1. Data preparation (5points)

1) Data download and preprocessing: Download the Flores200 dataset, which is a benchmark data for machine translation between English and low-resource languages.

In [1]:

```python
from datasets import load_dataset
dataset = load_dataset("Muennighoff/flores200", 'eng_Latn-fra_Latn')
```

In [2]:

```python
import random

selected_data = dataset['dev']

# Set a fixed seed for reproducibility
random.seed(123)

# Randomly choose 100 distinct items
rndm_indices = random.sample(range(len(selected_data)), 100)
rndm_samples = selected_data.select(rndm_indices)
```

2. Machine Translation with Seq2Seq model (65 points, 20 points for each model, 5 points for data statistics)

Please use the following models to perform machine translation on the data you prepared. • Feel free to directly utilize the existing implementations on Hugging Face. from datasets import load_dataset dataset = load_dataset("Muennighoff/flores200") 2 • There is no specific requirement for the parameter settings. You are encouraged to try and test different settings and report the results. For other settings that are not specified here, you have the flexibility to select. • If you want to perform the translation more efficiently with the following models, you can consider using CTranslate2. Its Github repository is available here. The documentation of the package is available here. M2M-100 and MBART-50 are under Fairseq. • During the implementation, for each model, you will need to specify the source language and target language or select the specific model for your source and target languages.

MODEL 1 (OPUS-MT)

In [3]:

```python
# Import necessary modules from the transformers library
from transformers import AutoTokenizer, AutoModelForSeq2SeqLM

# Initialize the tokenizer for the English to French translation model
tokenizer = AutoTokenizer.from_pretrained("Helsinki-NLP/opus-mt-en-fr")
```

```
# Initialize the Seq2Seq model for the English to French translation
model = AutoModelForSeq2SeqLM.from_pretrained("Helsinki-NLP/opus-mt-en-fr")
```

In [4]:

```
def translate_text_1(input_text):
    # Tokenize input
    tokenized_text = tokenizer(input_text, return_tensors="pt", padding=True)

    # Generate translation
    translation_output = model.generate(**tokenized_text)

    # Decode and return translation
    return tokenizer.decode(translation_output[0], skip_special_tokens=True)
```

In [5]:

```
english=[]
# Use list comprehension to translate all English texts to French
translated_texts_1= [translate_text_1(item['sentence_eng_Latn']) for item in rndm_samp
english = [item['sentence_eng_Latn'] for item in rndm_samples]

# Extract all original French texts
original_texts = [item['sentence_fra_Latn'] for item in rndm_samples]
```

MODEL 2 (M2M-100)

In [6]:

```
# Import necessary modules from the transformers library
from transformers import AutoTokenizer, AutoModelForSeq2SeqLM

# Initialize the tokenizer for the m2m100_418M model
m2m_tokenizer = AutoTokenizer.from_pretrained("facebook/m2m100_418M")

# Initialize the Seq2Seq model for the m2m100_418M model
m2m_model = AutoModelForSeq2SeqLM.from_pretrained("facebook/m2m100_418M")

# Set the source language for the tokenizer
m2m_tokenizer.src_lang = "en"
```

In [7]:

```
def translate_text_2(model_name, input_text):
    # Tokenize the input text for translation
    encoded_text = m2m_tokenizer(input_text, return_tensors="pt", padding=True)

    # Generate the translation output
    translation_output = m2m_model.generate(**encoded_text, forced_bos_token_id=m2m_to

    # Return the decoded translated text
    return m2m_tokenizer.batch_decode(translation_output, skip_special_tokens=True)[0]
```

In [8]:

```
# Use list comprehension to translate all English texts to French
translated_texts_2 = [translate_text_2("facebook/m2m100_418M",\
                                       item['sentence_eng_Latn']) for item in rndm_san
```

Model 3 (MBART-50)

In [9]:

```
# Import necessary modules from the transformers library
from transformers import AutoTokenizer, AutoModelForSeq2SeqLM
```

```
# Initialize the tokenizer for the mBART-50 model
mbart_tokenizer = AutoTokenizer.from_pretrained("facebook/mbart-large-50-many-to-many-

# Initialize the Seq2Seq model for the mBART-50 model
mbart_model = AutoModelForSeq2SeqLM.from_pretrained("facebook/mbart-large-50-many-to-m

# Set the source language for the tokenizer
mbart_tokenizer.src_lang = "en_XX"
```

