

```

from transformers import BertForQuestionAnswering, BertTokenizer
import torch
from PyPDF2 import PdfReader

# Cargar modelo y tokenizador
model_name = 'bert-large-uncased-whole-word-masking-finetuned-squad'
model = BertForQuestionAnswering.from_pretrained(model_name)
tokenizer = BertTokenizer.from_pretrained(model_name)

Some weights of the model checkpoint at bert-large-uncased-whole-word-
masking-finetuned-squad were not used when initializing
BertForQuestionAnswering: ['bert.pooler.dense.weight',
'bert.pooler.dense.bias']
- This IS expected if you are initializing BertForQuestionAnswering
from the checkpoint of a model trained on another task or with another
architecture (e.g. initializing a BertForSequenceClassification model
from a BertForPreTraining model).
- This IS NOT expected if you are initializing
BertForQuestionAnswering from the checkpoint of a model that you
expect to be exactly identical (initializing a
BertForSequenceClassification model from a
BertForSequenceClassification model).

def extract_text_from_txt(txt_path, max_length=512):
    with open(txt_path, 'r', encoding='utf-8') as file:
        text = file.read()

    # Dividir el texto en fragmentos
    fragments = [text[i:i+max_length] for i in range(0, len(text),
max_length)]
    return fragments

# Función para realizar predicciones de preguntas y respuestas
def get_qa_predictions(context, questions):
    predictions = []
    for question in questions:
        inputs = tokenizer(question, context, return_tensors='pt')
        start_positions, end_positions = model(**inputs).values()
        start_index = torch.argmax(start_positions)
        end_index = torch.argmax(end_positions) + 1
        answer =
tokenizer.convert_tokens_to_string(tokenizer.convert_ids_to_tokens(inp
uts['input_ids'][0][start_index:end_index]))
        predictions.append(answer)
    return predictions

# Ruta al archivo de texto (.txt)
txt_path = 'hamlet.txt'

# Extraer texto del archivo de texto

```

```

txt_text = extract_text_from_txt(txt_path)

# Lista de preguntas
preguntas = [
    "Who is hamlet?",
    "What is a castle?",
    "Who is Fortinbras?",
    "Was Hamlet really in love with Ophelia?",
    "Where does the story take place?",
    "Who is Horatio?",
    "Who is Claudius?",
    "Who is Gertrude?",
    "What is the name of the castle?",
    "What is python?",
    "Quien es hamlet?",
    "¿Que es un castillo?",
    "¿Quien es Fortinbras?",
    "¿Hamlet estaba realmente enamorado de Ophelia?",
    "¿En donde se pasa la historia?",
    "¿Quien es Horatio?",
    "¿Quien es Claudius?",
    "¿Quien es Gertrude?",
    "¿Cual es el nombre del castillo?",
    "¿Que es python?"
]

# Obtener fragmentos de texto del archivo de texto
txt_fragments = extract_text_from_txt(txt_path)

# Inicializar lista para almacenar todas las respuestas
respuestas = []

# Iterar sobre los fragmentos y obtener predicciones
for fragment in txt_fragments:
    respuestas_fragmento = get_qa_predictions(fragment, preguntas)

    respuestas.extend(respuestas_fragmento)

# Imprimir preguntas y respuestas
for i in range(len(preguntas)):
    print(f"Pregunta {i + 1}: {preguntas[i]}")
    print(f"Respuesta {i + 1}: {respuestas[i]}")

Pregunta 1: Who is hamlet?
Respuesta 1: prince of denmark
Pregunta 2: What is a castle?
Respuesta 2: hamlet , prince of denmark
Pregunta 3: Who is Fortinbras?
Respuesta 3: francisco
Pregunta 4: Was Hamlet really in love with Ophelia?

```

Respuesta 4: hamlet , son to the former , and nephew to the present king .  
Pregunta 5: Where does the storie take place?  
Respuesta 5: the tragedy of hamlet , prince of denmark  
Pregunta 6: Who is Horatio?  
Respuesta 6: friend to hamlet  
Pregunta 7: Who is Claudius?  
Respuesta 7: king of denmark  
Pregunta 8: Who is Gertrude?  
Respuesta 8: hamlet  
Pregunta 9: What is the name of the castle?  
Respuesta 9: hamlet , prince of denmark  
Pregunta 10: What is python?  
Respuesta 10: the tragedy of hamlet , prince of denmark  
Pregunta 11: Quien es hamlet?  
Respuesta 11: hamlet , son to the former , and nephew to the present king  
Pregunta 12: ¿Que es un castillo?  
Respuesta 12: francisco  
Pregunta 13: ¿Quien es Fortinbras?  
Respuesta 13: francisco  
Pregunta 14: ¿Hamlet estaba realmente enamorado de Ophelia?  
Respuesta 14: hamlet  
Pregunta 15: ¿En donde se pasa la historia?  
Respuesta 15: the tragedy of hamlet , prince of denmark  
Pregunta 16: ¿Quien es Horatio?  
Respuesta 16: friend to hamlet  
Pregunta 17: ¿Quien es Claudius?  
Respuesta 17: claudius , king of denmark  
Pregunta 18: ¿Quien es Gertrude?  
Respuesta 18:  
Pregunta 19: ¿Cual es el nombre del castillo?  
Respuesta 19: francisco  
Pregunta 20: ¿Que es python?  
Respuesta 20: hamlet

#1¿Hubo alguna diferencia? Si, las respuestas en el idioma que no fue entrenado no llegan a ser coherentes.

#2¿Que lenguaje conviene mas y por que? Conviene mas en el lenguaje del que es el texto de origen, ya que presenta mayor coherencia a la hora de contestar, siendo con el otro idioma tener respuestas en blanco.

#3¿Cual era el tamaño del corpus? Fue un .txt de 188 KB.

#4¿Cuántas respuestas tienen coherencia? Cuatro respuestas en el idioma original y 1 en el otro idioma.

#5¿Si cambia el corpus y pregunta lo mismo recibirá una respuesta? No, ya que el texto a entrenar es distinto, y lo comprobe ya que lo hice con con otro corpus en español y fueron distintas las respuestas.

#6¿Cuántos lenguajes puede manejar el BERT para resolver preguntas? Con coherencia solo el idioma original, pero puede contestar en otro lenguaje pero es muy poco probable que de algo coherente.