ML for NLP2 2023-2024

Lab 04 Translating by Prompting a LLM

22210802 Haeeul HWANG

```
%pip install accelerate
%pip install sacrebleu

import torch
from transformers import pipeline
from tqdm import tqdm
import json
from sacrebleu.metrics import BLEU



# save the challenge set in json format as lists

references = []
sources = []

with open("Challenge_set-v2hA.json", 'r', encoding="utf-8") as f:
    for line in f:
        data = json.loads(line)
        references.append(data["reference"])
        sources.append(data["source"])

# remove all the punctuations from the references and sources

import string

def remove_punctuations(text):
    return text.translate(str.maketrans('', '', string.punctuation))

references_cleaned = [remove_punctuations(ref) for ref in references]
sources_cleaned = [remove_punctuations(src) for src in sources]
```

```
references_cleaned[:10]
```

```
['Les appels répétés de sa mère auraient dû nous alerter',
 'Le bruit soudain dans les chambres supérieures aurait dû nous alerter',
 'Leurs échecs répétés à signaler le problème auraient dû nous alerter',
 'Elle a demandé à son frère de ne pas se montrer arrogant',
 'Elle a promis à son frère de ne pas être arrogante',
 'Elle a promis à son médecin de demeurer active après sêtre retirée',
 'Ma mère a promis à mon père dêtre plus prudente sur la route',
 'La femme était très grande et extrêmement forte',
 'Leurs politiciens étaient plus ignorants que stupides',
 'Nous avons lancé une insulte et nous sommes partis brusquement']
```

```
sources_cleaned[:10]
```

```
['The repeated calls from his mother should have alerted us',
 'The sudden noise in the upper rooms should have alerted us',
 'Their repeated failures to report the problem should have alerted us',
 'She asked her brother not to be arrogant',
 'She promised her brother not to be arrogant',
 'She promised her doctor to remain active after retiring',
 'My mother promised my father to be more prudent on the road',
 'The woman was very tall and extremely strong',
 'Their politicians were more ignorant than stupid',
 'We shouted an insult and left abruptly']
```

1. **Which translation model is used when using an HuggingFace translation pipeline?**

   The default model is google t5.

```
# default translation model

translator = pipeline("translation_en_to_fr")
```

```
No model was supplied, defaulted to google-t5/t5-base and revision 686f1db (https://huggingface.co/google-t5/t5-base).
Using a pipeline without specifying a model name and revision in production is not recommended.
```

2. **What is the BLEU score achieved on the challenge set by an LLM ? by the translation pipeline ?**

