

finetuned-baaibge-large

February 26, 2025

```
[ ]: pip install datasets
```

```
Requirement already satisfied: datasets in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (3.2.0)
Requirement already satisfied: filelock in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from datasets)
(3.16.1)
Requirement already satisfied: numpy>=1.17 in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from datasets)
(2.2.1)
Requirement already satisfied: pyarrow>=15.0.0 in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from datasets)
(18.1.0)
Requirement already satisfied: dill<0.3.9,>=0.3.0 in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from datasets)
(0.3.8)
Requirement already satisfied: pandas in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from datasets)
(2.2.3)
Requirement already satisfied: requests>=2.32.2 in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from datasets)
(2.32.3)
Requirement already satisfied: tqdm>=4.66.3 in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from datasets)
(4.67.1)
Requirement already satisfied: xxhash in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from datasets)
(3.5.0)
Requirement already satisfied: multiprocessing<0.70.17 in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from datasets)
(0.70.16)
Requirement already satisfied: fsspec<=2024.9.0,>=2023.1.0 in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from
fsspec[http]<=2024.9.0,>=2023.1.0->datasets) (2024.9.0)
Requirement already satisfied: aiohttp in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from datasets)
(3.11.11)
Requirement already satisfied: huggingface-hub>=0.23.0 in
```

c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from datasets) (0.27.0)

Requirement already satisfied: packaging in c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from datasets) (24.2)

Requirement already satisfied: pyyaml>=5.1 in c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from datasets) (6.0.2)

Requirement already satisfied: aiohappyeyeballs>=2.3.0 in c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from aiohttp->datasets) (2.4.4)

Requirement already satisfied: aiosignal>=1.1.2 in c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from aiohttp->datasets) (1.3.2)

Requirement already satisfied: async-timeout<6.0,>=4.0 in c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from aiohttp->datasets) (5.0.1)

Requirement already satisfied: attrs>=17.3.0 in c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from aiohttp->datasets) (24.3.0)

Requirement already satisfied: frozenlist>=1.1.1 in c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from aiohttp->datasets) (1.5.0)

Requirement already satisfied: multidict<7.0,>=4.5 in c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from aiohttp->datasets) (6.1.0)

Requirement already satisfied: propcache>=0.2.0 in c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from aiohttp->datasets) (0.2.1)

Requirement already satisfied: yarll<2.0,>=1.17.0 in c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from aiohttp->datasets) (1.18.3)

Requirement already satisfied: typing-extensions>=3.7.4.3 in c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from huggingface-hub>=0.23.0->datasets) (4.12.2)

Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from requests>=2.32.2->datasets) (3.4.1)

Requirement already satisfied: idna<4,>=2.5 in c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from requests>=2.32.2->datasets) (3.10)

Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from requests>=2.32.2->datasets) (2.3.0)

Requirement already satisfied: certifi>=2017.4.17 in c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from requests>=2.32.2->datasets) (2024.12.14)

Requirement already satisfied: colorama in

```

c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from
tqdm>=4.66.3->datasets) (0.4.6)
Requirement already satisfied: python-dateutil>=2.8.2 in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from
pandas->datasets) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from
pandas->datasets) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from
pandas->datasets) (2024.2)
Requirement already satisfied: six>=1.5 in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from python-
dateutil>=2.8.2->pandas->datasets) (1.17.0)
Note: you may need to restart the kernel to use updated packages.

```

```
[ ]: !pip install transformers==4.45.2 sentence-transformers==3.1.1
```

```

Requirement already satisfied: transformers==4.45.2 in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (4.45.2)
Requirement already satisfied: sentence-transformers==3.1.1 in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (3.1.1)
Requirement already satisfied: filelock in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from
transformers==4.45.2) (3.16.1)
Requirement already satisfied: huggingface-hub<1.0,>=0.23.2 in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from
transformers==4.45.2) (0.27.0)
Requirement already satisfied: numpy>=1.17 in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from
transformers==4.45.2) (2.2.1)
Requirement already satisfied: packaging>=20.0 in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from
transformers==4.45.2) (24.2)
Requirement already satisfied: pyyaml>=5.1 in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from
transformers==4.45.2) (6.0.2)
Requirement already satisfied: regex!=2019.12.17 in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from
transformers==4.45.2) (2024.11.6)
Requirement already satisfied: requests in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from
transformers==4.45.2) (2.32.3)
Requirement already satisfied: safetensors>=0.4.1 in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from
transformers==4.45.2) (0.5.0)
Requirement already satisfied: tokenizers<0.21,>=0.20 in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from

