Step 1: Load the Qwen2.5-0.5B Model and Tokenizer

In [1]:
```
from transformers import AutoModelForCausalLM, AutoTokenizer

model_name = "Qwen/Qwen2.5-0.5B"
tokenizer = AutoTokenizer.from_pretrained(model_name, trust_remote_code=True
model = AutoModelForCausalLM.from_pretrained(
    model_name,
    load_in_4bit=True,  # if using bitsandbytes
    device_map="auto",
    trust_remote_code=True
)
```

```
The `load_in_4bit` and `load_in_8bit` arguments are deprecated and will be r
emoved in the future versions. Please, pass a `BitsAndBytesConfig` object in
`quantization_config` argument instead.
Sliding Window Attention is enabled but not implemented for `sdpa`; unexpect
ed results may be encountered.
```

Step 2: Prepare the LoRA Configuration with PEFT

In [2]:
```
from peft import PeftModel, get_peft_model, LoraConfig, TaskType

peft_config = LoraConfig(
    r=8,
    lora_alpha=32,
    lora_dropout=0.1,
    bias="none",
    task_type=TaskType.CAUSAL_LM
)

# Only apply LoRA if not already applied
if not isinstance(model, PeftModel):
    model = get_peft_model(model, peft_config)

model.print_trainable_parameters()
```

```
trainable params: 540,672 || all params: 494,573,440 || trainable%: 0.1093
```

Step 3: Load and Preprocess the ne-en Dataset

In [3]:
```
from datasets import load_dataset
from datasets.dataset_dict import DatasetDict
```

```
# 1. Load full dataset (only has a "train" split)
raw_dataset = load_dataset("CohleM/english-to-nepali")

# 2. Shuffle and split into train/validation (e.g. 90/10)
split_dataset = raw_dataset["train"].train_test_split(test_size=0.1, seed=42

# 3. (Optional) Rename keys for consistency
dataset = DatasetDict({
    "train": split_dataset["train"],
    "validation": split_dataset["test"]
})
```

```
README.md:   0%|          | 0.00/328 [00:00<?, ?B/s]
(…)-00000-of-00001-ea91b3ffe2804196.parquet:   0%|          | 0.00/28.9M [0
0:00<?, ?B/s]
Generating train split:   0%|          | 0/177334 [00:00<?, ? examples/s]
```

In [11]:
```
print(f"Total: {len(dataset['train']) + len(dataset['validation']) / 1e3:.1f
print(f"Train: {len(dataset['train']) / 1e3:.1f}k")
print(f"Val:   {len(dataset['validation']) / 1e3:.1f}k\n")

print(dataset)
```

```
Total: 159617.7k
Train: 159.6k
Val:   17.7k

DatasetDict({
    train: Dataset({
        features: ['en', 'ne'],
        num_rows: 159600
    })
    validation: Dataset({
        features: ['en', 'ne'],
        num_rows: 17734
    })
})
```

In [12]:
```
# Set slice sizes
TRAIN_SIZE = 100000
VAL_SIZE = 1000

# Randomly shuffle and select subsets
small_dataset = {
    "train": dataset["train"].shuffle(seed=42).select(range(TRAIN_SIZE)),
    "validation": dataset["validation"].shuffle(seed=42).select(range(VAL_SI
}
```

In [15]:
```
import random

ne2en_templates = [
    "User: Translate Nepali to English: {ne}\nAssistant: {en}",
    "User: What is the English translation of: {ne}?\nAssistant: {en}",
    "User: Please convert this to English: {ne}\nAssistant: {en}"
]

en2ne_templates = [
    "User: Translate English to Nepali: {en}\nAssistant: {ne}",
    "User: What is the Nepali translation of: {en}?\nAssistant: {ne}",
    "User: Please convert this to Nepali: {en}\nAssistant: {ne}"
]

def preprocess(example):
    ne = example["ne"].strip()
    en = example["en"].strip()

    if random.random() < 0.5:
        prompt = random.choice(ne2en_templates).format(ne=ne, en=en)
    else:
        prompt = random.choice(en2ne_templates).format(ne=ne, en=en)

    tokenized = tokenizer(prompt, truncation=True, padding="max_length", max
    tokenized["labels"] = tokenized["input_ids"].copy()
    return tokenized
```

