Question 1

```
# Install required libraries
```

```
!pip install nltk
```

```
Requirement already satisfied: nltk in /usr/local/lib/python3.11/dist-packages (3.9.1)
Requirement already satisfied: click in /usr/local/lib/python3.11/dist-packages (from nltk) (8.1.8)
Requirement already satisfied: joblib in /usr/local/lib/python3.11/dist-packages (from nltk) (1.4.2)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.11/dist-packages (from nltk) (2024.11.6)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from nltk) (4.67.1)
```

```
!pip install sacrebleu
```

```
Requirement already satisfied: sacrebleu in /usr/local/lib/python3.11/dist-packages (2.5.1)
Requirement already satisfied: portalocker in /usr/local/lib/python3.11/dist-packages (from sacrebleu) (3.1.1)
Requirement already satisfied: regex in /usr/local/lib/python3.11/dist-packages (from sacrebleu) (2024.11.6)
Requirement already satisfied: tabulate>=0.8.9 in /usr/local/lib/python3.11/dist-packages (from sacrebleu) (0.9.0)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.11/dist-packages (from sacrebleu) (1.26.4)
Requirement already satisfied: colorama in /usr/local/lib/python3.11/dist-packages (from sacrebleu) (0.4.6)
Requirement already satisfied: lxml in /usr/local/lib/python3.11/dist-packages (from sacrebleu) (5.3.1)
```

```
!pip install transformers torch
```

```
Downloading nvidia_cuda_nvrtc_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl (24.6 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 24.6/24.6 MB 20.0 MB/s eta 0:00:00
Downloading nvidia_cuda_runtime_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl (883 kB)
```

```
    Found existing installation: nvidia-cusolver-cu12 11.6.3.83
    Uninstalling nvidia-cusolver-cu12-11.6.3.83:
      Successfully uninstalled nvidia-cusolver-cu12-11.6.3.83
Successfully installed nvidia-cublas-cu12-12.4.5.8 nvidia-cuda-cupti-cu12-12.4.127 nvidia-cuda-nvrtc-cu12-12.4.127 nvidia-cu
```

```
!pip install datasets
```

```
Collecting datasets
  Downloading datasets-3.3.2-py3-none-any.whl.metadata (19 kB)
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (from datasets) (3.17.0)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.11/dist-packages (from datasets) (1.26.4)
Requirement already satisfied: pyarrow>=15.0.0 in /usr/local/lib/python3.11/dist-packages (from datasets) (18.1.0)
Collecting dill<0.3.9,>=0.3.0 (from datasets)
  Downloading dill-0.3.8-py3-none-any.whl.metadata (10 kB)
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (from datasets) (2.2.2)
Requirement already satisfied: requests>=2.32.2 in /usr/local/lib/python3.11/dist-packages (from datasets) (2.32.3)
Requirement already satisfied: tqdm>=4.66.3 in /usr/local/lib/python3.11/dist-packages (from datasets) (4.67.1)
Collecting xxhash (from datasets)
  Downloading xxhash-3.5.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (12 kB)
Collecting multiprocess<0.70.17 (from datasets)
  Downloading multiprocess-0.70.16-py311-none-any.whl.metadata (7.2 kB)
Requirement already satisfied: fsspec<=2024.12.0,>=2023.1.0 in /usr/local/lib/python3.11/dist-packages (from fsspec[http]<=2
Requirement already satisfied: aiohttp in /usr/local/lib/python3.11/dist-packages (from datasets) (3.11.13)
Requirement already satisfied: huggingface-hub>=0.24.0 in /usr/local/lib/python3.11/dist-packages (from datasets) (0.28.1)
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (from datasets) (24.2)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.11/dist-packages (from datasets) (6.0.2)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (1.3.2)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (25.1.0)
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (1.5.0)
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (6.1.
Requirement already satisfied: propcache>=0.2.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (0.3.0)
Requirement already satisfied: yarl<2.0,>=1.17.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (1.18.3
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests>=2.32.2->d
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests>=2.32.2->datasets) (3.
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests>=2.32.2->dataset
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests>=2.32.2->dataset
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas->datasets) (2.
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas->datasets) (2025.1)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas->datasets) (2025.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas->dat
Downloading datasets-3.3.2-py3-none-any.whl (485 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 485.4/485.4 kB 12.3 MB/s eta 0:00:00
Downloading dill-0.3.8-py3-none-any.whl (116 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 116.3/116.3 kB 8.1 MB/s eta 0:00:00
Downloading multiprocess-0.70.16-py311-none-any.whl (143 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 143.5/143.5 kB 9.9 MB/s eta 0:00:00
Downloading xxhash-3.5.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (194 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 194.8/194.8 kB 11.8 MB/s eta 0:00:00
Installing collected packages: xxhash, dill, multiprocess, datasets
Successfully installed datasets-3.3.2 dill-0.3.8 multiprocess-0.70.16 xxhash-3.5.0
```

```
!pip install sacrebleu
```

```
Collecting sacrebleu
  Downloading sacrebleu-2.5.1-py3-none-any.whl.metadata (51 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 51.8/51.8 kB 3.4 MB/s eta 0:00:00
Collecting portalocker (from sacrebleu)
  Downloading portalocker-3.1.1-py3-none-any.whl.metadata (8.6 kB)
Requirement already satisfied: regex in /usr/local/lib/python3.11/dist-packages (from sacrebleu) (2024.11.6)
Requirement already satisfied: tabulate>=0.8.9 in /usr/local/lib/python3.11/dist-packages (from sacrebleu) (0.9.0)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.11/dist-packages (from sacrebleu) (1.26.4)
Collecting colorama (from sacrebleu)
  Downloading colorama-0.4.6-py2.py3-none-any.whl.metadata (17 kB)
Requirement already satisfied: lxml in /usr/local/lib/python3.11/dist-packages (from sacrebleu) (5.3.1)
Downloading sacrebleu-2.5.1-py3-none-any.whl (104 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 104.1/104.1 kB 6.8 MB/s eta 0:00:00
Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Downloading portalocker-3.1.1-py3-none-any.whl (19 kB)
Installing collected packages: portalocker, colorama, sacrebleu
Successfully installed colorama-0.4.6 portalocker-3.1.1 sacrebleu-2.5.1
```