```

transformers==4.45.2) (0.20.3)
Requirement already satisfied: tqdm>=4.27 in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from
transformers==4.45.2) (4.67.1)
Requirement already satisfied: torch>=1.11.0 in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from sentence-
transformers==3.1.1) (2.5.0)
Requirement already satisfied: scikit-learn in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from sentence-
transformers==3.1.1) (1.6.0)
Requirement already satisfied: scipy in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from sentence-
transformers==3.1.1) (1.15.0)
Requirement already satisfied: Pillow in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from sentence-
transformers==3.1.1) (10.4.0)
Requirement already satisfied: fsspec>=2023.5.0 in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from huggingface-
hub<1.0,>=0.23.2->transformers==4.45.2) (2024.9.0)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from huggingface-
hub<1.0,>=0.23.2->transformers==4.45.2) (4.12.2)
Requirement already satisfied: networkx in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from
torch>=1.11.0->sentence-transformers==3.1.1) (3.4.2)
Requirement already satisfied: jinja2 in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from
torch>=1.11.0->sentence-transformers==3.1.1) (3.1.5)
Requirement already satisfied: sympy==1.13.1 in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from
torch>=1.11.0->sentence-transformers==3.1.1) (1.13.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from
sympy==1.13.1->torch>=1.11.0->sentence-transformers==3.1.1) (1.3.0)
Requirement already satisfied: colorama in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from
tqdm>=4.27->transformers==4.45.2) (0.4.6)
Requirement already satisfied: charset_normalizer<4,>=2 in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from
requests->transformers==4.45.2) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from
requests->transformers==4.45.2) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from
requests->transformers==4.45.2) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from

```
requests->transformers==4.45.2) (2024.12.14)
Requirement already satisfied: joblib>=1.2.0 in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from scikit-
learn->sentence-transformers==3.1.1) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from scikit-
learn->sentence-transformers==3.1.1) (3.5.0)
Requirement already satisfied: MarkupSafe>=2.0 in
c:\users\raviksh\miniconda3\envs\chat_arena\lib\site-packages (from
jinja2->torch>=1.11.0->sentence-transformers==3.1.1) (3.0.2)
```

```
[ ]: from datasets import load_dataset
      from sentence_transformers import (
          SentenceTransformer,
          SentenceTransformerTrainer,
      )
      from sentence_transformers.losses import MultipleNegativesRankingLoss
      from sentence_transformers.evaluation import TripletEvaluator
```

```
c:\Users\Raviksh\miniconda3\envs\chat_arena\lib\site-packages\tqdm\auto.py:21:
TqdmWarning: IProgress not found. Please update jupyter and ipywidgets. See
https://ipywidgets.readthedocs.io/en/stable/user_install.html
    from .autonotebook import tqdm as notebook_tqdm
```

```
[ ]: # for my custom dataset

import pandas as pd

train = pd.read_parquet('./train.parquet')
```

```
[ ]: import pandas as pd

# Initialize an empty list to store rows for the DataFrame
data = []

# Loop through the dataset to create triplets
n_examples = len(train)
for i in range(n_examples):
    prompt = train.iloc[i]['prompt']
    response_a = train.iloc[i]['response_a']
    response_b = train.iloc[i]['response_b']
    label = train.iloc[i]['winner']

    if label == 'model_a':
        data.append({
            "anchor": prompt,
            "positive": response_a,
```

```

        "negative": response_b
    })
    elif label == 'model_b':
        data.append({
            "anchor": prompt,
            "positive": response_b,
            "negative": response_a
        })
    else:
        print(f"Error in row {i}: Unexpected label value '{label}'")

# Convert the list of dictionaries into a DataFrame
triplets_df = pd.DataFrame(data)

# Display the first few rows of the DataFrame
print(triplets_df.head())

```

```

                                anchor \
0                                vieš po Slovensky?
1 You will be given a piece of news. Analyze it ...
2 Dört basamaklı, rakamları birbirinden ve sıfır...
3         3      Cabaletta Bio (CABA), Rocket Ph...
4                                Please be boring

```

```

                                positive \
0      Áno, hovorím po slovensky. Ako vám môžem pomôcť?
1 Let's break down the news and analyze it accor...
2 Bu soruyu çözmek için, verilen koşulları adım ...
3         3      ...
4 Alright, I'll be as boring as possible.\n\nTod...

```

```

                                negative
0      Áno, ved' som tu! Môžem ti pomôcť s otázkami al...
1 ``json\n{\n  "contains_orgs": true,\n  "orgs"...
2 Bu problemi adım adım çözelim:\n\n1) ABCD - DC...
3         .      AI      ,      ...
4 Understood. Here is a straightforward, unadorn...

