 7; Score: 0.5164
There are over 120 ethnic groups, each with its own language or dialect. Indigenous T

Top 8; Score: 0.5094
the two official languages every country in the world has official languages that are

Top 9; Score: 0.4811
Fiji Language. The Fijian language spoken in Fiji is a type of Austronesian langauge

Top 10; Score: 0.4547
Upon Uganda's independence in 1962, English was maintained as the official language,
```

Relevant Questions

Module: Retriever

❖ Idea



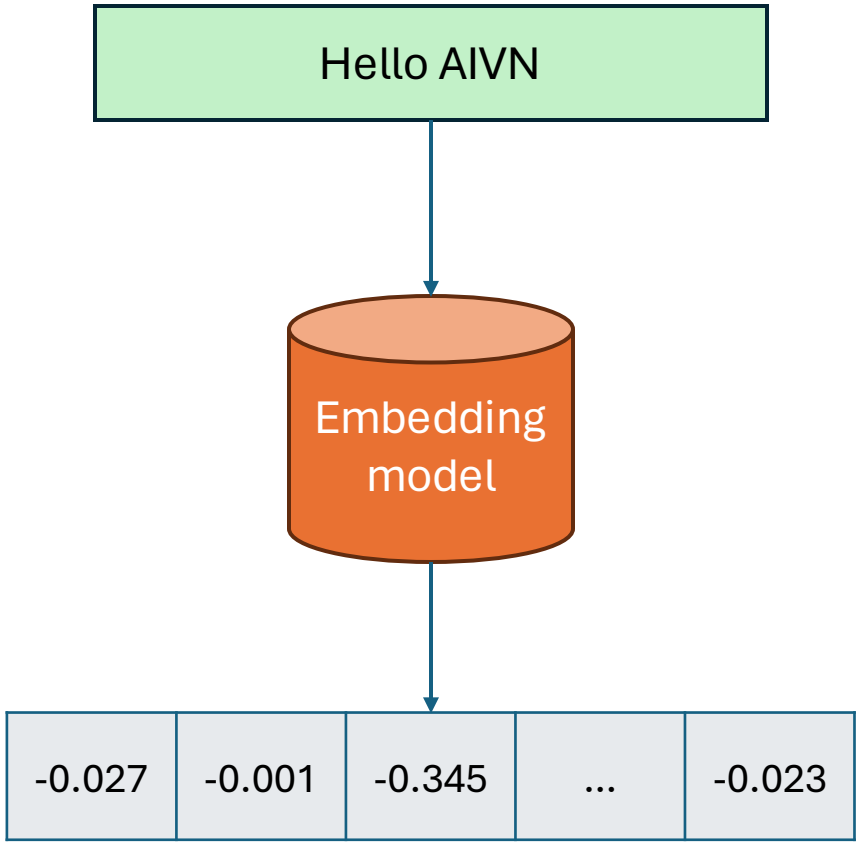
