1. Find an baseline NMT model (e.g. Opus-MT)
2. Find a open-source dataset that suits a specific area (e.g. law/history/movie/sci-fi)
3. Test its MT performance on the test dataset (use BLEU/sacredBLEU)
4. Improve
   1. Find some points that we can optimize, and analyse its result (good or bad, and why it would behave like that)
   2. Use similar datasets to fine-tune it. We can find a dataset and split it into training/testing, then we train on the training set, and then compare the performance between the training set and the testing set.
5. Make a conclusion
6. @shalong find a baseline model and an open-source dataset
   1. Baseline model: Helsinki-NLP/opus-mt-en-de
   2. Law translation datasets:
      1. JRC-Acquis
         1. <https://joint-research-centre.ec.europa.eu/language-technology-resources/jrc-acquis_en>
      2. DGT Translation Memory
         1. <https://joint-research-centre.ec.europa.eu/language-technology-resources/dgt-translation-memory_en>

Code:

from transformers import pipeline

# Use Helsinki-NLP's opus-MT model for English to German translation

translator = pipeline("translation", model="Helsinki-NLP/opus-mt-en-de")

# Sentence to translate (in English)

english\_text = "Hello, world!"

# Translate the sentence; set max\_length to ensure the output is long enough

result = translator(english\_text, max\_length=40)

# Print the translation result

print(result[0]['translation\_text'])

1. Test its performance on the test dataset, make a judging criteria (e.g. BLEU or some other criteria) @sudhakar
   1. Subtask: provide a get\_training\_set() and a get\_test\_set() function so that we can reuse to keep consistency
2. Improve
   1. May not be able to find an optimization point @one person @jiedeng
   2. The fine-tune may cost much resources that we cannot afford @weichen & litian
3. Compare the results (using the same criteria from step 2), and then conclude (1 person)