```

```

[ ]: from sklearn.model_selection import train_test_split

# Split the DataFrame into train and test sets
train_df, test_df = train_test_split(triplets_df, test_size=0.2,
    ↪ random_state=42)

# Display the first few rows of each split
print("Train Set:")
print(train_df.head())

```

```
print("\nTest Set:")
print(test_df.head())
```

Train Set:

```

                                anchor \
10277 Can you group these words in groups of 4: HAT ...
1696 ...
7416 rewrite: \nFuel your tech adventures with the ...
11168 do a summary of "STATISTICAL PARADISES AND PAR...
6865 Viết 1 đoạn văn giới thiệu về thành phố Long X...
```

```

                                positive \
10277 Here are the grouped words in categories of 4:...
1696 ...
7416 **Elevate Your iPhone 16 Experience with Unmat...
11168 ## Statistical Paradises and Paradoxes in Big ...
6865 Thành phố Long Xuyên, thủ phủ của tỉnh An Gian...
```

```

                                negative
10277 Here's one possible grouping of the words into...
1696 ...
7416 Embark on your technological journey with the ...
11168 ## Statistical Paradises & Paradoxes in Big D...
6865 Thành phố Long Xuyên, thuộc tỉnh An Giang, là ...
```

Test Set:

```

                                anchor \
28966 c++
36858 Case reports (entirely fictional and only for ...
44109 ...
1219 ...
29884 You are the leadership team of Anderson Univer...
```

```

                                positive \
28966 C++ ...
36858 ### Easy Cases\n\n#### Case 1: 40-year-old Fem...
44109 Mac , ...
1219 ...
29884 **Strategic Plan for Anderson University**\n\n...
```

```

                                negative
28966 ``cpp\n#include <iostream>\n\nusing namespace...
36858 ### Evaluation of Theodosian Surgery for Ficti...
44109 MacOS ...
1219 ...
29884 To address the challenges and leverage the opp...
```

```
[ ]: from sentence_transformers import SentenceTransformer
model = SentenceTransformer("C:
    ↳\\Users\\Raviksh\\Downloads\\finetune_sentence_transformer\\BAAI_bge_large_en_v1.
    ↳5")

[ ]: loss = MultipleNegativesRankingLoss(model)

[ ]: from datasets import Dataset
tr = Dataset.from_dict({
    "anchor": list(train_df['anchor']),
    "positive": list(train_df['positive']),
    "negative": list(train_df['negative'])
})

te = Dataset.from_dict({
    "anchor": list(test_df['anchor']),
    "positive": list(test_df['positive']),
    "negative": list(test_df['negative'])
})

[ ]: train_dataset = tr # Your training dataset
eval_dataset = te # Your evaluation dataset

[ ]: from sentence_transformers import SentenceTransformer, InputExample, losses,
    ↳SentenceTransformerTrainer, SentenceTransformerTrainingArguments
from torch.optim import AdamW
#from transformers import get_linear_schedule_with_warmup

optimizer = AdamW(model.parameters(), lr=6e-5) # Adjust the learning rate as
    ↳needed

[ ]: # num_training_steps = len(train_dataset) * 5 # Assuming 5 epochs
# num_warmup_steps = int(0.1 * num_training_steps) # Warmup steps (10% of
    ↳total steps)
# scheduler = get_linear_schedule_with_warmup(optimizer,
    ↳num_warmup_steps=num_warmup_steps, num_training_steps=num_training_steps)

[ ]: from transformers import get_cosine_schedule_with_warmup
batch_size = 8
num_epochs = 5
num_training_steps = (len(train_dataset) // batch_size) * num_epochs # Total
    ↳steps
num_warmup_steps = int(0.1 * num_training_steps) # 10% warm-up (adjustable)

scheduler = get_cosine_schedule_with_warmup(
    optimizer,
    num_warmup_steps=num_warmup_steps,
```



```

    num_training_steps=num_training_steps
)

```

```

[ ]: training_args = SentenceTransformerTrainingArguments(
    output_dir="./results",           # Directory to save results
    num_train_epochs=5,               # Number of epochs
    per_device_train_batch_size=8,    # Batch size for training
    logging_dir="./logs",            # Directory for storing logs
    evaluation_strategy="epoch",      # Evaluate at the end of each epoch
    save_strategy="epoch",           # Save model at the end of each epoch
    save_total_limit=5,              # Keep only 2 checkpoints
    metric_for_best_model="eval_loss", # Use eval loss to select the best
    ↪model
    greater_is_better=False,         # Lower eval loss is better
    load_best_model_at_end=True,     # Load the best model after training
)

```

```

c:\Users\Raviksh\miniconda3\envs\chat_arena\lib\site-
packages\transformers\training_args.py:1545: FutureWarning:
`evaluation_strategy` is deprecated and will be removed in version 4.46 of
Transformers. Use `eval_strategy` instead
    warnings.warn(

```

```

[ ]: trainer = SentenceTransformerTrainer(
    model=model,
    args=training_args,
    train_dataset=train_dataset,
    eval_dataset=eval_dataset,
    loss=loss,
    optimizers=(optimizer, scheduler) # Pass the custom optimizer and scheduler
)

```

```

[ ]: trainer.train()

```

```

2%|          | 500/24220 [25:35<20:01:53, 3.04s/it]
{'loss': 0.9917, 'grad_norm': 14.937872886657715, 'learning_rate':
1.2391573729863692e-05, 'epoch': 0.1}

4%|          | 1000/24220 [51:27<19:57:41, 3.09s/it]
{'loss': 0.844, 'grad_norm': 8.561187744140625, 'learning_rate':
2.4783147459727385e-05, 'epoch': 0.21}

6%|          | 1500/24220 [1:17:16<19:32:44, 3.10s/it]
{'loss': 0.8357, 'grad_norm': 12.466208457946777, 'learning_rate':
3.717472118959108e-05, 'epoch': 0.31}

8%|          | 2000/24220 [1:43:07<19:08:58, 3.10s/it]

