Installing Modules

```
1 !pip install spacy==3
2 !python -m spacy download en_core_web_sm
3 !pip install pytorch_lightning torchmetrics tableprint
4 !python -m spacy download de_core_news_sm
5
```

```
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /usr/local/
Requirement already satisfied: google-auth<2,>=1.6.3 in /usr/local/lib/pyth
Requirement already satisfied: setuptools>=41.0.0 in /usr/local/lib/python3
Requirement already satisfied: werkzeug>=0.11.15 in /usr/local/lib/python3.
Requirement already satisfied: wheel>=0.26; python_version >= "3" in /usr/l
Requirement already satisfied: grpcio>=1.24.3 in /usr/local/lib/python3.7/d
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /usr/loc
Requirement already satisfied: pyparsing>=2.0.2 in /usr/local/lib/python3.7/
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python
Requirement already satisfied: async-timeout<4.0,>=3.0 in /usr/local/lib/py
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.7/d
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.7/dis
Requirement already satisfied: importlib-metadata; python_version < "3.8" i
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/pyth
Requirement already satisfied: rsa<5,>=3.1.4; python_version >= "3.6" in /us
Requirement already satisfied: cachetools<5.0,>=2.0.0 in /usr/local/lib/pyt
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/p
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-pa
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /usr/local/lib/pytho
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.7/
2021-06-18 04:56:44.121211: I tensorflow/stream_executor/platform/default/ds

Requirement already satisfied: de-core-news-sm==3.0.0 from https://github.c
Requirement already satisfied: spacy<3.1.0,>=3.0.0 in /usr/local/lib/python
Requirement already satisfied: pydantic<1.8.0,>=1.7.1 in /usr/local/lib/pyt
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/pyth
Requirement already satisfied: typer<0.4.0,>=0.3.0 in /usr/local/lib/python
Requirement already satisfied: pathy in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: catalogue<2.1.0,>=2.0.1 in /usr/local/lib/py
Requirement already satisfied: wasabi<1.1.0,>=0.8.1 in /usr/local/lib/pytho
Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.7/
Requirement already satisfied: typing-extensions>=3.7.4; python_version < "
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/
Requirement already satisfied: thinc<8.1.0,>=8.0.0 in /usr/local/lib/python
Requirement already satisfied: numpy>=1.15.0 in /usr/local/lib/python3.7/dis
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.0 in /usr/local/lib/
Requirement already satisfied: srsly<3.0.0,>=2.4.0 in /usr/local/lib/python
Requirement already satisfied: jinja2 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python
Requirement already satisfied: blis<0.8.0,>=0.4.0 in /usr/local/lib/python3
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python
Requirement already satisfied: importlib-metadata>=0.20; python_version < "
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/py
Requirement already satisfied: click<7.2.0,>=7.1.1 in /usr/local/lib/python
```

```
Requirement already satisfied: click<7.2.0,>=7.1.1 in /usr/local/lib/python3.
Requirement already satisfied: smart-open<4.0.0,>=2.2.0 in /usr/local/lib/py
Requirement already satisfied: zipp>=0.5; python_version < "3.8" in /usr/lo
Requirement already satisfied: pyparsing>=2.0.2 in /usr/local/lib/python3.7/
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.7/
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dis
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3
✔ Download and installation successful
```

Imports

```
# Import Library
import random
import torch, torchtext
from torchtext.legacy import data
import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim


import pandas as pd
import sys, os, pickle
import numpy as np
import math
import matplotlib.pyplot  as plt

import spacy

import pytorch_lightning as pl
import torchmetrics

from pytorch_lightning.loggers import CSVLogger
from pytorch_lightning.callbacks import ModelCheckpoint
from sklearn.metrics import confusion_matrix
import tableprint as tp

import collections

# Manual Seed
SEED = 43
torch.manual_seed(SEED)
```

```
<torch._C.Generator at 0x7f24c34198b0>
```