In [10]:

```
def translate_text_3(input_text):
    # Tokenize the input text for translation
    encoded_text = mbart_tokenizer(input_text, return_tensors="pt", padding=True)

    # Generate the translation output
    translation_output = mbart_model.generate(**encoded_text, forced_bos_token_id=mbar

    # Return the decoded translated text
    return mbart_tokenizer.batch_decode(translation_output, skip_special_tokens=True)[
```

In [11]:

```
# Initialize list for storing translations
translated_texts_3 = []

# Translate each text in the random samples
for item in rndm_samples:
    # Extract English and original French text
    eng_text = item['sentence_eng_Latn']


    # Translate English text to French
    trans_french = translate_text_3(eng_text)

    # Append translated French text to the list
    translated_texts_3.append(trans_french)
```

Data statistics

In [13]:

```
import statistics

# Calculate lengths of English and French sentences
eng_lengths = [len(mbart_tokenizer(item['sentence_eng_Latn']).input_ids) for item in r
french_lengths = [len(mbart_tokenizer(item['sentence_fra_Latn']).input_ids) for item i

# Compute basic statistics for English sentences
min_length_eng = min(eng_lengths)
avg_length_eng = sum(eng_lengths) / len(eng_lengths)
max_length_eng = max(eng_lengths)
median_length_eng = statistics.median(eng_lengths)
std_dev_length_eng = statistics.stdev(eng_lengths)
variance_length_eng = statistics.variance(eng_lengths)
mode_length_eng = statistics.mode(eng_lengths)

# Compute basic statistics for French sentences
min_length_french = min(french_lengths)
avg_length_french = sum(french_lengths) / len(french_lengths)
max_length_french = max(french_lengths)
median_length_french = statistics.median(french_lengths)
```

```
std_dev_length_french = statistics.stdev(french_lengths)
variance_length_french = statistics.variance(french_lengths)
mode_length_french = statistics.mode(french_lengths)

print(f"English Sentences: \nMin Length = {min_length_eng}, \nAverage Length = {avg_le
      \nMax Length = {max_length_eng}, \nMedian Length = {median_length_eng}, \nStanda
      \nVariance = {variance_length_eng}, \nMode = {mode_length_eng}\n\n")
print(f"French Sentences: \nMin Length = {min_length_french}, \nAverage Length = {avg_
      \nMax Length = {max_length_french}, \nMedian Length = {median_length_french}, \
      \nStandard Deviation = {std_dev_length_french}, \nVariance = {variance_length_fr
      \nMode = {mode_length_french}")
```

```
English Sentences:
Min Length = 12,
Average Length = 31.48,
Max Length = 69,
Median Length = 31.0,
Standard Deviation = 9.701067316058943,
Variance = 94.11070707070706,
Mode = 31


French Sentences:
Min Length = 13,
Average Length = 39.27,
Max Length = 72,
Median Length = 38.0,
Standard Deviation = 11.922184566952687,
Variance = 142.13848484848484,
Mode = 33
```

3. Results analysis and evaluation (30 points, 10 points for each sub-question)

1) Please use the BLUE score to evaluate the performance of the models. You can find the details and examples about how to get BLUE scores in this link.

In [14]:

```
# Import the evaluation module
import evaluate

# Initialize the BLEU metric
bleu_metric = evaluate.load("bleu")

# Compute the BLEU score for OPUS-MT model
bleu_score = bleu_metric.compute(predictions=translated_texts_1, references=original_t

# Print the BLEU score for OPUS-MT model
print(f"BLEU Score for Translation Model:\n{bleu_score}\n")

# Compute the BLEU score for M2M-100 texts
bleu_score = bleu_metric.compute(predictions=translated_texts_2, references=original_t

# Print the BLEU score for M2M-100 model
print(f"BLEU Score for Translation Model:\n{bleu_score}\n")

# Compute the BLEU score for MBART-50 texts
```

```
bleu_score = bleu_metric.compute(predictions=translated_texts_3, references=original_t

# Print the BLEU score for MBART-50 model
print(f"BLEU Score for Translation Model:\n{bleu_score}\n")
```

```
BLEU Score for Translation Model:
{'bleu': 0.4382318045614185, 'precisions': [0.703788748564868, 0.4986072423398329, 0.
38126813095731454, 0.29442282749675747], 'brevity_penalty': 0.9836784826124793, 'leng
th_ratio': 0.9838102409638554, 'translation_length': 2613, 'reference_length': 2656}

BLEU Score for Translation Model:
{'bleu': 0.38514913523032057, 'precisions': [0.6721439749608764, 0.4531758957654723,
0.3331918505942275, 0.25354609929078015], 'brevity_penalty': 0.9616318146086646, 'len
gth_ratio': 0.9623493975903614, 'translation_length': 2556, 'reference_length': 2656}

BLEU Score for Translation Model:
{'bleu': 0.3961594263710492, 'precisions': [0.6709973251815056, 0.4604688120778705,
0.33802234174596607, 0.2503236944324558], 'brevity_penalty': 0.9852079334065056, 'len
gth_ratio': 0.985316265060241, 'translation_length': 2617, 'reference_length': 2656}
```

2) Provide the comparison discussions and analysis based on the evaluation results you obtained from each model. For example, which model performs best, which performs worst, and what is the possible reason for such results.

Based on the BLEU scores provided, the first model performs the best with a BLEU score of 0.4382, followed by the third model with a BLEU score of 0.3961, and the second model performs the worst with a BLEU score of 0.3851.

The BLEU score is a measure of the quality of machine-generated translations, with a higher score indicating a better match with the reference translations. The score ranges from 0 to 1, with 1 being a perfect match.

The first model might have performed the best due to a variety of factors such as the architecture of the model, the size and quality of the training data, and the optimization techniques used during training. The second model might have performed the worst due to limitations in these areas.

3) Select two data samples and compare the translation obtained by the three models with the ground truth. Provide some discussions based on your findings on these two examples.

In [15]:

```
# Sample 1
print(f"English: {english[0]}\n")
print(f"French Ground truth: {original_texts[0]}\n")
print(f"OPUS-MT Translation: {translated_texts_1[0]}\n")
print(f"M2M-100 Translation: {translated_texts_2[0]}\n")
print(f"MBART-50 Translation: {translated_texts_3[0]}\n")
```

```
English: The hearing also marks the date for the suspect’s right to a speedy trial.

French Ground truth: L'audience marque également la date pour le droit du suspect à u
n procès rapide.

OPUS-MT Translation: L'audience marque également la date à laquelle le suspect a droi
t à un procès rapide.

M2M-100 Translation: L’audition marque également la date pour le droit du suspect à u
n procès rapide.

MBART-50 Translation: L’audience marque également la date du droit du suspect à un pr
ocès rapide.
```

In [16]:

```
# Sample 2
print(f"English: {english[1]}\n")
print(f"French Ground truth: {original_texts[1]}\n")
print(f"OPUS-MT Translation: {translated_texts_1[1]}\n")
print(f"M2M-100 Translation: {translated_texts_2[1]}\n")
print(f"MBART-50 Translation: {translated_texts_3[1]}\n")
```

```
English: Leslie Aun, a spokesperson for the Komen Foundation, said the organization a
dopted a new rule that does not allow grants or funding to be awarded to organization
s that are under legal investigation.

French Ground truth: Leslie Aun, porte-parole de la Fondation Komen, a déclaré que
l'organisation a adopté une nouvelle règle qui ne permet pas d'accorder de subvention
s ou de financements aux organisations qui font l'objet d'une enquête judiciaire.

OPUS-MT Translation: Leslie Aun, porte-parole de la Fondation Komen, a déclaré que
l'organisation a adopté une nouvelle règle qui ne permet pas l'octroi de subventions
ou de fonds à des organisations faisant l'objet d'une enquête judiciaire.

M2M-100 Translation: Leslie Aun, porte-parole de la Fondation Komen, a déclaré que
l'organisation a adopté une nouvelle règle qui ne permet pas que des subventions ou d
es fonds soient accordés à des organisations qui sont en cours d'enquête juridique.

MBART-50 Translation: Leslie Aun, porte-parole de la Fondation Komen, a déclaré que
l'organisation avait adopté une nouvelle règle qui ne permet pas que des subventions
ou des fonds soient accordés aux organisations faisant l'objet d'une enquête judiciai
re.
```

In both samples, all three models (OPUS-MT, M2M-100, and MBART-50) provide accurate translations. However, there are slight differences in phrasing and word choice. The best model depends on the specific requirements of the translation task. For instance, MBART-50 maintains the original sentence structure better in these examples.

References:

Huggingface model documentation, ChatGPT, Github