```

```

{'loss': 0.8288, 'grad_norm': 10.600460052490234, 'learning_rate':
4.956629491945477e-05, 'epoch': 0.41}

10%|          | 2500/24220 [2:08:52<18:36:53, 3.09s/it]

{'loss': 0.8637, 'grad_norm': 7.649990081787109, 'learning_rate':
5.999805478952071e-05, 'epoch': 0.52}

12%|          | 3000/24220 [2:34:32<18:09:46, 3.08s/it]

{'loss': 0.8468, 'grad_norm': 6.945950031280518, 'learning_rate':
5.989557076021444e-05, 'epoch': 0.62}

14%|          | 3500/24220 [3:00:10<17:37:56, 3.06s/it]

{'loss': 0.8498, 'grad_norm': 6.96848726272583, 'learning_rate':
5.9637853499770356e-05, 'epoch': 0.72}

17%|          | 4000/24220 [3:25:44<17:09:17, 3.05s/it]

{'loss': 0.8556, 'grad_norm': 7.884798049926758, 'learning_rate':
5.9226241209200886e-05, 'epoch': 0.83}

19%|          | 4500/24220 [3:51:19<16:52:14, 3.08s/it]

{'loss': 0.8614, 'grad_norm': 4.905048370361328, 'learning_rate':
5.866287119193313e-05, 'epoch': 0.93}

20%|          | 4844/24220 [4:12:13<16:20:32, 3.04s/it]

{'eval_loss': 0.8219038248062134, 'eval_runtime': 199.1762,
'eval_samples_per_second': 48.64, 'eval_steps_per_second': 6.08, 'epoch': 1.0}

21%|          | 5000/24220 [4:20:19<16:23:56, 3.07s/it]

{'loss': 0.8081, 'grad_norm': 10.822230339050293, 'learning_rate':
5.7950668755826526e-05, 'epoch': 1.03}

23%|          | 5500/24220 [4:45:53<16:01:26, 3.08s/it]

{'loss': 0.9177, 'grad_norm': 25.40943145751953, 'learning_rate':
5.7093332023464756e-05, 'epoch': 1.14}

25%|          | 6000/24220 [5:11:26<15:35:27, 3.08s/it]

{'loss': 0.9405, 'grad_norm': 39.461612701416016, 'learning_rate':
5.609531272959494e-05, 'epoch': 1.24}

27%|          | 6500/24220 [5:37:02<15:04:46, 3.06s/it]

{'loss': 1.6099, 'grad_norm': 16.350177764892578, 'learning_rate':
5.496179310542342e-05, 'epoch': 1.34}

29%|          | 7000/24220 [6:02:36<14:43:15, 3.08s/it]

{'loss': 1.3897, 'grad_norm': 12.730401039123535, 'learning_rate':
5.3698658969797064e-05, 'epoch': 1.45}

```

31%| | 7500/24220 [6:28:12<14:19:37, 3.08s/it]
{'loss': 0.8337, 'grad_norm': 13.508018493652344, 'learning_rate':
5.2312469166994395e-05, 'epoch': 1.55}

33%| | 8000/24220 [6:53:46<13:56:16, 3.09s/it]
{'loss': 1.3375, 'grad_norm': 6.692089080810547, 'learning_rate':
5.081042150982137e-05, 'epoch': 1.65}

35%| | 8500/24220 [7:19:20<13:24:37, 3.07s/it]
{'loss': 0.7632, 'grad_norm': 6.210781574249268, 'learning_rate':
4.9200315404852826e-05, 'epoch': 1.75}

37%| | 9000/24220 [7:44:57<12:51:58, 3.04s/it]
{'loss': 0.748, 'grad_norm': 9.766450881958008, 'learning_rate':
4.749051135388865e-05, 'epoch': 1.86}

39%| | 9500/24220 [8:10:31<12:16:45, 3.00s/it]
{'loss': 0.7391, 'grad_norm': 16.904808044433594, 'learning_rate':
4.568988754191393e-05, 'epoch': 1.96}

40%| | 9688/24220 [8:23:09<12:07:01, 3.00s/it]
{'eval_loss': 0.8427066206932068, 'eval_runtime': 196.5122,
'eval_samples_per_second': 49.3, 'eval_steps_per_second': 6.162, 'epoch': 2.0}

41%| | 10000/24220 [8:38:52<11:47:50, 2.99s/it]
{'loss': 0.9193, 'grad_norm': 3.257291555404663, 'learning_rate':
4.380779373698084e-05, 'epoch': 2.06}

43%| | 10500/24220 [9:03:48<11:20:45, 2.98s/it]
{'loss': 0.7334, 'grad_norm': 16.266300201416016, 'learning_rate':
4.185400274138806e-05, 'epoch': 2.17}