Loading Data

Since the Multi30k dataset is part of the legacy code (and might be deprecated), I downloaded the original files from github

```
!wget https://raw.githubusercontent.com/multi30k/dataset/master/data/task1/raw/
!wget https://raw.githubusercontent.com/multi30k/dataset/master/data/task1/raw/
!wget https://raw.githubusercontent.com/multi30k/dataset/master/data/task1/raw/
!wget https://raw.githubusercontent.com/multi30k/dataset/master/data/task1/raw/
```

```
--2021-06-18 04:56:51--  https://raw.githubusercontent.com/multi30k/dataset/m
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.10
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.1
HTTP request sent, awaiting response... 200 OK
Length: 637044 (622K) [application/octet-stream]
Saving to: ‘train.de.gz’

train.de.gz         100%[===================>] 622.11K  --.-KB/s    in 0.02s

2021-06-18 04:56:52 (34.1 MB/s) - ‘train.de.gz’ saved [637044/637044]

--2021-06-18 04:56:52--  https://raw.githubusercontent.com/multi30k/dataset/m
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.10
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.1
HTTP request sent, awaiting response... 200 OK
Length: 568929 (556K) [application/octet-stream]
Saving to: ‘train.en.gz’

train.en.gz         100%[===================>] 555.59K  --.-KB/s    in 0.01s

2021-06-18 04:56:52 (38.9 MB/s) - ‘train.en.gz’ saved [568929/568929]

--2021-06-18 04:56:52--  https://raw.githubusercontent.com/multi30k/dataset/m
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.10
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.1
HTTP request sent, awaiting response... 200 OK
Length: 24681 (24K) [application/octet-stream]
Saving to: ‘val.de.gz’

val.de.gz           100%[===================>]  24.10K  --.-KB/s    in 0s

2021-06-18 04:56:52 (86.8 MB/s) - ‘val.de.gz’ saved [24681/24681]

--2021-06-18 04:56:52--  https://raw.githubusercontent.com/multi30k/dataset/m
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.10
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.1
HTTP request sent, awaiting response... 200 OK
Length: 21650 (21K) [application/octet-stream]
Saving to: ‘val.en.gz’

val.en.gz           100%[===================>]  21.14K  --.-KB/s    in 0s

2021-06-18 04:56:52 (114 MB/s) - ‘val.en.gz’ saved [21650/21650]
```

```
!gunzip val.en.gz
!gunzip val.de.gz
!gunzip train.en.gz
!gunzip train.de.gz
```

```
gzip: val.en already exists; do you wish to overwrite (y or n)? y
gzip: val.de already exists; do you wish to overwrite (y or n)? y
```

```
gzip: train.en already exists; do you wish to overwrite (y or n)? ^C
gzip: train.de already exists; do you wish to overwrite (y or n)? ^C
```

```
train_df_src = pd.read_csv('train.en', sep="\t", encoding="utf8", header=None, r
train_df_trg = pd.read_csv('train.de', sep="\t", encoding="utf8", header=None, r
test_df_src = pd.read_csv('val.en', sep="\t", encoding="utf8", header=None, name
test_df_trg = pd.read_csv('val.de', sep="\t", encoding="utf8", header=None, name
train_df_src.head()
```

	sentence
0	Two young, White males are outside near many b...
1	Several men in hard hats are operating a giant...
2	A little girl climbing into a wooden playhouse.
3	A man in a blue shirt is standing on a ladder ...
4	Two men are at the stove preparing food.

```
train_df = pd.concat([train_df_src, train_df_trg], axis=1)
test_df = pd.concat([test_df_src, test_df_trg], axis=1)
train_df.columns = ['src', 'trg']
test_df.columns = ['src', 'trg']
```

```
train_df.head()
```

	src	trg
0	Two young, White males are outside near many b...	Zwei junge weiße Männer sind im Freien in der ...
1	Several men in hard hats are operating a giant...	Mehrere Männer mit Schutzhelmen bedienen ein A...
2	A little girl climbing into a wooden playhouse.	Ein kleines Mädchen klettert in ein Spielhaus ...
3	A man in a blue shirt is standing on a ladder ...	Ein Mann in einem blauen Hemd steht auf einer ...