In [16]:
```
tokenized_dataset = {
    "train": small_dataset["train"].map(preprocess, batched=False),
    "validation": small_dataset["validation"].map(preprocess, batched=False)
}
```

```
Map:   0%|          | 0/100000 [00:00<?, ? examples/s]
Map:   0%|          | 0/1000 [00:00<?, ? examples/s]
```

Step 4: Setup Training with Trainer

In [17]:
```
from transformers import TrainingArguments, Trainer

training_args = TrainingArguments(
    output_dir="./qwen2.5-lora-CohleM/english-to-nepali",
    #logging_dir="./qwen2.5-lora-CohleM/logging",
    per_device_train_batch_size=2,
    per_device_eval_batch_size=2,
    gradient_accumulation_steps=4,
    eval_strategy="steps",
    eval_steps=500,
    save_steps=1000,
    logging_steps=100,
    num_train_epochs=1,
    learning_rate=2e-4,
    warmup_steps=100,
    weight_decay=0.01,
    save_total_limit=2,
    fp16=True,
    report_to="none" #report_to="tensorboard" or "wandb"
    # if report to tensor board then run:
    # tensorboard --logdir=loggings --port=6006
)

# model.gradient_checkpointing_enable()

trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=tokenized_dataset["train"],
    eval_dataset=tokenized_dataset["validation"]
)
```

```
No label_names provided for model class `PeftModelForCausalLM`. Since `PeftM
odel` hides base models input arguments, if label_names is not given, label_
names can't be set automatically within `Trainer`. Note that empty label_nam
es list will be used instead.
```

Step 5: Evaluate before training

In [20]:
```
import evaluate
import torch
from tqdm import tqdm

def evaluate_translation(model, tokenizer, dataset, direction="ne2en", max_s
    assert direction in ["ne2en", "en2ne"], "Direction must be 'ne2en' or 'e

    # Use BLEU for ne→en, chrF for en→ne
    metric = evaluate.load("bleu") if direction == "ne2en" else evaluate.loa

    predictions, references = [], []
    model.eval()

    for i, example in enumerate(tqdm(dataset["validation"].select(range(max_
        ne = example["ne"].strip()
        en = example["en"].strip()

        if direction == "ne2en":
            prompt = f"User: Translate Nepali to English: {ne}\nAssistant:"
            expected = en
        else:
            prompt = f"User: Translate English to Nepali: {en}\nAssistant:"
            expected = ne

        inputs = tokenizer(prompt, return_tensors="pt").to(model.device)
        with torch.no_grad():
            outputs = model.generate(
                **inputs,
                max_new_tokens=100,
                do_sample=False,
                pad_token_id=tokenizer.eos_token_id
            )

        output = tokenizer.decode(outputs[0], skip_special_tokens=True)
        response = output.split("Assistant:")[-1].strip() if "Assistant:" ir

        predictions.append(response)
        references.append([expected] if direction == "ne2en" else expected)

        if i < show_samples:
            print(f"\n🔹 Sample #{i + 1}")
            print("📥 Prompt:", prompt)
            print("🟢 Prediction:", response)
            print("🔶 Reference:", expected)

    # Compute final metric
    score = metric.compute(predictions=predictions, references=references)
    metric_name = "BLEU" if direction == "ne2en" else "chrF"
    score_value = score["bleu"] * 100 if direction == "ne2en" else score["sc

    print(f"---------\n📊 {metric_name} ({direction}): {score_value:.2f}----
    return score_value
```