45%| | 11000/24220 [9:28:45<11:00:38, 3.00s/it]
{'loss': 0.6299, 'grad_norm': 11.509878158569336, 'learning_rate':
3.9838659646247754e-05, 'epoch': 2.27}

47%| | 11500/24220 [9:53:42<10:32:02, 2.98s/it]
{'loss': 0.6552, 'grad_norm': 5.4828314781188965, 'learning_rate':
3.7772229152937264e-05, 'epoch': 2.37}

50%| | 12000/24220 [10:18:41<10:09:55, 2.99s/it]
{'loss': 0.6462, 'grad_norm': 12.842888832092285, 'learning_rate':
3.566544123496908e-05, 'epoch': 2.48}

52%| | 12500/24220 [10:43:39<9:47:13, 3.01s/it]

```

{'loss': 0.8843, 'grad_norm': 11.981834411621094, 'learning_rate':
3.352923542243117e-05, 'epoch': 2.58}

54%|          | 13000/24220 [11:08:35<9:22:55, 3.01s/it]

{'loss': 0.6229, 'grad_norm': 10.547221183776855, 'learning_rate':
3.1374703998301354e-05, 'epoch': 2.68}

56%|          | 13500/24220 [11:33:32<8:54:29, 2.99s/it]

{'loss': 0.6116, 'grad_norm': 8.999836921691895, 'learning_rate':
2.9213034401589843e-05, 'epoch': 2.79}

58%|          | 14000/24220 [11:58:32<8:31:55, 3.01s/it]

{'loss': 0.6055, 'grad_norm': 10.036020278930664, 'learning_rate':
2.7055451136382616e-05, 'epoch': 2.89}

60%|          | 14500/24220 [12:23:33<8:06:18, 3.00s/it]

{'loss': 0.6195, 'grad_norm': 11.372234344482422, 'learning_rate':
2.491315748842366e-05, 'epoch': 2.99}

60%|          | 14532/24220 [12:28:25<8:02:41, 2.99s/it]

{'eval_loss': 0.8828319311141968, 'eval_runtime': 197.6183,
'eval_samples_per_second': 49.024, 'eval_steps_per_second': 6.128, 'epoch': 3.0}

62%|          | 15000/24220 [12:52:00<7:40:06, 2.99s/it]

{'loss': 0.4185, 'grad_norm': 16.031789779663086, 'learning_rate':
2.2797277351873956e-05, 'epoch': 3.1}

64%|          | 15500/24220 [13:17:00<7:15:21, 3.00s/it]

{'loss': 0.4409, 'grad_norm': 854.455078125, 'learning_rate':
2.0718797468312442e-05, 'epoch': 3.2}

66%|          | 16000/24220 [13:42:00<6:51:31, 3.00s/it]

{'loss': 0.4271, 'grad_norm': 8.762587547302246, 'learning_rate':
1.8688510377903952e-05, 'epoch': 3.3}

68%|          | 16500/24220 [14:07:01<6:29:10, 3.02s/it]

{'loss': 0.4036, 'grad_norm': 32.71707534790039, 'learning_rate':
1.6716958378961136e-05, 'epoch': 3.41}

70%|          | 17000/24220 [14:32:00<6:00:18, 2.99s/it]

{'loss': 0.4106, 'grad_norm': 10.449934005737305, 'learning_rate':
1.4814378786891255e-05, 'epoch': 3.51}

72%|          | 17500/24220 [14:57:01<5:36:18, 3.00s/it]

{'loss': 0.386, 'grad_norm': 18.833757400512695, 'learning_rate':
1.2990650776771533e-05, 'epoch': 3.61}

```

74%| | 18000/24220 [15:22:03<5:11:18, 3.00s/it]
{'loss': 0.4274, 'grad_norm': 81.62264251708984, 'learning_rate':
1.1255244085573986e-05, 'epoch': 3.72}

76%| | 18500/24220 [15:50:05<4:45:37, 3.00s/it]
{'loss': 0.4126, 'grad_norm': 7.445626258850098, 'learning_rate':
9.617169840404147e-06, 'epoch': 3.82}

78%| | 19000/24220 [16:14:58<4:20:58, 3.00s/it]
{'loss': 0.3955, 'grad_norm': 19.070829391479492, 'learning_rate':
8.084933768078922e-06, 'epoch': 3.92}

80%| | 19376/24220 [16:39:44<8:02:23, 5.98s/it]
{'eval_loss': 0.9970029592514038, 'eval_runtime': 261.4252,
'eval_samples_per_second': 37.058, 'eval_steps_per_second': 4.632, 'epoch': 4.0}

81%| | 19500/24220 [16:46:02<3:55:21, 2.99s/it]
{'loss': 0.3575, 'grad_norm': 12.04888916015625, 'learning_rate':
6.666492029003463e-06, 'epoch': 4.03}