Given two vectors, we can calculate the similarity between them by using a similarity formula. E.g: L2 Distance, Cosine Similarity...

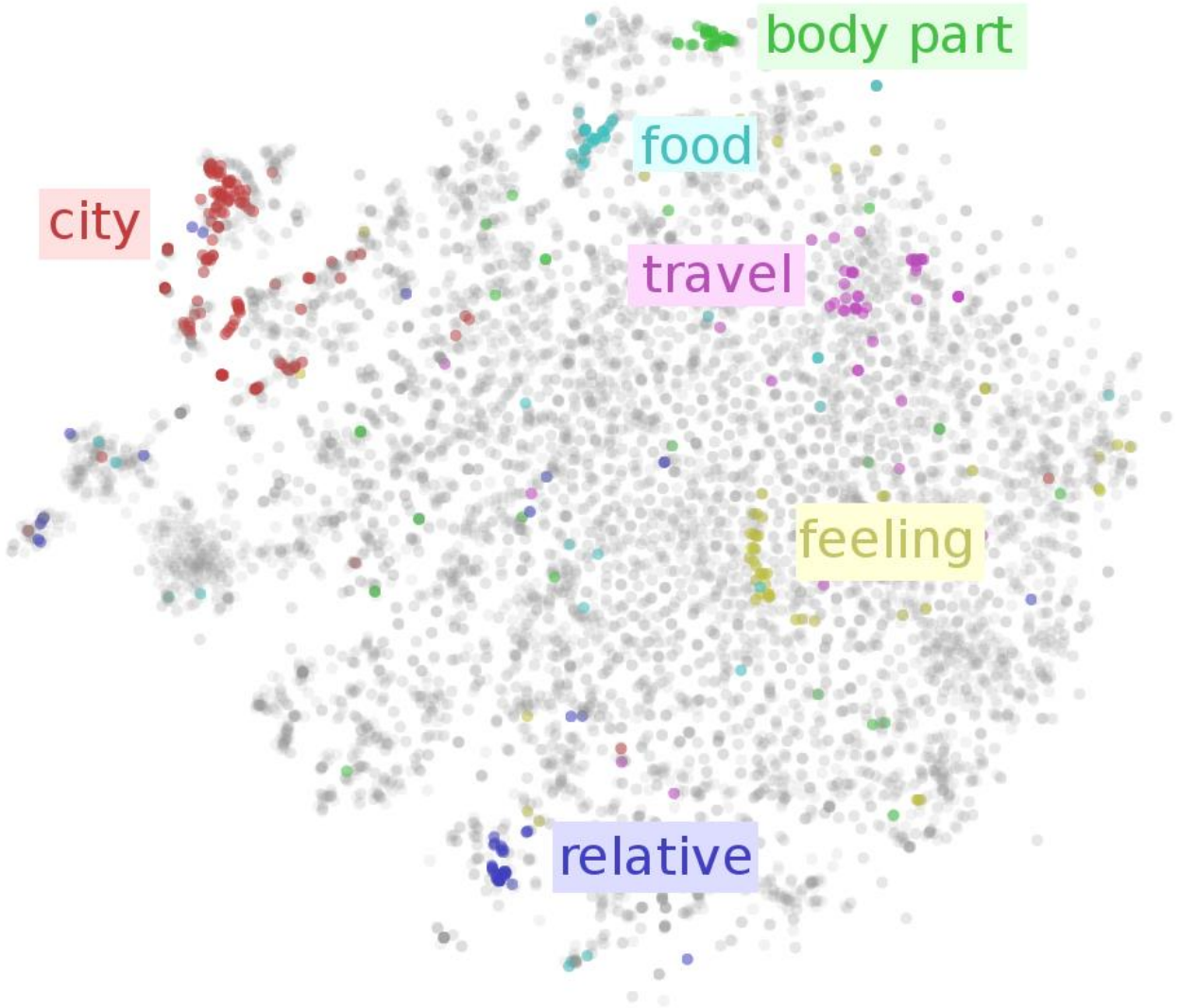
$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

https://www.researchgate.net/figure/Sample-cosine-similarity-results_fig2_347682908

Module: Retriever

❖ Idea

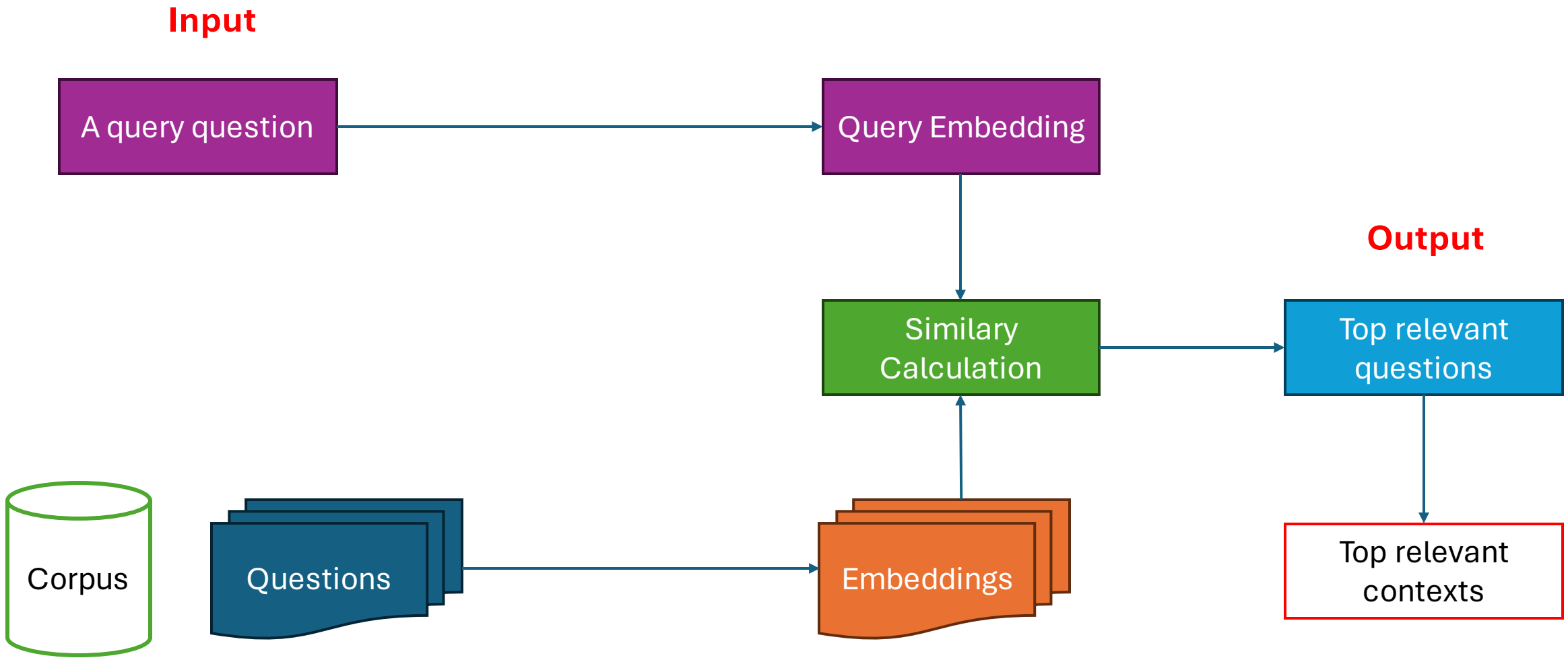




https://www.analyticsvidhya.com/blog/2020/03/pretrained-word-embeddings-nlp/

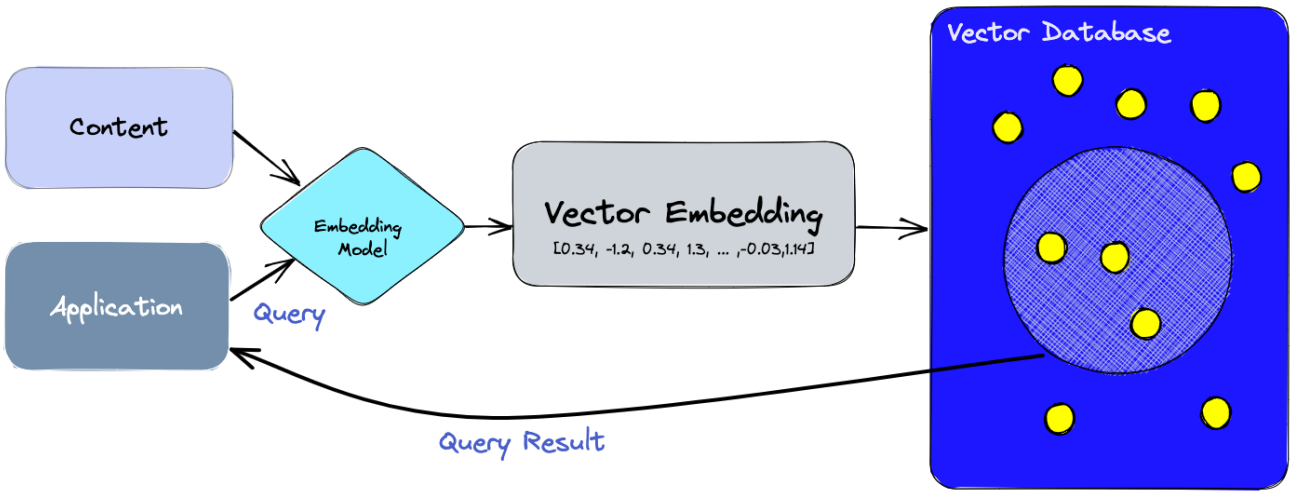
Module: Retriever

❖ Idea



Module: Retriever

