Entrenamiento de un transformer para Q&A

Equipo 5:

- Héctor Manuel Cárdenas Yáñez
- Alejandro Pizarro Chávez
- Diego Rosas
- Fausto Alejandro Palma Cervantes
- Alan Ricardo Vilchis Arceo

Corpus: Caperucita Roja

Cuento de hadas, adaptación de Watty Piper

Objetivo

El objetivo de esta actividad es emplear un modelo preentrenado 'BertForQuestionAnswering' para responder diez preguntas planteadas sobre un corpus elegido por nosotros. En nuestro caso, hemos decidido utilizar el cuento de hadas Caperucita Roja debido a su simplicidad y a que es ampliamente conocido. Buscamos obtener respuestas en español e inglés utilizando el mismo modelo preentrenado, y evaluaremos las diferencias entre las respuestas generadas en ambos idiomas, su coherencia y el impacto en el tamaño del corpus.

Implementación del código

Importar bibliotecas y descargar el corpus

In [1]:
```
import torch
from transformers import BertForQuestionAnswering, BertTokenizer
import requests
```

In [2]:
```
# Descargamos nuestro corpus
url = "https://raw.githubusercontent.com/AlejandroPizarroo/Transformer-Q-A-Corpus/main/LITTLE%20RED%20RIDING%20HOOD"
response = requests.get(url)
text = response.text

# Imprimir una parte del texto
print(text[:500])
```

```
The Project Gutenberg eBook, Children's Hour with Red Riding Hood and
Other Stories, Edited by Watty Piper


This eBook is for the use of anyone anywhere at no cost and with
almost no restrictions whatsoever.  You may copy it, give it away or
re-use it under the terms of the Project Gutenberg License included
with this eBook or online at www.gutenberg.org




Title: Children's Hour with Red Riding Hood and Other Stories

Editor: Watty Piper

Release Date: March 15, 2004  [eBook #11592]

Languag
```

Preprocesamiento del texto

In [3]:
```
# Dividir el texto en fragmentos mas pequeños
max_chunk = 4000
chunks = [text[i:i+max_chunk] for i in range(0, len(text), max_chunk)]
```

In [4]:
```
# Tokenizacion del texto
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
inputs = tokenizer(chunks, return_tensors='pt', padding=True, truncation=True)
```

Definir las preguntas

In [5]:
```python
# Definir las preguntas
questions = [
    "What color is Little Red Riding Hood's hood?",
    "Who visits the grandmother before Little Red Riding Hood in the story?",
    "What does Little Red Riding Hood carry in her basket for her grandmother?",
    "Who warns Little Red Riding Hood about talking to strangers in the woods?",
    "Who did the wolf pretend to be when he reached the grandmother's cottage?",
    "Where did Little Red Riding Hood's grandmother live?",
    "Who ultimately rescued Little Red Riding Hood from the wolf?",
    "What observations did Little Red Riding Hood make about the wolf disguised as her grandmother?",
    "How did the story end for the wolf?",
    "Who wrote this version of the fairy tale of Little Red Riding Hood?"
]
```

Entrenamiento del modelo

In [ ]:
```python
# Entrenar el modelo BertForQuestionAnswering
model = BertForQuestionAnswering.from_pretrained('bert-large-uncased-whole-word-masking-finetuned-squad')  # Modelo
answers = []

for question in questions:
    answers_for_question = []
    for i in range(len(chunks)):
        input_ids = tokenizer.encode(question, chunks[i], max_length=512, truncation=True, return_tensors='pt')
        with torch.no_grad():
            output = model(input_ids=input_ids)

        start_scores = output.start_logits
        end_scores = output.end_logits

        all_tokens = tokenizer.convert_ids_to_tokens(input_ids[0])
        # Filtrar los tokens especiales [SEP] y [CLS] de la respuesta
        answer_tokens = all_tokens[torch.argmax(start_scores) : torch.argmax(end_scores) + 1]
        answer_tokens = [token for token in answer_tokens if token not in ['[CLS]', '[SEP]']]

        answer = tokenizer.convert_tokens_to_string(answer_tokens)
```

```
        answers_for_question.append(answer.strip())

    # Unir las respuestas de los fragmentos en una sola respuesta para la pregunta
    final_answer = ' '.join(answers_for_question)
    answers.append(final_answer)
```

