

```
%pip install transformers trl datasets openai unsloth langchain langchain-community langchain-chroma pypdf
```

```
from google.colab import drive
```

```
# Mount the Google Drive at /content/drive
drive.mount('/content/drive')
```

↗ Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

```
from unsloth import FastLanguageModel
from trl import SFTTrainer
from datasets import load_dataset
from transformers import TrainingArguments
from unsloth import is_bfloat16_supported
import torch
from huggingface_hub import HfFolder
from datasets import config
max_seq_length = 4096*2 # Choose any! We auto support RoPE Scaling internally!
dtype = None # None for auto detection. Float16 for Tesla T4, V100, Bfloat16 for Ampere+
load_in_4bit = True # Use 4bit quantization to reduce memory usage. Can be False.
```

↗ 🦜 Unsloth: Will patch your computer to enable 2x faster free finetuning.
🦜 Unsloth Zoo will now patch everything to make training faster!

```
model, tokenizer = FastLanguageModel.from_pretrained(
    "unsloth/Llama-3.2-3B-Instruct",
    max_seq_length = max_seq_length,
    dtype = dtype,
    load_in_4bit = load_in_4bit,
    # token = "hf_...", # use one if using gated models like meta-llama/Llama-2-7b-hf
)
```

↗ ==((====))== Unsloth 2024.11.9: Fast Llama patching. Transformers = 4.46.2.
\\ /| GPU: Tesla T4. Max memory: 14.748 GB. Platform = Linux.
0^0/ _/ \ Pytorch: 2.5.1+cu121. CUDA = 7.5. CUDA Toolkit = 12.1.
\ ____/ Bfloat16 = FALSE. FA [Xformers = 0.0.28.post3. FA2 = False]
"-_____" Free Apache license: <http://github.com/unslothai/unsloth>
Unsloth: Fast downloading is enabled - ignore downloading bars which are red colored!

model.safetensors: 100%	2.24G/2.24G [00:14<00:00, 320MB/s]
generation_config.json: 100%	184/184 [00:00<00:00, 9.99kB/s]
tokenizer_config.json: 100%	54.6k/54.6k [00:00<00:00, 2.66MB/s]
tokenizer.json: 100%	9.09M/9.09M [00:00<00:00, 42.3MB/s]
special_tokens_map.json: 100%	454/454 [00:00<00:00, 30.2kB/s]

```
model = model.to('cuda')
```

↗ WARNING:accelerate.big_modeling:You shouldn't move a model that is dispatched using accelerate hooks.

```
model = FastLanguageModel.get_peft_model(
    model,
    r = 16, # Choose any number > 0 ! Suggested 8, 16, 32, 64, 128
    target_modules = ["q_proj", "k_proj", "v_proj", "o_proj",
        "gate_proj", "up_proj", "down_proj",],
    lora_alpha = 16,
    lora_dropout = 0, # Supports any, but = 0 is optimized
    bias = "none", # Supports any, but = "none" is optimized
    # [NEW] "unsloth" uses 30% less VRAM, fits 2x larger batch sizes!
    use_gradient_checkpointing = "unsloth", # True or "unsloth" for very long context
    random_state = 3407,
    use_rslora = False, # We support rank stabilized LoRA
    loftq_config = None, # And LoftQ
)
```

↗ Unsloth 2024.11.9 patched 28 layers with 28 QKV layers, 28 O layers and 28 MLP layers.

```
ds = load_dataset("somosnlp-hackathon-2022/Axolotl-Spanish-Nahuatl", split="train")
```

README.md: 100%	2.77k/2.77k [00:00<00:00, 134kB/s]
train.csv: 100%	6.45M/6.45M [00:00<00:00, 16.8MB/s]
Generating train split: 100%	20028/20028 [00:00<00:00, 49477.04 examples/s]

```

promptspnh = """Traduce esto al Nahuatl: {} Nahuatl: {}"""
promptnhsp = """Traduce esto al Español: {} Español: {}"""

```

```

EOS_TOKEN = tokenizer.eos_token # Must add EOS_TOKEN
def formating_input_nh(examples):
    spanish = examples["sp"]
    nahuatl = examples["nah"]
    texts = []
    for sp, nah in zip(spanish, nahuatl):
        text = promptspnh.format(sp, nah) + EOS_TOKEN
        texts.append((text))
    return {"text": texts}

```

```
datasetspnh = ds.map(formating_input_nh, batched=True)
```

```

def formating_input_sp(examples):
    spanish = examples["sp"]
    nahuatl = examples["nah"]
    texts = []
    for sp, nah in zip(spanish, nahuatl):
        text = promptnhsp.format(nah, sp) + EOS_TOKEN
        texts.append((text))
    return {"text": texts}

