Possible Answers :

0

Sub-Section Number :	7
Sub-Section Id :	640653146277
Question Shuffling Allowed :	Yes

Question Number : 293 Question Id : 640653993606 Question Type : SA
Correct Marks : 3

Question Label : Short Answer Question

Consider the embedding vector for a word, $x = [0.1, 0.2, -0.3, 0.4]$. Suppose the word is at position 2 in the given sentence. Add the corresponding position embedding p to the word embedding to get h, i.e. the sum of the elements in $h = x + p$. Use the fixed-sinusoidal position embedding vector calculated using the formula given below

$$PE(pos, 2i) = sin\left(\frac{pos}{10000^{2i/dmodel}}\right)$$

$$PE(pos, 2i+1) = cos\left(\frac{pos}{10000^{2i/dmodel}}\right)$$

What is $h[0] + h[1]$ i.e. sum of first two elements of h?

Response Type : Numeric
Evaluation Required For SA : Yes
Show Word Count : Yes
Answers Type : Range
Text Areas : PlainText
Possible Answers :

0.75 to 0.85

DLP

Section Id :	64065369324
Section Number :	15
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	14
Number of Questions to be attempted :	14
Section Marks :	50
Display Number Panel :	Yes
Section Negative Marks :	0

Group All Questions :	No
Enable Mark as Answered Mark for Review and Clear Response :	No
Section Maximum Duration :	0
Section Minimum Duration :	0
Section Time In :	Minutes
Maximum Instruction Time :	0
Sub-Section Number :	1
Sub-Section Id :	640653146278
Question Shuffling Allowed :	No

Question Number : 294 Question Id : 640653993622 Question Type : MCQ

Correct Marks : 0

Question Label : Multiple Choice Question

THIS IS QUESTION PAPER FOR THE SUBJECT "DEGREE LEVEL : DEEP LEARNING PRACTICE (COMPUTER BASED EXAM)"

ARE YOU SURE YOU HAVE TO WRITE EXAM FOR THIS SUBJECT?
CROSS CHECK YOUR HALL TICKET TO CONFIRM THE SUBJECTS TO BE WRITTEN.

(IF IT IS NOT THE CORRECT SUBJECT, PLS CHECK THE SECTION AT THE TOP FOR THE SUBJECTS REGISTERED BY YOU)

Options :

6406533355946. ✔ YES

6406533355947. ✖ NO

Sub-Section Number :	2
Sub-Section Id :	640653146279
Question Shuffling Allowed :	Yes

Question Number : 295 Question Id : 640653993630 Question Type : MCQ

Correct Marks : 4

Question Label : Multiple Choice Question

Suppose we want to feed the input to the model in the following format

```
[[CLS],token-1,token-2, token-3,[IMG],token-5,[REF], token-6, [EOS]]
```

Which of the following components in the tokenization pipeline help us achieve this in Hugging Face?

Options :

6406533355972. ✖ Pre-Tokenization

6406533355973. ✖ Normalization

6406533355974. ✔ Post Processor

after Tokenizer our output looks like
t1, t2, t3, t5, t6
the post processor adds the relevant tokens and then the

final input is sent to model.

6406533355975. ✖ Tokenization Algorithm

6406533355976. ✖ Decoder

Sub-Section Number : 3

Sub-Section Id : 640653146280

Question Shuffling Allowed : Yes

Question Number : 296 Question Id : 640653993627 Question Type : MCQ

Correct Marks : 3

Question Label : Multiple Choice Question

Choose the Hugging Face module that helps us train a tokenizer from scratch on a specific dataset.

Options :

pip install tokenizers

6406533355962. ✔ tokenizers

↳ it helps to build a tokenizer from scratch, build a vocab (depending on the needs) also, put spl. tokens depending on the models.