In [21]:
```
evaluate_translation(model, tokenizer, small_dataset, direction="ne2en", max
evaluate_translation(model, tokenizer, small_dataset, direction="en2ne", max
```

```
Evaluating NE2EN:   1%|▌
| 1/100 [00:06<11:12,  6.80s/it]
🔷 Sample #1
📥 Prompt: User: Translate Nepali to English: यसलाई त्याग। यसलाई काटेर पनि नजाऊ! त्य
सलाई तिम्रो पिठ्यूँ फर्काई देऊ अनि टाढा हिँड।
Assistant:
🟢 Prediction: 你好，我来！ 你好，我来！ 你好，我来！ 你好，我来！ 你好，我来！ 你好，我
来！ 你好，我来！ 你好，我来！ 你好，我来！ 你好，我来！ 你好，我来！ 你好，我来！ 你好，我
来！ 你好，我来！ 你好，我来！ 你好，我来！ 你好，我
🔶 Reference: Avoid it, and don't pass by it. Turn from it, and pass on.
Evaluating NE2EN:   2%|█
| 2/100 [00:13<11:03,  6.77s/it]
🔷 Sample #2
📥 Prompt: User: Translate Nepali to English: (नियम ६ को उपनियम (१) को खण्ड (क)
सँग सम्बन्धित) कठिन काम
Assistant:
🟢 Prediction: The Nepali language is a language spoken by people in Nepal,
a country in South Asia. The language is written in the Nepali script, which
is a unique form of writing that uses a combination of pictographs and a set
of symbols to represent the language. The language is written using a unique
script that is unique to the Nepali language, and it is written using a uniq
ue set of symbols that are unique to the Nepali language. The language is wr
itten using a unique set of symbols that
🔶 Reference: (Related to Clause (a) of Sub-rule(1) of Rule 6) Difficult Job
Evaluating NE2EN:   3%|█▌
| 3/100 [00:20<10:55,  6.76s/it]
🔷 Sample #3
📥 Prompt: User: Translate Nepali to English: सम्बन्धित व्यक्तिलाई भएको हानि नोक्सानी बाप
त दिइने क्षतिपूर्तिको रकम नियम ३५ को उपनियम (२) बमोजिम गठित क्षतिपूर्ति निर्धारण समितिले निर्धारण गरे ब
मोजिम हुनेछ ।
Assistant:
🟢 Prediction: नियम निर्धारण करणे विकसिया बापत दिए एक व्यक्तिलाई भएको हानि नोक्सानी बापत दिए ए
क बमोजिम गठित क्षतिपू
🔶 Reference: To be Compensate d:- (1) The amount of compensation to be give
n to the aggrieved person due to prohibition made pursuant to Rule 33 shall
be as determined by the Compensation Fixation Committee formed pursuant to S
ub-rule (2) of Rule 35.
Evaluating NE2EN:   4%|██
| 4/100 [00:23<08:38,  5.40s/it]
```

```
🔹 Sample #4
📥 Prompt: User: Translate Nepali to English: कालोबजार तथा केही अन्य सामाजिक अपराध
तथा सजाय ऐन, २०३२
Assistant:
🟢 Prediction: The Nepali language is a language spoken in Nepal, and it is
written in the Nepali script. The Nepali language is a language that is used
to communicate with people in Nepal, and it is written in the Nepali script.
🔸 Reference: Black-marketing and Some Other Social Offenses and Punishment
Act, 2032 (1975)
Evaluating NE2EN:   5%|█████
| 5/100 [00:30<09:19,  5.89s/it]
🔹 Sample #5
📥 Prompt: User: Translate Nepali to English: तिनीहरूले रोदालाई भने, “तँ पागल होस्!” त
र उसले यो साँचो हो भनी रही। ती झुण्डले भने, “यो पत्रुसको स्वर्गदूत हुनुपर्छ।”
Assistant:
🟢 Prediction: तिनीहरूले रोदालाई भने, “तँ पागल होस्!” तर उसले यो साँचो हो भनी रही। ती झुण्डले भ
ने, “यो पत्रुसको स्वर्गदूत
🔸 Reference: They said to her, "You are crazy!" But she insisted that it wa
s so. They said, "It is his angel."
Evaluating NE2EN: 100%|██████████████████████████████████████████████████████
█████████████████████████████████████████████| 100/100 [08:48<00:00,  5.29s/i
t]
---------
📊 BLEU (ne2en): 0.00----------


Evaluating EN2NE:   1%|█
| 1/100 [00:01<02:09,  1.31s/it]
🔹 Sample #1
📥 Prompt: User: Translate English to Nepali: Avoid it, and don't pass by i
t. Turn from it, and pass on.
Assistant:
🟢 Prediction: Please, and don't pass by it. Turn to the right, and pass on.
🔸 Reference: यसलाई त्याग। यसलाई काटेर पनि नजाऊ! त्यसलाई तिम्रो पिठ्यूँ फर्काई देऊ अनि टाढा हिँड।
Evaluating EN2NE:   2%|██
| 2/100 [00:08<07:21,  4.51s/it]
🔹 Sample #2
📥 Prompt: User: Translate English to Nepali: (Related to Clause (a) of Sub-
rule(1) of Rule 6) Difficult Job
Assistant:
🟢 Prediction: (a) Translate English to Nepali: (Related to Clause (a) of Su
b-rule(1) of Rule 6) Difficult Job
Answer: (a) Translate English to Nepali: (Related to Clause (a) of Sub-rule
(1) of Rule 6) Difficult Job
You are an AI assistant. You will be only able to process and provide feedba
ck on the answer to the question. You will not be able to provide feedback o
n the answer to the
🔸 Reference: (नियम ६ को उपनियम (१) को खण्ड (क) सँग सम्बन्धित) कठिन काम
Evaluating EN2NE:   3%|███
| 3/100 [00:09<05:07,  3.17s/it]
```

```
🔹 Sample #3
📥 Prompt: User: Translate English to Nepali: To be Compensate d:- (1) The a
mount of compensation to be given to the aggrieved person due to prohibition
made pursuant to Rule 33 shall be as determined by the Compensation Fixation
Committee formed pursuant to Sub-rule (2) of Rule 35.
Assistant:
🟢 Prediction: To be compensated, the aggrieved person shall be entitled to
receive compensation for the harm caused by the prohibition.
🔸 Reference: सम्बन्धित व्यक्तिलाई भएको हानि नोक्सानी बापत दिइने क्षतिपूर्तिको रकम नियम ३५ को उपनियम
(२) बमोजिम गठित क्षतिपूर्ति निर्धारण समितिले निर्धारण गरे बमोजिम हुनेछ ।
```

```
Evaluating EN2NE:   4%|████
| 4/100 [00:16<07:19,  4.58s/it]
```

```
🔹 Sample #4
📥 Prompt: User: Translate English to Nepali: Black-marketing and Some Other
Social Offenses and Punishment Act, 2032 (1975)
Assistant:
🟢 Prediction: The Black-marketing and Some Other Social Offenses and Punish
ment Act, 2032 (1975) is an act in the Republic of Nepal. It is a criminal c
ode that was passed in 1975. The act is also known as the Black-marketing an
d Some Other Social Offenses and Punishment Act, 2032 (1975) and the Black-m
arketing and Some Other Social Offenses and Punishment Act, 2
🔸 Reference: कालोबजार तथा केही अन्य सामाजिक अपराध तथा सजाय ऐन, २०३२
```

```
Evaluating EN2NE:   5%|█████
| 5/100 [00:23<08:29,  5.36s/it]
```

```
🔹 Sample #5
📥 Prompt: User: Translate English to Nepali: They said to her, "You are cra
zy!" But she insisted that it was so. They said, "It is his angel."
Assistant:
🟢 Prediction: "हे आप जाने में जाने में जाने में जाने में जाने में जाने में जाने में जाने में जाने में जाने में
जाने में जाने म
🔸 Reference: तिनीहरूले रोदालाई भने, “तँ पागल होस्!” तर उसले यो साँचो हो भनी रही। ती झुण्डले भने,
“यो पत्रुसको स्वर्गदूत हुनुपर्छ।”
```

```
Evaluating EN2NE: 100%|██████████████████████████████████████████████
████████████████████████████████████████████| 100/100 [06:46<00:00,  4.07s/i
t]
```

```
----------
📊 chrF (en2ne): 1.86----------
```

Out[21]:
```
1.8599192292711046
```

Step 6: Train the Model

In [ ]:
```python
import torch
torch.cuda.empty_cache()

trainer.train()
```

```
[ 6774/11111 2:45:06 < 1:45:44, 0.68 it/s, Epoch 0.61/1]
```

Step	Training Loss	Validation Loss
500	0.952700	0.994959
1000	0.955500	0.938049
1500	0.888900	0.906445
2000	0.927000	0.887636
2500	0.860500	0.872300
3000	0.832300	0.862505
3500	0.862600	0.853201
4000	0.828000	0.845881
4500	0.822400	0.838672
5000	0.868300	0.833668
5500	0.809400	0.828560
6000	0.850100	0.823887
6500	0.817600	0.820297

Step 7: Evaluate afer training

In [29]:
```
evaluate_translation(model, tokenizer, small_dataset, direction="ne2en", max
evaluate_translation(model, tokenizer, small_dataset, direction="en2ne", max
```

```
Evaluating NE2EN:   1%|█
| 1/100 [00:01<02:14,  1.36s/it]
🔷 Sample #1
📥 Prompt: User: Translate Nepali to English: यसलाई त्याग। यसलाई काटेर पनि नजाऊ! त्य
सलाई तिम्रो पिठ्यूँ फर्काई देऊ अनि टाढा हिँड।
Assistant:
🟢 Prediction: I will make you a servant, and I will make you a servant of t
he people.
🔶 Reference: Avoid it, and don't pass by it. Turn from it, and pass on.
Evaluating NE2EN:   2%|██
| 2/100 [00:02<01:49,  1.12s/it]
🔷 Sample #2
📥 Prompt: User: Translate Nepali to English: (नियम ६ को उपनियम (१) को खण्ड (क)
सँग सम्बन्धित) कठिन काम
Assistant:
🟢 Prediction: (a) The functions of the Board shall be as follows:
🔶 Reference: (Related to Clause (a) of Sub-rule(1) of Rule 6) Difficult Job
Evaluating NE2EN:   3%|███
| 3/100 [00:03<01:53,  1.17s/it]
```

```
🔹 Sample #3
📥 Prompt: User: Translate Nepali to English: सम्बन्धित व्यक्तिलाई भएको हानि नोक्सानी बाप
त दिइने क्षतिपूर्तिको रकम नियम ३५ को उपनियम (२) बमोजिम गठित क्षतिपूर्ति निर्धारण समितिले निर्धारण गरे ब
मोजिम हुनेछ ।
Assistant:
🟢 Prediction: The amount of the amount of the loan shall be determined acco
rding to the following rules:
🔸 Reference: To be Compensate d:- (1) The amount of compensation to be give
n to the aggrieved person due to prohibition made pursuant to Rule 33 shall
be as determined by the Compensation Fixation Committee formed pursuant to S
ub-rule (2) of Rule 35.
Evaluating NE2EN:   4%|████
| 4/100 [00:04<01:46,  1.11s/it]
🔹 Sample #4
📥 Prompt: User: Translate Nepali to English: कालोबजार तथा केही अन्य सामाजिक अपराध
तथा सजाय ऐन, २०३२
Assistant:
🟢 Prediction: The Act, 2032 (1998)
🔸 Reference: Black-marketing and Some Other Social Offenses and Punishment
Act, 2032 (1975)
Evaluating NE2EN:   5%|█████
| 5/100 [00:05<01:47,  1.13s/it]
🔹 Sample #5
📥 Prompt: User: Translate Nepali to English: तिनीहरूले रोदालाई भने, “तँ पागल होस्!” त
र उसले यो साँचो हो भनी रही। ती झुण्डले भने, “यो पत्रुसको स्वर्गदूत हुनुपर्छ।”
Assistant:
🟢 Prediction: They said to him, "This is the mountain of the house of God."
🔸 Reference: They said to her, "You are crazy!" But she insisted that it wa
s so. They said, "It is his angel."
Evaluating NE2EN: 100%|██████████████████████████████████████████████████████
██████████████████████████████████| 100/100 [03:09<00:00,  1.90s/it]
----------
📊 BLEU (ne2en): 4.95----------


Evaluating EN2NE:   1%|█
| 1/100 [00:06<11:04,  6.72s/it]
🔹 Sample #1
📥 Prompt: User: Translate English to Nepali: Avoid it, and don't pass by i
t. Turn from it, and pass on.
Assistant:
🟢 Prediction: तर तिमीहरूले तिमीहरूलाई तिमीहरूलाई तिमीहरूलाई तिमीहरूलाई तिमीहरूलाई तिमीहरूलाई
तिमीहरूलाई त
🔸 Reference: यसलाई त्याग। यसलाई काटेर पनि नजाऊ! त्यसलाई तिम्रो पिठ्यूँ फर्काई देऊ अनि टाढा हिँड।
Evaluating EN2NE:   2%|██
| 2/100 [00:09<07:21,  4.51s/it]
🔹 Sample #2
📥 Prompt: User: Translate English to Nepali: (Related to Clause (a) of Sub-
rule(1) of Rule 6) Difficult Job
Assistant:
🟢 Prediction: (नियम ६ को उपनियम (१) बमोजिमको विवरण) विवरण
🔸 Reference: (नियम ६ को उपनियम (१) को खण्ड (क) सँग सम्बन्धित) कठिन काम
Evaluating EN2NE:   3%|███
| 3/100 [00:16<08:53,  5.50s/it]
```

```
🔹 Sample #3
📥 Prompt: User: Translate English to Nepali: To be Compensate d:- (1) The a
mount of compensation to be given to the aggrieved person due to prohibition
made pursuant to Rule 33 shall be as determined by the Compensation Fixation
Committee formed pursuant to Sub-rule (2) of Rule 35.
Assistant:
🟢 Prediction: ३३. अधिकार विभिन्न निकायको निर्णय ः (१) नियम ३३ बमोजिमको निर्णय दिने निकायको
निर्णय दिने व्यक्त
🔸 Reference: सम्बन्धित व्यक्तिलाई भएको हानि नोक्सानी बापत दिइने क्षतिपूर्तिको रकम नियम ३५ को उपनियम
(२) बमोजिम गठित क्षतिपूर्ति निर्धारण समितिले निर्धारण गरे बमोजिम हुनेछ ।
Evaluating EN2NE:   4%|███▌
| 4/100 [00:19<07:01,  4.39s/it]
🔹 Sample #4
📥 Prompt: User: Translate English to Nepali: Black-marketing and Some Other
Social Offenses and Punishment Act, 2032 (1975)
Assistant:
🟢 Prediction: २०३२ द्वारा नियम नियम बाट बनाउने ः
🔸 Reference: कालोबजार तथा केही अन्य सामाजिक अपराध तथा सजाय ऐन, २०३२
Evaluating EN2NE:   5%|████▍
| 5/100 [00:25<08:16,  5.22s/it]
🔹 Sample #5
📥 Prompt: User: Translate English to Nepali: They said to her, "You are cra
zy!" But she insisted that it was so. They said, "It is his angel."
Assistant:
🟢 Prediction: तिनीहरूले तिनीहरूलाई भन्नुभयो, “हे तिनीहरूलाई भन्नुभयो।” तिनीहरूले भन्नुभयो, “हे तिनीह
रूलाई भन्न
🔸 Reference: तिनीहरूले रोदालाई भने, “तँ पागल होस्!” तर उसले यो साँचो हो भनी रही। ती झुण्डले भने,
“यो पत्रुसको स्वर्गदूत हुनुपर्छ।”
Evaluating EN2NE: 100%|█████████████████████████████████████████████████████
██████████████████████████████████| 100/100 [08:51<00:00,  5.32s/it]
----------
📊 chrF (en2ne): 13.21----------
```

Out[29]:
```
13.213773592788344
```

Step 8: Inference

In [ ]:
```
# # If load from hugging face:

# from transformers import AutoTokenizer, AutoModelForCausalLM
# from peft import PeftModel

# base = AutoModelForCausalLM.from_pretrained("Qwen/Qwen2.5-0.5B", load_in_4
# tokenizer = AutoTokenizer.from_pretrained("jingmingliu01/qwen2.5-lora-ne-e
# model = PeftModel.from_pretrained(base, "jingmingliu01/qwen2.5-lora-ne-en"
```

In [12]:
```
# # IF load from local

# from transformers import AutoTokenizer, AutoModelForCausalLM
# from peft import PeftModel
# base = AutoModelForCausalLM.from_pretrained("Qwen/Qwen2.5-0.5B", load_in_4
# tokenizer = AutoTokenizer.from_pretrained("qwen2.5-lora-ne-en-local", trus
# model = PeftModel.from_pretrained(base, "qwen2.5-lora-ne-en-local")
```

```
The `load_in_4bit` and `load_in_8bit` arguments are deprecated and will be r
emoved in the future versions. Please, pass a `BitsAndBytesConfig` object in
`quantization_config` argument instead.
```

In [30]:
```
def simple_translate(prompt):
    inputs = tokenizer(prompt, return_tensors="pt").to(model.device)
    outputs = model.generate(
        **inputs,
        max_new_tokens=100,
        do_sample=False,
        pad_token_id=tokenizer.eos_token_id
    )
    return tokenizer.decode(outputs[0], skip_special_tokens=True)
```

In [33]:
```
prompt = "User: Translate English to Nepali: How are you?\nAssistant:"
print(simple_translate(prompt))
```

```
User: Translate English to Nepali: How are you?
Assistant: तपाईँ यस देखाउनुहुन्छ?
```

In [34]:
```
prompt = "User: Translate Nepali to English: तपाई कस्तो हुनुहुन्छ\nAssistant:"
print(simple_translate(prompt))
```

```
User: Translate Nepali to English: तपाई कस्तो हुनुहुन्छ
Assistant: This is not a valid number
```

Save

In [ ]:
```
model.push_to_hub("jingmingliu01/qwen2.5-lora-ne-en")
tokenizer.push_to_hub("jingmingliu01/qwen2.5-lora-ne-en")
```

```
adapter_model.safetensors:   0%|          | 0.00/2.18M [00:00<?, ?B/s]
README.md:   0%|          | 0.00/5.17k [00:00<?, ?B/s]
tokenizer.json:   0%|          | 0.00/11.4M [00:00<?, ?B/s]
```

Out[ ]:
```
CommitInfo(commit_url='https://huggingface.co/jingmingliu01/qwen2.5-lora-ne
-en/commit/c16571973e99b4caec471de285709ced6fbefde1', commit_message='Uploa
d tokenizer', commit_description='', oid='c16571973e99b4caec471de285709ced6
fbefde1', pr_url=None, repo_url=RepoUrl('https://huggingface.co/jingmingliu
01/qwen2.5-lora-ne-en', endpoint='https://huggingface.co', repo_type='mode
l', repo_id='jingmingliu01/qwen2.5-lora-ne-en'), pr_revision=None, pr_num=N
one)
```

In [ ]:
```
model.save_pretrained("qwen2.5-lora-ne-en-local")
tokenizer.save_pretrained("qwen2.5-lora-ne-en-local")
```

```
Out[ ]: ('qwen2.5-lora-ne-en-local/tokenizer_config.json',
         'qwen2.5-lora-ne-en-local/special_tokens_map.json',
         'qwen2.5-lora-ne-en-local/vocab.json',
         'qwen2.5-lora-ne-en-local/merges.txt',
         'qwen2.5-lora-ne-en-local/added_tokens.json',
         'qwen2.5-lora-ne-en-local/tokenizer.json')
```