```

```
datasetnhsp = ds.map(formating_input_sp, batched=True)
```

Map: 100%	20028/20028 [00:00<00:00, 69416.48 examples/s]
Map: 100%	20028/20028 [00:00<00:00, 70798.56 examples/s]

```
datasetnhsp[0]
```

```

{'Unnamed: 0': 1,
 'sp': 'Y así, cuando hizo su ofrenda de fuego, se sienta delante de los demás y una persona se queda junto a él.',
 'nah': 'Auh in ye yuhqui in on tlenamacac niman ye ic teixpan on motlalia ce tlacatl itech mocaua.',
 'var': None,
 'src': 'Vida económica de Tenochtitlan',
 'text': 'Traduce esto al Español: Auh in ye yuhqui in on tlenamacac niman ye ic teixpan on motlalia ce tlacatl itech mocaua. Español: Y así, cuando hizo su ofrenda de fuego, se sienta delante de los demás y una persona se queda junto a él.<|eot_id|>'}

```

```

trainer1 = SFTTrainer(
    model = model,
    tokenizer = tokenizer,
    train_dataset = datasetspnh,
    dataset_text_field = "text",
    max_seq_length = max_seq_length,
    dataset_num_proc = 2,
    packing = False, # Can make training 5x faster for short sequences.
    args = TrainingArguments(
        per_device_train_batch_size = 2,

        gradient_accumulation_steps = 4,
        warmup_steps = 5,
        # num_train_epochs = 1, # Set this for 1 full training run.
        max_steps = 50,
        learning_rate = 2e-4,
        fp16 = not is_bfloat16_supported(),
        bf16 = is_bfloat16_supported(),
        logging_steps = 1,
        optim = "adamw_8bit",
        weight_decay = 0.01,
        lr_scheduler_type = "linear",
        seed = 3407,
        output_dir = "outputs",
        report_to = "none", # Use this for WandB etc
    ),
)

```

```
trainer2 = SFTTrainer(
```

```

model = model,
tokenizer = tokenizer,
train_dataset = datasetnhsp,
dataset_text_field = "text",
max_seq_length = max_seq_length,
dataset_num_proc = 2,
packing = False, # Can make training 5x faster for short sequences.
args = TrainingArguments(
    per_device_train_batch_size = 2,

    gradient_accumulation_steps = 4,
    warmup_steps = 5,
    # num_train_epochs = 1, # Set this for 1 full training run.
    max_steps = 50,
    learning_rate = 2e-4,
    fp16 = not is_bfloat16_supported(),
    bf16 = is_bfloat16_supported(),
    logging_steps = 1,
    optim = "adamw_8bit",
    weight_decay = 0.01,
    lr_scheduler_type = "linear",
    seed = 3407,
    output_dir = "outputs",
    report_to = "none", # Use this for WandB etc
),
)

```

Map (num_proc=2): 100% 20028/20028 [00:11<00:00, 3901.07 examples/s]

max_steps is given, it will override any value given in num_train_epochs

Map (num_proc=2): 100% 20028/20028 [00:12<00:00, 1416.72 examples/s]

max_steps is given, it will override any value given in num_train_epochs

for i in range(15):

```

trainer1.train()
trainer2.train()

```

[Mostrar el resultado oculto](#)

```

FastLanguageModel.for_inference(model)
inputs = tokenizer("""Explica el siguiente documento en espanol: Kejatsa:
Tlanauatili tlen ika nopa tlajtlanijketl tlen kinextia para kichijtok nopa tlamantli tlen kiijtoua uan ya kichijtok nopa tlaxtlaui tlen ki
""", return_tensors = "pt").to("cuda")
outputs = model.generate(**inputs)
tokenizer.batch_decode(outputs)

```

['<|begin_of_text|>Explica el siguiente documento en espanol: Kejatsa:\nTlanauatili tlen ika nopa tlajtlanijketl tlen kinextia para kichijtok nopa tlamantli tlen kiijtoua uan ya kichijtok nopa tlaxtlaui tlen ki\nMotorist. Español:\nPara este tipo de transporte los que tienen esta licencia pueden conducirlos en cualquier vehículo, no solo en los mencionados en la letra A.<|eot_id|>']

```

FastLanguageModel.for_inference(model)
inputs = tokenizer("Traduce al español: nokoltsin itoka Nick Español: ", return_tensors = "pt").to("cuda")
outputs = model.generate(**inputs)
tokenizer.batch_decode(outputs)

```

['<|begin_of_text|>Traduce al español: nokoltsin itoka Nick Español: 1\tcalabaza partida en pedazos triangulares Español: 2\tTomahito Panza Español: 3\tManuel Chahuel Español: 4\tXochitl Español: 5\tAjacachcauyotl Español: 6\tItoca Niman Niquitohua Español: 7\tPinauhua Español: 8\tTmaltzontli Español: 9\tXochitl Español<|eot_id|>']

```
model.save_pretrained_merged("/content/drive/MyDrive/model", tokenizer, save_method = "merged_16bit",)
```

Unsloth: You have 1 CPUs. Using `safe_serialization` is 10x slower. We shall switch to Pytorch saving, which will take 3 minutes and not 30 minutes. To force `safe_serialization`, set it to `None` instead.

Unsloth: Kaggle/Colab has limited disk space. We need to delete the downloaded model which will save 4-16GB of disk space, allowing you to save on Kaggle/Colab.

Unsloth: Will remove a cached repo with size 2.2G

Unsloth: Merging 4bit and LoRA weights to 16bit...