6406533355963. ✖ transformers

6406533355964. ✖ evaluate

6406533355965. ✖ Autotrain

6406533355966. ✖ Accelerate

Question Number : 297 Question Id : 640653993628 Question Type : MCQ

Correct Marks : 3

Question Label : Multiple Choice Question

A dataset contains 10 billion words (separated by a single white space). Suppose we use a pre-trained tokenizer that has a vocabulary of size 10,000 to tokenize the dataset, then the number of tokens in the dataset will always be greater than or equal to the number of words in the dataset. The statement is

dataset = $10^9 \times 10$

vocab = 10,000

- tokenizer splits the dataset into the tokens

high possibility → many words will not be present in the vocab.

∴, total no. of tokens ≥ total no. of words.

Options :

6406533355967. ✔ True

6406533355968. ✖ False

Question Number : 298 Question Id : 640653993629 Question Type : MCQ

Correct Marks : 3

Question Label : Multiple Choice Question

Which of the following tokenization algorithms can be applied to languages that do not have any word delimiters?

Options :

BPE & Word can't be used since we do not have word delimiters like space.

6406533355969. ✖ BPE (Byte Pair Encoding)

6406533355970. ✖ Wordpiece

6406533355971. ✔ Sentencepiece

Best possible opⁿ is sentence piece. it handles languages w/o explicit word boundaries.

Question Number : 299 Question Id : 640653993631 Question Type : MCQ

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the Wikipedia dataset scraped from the web that contains 2 billion words. A team decided to use the BPE tokenization algorithm with the varying vocabulary size from 2K to 52K, then the statement that increasing the number of merges will increase the size of the vocabulary is

Options :

6406533355977. ✔ True

6406533355978. ✖ False

6406533355979. ✖ Insufficient information

Question Number : 300 Question Id : 640653993633 Question Type : MCQ

Correct Marks : 3

Question Label : Multiple Choice Question

Suppose that we pre-train a Causal Language Model. Choose the data collator function from the Hugging Face library that is suitable for this task

Options :

6406533355987. ✖ DataCollator(tokenizer)

6406533355988. ✖ DefaultDataCollator(tokenizer)

6406533355989. ✔ DataCollatorForLanguageModelling(tokenizer,mlm=False)

6406533355990. ✖ DataCollatorForCausalLanguageModelling(tokenizer)

6406533355991. ✖ DataLoader(tokenizer)

Sub-Section Number :	4
Sub-Section Id :	640653146281
Question Shuffling Allowed :	Yes

Question Number : 301 Question Id : 640653993625 Question Type : MSQ

Correct Marks : 4 Max. Selectable Options : 0

Question Label : Multiple Select Question

The IMDB dataset has 25000 samples in the training split. It contains two columns, named, *text and label*. Consider the code snippet given below and choose all the correct statements

```
from datasets import load_dataset
imdb_dataset = load_dataset("stanfordnlp/imdb",split='train')

def get_num_words(example):
    num_words = len(example["text"].split())
    return {'num_words':num_words}

ds = imdb_dataset.map(get_num_words)
```

Options :

6406533355952. ✔ The *ds* variable contains three columns named:text,label,num_words

6406533355953. ✖ The *ds* variable contains two columns named:text,labels

6406533355954. ✖ The *ds* variable contains only one column named:num_words

6406533355955. ✖ Executing the last statement *imdb_dataset.map(get_num_words)* raises an error

6406533355956. ✖ Executing *len(example["text"].split())* raises an error

6406533355957. ✔ The total number of samples in the variable *ds* is 25000

Question Number : 302 Question Id : 640653993626 Question Type : MSQ
Correct Marks : 4 Max. Selectable Options : 0
Question Label : Multiple Select Question