By the translation pipeline : 0.448
By the Mistral-based model(zephyr-7b-beta): 0.418
By the LLaMA-based model(Llama2-13b-Language-translate) : 0.444

```
# translate the sources (in French) in English

translated_sources = [translator(src) for src in tqdm(sources_cleaned)]
```

```
100%|██████████| 108/108 [03:14<00:00,  1.80s/it]
```

```
# keep only the tranlated sentences

translated_sources = [src[0]['translation_text'] for src in translated_sources]
translated_sources[:10]
```

```
['Les appels répétés de sa mère auraient dû nous avertir',
 'Le bruit soudain dans les chambres supérieures aurait dû nous alerter.',
 'Leur refus répété de signaler le problème aurait dû nous avertir',
 'Elle a demandé à son frère de ne pas être arrogant',
 'Elle a promis à son frère de ne pas être arrogante',
 'Elle a promis à son médecin de demeurer active après sa retraite',
 'Ma mère a promis à mon père d’être plus prudent sur la route.',
 'La femme était très grande et extrêmement forte',
 'Leurs politiques étaient plus ignorantes que stupides',
 'Nous avons crié une insulte et nous sommes brusquement partis.']
```

```
# bleu score of the default model

bleu = BLEU()
bleu_score_llm = bleu.corpus_score(translated_sources, references_cleaned)
bleu_score_llm.score
```

```
0.44822138964642555
```

```
# mistral based model

device = device = torch.device("cuda")
pipe = pipeline("text-generation", model="HuggingFaceH4/zephyr-7b-beta", torch_dtype=torch.bfloat16, device_map=device)
```

```python
# translate the sources to french

translated_sources_llm = []

for src in tqdm(sources_cleaned):
    message = [
        {
            "role" : "user",
            "content" : f"translate the following text to french: '{src}'"
        }
    ]

    prompt = pipe.tokenizer.apply_chat_template(message, tokenize=False, add_generation_prompt=True)

    output = pipe(prompt, truncation=True, max_new_tokens=256, max_length=100, do_sample=True, temperature=0.7, top_k=50, top_p=0.95, num_return_sequences=1)
    translated_sources_llm.append(output[0]["generated_text"])
```

```python
# save the translated sentences by the model in a list

translated_llm = []

for text in translated_sources_llm:
    extracted_sentence = text.split("\n")[3]
    translated_llm.append(extracted_sentence.strip())

translated_llm
```

```
['Les répétées appels de sa mère devraient nous avoir mis en garde" en français.',
 '"Le bruit soudain dans les chambres hautes nous devrait avoir mis en garde"',
 '"Ses répétées négligences à ne pas signaler le problème devraient nous avoir mis en garde"',
 '"Elle a demandé à son frère de ne pas être arrogant"',
 '"Elle a promis à son frère de ne pas être arrogante"',
 '"Elle a promis à son médecin de rester active après avoir pris sa retraite"',
 '"Ma mère a promis à mon père d\'être plus prudente sur la route"',
 '"La femme était très grande et extrêmement forte."',
 '"Les leurs politiques étaient plus ignorants que stupides"',
 'Nous criâmes un insulte et nous sommes partis brusquement.',
 '"Le chat et le chien devraient être surveillés"',
 '"Mon père et mon frère seront heureux demain"',
 '"Mon livre et mon crayon peuvent être volés"',
 '"Le vache et le poulet doivent être alimentés"',
 '"Mienne mère et ma sœur seront heureuses demain"',
 '"Mes chaussures et mes souliers seront trouvés"',
 '"Le chien et le bœuf sont nerveux"',
 '"My père et ma mère seront heureux demain"',
 '"Mes réfrigérateurs et mon table de cuisine ont été volés"',
 '"Paul et moi pouvons facilement être persuadés(es) de rejoindre vous"',
```

```python
# clean the translated sentences removing punctuations and words between parentheses.

import re

def remove_punctuation(sentence):
    sentence_without_punctuation = re.sub(r'[^\w\s\']', '', sentence)
    return sentence_without_punctuation

def filter_sentences(sentences):
    filtered_sentences = []
    for sentence in sentences:
        if '(' not in sentence and ')' not in sentence:
            filtered_sentences.append(remove_punctuation(sentence))
    return filtered_sentences

filtered_sentences = filter_sentences(translated_llm)

filtered_sentences
```

```
['Les répétées appels de sa mère devraient nous avoir mis en garde en français',
 'Le bruit soudain dans les chambres hautes nous devrait avoir mis en garde',
 'Ses répétées négligences à ne pas signaler le problème devraient nous avoir mis en garde',
 'Elle a demandé à son frère de ne pas être arrogant',
 'Elle a promis à son frère de ne pas être arrogante',
 'Elle a promis à son médecin de rester active après avoir pris sa retraite',
 "Ma mère a promis à mon père d'être plus prudente sur la route",
 'La femme était très grande et extrêmement forte',
 'Les leurs politiques étaient plus ignorants que stupides',
 'Nous criâmes un insulte et nous sommes partis brusquement',
 'Le chat et le chien devraient être surveillés',
 'Mon père et mon frère seront heureux demain',
 'Mon livre et mon crayon peuvent être volés',
 'Le vache et le poulet doivent être alimentés',
 'Mienne mère et ma sœur seront heureuses demain',
 'Mes chaussures et mes souliers seront trouvés',
 'Le chien et le bœuf sont nerveux',
 'My père et ma mère seront heureux demain',
 'Mes réfrigérateurs et mon table de cuisine ont été volés',
```

```python
# bleu score of the mistral based model

bleu = BLEU()
bleu_score_llm = bleu.corpus_score(filtered_sentences, references_cleaned)
bleu_score_llm.score
```

```
0.41800406854321887
```

```python
# LLaMA2-13b model

from transformers import AutoTokenizer, AutoModelForSeq2SeqLM

tokenizer = AutoTokenizer.from_pretrained("SnypzZz/Llama2-13b-Language-translate")
model = AutoModelForSeq2SeqLM.from_pretrained("SnypzZz/Llama2-13b-Language-translate")
```

```python
# translate the sources to french

translated_sources = []

for src in sources_cleaned:
    model_inputs = tokenizer(src, return_tensors="pt", padding=True, truncation=True)
    generated_tokens = model.generate(**model_inputs, forced_bos_token_id=tokenizer.lang_code_to_id["fr_XX"])
    output = tokenizer.batch_decode(generated_tokens, skip_special_tokens=True)
    translated_sources += output
    print("Translated source: ", output)
```

```
Translated source:  ['Les appels répétés de sa mère auraient dû nous avertir']
Translated source:  ['Le bruit soudain dans les chambres supérieures aurait dû nous alerter']
Translated source:  ['Leur incapacité répétée de déclarer le problème aurait dû nous avertir']
Translated source:  ['Elle a demandé à son frère de ne pas être arrogant.']
Translated source:  ['Elle a promis à son frère de ne pas être arrogant.']
Translated source:  ['Elle a promis à son médecin de rester actif après sa retraite.']
Translated source:  ["Ma mère a promis à mon père d'être plus prudent sur la route"]
Translated source:  ['La femme était très haute et extrêmement forte.']
Translated source:  ['Leurs politiciens étaient plus ignorants que stupides']
Translated source:  ['Nous avons crié un insulte et nous avons quitté brusquement.']
Translated source:  ['Le chat et le chien devraient être surveillés']
Translated source:  ['Mon père et mon frère seront heureux demain']
Translated source:  ['Mon livre et mon stylo pourraient être volés']
Translated source:  ['La vache et la poule doivent être nourries']
Translated source:  ['Ma mère et ma sœur seront heureuses demain']
Translated source:  ['Mes chaussures et mes souliers seront trouvés']
Translated source:  ['Le chien et la vache sont nerveux']
Translated source:  ['Mon père et ma mère seront heureux demain']
```

```python
# BLEU score

bleu = BLEU()
bleu_score = bleu.corpus_score(translated_sources, references_cleaned)
print("BLEU score: ", bleu_score.score)
```

```
BLEU score:  0.44420544193465444
```

3. **How long does it take to translation the test set with a LLM? with a specific model ?**

Default model : 3 minutes
Mistral-based model(zephyr-7b-beta) : 6 minutes
LLaMA-based model(Llama2-13b-Language-translate) : 6 minutes