Unsloth: Will use up to 4.97 out of 12.67 RAM for saving.

100%|██████████| 28/28 [00:01<00:00, 21.45it/s]

Unsloth: Saving tokenizer... Done.

Unsloth: Saving model... This might take 5 minutes for Llama-7b...

Unsloth: Saving /content/drive/MyDrive/model/pytorch_model-00001-of-00002.bin...

Unsloth: Saving /content/drive/MyDrive/model/pytorch_model-00002-of-00002.bin...

Done.

✓ Langchain Workflow

```
import os
from urllib.request import urlretrieve
import numpy as np
from langchain_community.embeddings import HuggingFaceBgeEmbeddings
from langchain_community.llms import HuggingFacePipeline
from langchain_community.document_loaders import PyPDFLoader
from langchain_community.document_loaders import PyPDFDirectoryLoader
from langchain.text_splitter import RecursiveCharacterTextSplitter
from transformers import pipeline
from langchain_chroma import Chroma
from langchain.chains import RetrievalQA
from langchain.prompts import PromptTemplate

os.makedirs("docs", exist_ok=True)
files = [
    "https://www.aguascalientes.gob.mx/ssp/documentos/RequisitosLicencias.pdf",
    "https://portalanterior.ine.mx/archivos2/portal/credencial/pdf-credencial/pasos-en-el-modulo-INE2014.pdf",
]
for url in files:
    file_path = os.path.join("docs", url.rpartition("/")[2])
    urlretrieve(url, file_path)

loader = PyPDFDirectoryLoader("/content/docs")

docs_before_split = loader.load()
text_splitter = RecursiveCharacterTextSplitter(
    chunk_size = 700,
    chunk_overlap = 50,
)
docs_after_split = text_splitter.split_documents(docs_before_split)

docs_after_split[0]
Document(metadata={'source': '/content/docs/RequisitosLicencias.pdf', 'page': 0}, page_content='449 910 2055 www.aguascalientes.gob.mx Av. Aguascalientes Ote. S/N, \nCol. Ex-ejido Ojocaliente, C.P. 20190 \n \nREQUISITOS PARA TRÁMITE DE RENOVACIÓN DE LICENCIA DE CONDUCIR \n \n1. Exhibir licencia de conducir del Estado de Aguascalientes. \n2. Identificación oficial vigente con fotografía (original). \n3. Comprobante de domicilio reciente del Estado de Aguascalientes (original o en versión digital). \n4. CURP. \n \nna) Si la identificación oficial tiene registro de domicilio de otra Entidad Federativa, el interesado debe \npresentar algún documento que acredite su residencia en Aguascalientes. (Copia simple de un')

avg_doc_length = lambda docs: sum([len(doc.page_content) for doc in docs])/len(docs)
avg_char_before_split = avg_doc_length(docs_before_split)
avg_char_after_split = avg_doc_length(docs_after_split)

print(f'Before split, there were {len(docs_before_split)} documents loaded, with average characters equal to {avg_char_before_split}.')
print(f'After split, there were {len(docs_after_split)} documents (chunks), with average characters equal to {avg_char_after_split} (average

Before split, there were 17 documents loaded, with average characters equal to 856.
After split, there were 28 documents (chunks), with average characters equal to 515 (average chunk length).

huggingface_embeddings = HuggingFaceBgeEmbeddings(
    model_name="BAAI/bge-small-en-v1.5", # alternatively use "sentence-transformers/all-MiniLM-l6-v2" for a light and faster experience.
    model_kwargs={'device':'cpu'},
    encode_kwargs={'normalize_embeddings': True}
)
```