a. Setup

```
import pandas as pd
import numpy as np
from transformers import MarianMTModel, MarianTokenizer
import datasets
import sacrebleu
```

```
# Loading the models

model_name = 'Helsinki-NLP/opus-mt-en-es'
model = MarianMTModel.from_pretrained(model_name)
tokenizer = MarianTokenizer.from_pretrained(model_name)
```

```
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
config.json: 100%                                   1.47k/1.47k [00:00<00:00, 65.8kB/s]
pytorch_model.bin: 100%                             312M/312M [00:02<00:00, 162MB/s]
model.safetensors:  87%                             273M/312M [00:02<00:00, 160MB/s]
generation_config.json: 100%                        293/293 [00:00<00:00, 9.67kB/s]
tokenizer_config.json: 100%                         44.0/44.0 [00:00<00:00, 1.95kB/s]
source.spm: 100%                                    802k/802k [00:00<00:00, 20.2MB/s]
target.spm: 100%                                    826k/826k [00:00<00:00, 15.7MB/s]
vocab.json: 100%                                    1.59M/1.59M [00:00<00:00, 16.2MB/s]
/usr/local/lib/python3.11/dist-packages/transformers/models/marian/tokenization_marian.py:175: UserWarning: Recommended: pip
  warnings.warn("Recommended: pip install sacremoses.")
```

```
# Explain why this model is a good choice for the task

'''We are using model Helsinki-NLP/opus-mt-en-es for English-to-Spanish translation,
 as it is good choice because it is a pre-trained for the English-to-Spanish
 translation specifically. This is also publicly available in the Hugging Face models.
Also, we understand that this model is built on MarianMT architecture,
which is optimized for machine translation tasks, which we are working on'''
```

```
'We are using model Helsinki-NLP/opus-mt-en-es for English-to-Spanish translation,\n as it is good choice because it is a p
re-trained for the English-to-Spanish\n translation specifically. This is also publicly available in the Hugging Face model
s.\nAlso, we understand that this model is built on MarianMT architecture,\nwhich is optimized for machine translation task
s, which we are working on'
```

b. **Implementation**

```
# Translation Function
def translate_to_spanish(sentence):
    # Tokenize the input sentence
    inputs = tokenizer.encode(sentence, return_tensors="pt", padding=True, truncation=True)

    # Generate the translation
    translated = model.generate(inputs, max_length=512)

    # Decode the translation to get the Spanish sentence
    translation = tokenizer.decode(translated[0], skip_special_tokens=True)

    return translation

# Batch Processing
def translate_batch_to_spanish(sentences):
    # Tokenize input sentences
    inputs = tokenizer(sentences, return_tensors="pt", padding=True, truncation=True, max_length=512)

    # Generate the translations for all sentences in the batch
    translated = model.generate(inputs['input_ids'], max_length=512)
```

```
    # Decode the translations to get the Spanish sentences
    translations = [tokenizer.decode(t, skip_special_tokens=True) for t in translated]

    return translations



# Example usage
sentences = [
    "Hello, how are you?",
    "I am learning machine translation.",
    "This task is very interesting."
]

# Get the translations
translations = translate_batch_to_spanish(sentences)

# Print original English sentences and their Spanish translations
for eng, esp in zip(sentences, translations):
    print(f"English: {eng}")
    print(f"Spanish: {esp}\n")
```

```
English: Hello, how are you?
Spanish: Hola, ¿cómo estás?

English: I am learning machine translation.
Spanish: Estoy aprendiendo traducción automática.

English: This task is very interesting.
Spanish: Esta tarea es muy interesante.
```

c. Evaluation

```
test_sentences = [
    "I am going to the store.",
    "The weather is amazing today.",
    "She quickly ran across the street.",
    "This is a piece of cake.",
    "I don't understand why he's so upset.",
    "Could you please pass me the salt?",
    "I am feeling under the weather today",
    "We have recorded the video, haven't we?",
    "How much alcohol is too much alcohol",
    "Could I be more sorry",
    "You can't judge a book by its cover."
]



# Translate the test dataset
test_translations = translate_batch_to_spanish(test_sentences)

# Display the results
for eng, esp in zip(test_sentences, test_tranlations):
    print(f"English: {eng}")
    print(f"Spanish: {esp}\n")
```

```
English: I am going to the store.
Spanish: Voy a la tienda.

English: The weather is amazing today.
Spanish: El tiempo es increíble hoy.

English: She quickly ran across the street.
Spanish: Corrió rápidamente a través de la calle.

English: This is a piece of cake.
Spanish: Esto es pan comido.

English: I don't understand why he's so upset.
Spanish: No entiendo por qué está tan molesto.

English: Could you please pass me the salt?
```

```
Spanish: ¿Podrías pasarme la sal, por favor?

English: I am feeling under the weather today
Spanish: Me siento bajo el clima hoy.

English: We have recorded the video, haven't we?
Spanish: Hemos grabado el vídeo, ¿no?

English: How much alcohol is too much alcohol
Spanish: Cuánto alcohol es demasiado alcohol

English: Could I be more sorry
Spanish: ¿Podría sentirlo más?

English: You can't judge a book by its cover.
Spanish: No puedes juzgar un libro por su portada.
```

```
# Analysis

# 1. Simple translations
# I am going to the store = Voy a la tienda.
# Transalation is perpect. Conveys the meaning correctly

# 2. Idioms
# This is a piece of cake = Esto es pan comido.
# "Piece of cake" is correctly translated as "pan comido," which is equivalent idiom in Spanish

# 3. Complex Transalations
# How much alcohol is too much alcohol? = Cuánto alcohol es demasiado alcohol?
# A better translation would be: "¿Cuánto alcohol es demasiado?" to avoid repetition
```

```
test_translations
```

```
['Voy a la tienda.',
 'El tiempo es increíble hoy.',
 'Corrió rápidamente a través de la calle.',
 'Esto es pan comido.',
 'No entiendo por qué está tan molesto.',
 '¿Podrías pasarme la sal, por favor?',
 'Me siento bajo el clima hoy.',
 'Hemos grabado el vídeo, ¿no?',
 'Cuánto alcohol es demasiado alcohol',
 '¿Podría sentirlo más?',
 'No puedes juzgar un libro por su portada.']
```

```
# Reference translations (for comparison)
reference_translations = [
    ['Voy a la tienda.'],
    ['El tiempo es increíble hoy.'],
    ['Corrió rápidamente a través de la calle.'],
    ['Esto es pan comido.'],
    ['No entiendo por qué está tan molesto.'],
    ['¿Podrías pasarme la sal, por favor?'],
    ['Me siento bajo el clima hoy.'],
    ['Hemos grabado el vídeo, ¿no?'],
    ['¿Cuánto alcohol es demasiado?'],
    ['¿Podría sentirlo más?'],
    ['No puedes juzgar un libro por su portada.']
]
```