Consider two datasets namely "ds1" and "ds2" . The structure of the dataset with the number of samples in each split is given below. Suppose we create a new dataset in the following ways. Assume necessary

```
DatasetDict({
    train: Dataset({
        features: ['text', 'label'],
        num_rows: 25000
    })
    test: Dataset({
        features: ['text', 'label'],
        num_rows: 25000
    })
    unsupervised: Dataset({
        features: ['text', 'label'],
        num_rows: 50000
    })
})
```

ds1

```
DatasetDict({
    train: Dataset({
        features: ['text', 'label'],
        num_rows: 8530
    })
    test: Dataset({
        features: ['text', 'label'],
        num_rows: 1066
    })
    unsupervised: Dataset({
        features: ['text', 'label'],
        num_rows: 1066
    })
})
```

ds2

imports and the statements are executed independently (i.e., an error in executing a statement does not affect the execution of other statements). Select all the correct statements.

```
ds3 = datasets.concatenate_datasets([ds1,ds2])
ds4 = datasets.concatenate_datasets([ds1['train'],ds2['train']])
ds5 = datasets.concatenate_datasets([ds1['train'],ds2['test']])
ds6 = datasets.concatenate_datasets(
            [ds1['train'],ds1['test'],
            ds2['train'],ds2['validation']])
```

Options :

6406533355958. ✖ The number of samples in each split of *ds3* is: {train:33,530,test:26,066,unsupervised:51,066}

6406533355959. ✔ The number of samples in *ds4* is 33,530

6406533355960. ✔ The number of samples in *ds5* is 26,066

6406533355961. ✔ The number of samples in *ds6* is 59,596

Sub-Section Number :	5
Sub-Section Id :	640653146282
Question Shuffling Allowed :	Yes

Question Number : 303 Question Id : 640653993623 Question Type : MSQ
Correct Marks : 3 Max. Selectable Options : 0
Question Label : Multiple Select Question

Consider downloading the IMDB dataset using the following code

```
from datasets import load_dataset
imdb_dataset = load_dataset("stanfordnlp/imdb",split='train[0:10000]')
```

choose all the correct statements.

Remember → if you don't use 'split' parameter it will be a datasetdict

Options :

6406533355948. ✔ The variable imdb_dataset would have 10000 samples

6406533355949. ✖ The data type of the variable imdb_dataset is *DatasetDict*

6406533355950. ✔ The data type of the variable imdb_dataset is *Dataset*

→ why? because you have already split

Question Number : 304 Question Id : 640653993632 Question Type : MSQ
Correct Marks : 3 Max. Selectable Options : 0
Question Label : Multiple Select Question

Which of the following attribute(s) is (are) returned by a tokenizer's `.encode_batch` method?

Options :

6406533355980. ✔ ids
6406533355981. ✔ tokens
6406533355982. ✔ offsets
6406533355983. ✔ attention_mask
6406533355984. ✔ special_token_mask
6406533355985. ✔ type_ids
6406533355986. ✖ vocab_size → more related to sth. else

it is related as a property of the ↑ tokenizer

Sub-Section Number : 6
Sub-Section Id : 640653146283
Question Shuffling Allowed : No

Question Id : 640653993634 Question Type : COMPREHENSION Sub Question Shuffling Allowed : No Group Comprehension Questions : No Question Pattern Type : NonMatrix
Question Numbers : (305 to 306)
Question Label : Comprehension

Here is a configuration of the GPTNeo model from the Hugging Face hub.


```
"bos_token_id": 50256,
"classifier_dropout": 0.1,
"embed_dropout": 0.0,
"eos_token_id": 50256,
"hidden_size": 2048,
"initializer_range": 0.02,
"intermediate_size": null,
"layer_norm_epsilon": 1e-05,
"max_position_embeddings": 2048,
"model_type": "gpt_neo",
"num_heads": 16,
"num_layers": 24,
"resid_dropout": 0.0,
"transformers_version": "4.44.2",
"use_cache": true,
"vocab_size": 50257,
"window_size": 256
```


Figure 1: GPTNeoConfig

Based on the above data, answer the given subquestions.