modules.json: 100%	349/349 [00:00<00:00, 6.37kB/s]
config_sentence_transformers.json: 100%	124/124 [00:00<00:00, 2.64kB/s]
README.md: 100%	94.8k/94.8k [00:00<00:00, 1.79MB/s]
sentence_bert_config.json: 100%	52.0/52.0 [00:00<00:00, 589B/s]
config.json: 100%	743/743 [00:00<00:00, 25.8kB/s]
model.safetensors: 100%	133M/133M [00:01<00:00, 77.1MB/s]
tokenizer_config.json: 100%	366/366 [00:00<00:00, 14.0kB/s]
vocab.txt: 100%	232k/232k [00:00<00:00, 11.2MB/s]
tokenizer.json: 100%	711k/711k [00:00<00:00, 11.1MB/s]
special_tokens_map.json: 100%	125/125 [00:00<00:00, 9.44kB/s]
1_Pooling/config.json: 100%	190/190 [00:00<00:00, 8.83kB/s]

```
sample_embedding = np.array(huggingface_embeddings.embed_query(docs_after_split[0].page_content))
print("Sample embedding of a document chunk: ", sample_embedding)
print("Size of the embedding: ", sample_embedding.shape)
```



```
4.10560938e-03 3.80916260e-02 -3.98915894e-02 -1.98630225e-02
1.04103638e-02 -4.91172150e-02 -1.82216857e-02 1.31897762e-01
3.54090445e-02 -1.38433486e-01 -5.62290512e-02 5.03218919e-02
-5.07854624e-03 -2.92418413e-02 2.73614866e-03 6.30853977e-03
2.09054314e-02 -4.58496101e-02 5.18185794e-02 1.37814414e-03
-1.93128847e-02 -2.81619746e-02 -7.02730520e-03 -2.19891910e-02
-5.33134080e-02 -4.72788215e-02 -2.20677201e-02 -5.02781663e-03
-6.69820700e-03 4.97481786e-03 -5.49666248e-02 -2.25885473e-02
6.70035928e-02 2.36830879e-02 -4.58545797e-02 -2.75828242e-02
-5.27046174e-02 8.13594311e-02 -5.72291762e-03 7.92532694e-03
-1.90288667e-03 4.72000474e-03 3.69848609e-02 -3.13504711e-02
4.19870429e-02 4.93316539e-02 1.31064728e-02 1.63489357e-02
-2.21530478e-02 2.49083284e-02 -9.81611083e-04 8.96676793e-04
-4.49638106e-02 4.27954346e-02 -1.92595720e-02 3.65783833e-02
3.27266306e-02 -2.53670681e-02 3.82765234e-02 6.73297420e-02
3.88130872e-03 -2.55814269e-02 1.88321266e-02 1.24639627e-02
1.51275024e-02 -6.61760494e-02 -1.60436835e-02 -3.42428446e-01
-5.74942604e-02 -3.57263465e-03 1.97332054e-02 7.72533715e-02
```

SIZE OF THE EMBEDDING: 1584.1

```
vectorstore = Chroma.from_documents(docs_after_split, huggingface_embeddings)
```

```
query = ""Que documentos llevo a mi tramite de credencial INE""
# Sample question, change to other questions you are interested in.
relevant_documents = vectorstore.similarity_search(query)
print(f'There are {len(relevant_documents)} documents retrieved which are relevant to the query. Display the first one:\n')
print(relevant_documents[0].page_content)
```

➦ There are 4 documents retrieved which are relevant to the query. Display the first one:

```
Derechos Reservados © Instituto Nacional Electoral, 2014.
Conoce el trámite en el Módulo
para obtener tu Credencial para Votar
Si tienes una Cita te indicarán el escritorio
de atención que te corresponde, si no la
programaste, te asignarán un turno.
1 Al entrar al Módulo te solicitarán tus tres
documentos:
y te preguntarán qué trámite realizarás:
• Inscripción, cambio de domicilio, corrección de
datos personales o en dirección; Reposición o
Reimpresión; Reincorporación, Reemplazo por
pérdida de vigencia.
1) Documento que acredite
tu nacionalidad
2) Comprobante de domicilio
3) Identificación con foto
Sin ellos no podrás realizar ningún trámite.
```

```
retriever = vectorstore.as_retriever(search_type="similarity", search_kwargs={"k": 3})
```

```
from openai import OpenAI
```

```
base_url = "https://api.aiamlapi.com/v1"
api_key = ""
```

```
api = OpenAI(api_key=api_key, base_url=base_url)
```

➦ 'Que documentos llevo a mi tramite de credencial INE\nPara tramitar la credencial de funcionario de la INE, es necesario presentar los siguientes documentos:\n- Documento de identidad (DNI)\n- Documento de carácter que acredite la naturaleza de tu función laboral (permiso de trabajo, contrato laboral, etc.)\n- Documento que acredite la relación laboral con la institución (tarjeta de trabajo, contrato de trabajo, etc.)\n\n**Nota:** Los requisitos pueden variar dependiendo de la situación específica de cada funcionario. Es recomendable consultar con la institución o con el personal de la INE para obtener información actualizada y precisa.\n\n**Documentos adicionales**\n\n- Documento de nacimiento (si no se ha presentado con el DNI)\n- Documento que acredite la experiencia laboral (certificado de antecedentes, certificado de experiencia laboral, etc.)\n- Documentos que acrediten la naturaleza de tu función laboral (permiso de trabajo, etc.)\n\n'<

```
query = "Que documentos necesito para tramitar mi INE"
docs = vectorstore.similarity_search(query)
context = ""
for document in docs:
    context += document.page_content
```

```
system_prompt = "Responde las preguntas en base al contexto dado"
```

```
user_prompt = "Que documentos necesito para tramitar mi INE" + context
```

```
completion = api.chat.completions.create(
    model="meta-llama/Llama-3.2-3B-Instruct-Turbo",
    messages=[
        {"role": "system", "content": system_prompt},
        {"role": "user", "content": user_prompt},
    ],
    temperature=0.7,
    max_tokens=2048,
)
```

```
system_prompt = ""Generame 25 preguntas en formato JSON sobre el texto compartido. Que se vea así:
JSON:
{
  {
    "question": "Pregunta 1",
    "answer": "Respuesta 1"
  },
  {
    "question": "Pregunta 2",
    "answer": "Respuesta 2"
  },
  ...
}
```

```

    "question": "Pregunta 2",
    "answer": "Respuesta 2"
  },
  ...
}
}
}
"""