83%| | 20000/24220 [17:10:57<3:30:31, 2.99s/it]
{'loss': 0.2405, 'grad_norm': 8.26357364654541, 'learning_rate':
5.369209904680655e-06, 'epoch': 4.13}

85%| | 20500/24220 [17:35:52<3:05:18, 2.99s/it]
{'loss': 0.2524, 'grad_norm': 13.261795043945312, 'learning_rate':
4.199823553368554e-06, 'epoch': 4.23}

87%| | 21000/24220 [18:11:09<4:01:57, 4.51s/it]
{'loss': 0.2345, 'grad_norm': 5.840431213378906, 'learning_rate':
3.164405032470541e-06, 'epoch': 4.34}

89%| | 21500/24220 [18:45:15<3:07:29, 4.14s/it]
{'loss': 0.2395, 'grad_norm': 13.854308128356934, 'learning_rate':
2.2683307692797293e-06, 'epoch': 4.44}

91%| | 22000/24220 [19:20:54<2:37:19, 4.25s/it]
{'loss': 0.2404, 'grad_norm': 10.368667602539062, 'learning_rate':
1.5162536437939778e-06, 'epoch': 4.54}

93%| | 22500/24220 [19:58:04<2:06:40, 4.42s/it]
{'loss': 0.2318, 'grad_norm': 32.48417282104492, 'learning_rate':
9.120788285617776e-07, 'epoch': 4.64}

95%| | 23000/24220 [20:35:16<1:31:25, 4.50s/it]

```
{'loss': 0.2395, 'grad_norm': 1.8681293725967407, 'learning_rate':
4.589435110110762e-07, 'epoch': 4.75}

97%|      | 23500/24220 [21:11:04<50:43, 4.23s/it]

{'loss': 0.2313, 'grad_norm': 20.09564971923828, 'learning_rate':
1.5920060355322786e-07, 'epoch': 4.85}

99%|      | 24000/24220 [21:46:43<15:47, 4.31s/it]

{'loss': 0.2339, 'grad_norm': 7.399160385131836, 'learning_rate':
1.4406526047496816e-08, 'epoch': 4.95}

100%|     | 24220/24220 [22:16:18<00:00, 3.91s/it]

{'eval_loss': 1.0698752403259277, 'eval_runtime': 846.1707,
'eval_samples_per_second': 11.449, 'eval_steps_per_second': 1.431, 'epoch': 5.0}

100%|     | 24220/24220 [22:16:44<00:00, 3.31s/it]

{'train_runtime': 80204.4935, 'train_samples_per_second': 2.416,
'train_steps_per_second': 0.302, 'train_loss': 0.6424064959109469, 'epoch': 5.0}
```

```
[ ]: TrainOutput(global_step=24220, training_loss=0.6424064959109469,
metrics={'train_runtime': 80204.4935, 'train_samples_per_second': 2.416,
'train_steps_per_second': 0.302, 'total_flos': 0.0, 'train_loss':
0.6424064959109469, 'epoch': 5.0})
```