In [7]:
```
# Imprimir respuestas
for i, answer in enumerate(answers):
    print(f"Question: {questions[i]}")
    # Imprimir solo la primera oración como respuesta
    sentences = answer.split('.')
    print(f"Answer: {sentences[0]}.")
```

```
Question: What color is Little Red Riding Hood's hood?
Answer: red wolf.
Question: Who visits the grandmother before Little Red Riding Hood in the story?
Answer: mother father.
Question: What does Little Red Riding Hood carry in her basket for her grandmother?
Answer: eggs , butter and cake .
Question: Who warns Little Red Riding Hood about talking to strangers in the woods?
Answer: mother .
Question: Who did the wolf pretend to be when he reached the grandmother's cottage?
Answer:  little red riding hood.
Question: Where did Little Red Riding Hood's grandmother live?
Answer: at the further end of the wood was another pretty cottage the cottage.
Question: Who ultimately rescued Little Red Riding Hood from the wolf?
Answer:  little red riding hood ' s father.
Question: What observations did Little Red Riding Hood make about the wolf disguised as her grandmother?
Answer:  everybody was happy that little red riding hood had escaped the wolf.
Question: How did the story end for the wolf?
Answer:  chopped off mr .
Question: Who wrote this version of the fairy tale of Little Red Riding Hood?
Answer: watty piper .
```

Obtener las respuestas de esas 10 preguntas en español e inglés

In [8]:
```
# Definir las preguntas
questions = [
    "What color is Little Red Riding Hood's hood?",
    "Who visits the grandmother before Little Red Riding Hood in the story?",
```

```
    "What does Little Red Riding Hood carry in her basket for her grandmother?",
    "Who warns Little Red Riding Hood about talking to strangers in the woods?",
    "Who did the wolf pretend to be when he reached the grandmother's cottage?",
    "Where did Little Red Riding Hood's grandmother live?",
    "Who ultimately rescued Little Red Riding Hood from the wolf?",
    "What observations did Little Red Riding Hood make about the wolf disguised as her grandmother?",
    "How did the story end for the wolf?",
    "Who wrote this version of the fairy tale of Little Red Riding Hood?",
    "¿De qué color es el capuchón de Caperucita Roja?",
    "¿Quién visita a la abuelita antes que Caperucita Roja en la historia?",
    "¿Qué lleva Caperucita Roja en su cesta para su abuela?",
    "¿Quién advierte a Caperucita Roja sobre hablar con extraños en el bosque?",
    "¿A quién fingió ser el lobo cuando llegó a la cabaña de la abuelita?",
    "¿Dónde vivía la abuela de Caperucita Roja?",
    "¿Quién finalmente rescató a Caperucita Roja del lobo?",
    "¿Qué observaciones hizo Caperucita Roja sobre el lobo disfrazado como su abuela?",
    "¿Cómo terminó la historia para el lobo?",
    "¿Quién escribió esta versión del cuento de hadas de Caperucita Roja?"
]
```

In [ ]:

```
# Entrenar el modelo BertForQuestionAnswering
model = BertForQuestionAnswering.from_pretrained('bert-large-uncased-whole-word-masking-finetuned-squad')  # Modelo
answers = []

for question in questions:
    answers_for_question = []
    for i in range(len(chunks)):
        input_ids = tokenizer.encode(question, chunks[i], max_length=512, truncation=True, return_tensors='pt')
        with torch.no_grad():
            output = model(input_ids=input_ids)

        start_scores = output.start_logits
        end_scores = output.end_logits

        all_tokens = tokenizer.convert_ids_to_tokens(input_ids[0])
        # Filtrar los tokens especiales [SEP] y [CLS] de la respuesta
        answer_tokens = all_tokens[torch.argmax(start_scores) : torch.argmax(end_scores) + 1]
        answer_tokens = [token for token in answer_tokens if token not in ['[CLS]', '[SEP]']]

        answer = tokenizer.convert_tokens_to_string(answer_tokens)
```

```
        answers_for_question.append(answer.strip())

    # Unir las respuestas de los fragmentos en una sola respuesta para la pregunta
    final_answer = ' '.join(answers_for_question)
    answers.append(final_answer)
```