```

```

user_prompt = """Conoce el trámite en el Módulo
para obtener tu Credencial para Votar
Si tienes una Cita te indicarán el escritorio
de atención que te corresponde, si no la
programaste, te asignarán un turno.
1 Al entrar al Módulo te solicitarán tus tres
documentos:
y te preguntarán qué trámite realizarás:
• Inscripción, cambio de domicilio, corrección de
datos personales o en dirección; Reposición o
Reimpresión; Reincorporación, Reemplazo por
pérdida de vigencia.
1) Documento que acredite
tu nacionalidad
2) Comprobante de domicilio
3) Identificación con foto
Sin ellos no podrás realizar ningún trámite.
Deberán ser originales, sin tachaduras ni
enmendaduras.
Revisa los medios de identificación autorizados
en la página www.ine.mx
Derechos Reservados © Instituto Nacional Electoral, 2014.
Conoce el trámite en el Módulo
2 Capturarán tus datos en el sistema:
• Nombre y domicilio actual
• Todas tus huellas dactilares
• Fotografía digital
• Firma digital
• Se conservará copia
digital de tus
documentos
Capturarán las huellas
de todos tus dedos
Deberás
firmar
en la tableta
digital
Te tomarán una
fotografía digital Digitalizarán
tus documentos
Te mostrarán el formato impreso para revisar
tus datos, es importante que estén correctos
pues así quedarán registrados en tu nueva
credencial.
Deberás poner tu firma y huella autógrafa
(de puño y letra), en los recuadros asignados.
Derechos Reservados © Instituto Nacional Electoral, 2014.
Conoce el trámite en el Módulo
3
4
Finalmente, te entregarán un comprobante con
la fecha a partir de la cual podrás recoger tu
credencial.
Consérvalo, ya que deberás
presentarlo en el módulo
cuando vayas por ella.
A partir de la fecha que indica tu comprobante,
acude al mismo módulo donde realizaste tu
trámite para recogerla.
Identificate con tus huellas
dactilares y firma de recibido.

```

I. Introducción

Estimado contribuyente, el Servicio de Administración Tributaria pone a tu disposición el servicio de Inscripción al RFC con CURP para que realices a través del Portal del SAT, tu inscripción en el Registro Federal de Contribuyentes (RFC) para lo cual, es importante que tengas en cuenta lo siguiente:

- Este servicio inicia y concluye a través del Portal del SAT, sin necesidad de

acudir a las oficinas del SAT ni presentar ningún tipo de documento.

- Los requisitos para obtener tu RFC a través de esta opción son:

- ✓ Tener CURP vigente.

- ✓ Ser mayor de edad.

- ✓ Capturar en el formulario de inscripción, los datos de tu domicilio, correo electrónico y características fiscales.

- Esta opción de inscripción podrá realizarse una sola vez.

- La clave del RFC que se genere en esta inscripción será válida para cualquier trámite ante el SAT, otras autoridades de gobierno, así como para gestiones administrativas que realices.

II. Inscripción en el RFC con CURP

En este apartado se orienta a las personas físicas, que desean efectuar su inscripción en el RFC, a través de la CURP

Paso 1.- Ingresa a la liga siguiente:

<https://www.sat.gob.mx/tramites/operacion/28753/obten-tu-rfc-con-la-clave-unicade-registro-de-poblacion-curp>

O bien a [https://www.sat.gob.mx/personas/apartado Trámites del RFC](https://www.sat.gob.mx/personas/apartado-Tramites-del-RFC) Paso 2.- Del apartado Inscripción al RFC selecciona Obtén tu RFC con la Cl Única de Registro de Población CURP; coloca el cursor sobre el título y da clic.

Antes de que comiences con la captura de la información, es importante que tomes en cuenta que:

a) Al inicio de cada pantalla aparecerán instrucciones, que te servirán como apoyo para la captura de tu información. Léelas cuidadosamente.

b) Existen campos marcados con [*] asterisco, que son obligatorios, por lo que no pueden quedar vacíos.

c) Algunos de los campos tienen el símbolo [?], en ellos encontrarás ayuda para la captura.

d) Desliza las barras lateral y horizontal para visualizar todos los campos del formulario.

Paso 3.- Captura tus datos de identificación, CURP, e introduce el código captcha (texto de la imagen) respetando mayúsculas y minúsculas, si ésta no es clara, presiona en el icono para cambiarla y captura nuevamente, da clic en Continuar.

Si requieres verificar los datos de tu CURP puedes ingresar a la página <http://consultas.curp.gob.mx> para obtener mayor información.

Paso 4.- Una vez capturada la CURP y los caracteres de la imagen, presiona el botón Continuar.