Tokenization and Building Dataset/DataLoader

```
print(f'Number of Train Examples: {len(train_df)}')
print(f'Number of Test Examples: {len(test_df)}')
```

```
Number of Train Examples: 29000
Number of Test Examples: 1014
```

```
from torchtext.data.utils import get_tokenizer
src_tokenizer = get_tokenizer('spacy', language='en_core_web_sm')
trg_tokenizer = get_tokenizer('spacy', language='de_core_news_sm')
```

```
def build_vocab(df, tokenizer, **vocab_kwarg):

    token_freqs = collections.Counter()

    for index, row  in df.iterrows():
        tokens = tokenizer(row['sentence'])
        token_freqs.update(tokens)

    vocab = torchtext.vocab.Vocab(token_freqs)


    return vocab
```

```
src_vocab = build_vocab(train_df_src, src_tokenizer)
trg_vocab = build_vocab(train_df_trg, trg_tokenizer)
```

```
print('Size of src vocab : ', len(src_vocab.freqs))
print('Size of trg vocab : ', len(trg_vocab.freqs))
```

```
Size of src vocab :  10833
Size of trg vocab :  19210
```

```
def data_process(df):
    data = []
    for index, row in df.iterrows():
      src_tensor_ = torch.tensor([src_vocab[token] for token in src_tokenizer(r
                                  dtype=torch.long)
      trg_tensor_ = torch.tensor([trg_vocab[token] for token in trg_tokenizer(r
                                  dtype=torch.long)
      data.append((src_tensor_, trg_tensor_))
    return data

train_dataset = data_process(train_df)
# val_dataset = data_process(val_df)
test_dataset = data_process(test_df)
```

```
PAD_IDX = src_vocab['<pad>']
print(PAD_IDX)
```

```
1
```

```
class Collator:
    def __init__(self, pad_idx):

        self.pad_idx = pad_idx

    def collate(self, batch):
        src_batch, trg_batch = [], []
        for src_item, trg_item in batch:
          src_batch.append(torch.cat([src_item], dim=0))
          trg_batch.append(torch.cat([trg_item], dim=0))
```

```
        src_batch = nn.utils.rnn.pad_sequence(src_batch, padding_value=self.pad_
        trg_batch = nn.utils.rnn.pad_sequence(trg_batch, padding_value=self.pad_

        return src_batch, trg_batch
```

```
collator = Collator(PAD_IDX)
```

```
batch_size = 32

train_loader = torch.utils.data.DataLoader(train_dataset,
                                           batch_size,
                                           shuffle = True,
                                           collate_fn = collator.collate,
                                           drop_last=True
                                          )

test_loader = torch.utils.data.DataLoader(test_dataset,
                                           batch_size,
                                           shuffle = False,
                                           collate_fn = collator.collate,
                                           drop_last=True
                                          )
```

Initializing GPU as the device

```
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
```

Save the vocabulary for later use

```
with open('src_tokenizer.pkl', 'wb') as tokens:
    pickle.dump(src_vocab.stoi, tokens)

with open('trg_tokenizer.pkl', 'wb') as tokens:
    pickle.dump(trg_vocab.stoi, tokens)

```

Defining Our Model

Boilerplate code

```
class TL(pl.LightningModule):
    def __init__(self):
        super(TL, self).__init__()

        self.train_acc =  torch.tensor(0.)
        self.avg_train_loss = torch.tensor(0.)
        self.table_context = None
```

```
        self.table_context = None


    def training_step(self, batch, batch_idx):
        src, trg = batch
        output = self(src, trg)
        output_dim = output.shape[-1]
        output = output[1:].view(-1, output_dim)
        trg = trg[1:].view(-1)
        loss_train = self.loss(output, trg)
        return loss_train

    def validation_step(self, batch, batch_idx):
        src, trg = batch
        output = self(src, trg, 0)
        output_dim = output.shape[-1]
        output = output[1:].view(-1, output_dim)
        trg = trg[1:].view(-1)
        loss_valid = self.loss(output, trg)
        return {"loss": loss_valid}

    def training_epoch_end(self, outputs):
        self.avg_train_loss = torch.stack([x['loss'] for x in outputs]).mean()

    def validation_epoch_end(self, outputs):
        if trainer.running_sanity_check:
            return
        avg_valid_loss = torch.stack([x['loss'] for x in outputs]).mean()
        metrics = {'epoch': self.current_epoch+1, 'Train PPL': math.exp(self.av
        if self.table_context is None:
            self.table_context = tp.TableContext(headers=['epoch', 'Train PPL',
            self.table_context.__enter__()
        self.table_context([self.current_epoch+1, math.exp(self.avg_train_loss.
        self.logger.log_metrics(metrics)
        if self.current_epoch == self.trainer.max_epochs - 1:
            self.validation_end(outputs)

    def validation_end(self, outputs):
        self.table_context.__exit__()
```

