Lab5: Translating by Prompting a LLM On the Importance of Parallel Data

He HUANG 22107447

November 23, 2023

1 Introduction

2 Your work

We will use the challenge set available [1] to evaluate the capacity of a LLM to translate (either one of `Mistral` LLM or `LLaMA`) and compare their performance to `mBART`.

1. What is the BLEU score achieved on the challenge set by an LLM ? by `mBART` ?
Mistral LLM :
To make it easier to clean up the generated text, here I set up the LLM model without much explanation.

```
# Mistral LLM
# Use the LLM prompt method and pipeline to generate the translation
pipe = pipeline("text-generation",
                model="HuggingFaceH4/zephyr-7b-alpha",
                torch_dtype=torch.bfloat16,
                device_map="auto")

def mistral_translate(dataset):
    translated = []

    for i in range(0, len(dataset)):
         # take the source sentence and add the prompt
         message = [{"role": "user","content": "translate into French and do not add explanations or notes: " + dataset['source'][i]}]
         # tokenize the prompt
         prompt = pipe.tokenizer.apply_chat_template(message,
                                              tokenize=False,
                                              add_generation_prompt=True)
        # generate the translation
         outputs = pipe(prompt,
              max_new_tokens=256,
              do_sample=True,
              temperature=0.7,
              top_k=50,
              top_p=0.95)

```

[1] https://aclanthology.org/D17-1263/

```
        translated.append(outputs[0]["generated_text"])
    return translated

# extract the content translated by Mistral
mistral_en_fr = []
for sent in translated_mistral:
    sent = re.split("\n", sent)
    if '<|assistant|>' in sent:
       ids = sent.index('<|assistant|>')
       mistral_en_fr.append(sent[ids+1])

# compute BLEU score for Mistral
mistral_results = bleu.compute(predictions=mistral_en_fr, references=corpus["
    reference"])
print(f"Bleu score of Mistral model : {mistral_results}")
```

mBART :

```
# define the mBart model and tokenizer
from transformers import MBartForConditionalGeneration, MBart50TokenizerFast

mbart_name = "facebook/mbart-large-50-many-to-many-mmt"
mbart_model = MBartForConditionalGeneration.from_pretrained(mbart_name).to(
    device)
mbart_tokenizer = MBart50TokenizerFast.from_pretrained(mbart_name, src_lang="
    en_XX")

# translate English sentences into French
def mbart_translate(en_source):
  # encode the french sentence
  inputs = mbart_tokenizer(en_source["source"], return_tensors="pt", truncation=
    True, max_length=512, padding=True).to(device)
  # generate
  generated_tokens = mbart_model.generate(**inputs, forced_bos_token_id=
    mbart_tokenizer.lang_code_to_id["fr_XX"])
  # decode the english sentence
  eng_sent = mbart_tokenizer.batch_decode(generated_tokens, skip_special_tokens=
    True)
  return {"mbart_en_fr": eng_sent}


# Compute BLEU score
import evaluate
bleu = evaluate.load("bleu")

mbart_results = bleu.compute(predictions=corpus["mbart_en_fr"], references=
    corpus["reference"])
print(f"Bleu score of mBART model : {mbart_results}")
```

Results :
As can be seen from the results, there is a difference between the bleu score of the LLM model and the mBART translation model, although it is not particularly large.

```
Output:
Bleu score of mBART model : {'bleu': 0.5017724846364539, 'precisions':
    [0.7444561774023232, 0.5578069129916567, 0.4322845417236662,
    0.35313001605136435], 'brevity_penalty': 1.0, 'length_ratio':
    1.0021164021164022, 'translation_length': 947, 'reference_length': 945}

Output:
```

```
Bleu score of Mistral model : {'bleu': 0.45006544291275874, 'precisions':
    [0.7083333333333334, 0.5048309178743962, 0.3861111111111111,
    0.3088235294117647], 'brevity_penalty': 0.9904306953846911, 'length_ratio':
    0.9904761904761905, 'translation_length': 936, 'reference_length': 945}
```

2. Compute for each major category of difficulty the proportion of sentences in which the difficulty (identified by square brackets) is correctly translated ?

And here are all the results :

	category_major	GNMT_correct
0	Lexico-Syntactic	56.097561
1	Morpho-Syntactic	72.413793
2	Syntactic	73.684211

Figure 1: Accuracy of GNMT model

	category_major	PBMT1_correct
0	Lexico-Syntactic	39.024390
1	Morpho-Syntactic	17.241379
2	Syntactic	28.947368

Figure 2: Accuracy of PBMT1 model

	category_major	PBMT2_correct
0	Lexico-Syntactic	43.902439
1	Morpho-Syntactic	17.241379
2	Syntactic	26.315789

Figure 3: Accuracy of PBMT2 model

	category_major	NMT_correct
0	Lexico-Syntactic	46.341463
1	Morpho-Syntactic	75.862069
2	Syntactic	34.210526

Figure 4: Accuracy of NMT model