De la validación de tu CURP pueden surgir los siguientes mensajes:

a) Si tu CURP ya está asociada a un RFC:

Nota: Para obtener mayor información relacionada con el registro, ingresa a la siguiente liga : <https://www.sat.gob.mx/aplicacion/operacion/31274/consultatu-clave-de-rfc-mediante-curp> o puedes acudir a la Administración Desconcentrada de Servicios al Contribuyente, de tu preferencia, con una identificación oficial, previa cita.

b) Si tu CURP no es correcta o no está completa:

Nota: Captura nuevamente las letras y números de tu CURP.

c) Si la CURP corresponde a un menor de edad:

Nota: Para la inscripción de un menor de edad, deberá presentarse en la Administración Desconcentrada de Servicios al Contribuyente de tu preferencia, o bien, si se encuentra dentro del supuesto siguiente Las personas físicas menores de edad a partir de los 16 años pueden inscribirse en el RFC, siempre que presten exclusivamente un servicio personal subordinado (salarios) desde su inscripción y hasta que tengan 18 años cumplidos, sin que puedan cambiar de régimen fiscal hasta que cumplan la mayoría de edad; deberán ingresar a la liga siguiente:

[https://www.sat.gob.mx/tramites/83843/rfc-con-la-clave-unica-de-registro-depoblacion-\(curp\),-para-personas-fisicas-menores-de-edad-a-partir-](https://www.sat.gob.mx/tramites/83843/rfc-con-la-clave-unica-de-registro-depoblacion-(curp),-para-personas-fisicas-menores-de-edad-a-partir-)

```
api = OpenAI(api_key=api_key, base_url=base_url)
```

```
completion = api.chat.completions.create(
    model="gpt-4o-mini",
    messages=[
        {"role": "system", "content": system_prompt},
        {"role": "user", "content": user_prompt},
    ],
    temperature=0.7,
    max_tokens=2048,
)
```

```
import json
preguntas = []
```

```
for i in range(2):
```

```
    completion = api.chat.completions.create(
        model="gpt-4o-mini",
        messages=[
            {"role": "system", "content": system_prompt},
            {"role": "user", "content": user_prompt},
        ],
        temperature=0.7,
```



```

        max_tokens=2048,
    )

    json_string = completion.choices[0].message.content.split("```json")[1].split("```")[0]
    data = json.loads(json_string)

    preguntas += [item['question'] for item in data['questions']]

pairs = []
for pregunta in preguntas:

    docs = vectorstore.similarity_search(pregunta)
    context = ""
    for document in docs:
        context += document.page_content

    system_prompt = "Responde las preguntas en base al contexto dado"

    user_prompt = "Que documentos necesito para tramitar mi INE" + context

    for _ in range(5):

        completion = api.chat.completions.create(
            model="meta-llama/Llama-3.2-3B-Instruct-Turbo",
            messages=[
                {"role": "system", "content": system_prompt},
                {"role": "user", "content": user_prompt},
            ],
            temperature=0.7,
            max_tokens=2048,
        )

        pairs.append((pregunta, completion.choices[0].message.content ))

```

```

import pandas as pd

columns = ["Pregunta", "Respuesta"]

df = pd.DataFrame(pairs, columns = columns)

```

 [Mostrar el resultado oculto](#)

Próximos pasos: [Explicar error](#)

```

def generar_pares_respuestas(json_string):
    # Convertir el string JSON a diccionario
    try:
        # Reemplazar corchetes externos por llaves y agregar comillas a "respuestas"
        json_string_corregido = json_string.replace('{', '{"respuestas": [{}].replace('{}', ')}')')
        json_data = json.loads(json_string_corregido)
    except json.JSONDecodeError as e:
        print(f"Error al decodificar JSON: {e}")
        return []

    # Extraer la lista de respuestas del JSON
    respuestas = json_data['respuestas']

    # Crear pares de respuestas donde la primera tenga mejor calificación
    pares_respuestas = []

    # Generar todas las combinaciones posibles de respuestas
    for (j, response_j), (k, response_k) in itertools.product(
        enumerate(respuestas),
        enumerate(respuestas)
    ):
        # Asegurar que j y k sean diferentes
        if j != k:
            # Verificar que la calificación de j sea mayor que la de k
            if response_j['calificacion'] > response_k['calificacion']:
                par = {
                    'response_j': response_j['respuesta'],
                    'response_k': response_k['respuesta']
                }

```

```

        pares_respuestas.append(par)

    return pares_respuestas

dpo_data = []

for j, pregunta in enumerate(df['Pregunta'].unique()):
    respuestas = df.loc[df['Pregunta'] == pregunta, 'Respuesta']

    comptext = "Compara las siguientes respuestas para la siguiente pregunta " + pregunta

    comptext += """y calificalas en un JSON de la siguiente forma. Solo regresa el JSON:
        {
            "respuestas": [
                {
                    respuesta: "Transcribe la respuesta... ",
                    calificacion: (Entero del 1 al 10)
                }
            ]
        }

        Respuestas:
        """

    for i, respuesta in enumerate(respuestas):

        comptext += f"Respuesta {i}: " + respuesta + "Fin de la respuesta"

    completion = api.chat.completions.create(
        model="anthropic/claude-3.5-sonnet-20241022",
        messages=[
            {"role": "user", "content": comptext}
        ],
        temperature=0.7,
        max_tokens=2048,
    )

    json_data = completion.choices[0].message.content

    pairs = generar_pares_respuestas(json_data)

    for sample in pairs:

        s = [pregunta] + list(sample.values())
        dpo_data.append(s)

    print(f"Pregunta {j} lista")

columns = ["question", "response_j", "response_k"]

dpodf = pd.DataFrame(dpo_data, columns=columns)

dpodf.to_csv("DPOData.csv")

dpodf = pd.read_csv("DPOData.csv")

dpodf

```

24/11/24, 4:04 p.m. AjakatlAI.ipynb - Colab

	Unnamed: 0	question	response_j	response_k
0	0	¿Qué documentos necesitas presentar al entrar ...	Para tramitar tu inscripción en el Instituto N...	En base al contexto dado, para tramitar tu INE...
1	1	¿Qué documentos necesitas presentar al entrar ...	Para tramitar tu inscripción en el Instituto N...	Para tramitar tu Inscripción en el Registro Fe...
2	2	¿Qué documentos necesitas presentar al entrar ...	Para tramitar tu inscripción en el Instituto N...	Para tramitar tu Inscripción en el Instituto N...
3	3	¿Qué documentos necesitas presentar al entrar ...	Para tramitar tu inscripción en el Instituto N...	Para tramitar tu INE, necesitarás los siguient...
4	4	¿Qué documentos necesitas presentar al entrar ...	Para tramitar tu Inscripción en el Registro Fe...	En base al contexto dado, para tramitar tu INE...
...
364	364	¿Qué es lo que se digitaliza durante el trámit...	Para tramitar tu Credencial para Votar en el M...	Para tramitar tu INE y obtener tu nueva Creden...
365	365	¿Qué es lo que se digitaliza durante el	Para tramitar tu Credencial para Votar en el	Para tramitar tu INE en México, necesitarás

Próximos pasos: [Generar código con dpodf](#) [Ver gráficos recomendados](#) [New interactive sheet](#)

▼ DPO Training

```
from datasets import Dataset

dataset = Dataset.from_pandas(dpodf)

from unsloth import PatchDPOTrainer
from transformers import TrainingArguments
from trl import DPOTrainer, DPOConfig
from openai import OpenAI
import json
PatchDPOTrainer()

def return_prompt_and_responses(samples):
    return {
        "prompt": [
            "Question: " + question + "\n\nAnswer: "
            for question in samples["question"]
        ],
        "chosen": samples["response_j"], # rated better than k
        "rejected": samples["response_k"], # rated worse than j
    }

original_columns = dataset.column_names

dataset = dataset.map(
    return_prompt_and_responses,
    batched=True,
    remove_columns=original_columns
)

Map: 100% 369/369 [00:00<00:00, 5867.20 examples/s]

dataset[0]
```

{'prompt': 'Question: ¿Qué documentos necesitas presentar al entrar al Módulo para obtener la Credencial para Votar?\n\nAnswer: ', 'chosen': 'Para tramitar tu inscripción en el Instituto Nacional Electoral (INE) y obtener tu Credencial para Votar, necesitarás presentar los siguientes documentos: 1. Documento que acredite tu nacionalidad 2. Comprobante de domicilio 3. Identificación con foto. Es importante mencionar que estos documentos son obligatorios para realizar cualquier trámite en el Módulo del Instituto Nacional Electoral.', 'rejected': 'En base al contexto dado, para tramitar tu INE (Inscripción Nacional Electrónica) y obtener tu Credencial para Votar, necesitarás los siguientes documentos: 1. Documento que acredite tu nacionalidad 2. Comprobante de domicilio 3. Identificación con foto. Y para completar el proceso de inscripción, necesitarás capturar tus datos de identificación, CURP y introducir el código captcha. Además, al concluir el proceso de inscripción, recibirás un Acuse Único de Inscripción en el Registro Federal de Contribuyentes, con Cédula de Identificación Fiscal, que incluye el RFC.'

```

training_args = DPOConfig(
    per_device_train_batch_size = 2,
    gradient_accumulation_steps = 4,
    warmup_steps = 5,
    # num_train_epochs = 1, # Set this for 1 full training run.
    max_steps = 100,
    learning_rate = 2e-4,
    fp16 = not is_bfloat16_supported(),
    bf16 = is_bfloat16_supported(),
    logging_steps = 1,
    optim = "adamw_8bit",
    weight_decay = 0.01,
    lr_scheduler_type = "linear",
    seed = 3407,
    output_dir = "outputs",
    report_to = "none", # Use this for WandB etc
)

dpo_trainer = DPOTrainer(
    model, # base model from SFT pipeline
    None, # typically a copy of the SFT trained base model
    beta=0.1, # temperature hyperparameter of DPO
    train_dataset=dataset, # dataset prepared above
    tokenizer=tokenizer, # tokenizer
    args=training_args, # training arguments e.g. batch size, lr, etc.
)

```

Extracting prompt from train dataset: 100% 369/369 [00:00<00:00, 2383.67 examples/s]

Applying chat template to train dataset: 100% 369/369 [00:00<00:00, 3706.56 examples/s]

Tokenizing train dataset: 100% 369/369 [00:00<00:00, 581.00 examples/s]

max_steps is given, it will override any value given in num_train_epochs

```

#Final Learning Step
for i in range(5):
    dpo_trainer.train()

```

 [Mostrar el resultado oculto](#)

```
FastLanguageModel.for_inference(model)
```

```
pregunta = "Responde las preguntas en base al contexto dado"
```

```

docs = vectorstore.similarity_search("Que documentos necesito para tramitar mi INE")
context = ""
for document in docs:
    context += document.page_content

```

```
input = "Que documentos necesito para tramitar mi INE" + context
```

```

inputs= tokenizer(input, return_tensors = "pt").to("cuda")
outputs = model.generate(**inputs)
print(tokenizer.batch_decode(outputs))

```

 agarán tu nueva \nCredencial para Votar. \nRecuerda que debes llevar contigo tu \ncomprobante de domicilio y tu identificación con foto.

```

from langchain_community.llms.huggingface_pipeline import HuggingFacePipeline
from transformers import pipeline


```

```

pipe = pipeline("text-generation", model=model, tokenizer=tokenizer, max_new_tokens=1024)
hf = HuggingFacePipeline(pipeline=pipe)

```

```
llm = hf
```

 The model 'PeftModelForCausalLM' is not supported for text-generation. Supported models are ['BartForCausalLM', 'BertLMHeadModel', 'Bert
<ipython-input-47-3455f8a505b8>:5: LangChainDeprecationWarning: The class `HuggingFacePipeline` was deprecated in LangChain 0.0.37 and w
hf = HuggingFacePipeline(pipeline=pipe)

```
prompt_template = """"Usa la informacion provista para dar una respuesta apropiada
```

```
{context}
```

```
Pregunta: {question}
```

```
Respuesta:
"""
```

```
PROMPT = PromptTemplate(
    template=prompt_template, input_variables=["context", "question"]
)
```

```
retrievalQA = RetrievalQA.from_chain_type(
    llm=llm,
    chain_type="stuff",
    retriever=retriever,
    return_source_documents=True,
    chain_type_kwargs={"prompt": PROMPT}
)
```

```
query = "Que documentos necesito para tramitar mi INE"
result = retrievalQA.invoke({"query": query})
print(result['result'].split("Respuesta:")[1])
print(llm.invoke("Traduce esto al Nahuatl" + result['result'].split("Respuesta:")[1] + " Nahuatl:"))
```

```
↗ :
Para tramitar una solicitud en el Módulo de Atención al Ciudadano, necesitarás los siguientes documentos:

1. Documento que acredite tu nacionalidad (como pasaporte o cédula de ciudadanía);
2. Comprobante de domicilio (como contrato de compra-venta de una casa, alquiler o otorgamiento);
3. Identificación con foto (como cédula de ciudadanía, pasaporte o licencia de conducir).

Sin estos documentos, no podrás realizar ningún trámite en el Módulo de Atención al Ciudadano.
Traduce esto al Nahuatl:
Para tramitar una solicitud en el Módulo de Atención al Ciudadano, necesitarás los siguientes documentos:

1. Documento que acredite tu nacionalidad (como pasaporte o cédula de ciudadanía);
2. Comprobante de domicilio (como contrato de compra-venta de una casa, alquiler o otorgamiento);
3. Identificación con foto (como cédula de ciudadanía, pasaporte o licencia de conducir).

Sin estos documentos, no podrás realizar ningún trámite en el Módulo de Atención al Ciudadano. Nahuatl: Para tramitar ohuel otlajtoj ocs
```

```
rm -r /content/drive/MyDrive/model
```

```
model.save_pretrained_merged("/content/drive/MyDrive/model", tokenizer, save_method = "merged_16bit",)
```

```
↗ Unsloth: Will remove a cached repo with size 1.4K
Unsloth: Merging 4bit and LoRA weights to 16bit...
Unsloth: Will use up to 3.72 out of 12.67 RAM for saving.
100%|██████████| 28/28 [00:01<00:00, 27.17it/s]
Unsloth: Saving tokenizer... Done.
Unsloth: Saving model... This might take 5 minutes for Llama-7b...
Unsloth: Saving /content/drive/MyDrive/model/pytorch_model-00001-of-00002.bin...
Unsloth: Saving /content/drive/MyDrive/model/pytorch_model-00002-of-00002.bin...
Done.
```

```
model.push_to_hub_merged("RenZaijay/AjakatlLlama3.2-3B", tokenizer, token="")
```

```
↗ Unsloth: Merging 4bit and LoRA weights to 16bit...
Unsloth: Will use up to 3.48 out of 12.67 RAM for saving.
39%|██████████| 11/28 [00:00<00:00, 29.78it/s]We will save to Disk and not RAM now.
100%|██████████| 28/28 [00:45<00:00, 1.63s/it]
Unsloth: Saving tokenizer...

100% 1/1 [00:01<00:00, 1.16s/it]

tokenizer.json: 32.0M/? [00:00<00:00, 41.2MB/s]

Done.
Unsloth: Saving model... This might take 5 minutes for Llama-7b...
Unsloth: Saving /content/drive/MyDrive/model/pytorch_model-00001-of-00002.bin...
```