1 best model i got using 19536 checkpoint i will retrain it for 3 epoch

```
[ ]: from sentence_transformers import SentenceTransformer
checkpoint_path = "C:
↪\\Users\\Raviksh\\Downloads\\finetune_sentence_transformer\\s_tr\\checkpoint-19376"
model = SentenceTransformer(checkpoint_path)
```

```
[ ]: loss = MultipleNegativesRankingLoss(model)
```

```
[ ]: from transformers import get_cosine_schedule_with_warmup
batch_size = 8
num_epochs = 3
num_training_steps = (len(train_dataset) // batch_size) * num_epochs # Total
↪steps
num_warmup_steps = int(0.15 * num_training_steps) # 10% warm-up (adjustable)
```

```
[ ]: from sentence_transformers import SentenceTransformer, InputExample, losses,
↪SentenceTransformerTrainer, SentenceTransformerTrainingArguments
```

```
from torch.optim import AdamW
```

```
[ ]: # Load optimizer and scheduler states if saved
optimizer = AdamW(model.parameters(), lr=2e-5)
scheduler = get_cosine_schedule_with_warmup(
    optimizer,
    num_warmup_steps=num_warmup_steps,
    num_training_steps=num_training_steps
)

# Update training arguments for retraining
training_args = SentenceTransformerTrainingArguments(
    output_dir="./results",
    num_train_epochs=3, # Number of additional epochs
    per_device_train_batch_size=8,
    logging_dir="./logs",
    evaluation_strategy="epoch",
    save_strategy="epoch",
    save_total_limit=3,
    metric_for_best_model="eval_loss",
    greater_is_better=False,
    load_best_model_at_end=True
)

# Resume training
trainer = SentenceTransformerTrainer(
    model=model,
    args=training_args,
    train_dataset=train_dataset,
    eval_dataset=eval_dataset,
    loss=loss,
    optimizers=(optimizer, scheduler) # Use saved or reinitialized optimizer/
    ↪ scheduler
)

trainer.train()
```

```
c:\Users\Raviksh\miniconda3\envs\chat_arena\lib\site-
packages\transformers\training_args.py:1545: FutureWarning:
`evaluation_strategy` is deprecated and will be removed in version 4.46 of
Transformers. Use `eval_strategy` instead
```

```
warnings.warn(
  3%|          | 500/14532 [27:31<12:24:36,  3.18s/it]
```

```
{'loss': 0.2896, 'grad_norm': 9.947051048278809, 'learning_rate':
4.589261128958238e-06, 'epoch': 0.1}
```

```
7%|          | 1000/14532 [54:31<12:11:17,  3.24s/it]
```

```

{'loss': 0.2533, 'grad_norm': 6.576684474945068, 'learning_rate':
9.178522257916476e-06, 'epoch': 0.21}

10%|          | 1500/14532 [1:21:21<11:42:14, 3.23s/it]

{'loss': 0.2366, 'grad_norm': 15.67342472076416, 'learning_rate':
1.3767783386874713e-05, 'epoch': 0.31}

14%|          | 2000/14532 [1:48:23<11:30:36, 3.31s/it]

{'loss': 0.2243, 'grad_norm': 6.9420166015625, 'learning_rate':
1.8357044515832952e-05, 'epoch': 0.41}

17%|          | 2500/14532 [2:17:21<11:00:52, 3.30s/it]

{'loss': 0.2127, 'grad_norm': 5.346223831176758, 'learning_rate':
1.9966680032133108e-05, 'epoch': 0.52}

21%|          | 3000/14532 [2:44:16<10:35:32, 3.31s/it]

{'loss': 0.1991, 'grad_norm': 3.4953625202178955, 'learning_rate':
1.9782708350867e-05, 'epoch': 0.62}

24%|          | 3500/14532 [3:11:14<9:42:52, 3.17s/it]

{'loss': 0.2039, 'grad_norm': 9.505681037902832, 'learning_rate':
1.9440692168308896e-05, 'epoch': 0.72}

28%|          | 4000/14532 [3:38:07<9:19:03, 3.18s/it]

{'loss': 0.1949, 'grad_norm': 5.094330310821533, 'learning_rate':
1.8946156925376327e-05, 'epoch': 0.83}

31%|          | 4500/14532 [4:05:05<9:10:15, 3.29s/it]

{'loss': 0.2046, 'grad_norm': 11.371742248535156, 'learning_rate':
1.8307092083988976e-05, 'epoch': 0.93}

33%|          | 4844/14532 [4:26:57<8:19:19, 3.09s/it]

{'eval_loss': 1.0686678886413574, 'eval_runtime': 198.6288,
'eval_samples_per_second': 48.774, 'eval_steps_per_second': 6.097, 'epoch': 1.0}

34%|          | 5000/14532 [4:35:30<8:39:33, 3.27s/it]

{'loss': 0.1726, 'grad_norm': 1.7706348896026611, 'learning_rate':
1.7533822053352127e-05, 'epoch': 1.03}

38%|          | 5500/14532 [5:03:04<8:12:59, 3.27s/it]

{'loss': 0.127, 'grad_norm': 17.961708068847656, 'learning_rate':
1.6638839394009634e-05, 'epoch': 1.14}

41%|          | 6000/14532 [5:30:17<7:48:22, 3.29s/it]

{'loss': 0.1497, 'grad_norm': 5.063749313354492, 'learning_rate':
1.5636602994337558e-05, 'epoch': 1.24}

```


45%| | 6500/14532 [5:57:28<7:11:38, 3.22s/it]
{'loss': 0.2007, 'grad_norm': 3.986812114715576, 'learning_rate':
1.4543304480037266e-05, 'epoch': 1.34}

48%| | 7000/14532 [6:24:17<6:47:32, 3.25s/it]
{'loss': 0.2104, 'grad_norm': 31.943180084228516, 'learning_rate':
1.3376606630398355e-05, 'epoch': 1.45}

52%| | 7500/14532 [6:51:06<6:23:40, 3.27s/it]
{'loss': 0.1567, 'grad_norm': 9.610870361328125, 'learning_rate':
1.215535802734633e-05, 'epoch': 1.55}

55%| | 8000/14532 [7:17:56<5:57:40, 3.29s/it]
{'loss': 0.1669, 'grad_norm': 17.893171310424805, 'learning_rate':
1.089928854726106e-05, 'epoch': 1.65}

58%| | 8500/14532 [7:44:47<5:28:43, 3.27s/it]
{'loss': 0.1203, 'grad_norm': 3.770479440689087, 'learning_rate':
9.628690615046587e-06, 'epoch': 1.75}

62%| | 9000/14532 [8:11:59<4:54:51, 3.20s/it]
{'loss': 0.1183, 'grad_norm': 24.373321533203125, 'learning_rate':
8.364091369950783e-06, 'epoch': 1.86}

65%| | 9500/14532 [8:39:28<4:32:54, 3.25s/it]
{'loss': 0.1076, 'grad_norm': 6.507205009460449, 'learning_rate':
7.125921039458415e-06, 'epoch': 1.96}