❖ Vector Database

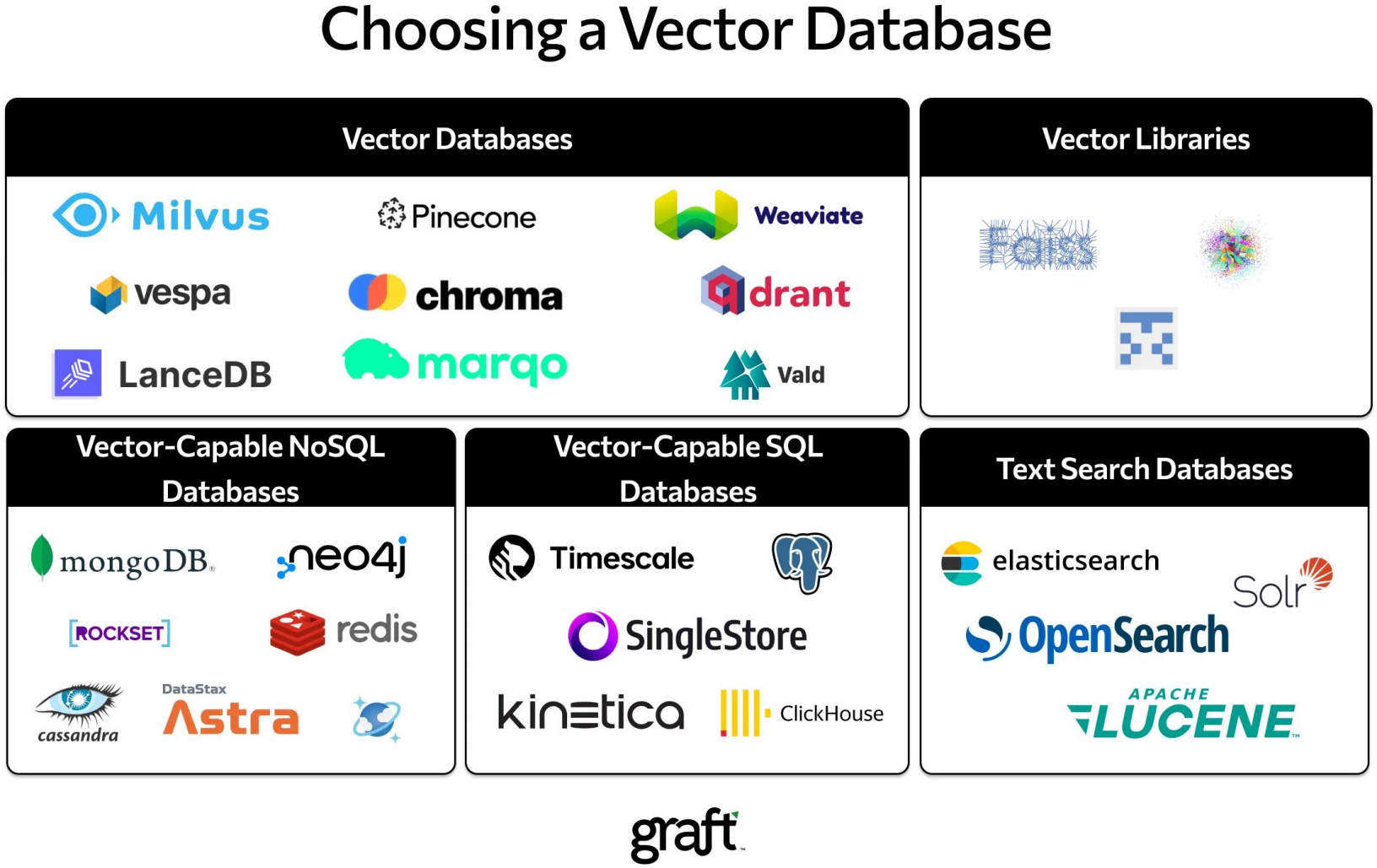


Vector database: A database that can store vectors (fixed-length lists of numbers) along with other data items.

https://www.pinecone.io/learn/vector-database/

Module: Retriever

❖ Vector Database



https://www.graft.com/blog/top-vector-databases-for-ai-projects

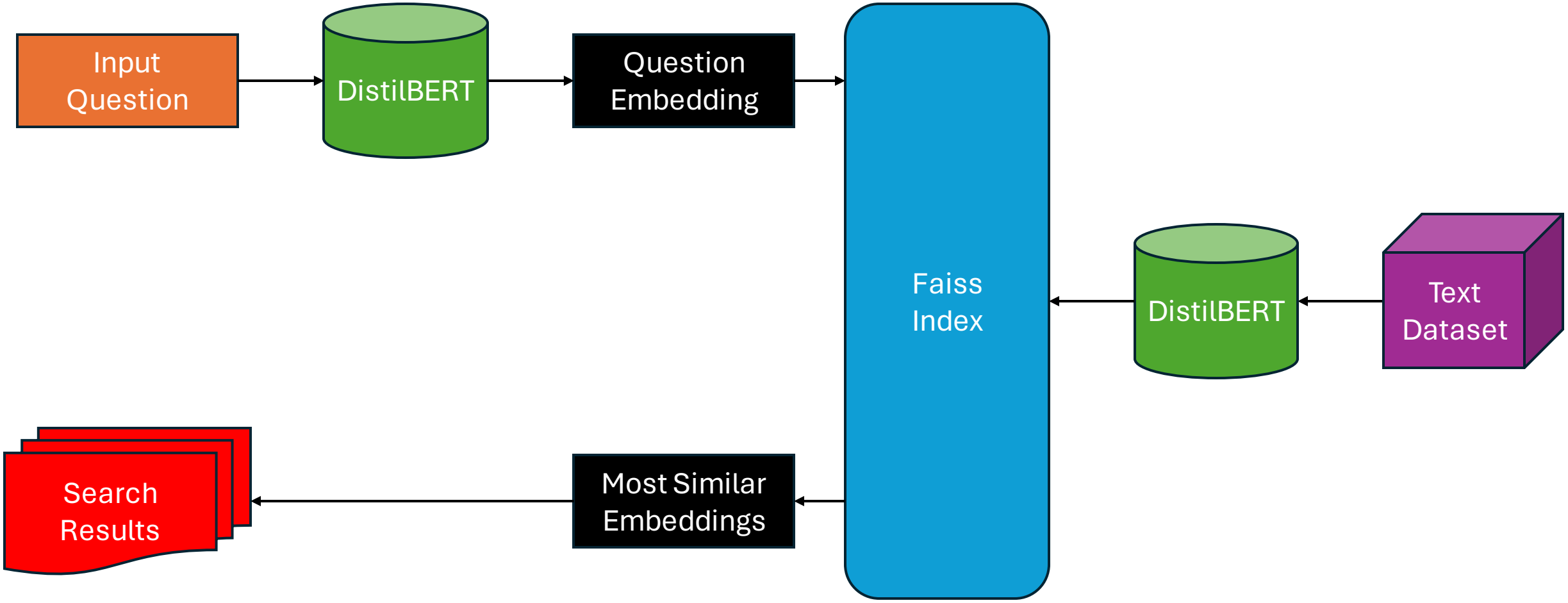
Module: Retriever

❖ Faiss

https://www.youtube.com/watch?v=sKyvsdEv6rk

Module: Retriever

❖ Faiss Pipeline



Module: Retriever

❖ Step 1: Install and import libraries