In [10]:
```
# Imprimir respuestas
for i, answer in enumerate(answers):
    print(f"Question: {questions[i]}")
    # Imprimir solo la primera oración como respuesta
    sentences = answer.split('.')
    print(f"Answer: {sentences[0]}.")
```

```
Question: What color is Little Red Riding Hood's hood?
Answer: red wolf.
Question: Who visits the grandmother before Little Red Riding Hood in the story?
Answer: mother father.
Question: What does Little Red Riding Hood carry in her basket for her grandmother?
Answer: eggs , butter and cake .
Question: Who warns Little Red Riding Hood about talking to strangers in the woods?
Answer: mother .
Question: Who did the wolf pretend to be when he reached the grandmother's cottage?
Answer:  little red riding hood.
Question: Where did Little Red Riding Hood's grandmother live?
Answer: at the further end of the wood was another pretty cottage the cottage.
Question: Who ultimately rescued Little Red Riding Hood from the wolf?
Answer:  little red riding hood ' s father.
Question: What observations did Little Red Riding Hood make about the wolf disguised as her grandmother?
Answer:  everybody was happy that little red riding hood had escaped the wolf.
Question: How did the story end for the wolf?
Answer:  chopped off mr .
Question: Who wrote this version of the fairy tale of Little Red Riding Hood?
Answer: watty piper .
Question: ¿De qué color es el capuchón de Caperucita Roja?
Answer: red .
Question: ¿Quién visita a la abuelita antes que Caperucita Roja en la historia?
Answer:  everybody was happy that little red riding hood had escaped the wolf .
Question: ¿Qué lleva Caperucita Roja en su cesta para su abuela?
Answer:  .
Question: ¿Quién advierte a Caperucita Roja sobre hablar con extraños en el bosque?
Answer:  .
```

```
Question: ¿A quién fingió ser el lobo cuando llegó a la cabaña de la abuelita?
Answer:  .
Question: ¿Dónde vivía la abuela de Caperucita Roja?
Answer:  everybody was happy that little red riding hood had escaped the wolf .
Question: ¿Quién finalmente rescató a Caperucita Roja del lobo?
Answer:  .
Question: ¿Qué observaciones hizo Caperucita Roja sobre el lobo disfrazado como su abuela?
Answer: almost no restrictions .
Question: ¿Cómo terminó la historia para el lobo?
Answer:  .
Question: ¿Quién escribió esta versión del cuento de hadas de Caperucita Roja?
Answer:  .
```

Resultados

¿Hubo alguna diferencia?

- La diferencia que se encuentra entre las preguntas en español e inglés es sumamente evidente. En las preguntas en inglés, el transformer logra contestar de manera correcta o parcialmente correcta nueve de las diez preguntas que se le realizaron. Sin embargo, en las preguntas en español solamente logra contestar de manera correcta la primera pregunta, la cual resulta ser la más sencilla de todas. Es verdad que responde a otras tres preguntas más, pero su respuesta no tiene sentido alguno, y en las otras preguntas simplemente no da una respuesta clara y concisa.

¿Qué lenguaje conviene más y por qué?

- En nuestro caso de prueba, el uso del idioma inglés resulta mucho más conveniente, principalmente debido al modelo que empleamos: el modelo 'bert-large-uncased-whole-word-masking-finetuned-squad', el cual se basa en BERT (Bidirectional Encoder Representations from Transformers) y está preentrenado en inglés. Esta es la razón principal por la que buscamos un corpus en inglés, para que fuera coherente con el modelo preentrenado que seleccionamos. Esto se puede evidenciar en los resultados obtenidos de las preguntas formuladas tanto en español como en inglés.

¿Cuál era el tamaño del corpus?

- El corpus consta de 741 palabras y 4,230 caracteres. Es importante mencionar que elegimos un corpus significativamente más pequeño. En las pruebas iniciales, habíamos optado por un corpus más extenso y complejo, pero nos enfrentamos a resultados extremadamente lentos de obtener. Además, carecían por completo de coherencia ya que mientras más complejo era el libro, menos

sentido tenían las respuestas que obteníamos. Los corpus que probamos anteriormente fueron 'La Metamorfosis' de Franz Kafka y 'Rebelión en la Granja' de George Orwell.

¿Cuántas respuestas tienen coherencia?

- De las diez preguntas formuladas en inglés, nueve muestran coherencia y tienen una respuesta correcta o parcialmente correcta.

Cambio de corpus, mismas preguntas

In [11]:
```
# Descargamos nuestro corpus
url = "https://gutenberg.net.au/ebooks01/0100011.txt"
response = requests.get(url)
text = response.text

# Imprimir una parte del texto
print(text[:500])
```

```
Project Gutenberg Australia



Title:      Animal Farm
Author:     George Orwell (pseudonym of Eric Blair) (1903-1950)
* A Project Gutenberg of Australia eBook *
eBook No.:  0100011.txt
Language:   English
Date first posted:          August 2001
Date most recently updated: March 2008

Project Gutenberg of Australia eBooks are created from printed editions
which are in the public domain in Australia, unless a copyright notice
is included. We do NOT keep any eBooks in compliance with a particula
```

In [12]:
```
# Dividir el texto en fragmentos mas pequeños
max_chunk = 4000
chunks = [text[i:i+max_chunk] for i in range(0, len(text), max_chunk)]
```

In [13]:
```python
# Definir las preguntas
questions = [
    "What color is Little Red Riding Hood's hood?",
    "Who visits the grandmother before Little Red Riding Hood in the story?",
    "What does Little Red Riding Hood carry in her basket for her grandmother?",
    "Who warns Little Red Riding Hood about talking to strangers in the woods?",
    "Who did the wolf pretend to be when he reached the grandmother's cottage?",
    "Where did Little Red Riding Hood's grandmother live?",
    "Who ultimately rescued Little Red Riding Hood from the wolf?",
    "What observations did Little Red Riding Hood make about the wolf disguised as her grandmother?",
    "How did the story end for the wolf?",
    "Who wrote this version of the fairy tale of Little Red Riding Hood?"
]
```