67%| | 9688/14532 [8:52:49<4:15:26, 3.16s/it]
{'eval_loss': 1.1973414421081543, 'eval_runtime': 197.1655,
'eval_samples_per_second': 49.136, 'eval_steps_per_second': 6.142, 'epoch': 2.0}

69%| | 10000/14532 [9:10:08<4:07:50, 3.28s/it]
{'loss': 0.0824, 'grad_norm': 15.091569900512695, 'learning_rate':
5.934182878841825e-06, 'epoch': 2.06}

72%| | 10500/14532 [9:37:21<3:35:34, 3.21s/it]
{'loss': 0.0753, 'grad_norm': 72.4637680053711, 'learning_rate':
4.808130008659242e-06, 'epoch': 2.17}

76%| | 11000/14532 [10:04:46<3:16:26, 3.34s/it]
{'loss': 0.0586, 'grad_norm': 14.270190238952637, 'learning_rate':
3.7659543710511724e-06, 'epoch': 2.27}

79%| | 11500/14532 [10:32:16<2:40:38, 3.18s/it]

```

{'loss': 0.0612, 'grad_norm': 2.437696695327759, 'learning_rate':
2.8244928298999953e-06, 'epoch': 2.37}

83%|      | 12000/14532 [10:59:07<2:14:45, 3.19s/it]

{'loss': 0.0604, 'grad_norm': 7.160985946655273, 'learning_rate':
1.9989551629502833e-06, 'epoch': 2.48}

86%|      | 12500/14532 [11:25:50<1:50:11, 3.25s/it]

{'loss': 0.073, 'grad_norm': 7.49570369720459, 'learning_rate':
1.3026783403120957e-06, 'epoch': 2.58}

89%|      | 13000/14532 [11:52:34<1:22:09, 3.22s/it]

{'loss': 0.0502, 'grad_norm': 0.5728941559791565, 'learning_rate':
7.469110590995865e-07, 'epoch': 2.68}

93%|      | 13500/14532 [12:19:23<54:31, 3.17s/it]

{'loss': 0.046, 'grad_norm': 1.505934238433838, 'learning_rate':
3.4063201515477484e-07, 'epoch': 2.79}

96%|      | 14000/14532 [12:46:08<27:56, 3.15s/it]

{'loss': 0.0426, 'grad_norm': 4.470372676849365, 'learning_rate':
9.040484776677183e-08, 'epoch': 2.89}

100%|     | 14500/14532 [13:12:36<01:41, 3.18s/it]

{'loss': 0.0563, 'grad_norm': 2.4138896465301514, 'learning_rate':
2.7210082640505284e-10, 'epoch': 2.99}

100%|     | 14532/14532 [13:17:42<00:00, 3.10s/it]

{'eval_loss': 1.2383840084075928, 'eval_runtime': 195.9146,
'eval_samples_per_second': 49.45, 'eval_steps_per_second': 6.181, 'epoch': 3.0}

100%|     | 14532/14532 [13:17:50<00:00, 3.29s/it]

{'train_runtime': 47870.8569, 'train_samples_per_second': 2.428,
'train_steps_per_second': 0.304, 'train_loss': 0.1430878928466963, 'epoch': 3.0}

[ ]: TrainOutput(global_step=14532, training_loss=0.1430878928466963,
metrics={'train_runtime': 47870.8569, 'train_samples_per_second': 2.428,
'train_steps_per_second': 0.304, 'total_flos': 0.0, 'train_loss':
0.1430878928466963, 'epoch': 3.0})

```

```
[ ]: 
```

```
[ ]: 
```

```
[ ]: from sentence_transformers import SentenceTransformer
from huggingface_hub import HfApi, Repository

# Path to the saved model directory
model_dir = "/kaggle/working/results/checkpoint-7500/"

# Load the trained SentenceTransformer model
model = SentenceTransformer(model_dir)

# Push the model to Hugging Face
model.push_to_hub("Raviksh/finetune_bge_large", token='#####')

print(f"Model uploaded to Hugging Face: https://huggingface.co/your-org-name/
↳your-model-name")
```

```
model.safetensors: 0%|          | 0.00/711M [00:00<?, ?B/s]
```

```
Model uploaded to Hugging Face: https://huggingface.co/your-org-name/your-model-
name
```

```
[ ]: import os

print(os.listdir('./results'))
```

```
['checkpoint-7500', 'checkpoint-8000']
```

```
[ ]: print(os.listdir('./results/checkpoint-7500'))
```

```
['README.md', 'scheduler.pt', 'model.safetensors', 'training_args.bin',
'optimizer.pt', 'modules.json', 'tokenizer_config.json', '1_Pooling',
'config.json', 'special_tokens_map.json', 'trainer_state.json',
'tokenizer.json', 'rng_state.pth', 'config_sentence_transformers.json',
'sentence_bert_config.json', 'vocab.txt']
```

```
[ ]:
```