```
!pip install -qq transformers[sentencepiece]==4.35.2 datasets==2.16.1 evaluate==0.4.1
```

```
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 507.1/507.1 kB 7.0 MB/s eta 0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 84.1/84.1 kB 6.6 MB/s eta 0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 115.3/115.3 kB 10.6 MB/s eta 0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 134.8/134.8 kB 8.1 MB/s eta 0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 134.8/134.8 kB 10.6 MB/s eta 0:00:00
```

```
!sudo apt-get install libomp-dev
!pip install -qq faiss-gpu
```

Module: Retriever

❖ Step 1: Install and import libraries

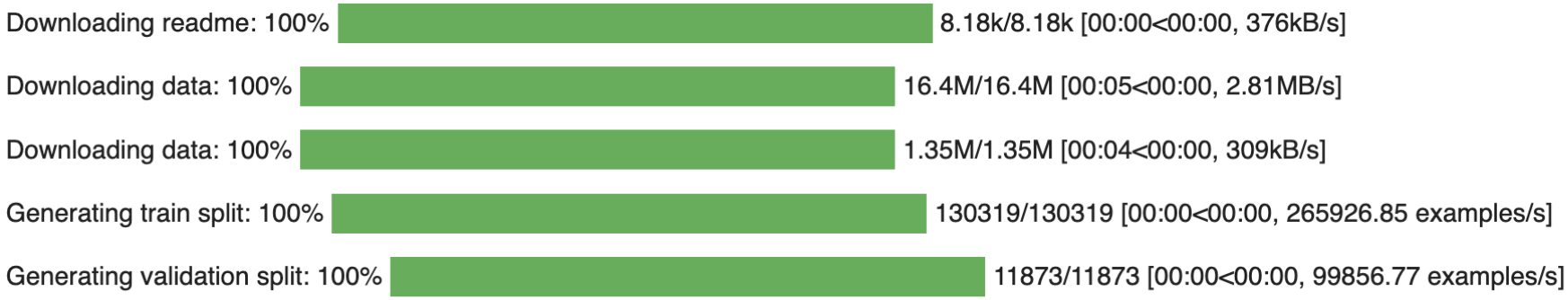
```python
import numpy as np
import collections
import torch
import faiss
import evaluate

from datasets import load_dataset
from transformers import AutoTokenizer, AutoModel
from transformers import AutoModelForQuestionAnswering
from transformers import TrainingArguments
from transformers import Trainer
from tqdm.auto import tqdm

device = torch.device("cuda") if torch.cuda.is_available() else torch.device("cpu")
```

Module: Retriever

❖ Step 2: Download dataset

```
1 DATASET_NAME = 'squad_v2'
2 raw_datasets = load_dataset(DATASET_NAME, split='train')
3 raw_datasets
```

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:88: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens),
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
```




```
Dataset({
    features: ['id', 'title', 'context', 'question', 'answers'],
    num_rows: 130319
})
```

Module: Retriever

❖ Step 2: Download dataset

```
1 raw_datasets = raw_datasets.filter(
2     lambda x: len(x['answers']['text']) > 0
3 )
4 raw_datasets
```

Filter: 100% 130319/130319 [00:04<00:00, 30967.75 examples/s]

```
Dataset({
    features: ['id', 'title', 'context', 'question', 'answers'],
    num_rows: 86821
})
```

context string · *lengths* 151 3.71k	question string · *lengths* 1 25.7k
Beyoncé Giselle Knowles-Carter (/bi…	When did Beyonce start becoming…
Beyoncé Giselle Knowles-Carter (/bi…	What areas did Beyonce compete in…
Beyoncé Giselle Knowles-Carter (/bi…	When did Beyonce leave Destiny's…
Beyoncé Giselle Knowles-Carter (/bi…	In what city and state did Beyonce…
Beyoncé Giselle Knowles-Carter (/bi…	In which decade did Beyonce become…
Beyoncé Giselle Knowles-Carter (/bi…	In what R&B group was she the lead…

Module: Retriever

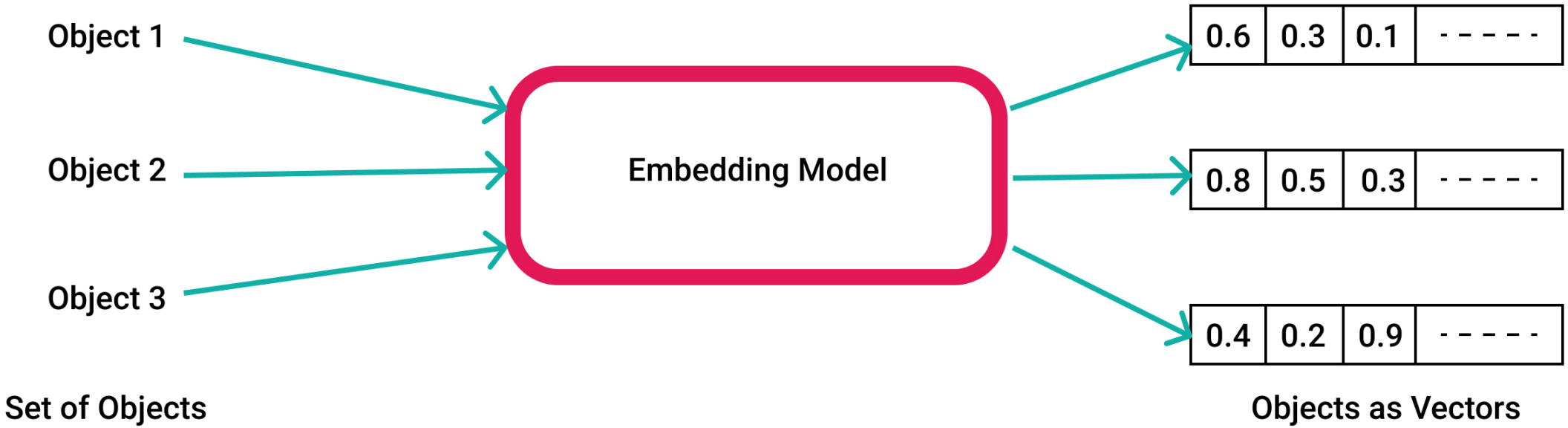
❖ Step 3: Initialize model

```
1 MODEL_NAME = "distilbert-base-uncased"
2 tokenizer = AutoTokenizer.from_pretrained(MODEL_NAME)
3 model = AutoModel.from_pretrained(MODEL_NAME).to(device)
```

```
tokenizer_config.json: 100% 28.0/28.0 [00:00<00:00, 649B/s]
config.json: 100% 483/483 [00:00<00:00, 14.3kB/s]
vocab.txt: 100% 232k/232k [00:00<00:00, 1.11MB/s]
tokenizer.json: 100% 466k/466k [00:00<00:00, 2.12MB/s]
model.safetensors: 100% 268M/268M [00:23<00:00, 14.3MB/s]
```

Module: Retriever

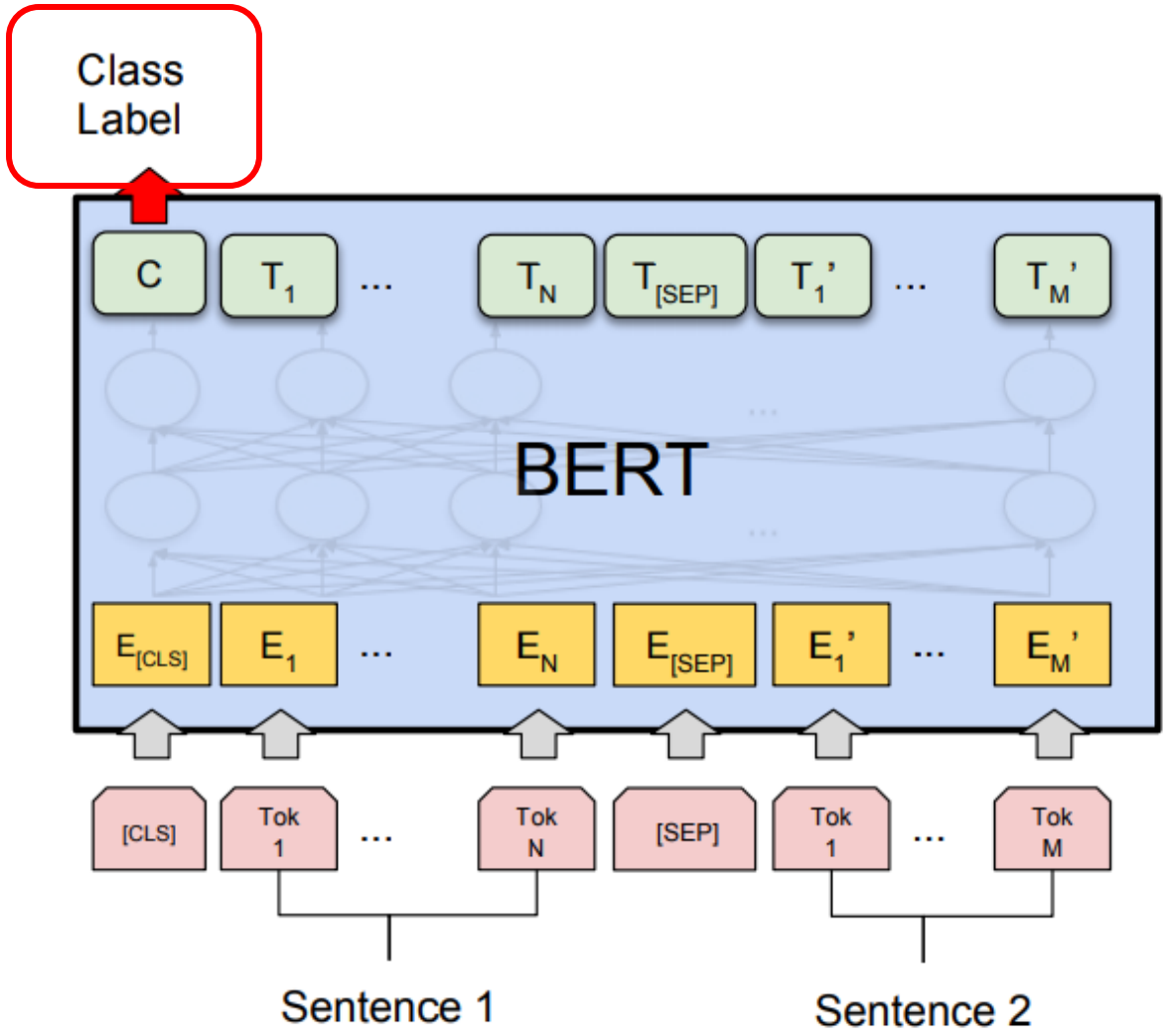
❖ Step 4: Create get embedding function



https://www.pinecone.io/learn/vector-embeddings/

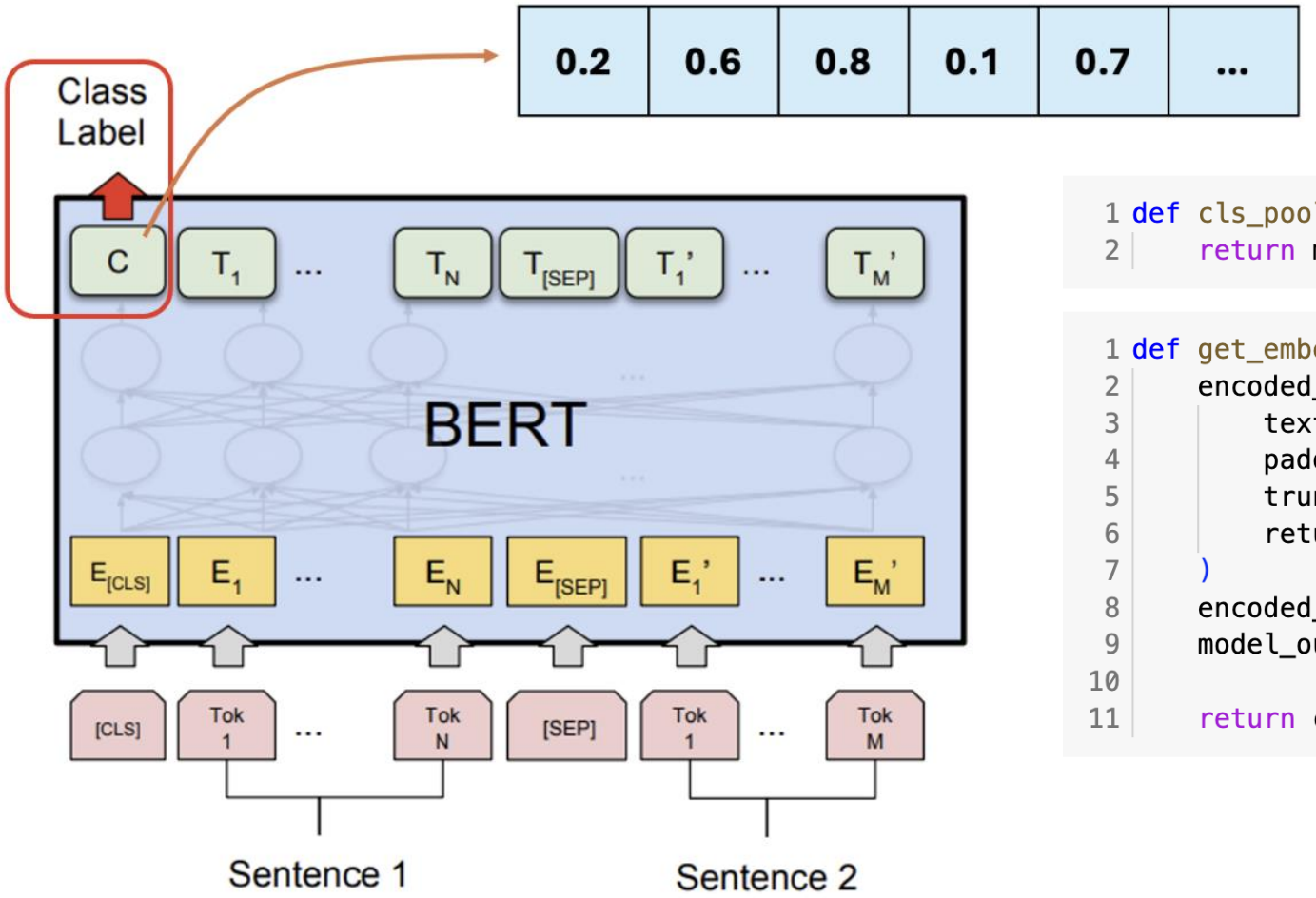
Module: Retriever

❖ Step 4: Create get embedding function



Module: Retriever

❖ Step 4: Create get embedding function

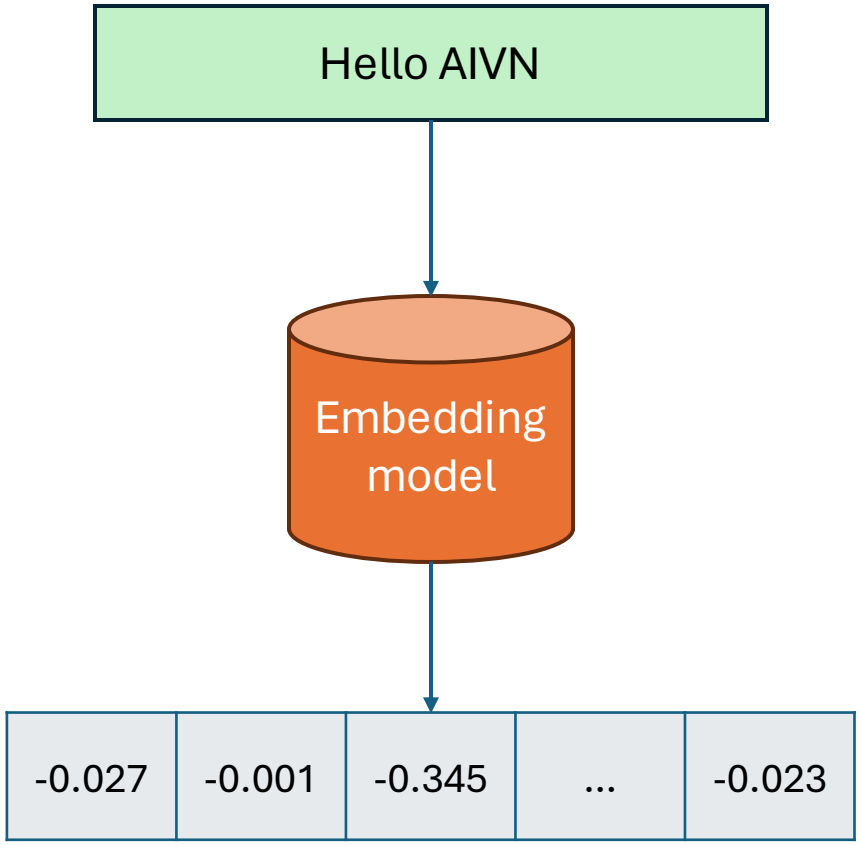



```python
def cls_pooling(model_output):
    return model_output.last_hidden_state[:, 0]
```

```python
def get_embeddings(text_list):
    encoded_input = tokenizer(
        text_list,
        padding=True,
        truncation=True,
        return_tensors='pt'
    )
    encoded_input = {k: v.to(device) for k, v in encoded_input.items()}
    model_output = model(**encoded_input)

    return cls_pooling(model_output)
```

Module: Retriever

❖ Step 4: Create get embedding function

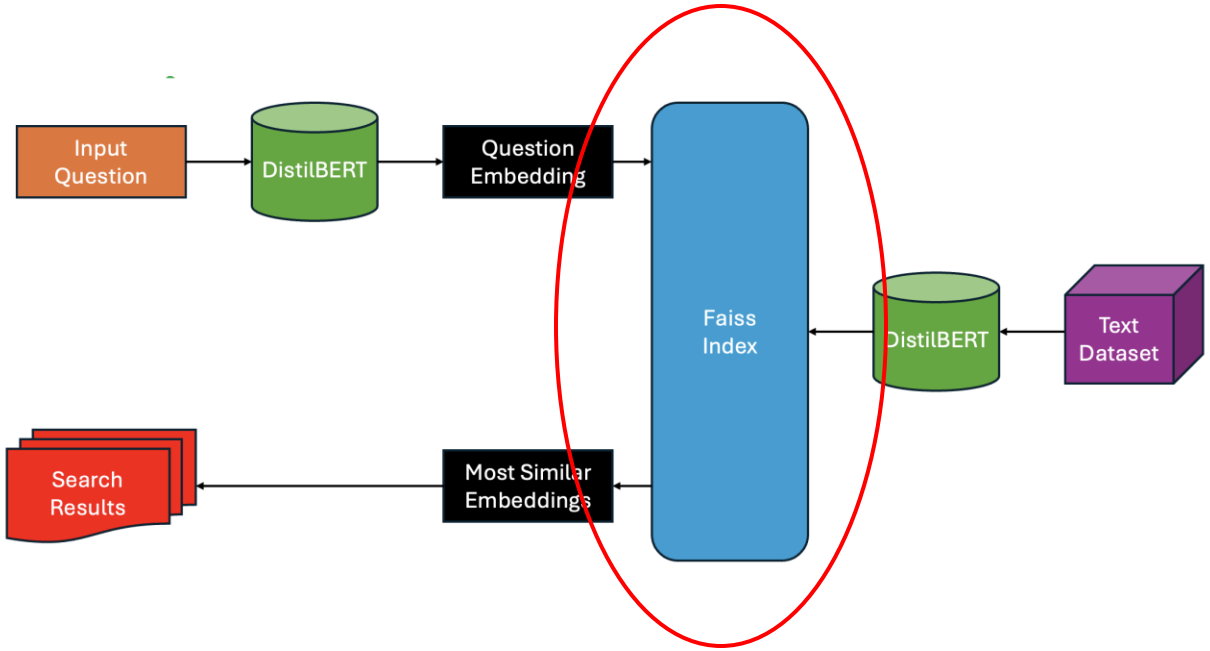



```
1 # Test functionality
2 embedding = get_embeddings(raw_datasets['question'][0])
3 embedding.shape
```

```
torch.Size([1, 768])
```

Module: Retriever

❖ Step 4: Create get embedding function




```python
# Convert to numpy array (required for HF Datasets)
EMBEDDING_COLUMN = 'question_embedding'
embeddings_dataset = raw_datasets.map(
    lambda x: {EMBEDDING_COLUMN: get_embeddings(x['question']).detach().cpu().numpy()[0]}
)
```

```
Map: 100% 86821/86821 [09:42<00:00, 142.06 examples/s]
```

```python
embeddings_dataset.add_faiss_index(column=EMBEDDING_COLUMN)
```

```
100% 87/87 [00:00<00:00, 109.40it/s]
Dataset({
    features: ['id', 'title', 'context', 'question', 'answers', 'question_embedding'],
    num_rows: 86821
})
```

Module: Retriever

❖ Step 5: Search

```
question = 'When did Beyonce start becoming popular?'

input_quest_embedding = get_embeddings([question]).cpu().detach().numpy()
input_quest_embedding.shape
```

```
(1, 768)
```

```
TOP_K = 5
scores, samples = embeddings_dataset.get_nearest_examples(
    EMBEDDING_COLUMN, input_quest_embedding, k=TOP_K
)
```

```
Top 1   Score: 0.0
Question: When did Beyonce start becoming popular?
Context: Beyoncé Giselle Knowles-Carter (/biːˈjɒnseɪ/ bee-YON-say) (bo
Answer: {'text': ['in the late 1990s'], 'answer_start': [269]}

Top 2   Score: 2.6135313510894775
Question: When did Beyoncé rise to fame?
Context: Beyoncé Giselle Knowles-Carter (/biːˈjɒnseɪ/ bee-YON-say) (bo
Answer: {'text': ['late 1990s'], 'answer_start': [276]}

Top 3   Score: 4.859482288360596
Question: When did Beyoncé release Formation?
Context: On February 6, 2016, one day before her performance at the Su
Answer: {'text': ['February 6, 2016'], 'answer_start': [3]}

Top 4   Score: 5.054221153259277
Question: In which decade did Beyonce become famous?
Context: Beyoncé Giselle Knowles-Carter (/biːˈjɒnseɪ/ bee-YON-say) (bo
Answer: {'text': ['late 1990s'], 'answer_start': [276]}

Top 5   Score: 5.170375347137451
Question: When did Beyonce begin her deals with name brands?
Context: The release of a video-game Starpower: Beyoncé was cancelled
Answer: {'text': ['since the age of 18'], 'answer_start': [433]}
```

Module: Retriever

❖ Step 5: Search with QA

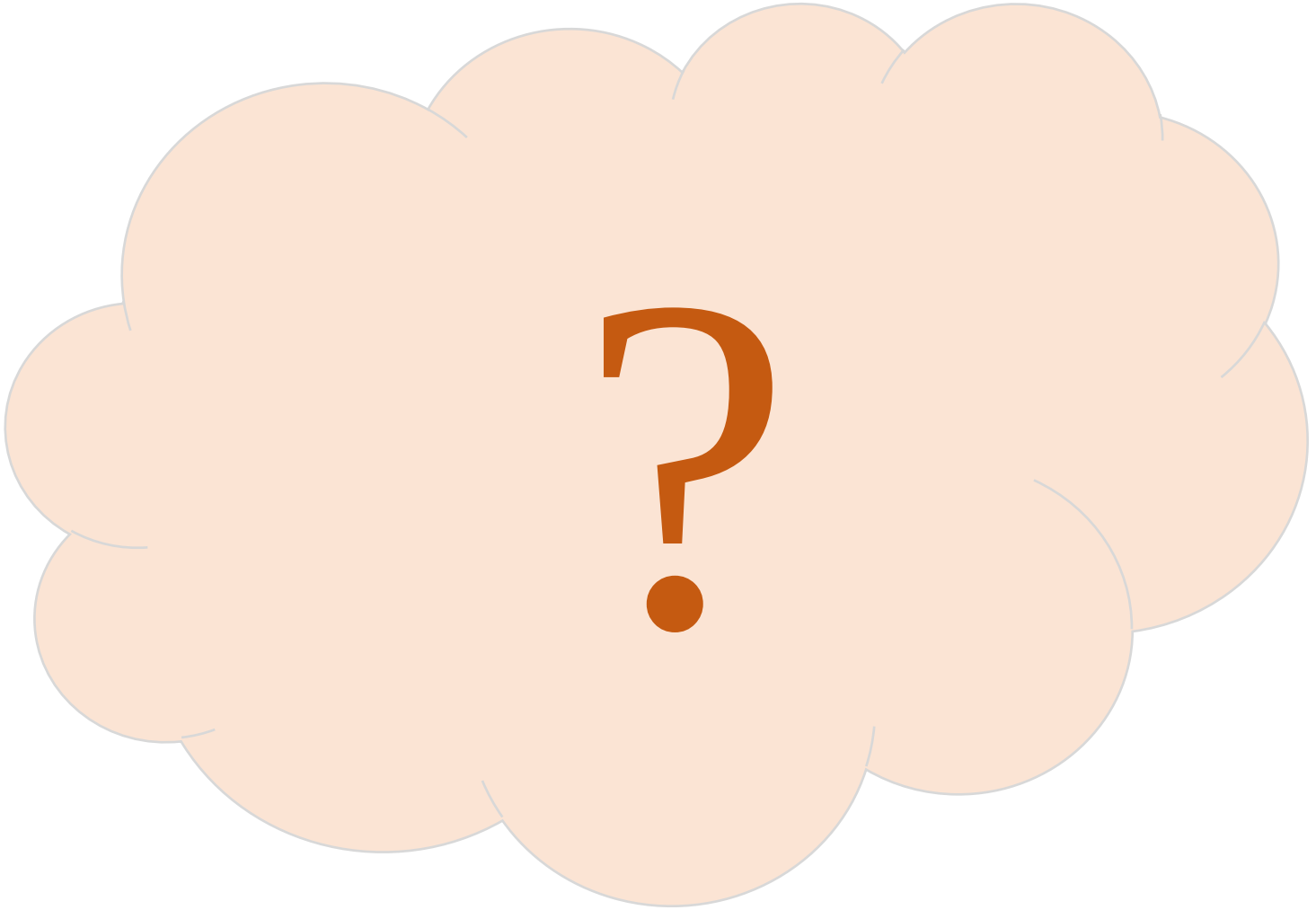
```
question = 'When did Beyonce start becoming popular?'

input_quest_embedding = get_embeddings([question]).cpu().detach().numpy()
input_quest_embedding.shape
```

```
(1, 768)
```

```
TOP_K = 3
for idx, input_question in enumerate(embeddings_dataset['question'][200:210]):
    input_quest_embedding = get_embeddings([input_question]).cpu().detach().numpy()
    scores, samples = embeddings_dataset.get_nearest_examples(
        EMBEDDING_COLUMN, input_quest_embedding, k=TOP_K
    )
    print(f'Question {idx + 1}: {input_question}')
    for jdx, score in enumerate(scores):
        print(f'Top {jdx + 1}\tScore: {score}')
        question = samples['question'][jdx]
        context = samples['context'][jdx]
        answer = pipe(
            question=question,
            context=context
        )
        print(f'Context: {context}')
        print(f'Answer: {answer}')
        print()
    print()
```

```
Question 1: How many awards was Beyonce nominated for at the 52nd Grammy Awards?
Top 1   Score: 0.0
Context: At the 52nd Annual Grammy Awards, Beyoncé received ten nominations, includ
Answer: {'score': 0.684537410736084, 'start': 51, 'end': 54, 'answer': 'ten'}

Top 2   Score: 2.3847267627716064
Context: At the 57th Annual Grammy Awards in February 2015, Beyoncé was nominated f
Answer: {'score': 0.3522382378578186, 'start': 77, 'end': 80, 'answer': 'six'}

Top 3   Score: 4.536745548248291
Context: Beyoncé's first solo recording was a feature on Jay Z's "'03 Bonnie & Clyd
Answer: {'score': 0.8402150273323059, 'start': 747, 'end': 751, 'answer': 'five'}


Question 2: Beyonce tied with which artist for most nominations by a female artist?
Top 1   Score: 0.0
Context: At the 52nd Annual Grammy Awards, Beyoncé received ten nominations, including Alb
Answer: {'score': 0.9431203603744507, 'start': 242, 'end': 253, 'answer': 'Lauryn Hill'}

Top 2   Score: 8.78997802734375
Context: At the 52nd Annual Grammy Awards, Beyoncé received ten nominations, including Alb
Answer: {'score': 0.6438630819320679, 'start': 517, 'end': 529, 'answer': 'Mariah Carey'}

Top 3   Score: 11.078895568847656
Context: At the 52nd Annual Grammy Awards, Beyoncé received ten nominations, including Alb
Answer: {'score': 0.9386846423149109, 'start': 242, 'end': 253, 'answer': 'Lauryn Hill'}
```

Question




Thank you!