In [ ]:
```python
# Entrenar el modelo BertForQuestionAnswering
model = BertForQuestionAnswering.from_pretrained('bert-large-uncased-whole-word-masking-finetuned-squad')  # Modelo
answers = []

for question in questions:
    answers_for_question = []
    for i in range(len(chunks)):
        input_ids = tokenizer.encode(question, chunks[i], max_length=512, truncation=True, return_tensors='pt')
        with torch.no_grad():
            output = model(input_ids=input_ids)

        start_scores = output.start_logits
        end_scores = output.end_logits

        all_tokens = tokenizer.convert_ids_to_tokens(input_ids[0])
        # Filtrar los tokens especiales [SEP] y [CLS] de la respuesta
        answer_tokens = all_tokens[torch.argmax(start_scores) : torch.argmax(end_scores) + 1]
        answer_tokens = [token for token in answer_tokens if token not in ['[CLS]', '[SEP]']]

        answer = tokenizer.convert_tokens_to_string(answer_tokens)
        answers_for_question.append(answer.strip())

    # Unir las respuestas de los fragmentos en una sola respuesta para la pregunta
    final_answer = ' '.join(answers_for_question)
    answers.append(final_answer)
```

In [15]:
```python
# Imprimir respuestas
for i, answer in enumerate(answers):
    print(f"Question: {questions[i]}")
    # Imprimir solo la primera oración como respuesta
    sentences = answer.split('.')
    print(f"Answer: {'.'.join(sentences[:5])}.")
```

```
Question: What color is Little Red Riding Hood's hood?
Answer:                                                .
Question: Who visits the grandmother before Little Red Riding Hood in the story?
Answer:                                                .
Question: What does Little Red Riding Hood carry in her basket for her grandmother?
Answer:                                                .
Question: Who warns Little Red Riding Hood about talking to strangers in the woods?
Answer:                                                .
Question: Who did the wolf pretend to be when he reached the grandmother's cottage?
Answer:                                                .
Question: Where did Little Red Riding Hood's grandmother live?
Answer:                                                .
Question: Who ultimately rescued Little Red Riding Hood from the wolf?
Answer:       .      who ultimately rescued little red riding hood from the wolf ? as they imagined , their enemies
in flight , and they rushed after them in disorder . this was just what snowball had intended . as soon as they were
well inside the yard , the three horses , the three cows , and the rest of the pigs , who had been lying in ambush i
n the cowshed , suddenly emerged in their rear , cutting them off . snowball now gave the signal for the charge .
Question: What observations did Little Red Riding Hood make about the wolf disguised as her grandmother?
Answer:                                                .
Question: How did the story end for the wolf?
Answer:       .                                         .
Question: Who wrote this version of the fairy tale of Little Red Riding Hood?
Answer: eric blair            snowball                                 .
```

¿Si cambia el corpus y pregunta lo mismo recibirá una respuesta? Demuestre

- Después de cambiar el corpus y realizar las mismas preguntas, descubrimos que de las diez, solo se pueden responder dos, y estas carecen de sentido alguno. En la pregunta siete, se intenta desarrollar una idea, pero al final, el texto carece de coherencia, pareciendo simplemente tomar un fragmento del corpus. Es importante mencionar que se limitó la salida de las respuestas para mantener la legibilidad, lo que nos deja solo con las primeras ideas. No obstante, esto proporciona suficiente evidencia para determinar que el modelo preentrenado no puede responder preguntas sobre el cuento de hadas Caperucita Roja cuando se utiliza un corpus de 'Rebelión en la Granja' de George Orwell.

¿Cuántos lenguajes puede manejar el BERT para resolver preguntas?

- BERT puede entender y procesar preguntas en más de 100 idiomas diferentes, aunque su entrenamiento principal se llevó a cabo en inglés. Es importante mencionar que al menos en nuestra demostración recomendaría hacer las preguntas en el mismo idioma en el que está el corpus ya que de esta manera se obtienen resultados más coherentes y precisos.

Conclusiones

Tras realizar múltiples pruebas con el modelo preentrenado BERT, hemos descubierto que constituye una práctica excelente para el desarrollo de un transformer de preguntas y respuestas (Q&A). Si bien es cierto que hoy en día estamos acostumbrados a herramientas como ChatGPT, que nos ofrecen respuestas más precisas y coherentes gracias a su sofisticación, BERT, siendo un modelo más básico, puede proporcionar resultados aceptables.

In [ ]: