In [ ]:
```
!pip install -U easynmt
!pip install sacrebleu
```

In [ ]:
```
# Import libraries
import pandas as pd
import sacrebleu
from easynmt import EasyNMT
import nltk

nltk.download("punkt_tab")
```

Question 2 (Opus-MT)

Translate Spanish sentences to English using Opus-MT model

In [ ]:
```
# Load opus-mt model
opus_model = EasyNMT("opus-mt")
```

```
100%|██████████| 11.9k/11.9k [00:00<00:00, 3.74MB/s]
```

In [ ]:
```
# Load Spanish dataset
dataset_filepath = (
    "/content/drive/MyDrive/jhu_lab6/flores101.test.es.txt"
)
with open(dataset_filepath, "r", encoding="utf-8") as f:
    spanish_sentences = f.readlines()

# Translate spanish to english
opus_eng_translations = opus_model.translate(
    spanish_sentences, source_lang="es", target_lang="en"
)

# Print first 10 sentences and their translations
sentences = spanish_sentences[:10]
translations = opus_eng_translations[:10]
results_df = pd.DataFrame(
    {
        "Spanish Sentence": sentences,
        "English Translation": translations,
    }
)
results_df
```

Question 2 (M2M100)

Translate Spanish sentences to English using M2M100 model

In [ ]:
```
# Load m2m100 model
m2m_model = EasyNMT("m2m_100_418M")
```

In [ ]:
```
# Load Spanish dataset
dataset_filepath = (
```

```
    "/content/drive/MyDrive/jhu_lab6/flores101.test.es.txt"
)
with open(dataset_filepath, "r", encoding="utf-8") as f:
    spanish_sentences = f.readlines()

# Translate spanish to english
m2m_eng_translations = m2m_model.translate(
    spanish_sentences, source_lang="es", target_lang="en"
)

# Print first 10 sentences and their translations
sentences = spanish_sentences[:10]
translations = m2m_eng_translations[:10]
results_df = pd.DataFrame(
    {
        "Spanish Sentence": sentences,
        "English Translation": translations,
    }
)
results_df
```

Question 3 (Opus-MT)

Evaluate lower-cased scores for full set of Opus-MT translations

In [ ]:
```
# Load reference English translations
reference_filepath = (
    "/content/drive/MyDrive/jhu_lab6/flores101.test.en.txt"
)
with open(reference_filepath, "r", encoding="utf-8") as f:
    reference_eng_sentences = f.readlines()

# Calculate BLEU, chrF and TER score
bleu = sacrebleu.corpus_bleu(
    opus_eng_translations, [reference_eng_sentences]
)
chrf = sacrebleu.corpus_chrf(
    opus_eng_translations, [reference_eng_sentences]
)
ter = sacrebleu.corpus_ter(
    opus_eng_translations, [reference_eng_sentences]
)

print(f"BLEU: {bleu.score}")
print(f"chrF: {chrf.score}")
print(f"TER: {ter.score}")
```

Question 3 (M2M100)

Evaluate lower-cased scores for full set of M2M100 translations

In [ ]:
```
# Load reference English translations
reference_filepath = (
    "/content/drive/MyDrive/jhu_lab6/flores101.test.en.txt"
```

```
)
with open(reference_filepath, "r", encoding="utf-8") as f:
    reference_eng_sentences = f.readlines()

# Calculate BLEU, chrF and TER score
bleu = sacrebleu.corpus_bleu(
    m2m_eng_translations, [reference_eng_sentences]
)
chrf = sacrebleu.corpus_chrf(
    m2m_eng_translations, [reference_eng_sentences]
)
ter = sacrebleu.corpus_ter(
    m2m_eng_translations, [reference_eng_sentences]
)

print(f"BLEU: {bleu.score}")
print(f"chrF: {chrf.score}")
print(f"TER: {ter.score}")
```

Question 4

Translate English sentences to Spanish, then French, then Russian. Finally translate Russian back to English.

In [ ]:
```
# Use opus-mt model to translate English to Spanish, then French, then Ru
sentences = [
    "Berlin straddles the banks of the River Spree, which flows into the
    "Among the city's main topographical features are the many lakes in t
    "Due to its location in the European Plain, Berlin is influenced by a
]

translations = [
    ("en", "es"),
    ("es", "fr"),
    ("fr", "ru"),
    ("ru", "en"),
]
for trans in translations:
    sentences = opus_model.translate(
        sentences,
        source_lang=trans[0],
        target_lang=trans[1],
    )

    print(
        f"Source lang: {trans[0]}, Target lang: {trans[1]}..."
    )
    for s in sentences:
        print(f"\nTranslated: {s}\n")
```