LANGUAGE TRANSLATION WITH

NN.TRANSFORMER & TORCH TEXT

Data Sourcing & Processing

- how to use torchtext's inbuilt datasets
- tokenize a raw text sentence
- build vocabulary
- numericalize tokens into tensor

```
from torchtext.data.utils import get_tokenizer
from torchtext.vocab import build_vocab_from_iterator
from torchtext.datasets import Multi30k
from typing import Iterable, List
```

```
SRC_LANGUAGE = 'de'
TGT_LANGUAGE = 'en'

# Place-holders
token_transform = {}
vocab_transform = {}

# Create source & target language tokenizer.
# Make sure to install the dependencies.

# pip install -U spacy
# python -m spacy download en_core_web_sm
# python -m spacy download de_core_news_sm
token_transform[SRC_LANGUAGE] = get_tokenizer('spacy', language='de_core_news_sm')
token_transform[TGT_LANGUAGE] = get_tokenizer('spacy', language='en_core_web_sm')
```

```
# helper function to yield list of tokens
def yield_tokens(data_iter: Iterable, language: str) -> List[str]:
    language_index = {SRC_LANGUAGE: 0, TRT_LANGUAGE: 1}

    for data_sample in data_iter:
        yield token_transform[language](data_sample[language_index[language]])
```

get_tokenizer() 함수

SRC / TGT의 data_sample을 tokenizing.

Language index

data_sample[0] or data_sample[1]

```
# Define special symbols and indices
UNK_IDX, PAD_IDX, BOS_IDX, EOS_IDX = 0, 1, 2, 3
special_symbols = ['<unk>', '<pad>', '<bos>', '<EOS>']

for ln in [SRC_LANGUAGE, TGT_LANGUAGE]:
    # Training data Iterator
    train_iter = Multi30k (split = 'train', language_pair = (SRC_LANGUAGE, TGT_LANGUAGE))
    # Create torchtext's Vocab object
    vocab_transform[ln] = build_vocab_from_iterator (yield_tokens (train_iter, ln),
                                                     min_freq = 1,
                                                     specials = special_symbols,
                                                     special_first = True)

# Set UNK_IDX as the default index.
# This index is returned when the token is not found.
# If not set, it throws RuntimeError when the queried token is not found in the vocabulary.
for ln in [SRC_LANGUAGE, TGT_LANGUAGE]:
    vocab_transform[ln].set_default_index(UNK_IDX)
```