```
!pip install -q transformers
#!pip install -q sentencepiece

import transformers
from transformers import pipeline
import torch

# Comprobar si hay una GPU disponible
device = 0 if torch.cuda.is_available() else -1

# https://huggingface.co/docs/transformers/main_classes/logging
transformers.utils.logging.set_verbosity_error()


# análisis sentimiento
from transformers import pipeline

nlp = pipeline("text-classification", model="nlptown/bert-base-multilingual-uncased-sentiment", device=device)

# Analizar el sentimiento del texto
result = nlp("vi cosas peores")[0]
print(f"label: {result['label']}, with score: {round(result['score'], 4)}")

# Analizar el sentimiento del texto
result = nlp("No estaba nada mal")[0]
print(f"label: {result['label']}, with score: {round(result['score'], 4)}")

# Analizar el sentimiento del texto
result = nlp("mejor dar un paseo")[0]
print(f"label: {result['label']}, with score: {round(result['score'], 4)}")
```

```
label: 1 star, with score: 0.6107
label: 3 stars, with score: 0.3773
label: 5 stars, with score: 0.3502
```

```
# traducción
from transformers import pipeline

# https://github.com/facebookresearch/flores/blob/main/flores200/README.md#languages-in-flores-200
nlp = pipeline("translation", model="facebook/nllb-200-distilled-600M", src_lang="spa_Latn", tgt_lang="eng_Latn

# traducción del texto
result = nlp("todo se estropeó")
print(result)
```

```
config.json: 100%  846/846 [00:00<00:00, 25.4kB/s]
pytorch_model.bin: 100%  2.46G/2.46G [00:31<00:00, 178MB/s]
generation_config.json: 100%  189/189 [00:00<00:00, 9.30kB/s]
tokenizer_config.json: 100%  564/564 [00:00<00:00, 26.2kB/s]
sentencepiece.bpe.model: 100%  4.85M/4.85M [00:00<00:00, 93.0MB/s]
tokenizer.json: 100%  17.3M/17.3M [00:00<00:00, 132MB/s]
special_tokens_map.json: 100%  3.55k/3.55k [00:00<00:00, 127kB/s]
[{'translation_text': 'Everything went wrong.'}]
```

```
# resumen
nlp = pipeline("summarization", model="csebuetnlp/mT5_multilingual_XLSum", device=device)

texto="""
    Había una vez una muchacha, cuyo padre era lechero, con un cántaro de leche en la cabeza.
    Caminaba ligera y dando grandes zancadas para llegar lo antes posible a la ciudad, a donde iba para vender
    Por el camino empezó a pensar lo que haría con el dinero que le darían a cambio de la leche.
    -Compraré un centenar de huevos. O no, mejor tres pollos. ¡Sí, compraré tres pollos!
    La muchacha seguía adelante poniendo cuidado de no tropezar mientras su imaginación iba cada vez más y más
```

```
    -Criaré los pollos y tendré cada vez más, y aunque aparezca por ahí el zorro y mate algunos, seguro que ten
    Pero de repente, la muchacha tropezó, el cántaro se rompió y con él se fueron la ternera, la vaca, el cerdo
"""

resumen = nlp(texto)

print(resumen)

del nlp
```

```
    [{'summary_text': 'Hace unos meses, la revista británica The New York Times publicó un reportaje sobre el negocio
```

```
# otra prueba con otro
import torch
from transformers import BertTokenizerFast, EncoderDecoderModel
device = 'cuda' if torch.cuda.is_available() else 'cpu'
ckpt = 'mrm8488/bert2bert_shared-spanish-finetuned-summarization'
tokenizer = BertTokenizerFast.from_pretrained(ckpt)
model = EncoderDecoderModel.from_pretrained(ckpt).to(device)

def generate_summary(text):

   inputs = tokenizer([text], padding="max_length", truncation=True, max_length=512, return_tensors="pt")
   input_ids = inputs.input_ids.to(device)
   attention_mask = inputs.attention_mask.to(device)
   output = model.generate(input_ids, attention_mask=attention_mask)
   return tokenizer.decode(output[0], skip_special_tokens=True)

generate_summary(texto)
```

```
    'La muchacha, cuyo padre era lechero, se rompió y con él se fueron la ternera, la vaca, el cerdo y los pollos'
```