Using BLEU

```
# BLEU is the most commonly used metric for machine translation evaluation
# It measures precision by comparing n-grams between the reference and test sentences
# It checks the overlap of unigrams, bigrams, trigrams, and four-grams between the reference and test translations


# 0.0: Worst Score (No overlap between the n-grams in test and reference translations)
# 1 means that every n-gram in the test translation matches exactly with reference translation(s)
```

+ Code + Text

```
import sacrebleu
```

```
# Calculate BLEU score for each sentence individually
for i in range(len(reference_translations)):
    bleu_score = sacrebleu.corpus_bleu([test_translations[i]], [reference_translations[i]])
    print(f"BLEU Score for sentence {i+1}: {bleu_score.score}")
```

```
BLEU Score for sentence 1: 100.00000000000004
BLEU Score for sentence 2: 100.00000000000004
BLEU Score for sentence 3: 100.00000000000004
BLEU Score for sentence 4: 100.00000000000004
BLEU Score for sentence 5: 100.00000000000004
BLEU Score for sentence 6: 100.00000000000004
BLEU Score for sentence 7: 100.00000000000004
BLEU Score for sentence 8: 100.00000000000004
BLEU Score for sentence 9: 39.76353643835252
BLEU Score for sentence 10: 100.00000000000004
BLEU Score for sentence 11: 100.00000000000004
```

Using TER (Translation Edit Rate)

```
# Translation Edit Rate (TER) is a comparison metric used to measure the edit distance between a test and reference translation
# It represents the minimum number of edits required to change the test translation into the reference translation


# Range of TER: 0.0: Perfect Translation
# A TER score of 0 means that no edits are necessary to make in test translation
# to match the reference translation


import sacrebleu

# Calculate TER score for each sentence individually
for i in range(len(reference_translations)):
    # Using test_translations instead of hypothesis_translations
    ter_score = sacrebleu.sentence_ter(test_translations[i], reference_translations[i])
    print(f"TER Score for sentence {i+1}: {ter_score}")
```

```
TER Score for sentence 1: TER = 0.00
TER Score for sentence 2: TER = 0.00
TER Score for sentence 3: TER = 0.00
TER Score for sentence 4: TER = 0.00
TER Score for sentence 5: TER = 0.00
TER Score for sentence 6: TER = 0.00
TER Score for sentence 7: TER = 0.00
TER Score for sentence 8: TER = 0.00
TER Score for sentence 9: TER = 75.00
TER Score for sentence 10: TER = 0.00
TER Score for sentence 11: TER = 0.00
```

```
# Strengths and Weaknesses

# BLEU Score
# Strengths:
# The majority of the sentences have BLEU scores of 100%
# Weakness:
# Sentence 9 stands out with a BLEU score of 39.76%, The BLEU score is sensitive to n-gram matches
# In this case, the translation has fewer or less accurate n-grams matching the reference, leading to a lower score

# TER Score (Translation Edit Rate)
# Strengths:
# Sentences having TER score of 0.00, there are no edits were required to align with the reference translations.
# Weakness:
# Sentence 9, however, has a TER score of 75.00, which indicates that a large number of edits (75%) were required.
```

d. **Analysis**

```
# Advantages

# Pre-trained models eliminate the need for training from scratch. These models are trained on diverse datasets,
# allowing them to handle texts and contexts effectively.
# Since there is no training needed, they have reduced computational requirements and enable rapid prototyping

# Limitations
# Pre-trained models have biases from training data and may not adapt to evolving language trends
# Handling Domain-Specific Vocabulary: Struggles with rare words or technical jargon, leading to inaccurate translations
```

```
# Idiomatic & Cultural Nuances: Often translates idioms literally, missing context or cultural meanings

# Suggestions for Improvement
# Training on specialized datasets improves accuracy in specific fields
```