Sub questions

Question Number : 305 Question Id : 640653993635 Question Type : SA

Correct Marks : 3

Question Label : Short Answer Question

Enter the number of parameters in the embedding layer of the model in millions. For example, if the answer is 1234567. Then enter it as 1.23

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Range

Text Areas : PlainText

Possible Answers :

102 to 103

no. of param = vocab x hidden size

= 50257 x 2048

= 102.960 millions

Question Number : 306 Question Id : 640653993636 Question Type : SA

Correct Marks : 3

Question Label : Short Answer Question

Enter the context length.

this is same as hidden size

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Equal

Text Areas : PlainText

Possible Answers :

2048

Sub-Section Number :	7
Sub-Section Id :	640653146284
Question Shuffling Allowed :	No

Question Id : 640653993637 Question Type : COMPREHENSION Sub Question Shuffling Allowed : No Group Comprehension Questions : No Question Pattern Type : NonMatrix

Question Numbers : (307 to 308)

Question Label : Comprehension

Here is a set of training arguments used by the GPT-2 model that was pre-trained on a dataset that contains 10 billion tokens. The context length of the model is modified to 2048, the vocabulary size is 50,257 and the embedding dimension is 768. The length of all the samples in a batch is equal to the context length of the model.

```
training_args = TrainingArguments( output_dir='out',
                                   evaluation_strategy="steps",
                                   eval_steps=500,
                                   num_train_epochs=1,
                                   per_device_train_batch_size=16,
                                   per_device_eval_batch_size=16,
                                   tf32=True,
                                   gradient_accumulation_steps=2,
                                   adam_beta1=0.9,
                                   adam_beta2=0.999,
                                   learning_rate=2e-5,
                                   weight_decay=0.01,
                                   logging_dir='logs',
                                   logging_strategy="steps",
                                   logging_steps = 500,
                                   save_steps=5000,
                                   save_total_limit=20,
                                   report_to='wandb',
```

batch = 16
gradientsteps = 2
Context_len = 2048
no. of steps = 1000

Based on the above data, answer the given subquestions.

Sub questions Tokens after 1000 steps →

Question Number : 307 Question Id : 640653993638 Question Type : SA

Correct Marks : 5

Question Label : Short Answer Question

Enter the number of tokens (in millions) processed by the model after 1000 steps. Enter the

answer to 2 decimal places. For example, if your answer is 123456789, then enter it as 123.45.

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Range

Text Areas : PlainText

Possible Answers :

65.3 to 65.7

32×2048000

$65,536,000$

↓ in millions

65.536

Question Number : 308 Question Id : 640653993639 Question Type : SA

Correct Marks : 3

Question Label : Short Answer Question

How many steps does it take to complete one epoch of training? Enter the answer in thousands (round down to an integer). For example, if your answer is 1234567.89, then enter it as 1234567.

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Range

Text Areas : PlainText

Possible Answers :

152 to 153

no. of samples = total dataset / context size = $\frac{10 \times 10^9}{2048}$

$= 4,882,812$ samples

final batch size = per device batch size × gradient accumulation

$= 16 \times 2 = 32$

steps = samples / batch size

$= \frac{4882812}{32} = 152,600$

in thousands

152.6

Sub-Section Number : 8

Sub-Section Id : 640653146285

Question Shuffling Allowed : Yes

Question Number : 309 Question Id : 640653993624 Question Type : SA

Correct Marks : 3

Question Label : Short Answer Question

The IMDB dataset has 25000 samples in the training split.
How many samples are there in the variable *small_imdb_ds* after executing the code below? If you think it raises an error, then enter -1.

```
from datasets import load_dataset
imdb_dataset = load_dataset("stanfordnlp/imdb",split='train[0:10000]')
small_imdb_ds = imdb_dataset.select(range(0,1000,2))
```

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Equal

Text Areas : PlainText

Possible Answers :

0 → 1000, 2

imdb = 10000 samples

$\frac{1000}{2} = 500$

500