Encoder

```
class Encoder(pl.LightningModule):
    def __init__(self, input_dim, emb_dim, hid_dim, n_layers, dropout):
        super().__init__()

        self.hid_dim = hid_dim
        self.n_layers = n_layers

        self.embedding = nn.Embedding(input_dim, emb_dim)
        self.rnn = nn.LSTM(emb_dim, hid_dim, n_layers, dropout = dropout, batch_
        self.dropout = nn.Dropout(dropout)

```

```
    def forward(self, src):
        embedded = self.dropout(self.embedding(src))
        output, (hidden,cell) = self.rnn(embedded)

        return hidden,cell
```

Decoder

```
class Decoder(pl.LightningModule):
    def __init__(self, emb_dim, hid_dim, n_layers, dropout, output_dim):
        super().__init__()

        self.hid_dim = hid_dim
        self.n_layers = n_layers
        self.output_dim = output_dim
        self.embedding = nn.Embedding(output_dim, emb_dim)
        self.rnn = nn.LSTM(emb_dim, hid_dim, n_layers, dropout = dropout, batch_
        self.fc_out = nn.Linear(hid_dim, output_dim)
        self.dropout = nn.Dropout(dropout)

    def forward(self, input, hidden, cell):
        input = input.unsqueeze(0)
        embedded = self.dropout(self.embedding(input))
        output, (hidden,cell) = self.rnn(embedded, (hidden,cell))
        prediction = self.fc_out(output.squeeze(0))

        return prediction, hidden, cell
```

Seq2Seq Model

```
# Define the model

class Seq2Seq(TL):
    def __init__(self, encoder, decoder, device):
        super(Seq2Seq, self).__init__()

        TRG_PAD_IDX = trg_vocab['<PAD>']
        self.loss = nn.CrossEntropyLoss(ignore_index=TRG_PAD_IDX)
        self.lr = 1e-3

        self.encoder = encoder
        self.decoder = decoder
        # self.device = device # Doesn't work in PyTorchLightning since it is a

        assert encoder.hid_dim == decoder.hid_dim, "Hidden Dimensions of Encode
        assert encoder.n_layers == decoder.n_layers, "Encoder and Decoder must h

    def forward(self, src, trg, teacher_forcing_ratio = 0.5):

        batch_size = trg.shape[1]
        trg_len = trg.shape[0]
```

```
        trg_vocab_size = self.decoder.output_dim
        outputs = torch.zeros(trg_len, batch_size, trg_vocab_size).to(self.devi

        hidden, cell = self.encoder(src)

        input = trg[0,:]

        for t in range(1, trg_len):

            output, hidden,cell = self.decoder(input, hidden, cell)

            outputs[t] = output

            teacher_force = random.random() < teacher_forcing_ratio

            top1 = output.argmax(1)

            input = trg[t] if teacher_force else top1

        return outputs

    def configure_optimizers(self):
        optim = torch.optim.Adam(self.parameters())
        return optim
```

```
INPUT_DIM = len(src_vocab)
OUTPUT_DIM = len(trg_vocab)
ENC_EMB_DIM = 256
DEC_EMB_DIM = 256
HID_DIM = 512
N_LAYERS = 2
ENC_DROPOUT = 0.5
DEC_DROPOUT = 0.5

enc = Encoder(INPUT_DIM, ENC_EMB_DIM, HID_DIM, N_LAYERS, ENC_DROPOUT)
dec = Decoder(DEC_EMB_DIM, HID_DIM, N_LAYERS, DEC_DROPOUT, OUTPUT_DIM)

model = Seq2Seq(enc, dec, device).to(device)
```

Model Checkpoint

This saves the best model (best => model with lowest val loss)

```
checkpoint_callback = ModelCheckpoint(
    monitor='val_loss',
    dirpath='/content',
    filename='sst-{epoch:02d}-{val_loss:.2f}',
    mode='min'
)
```

```
!rm -rf csv_logs
```

```
2 csvlogger = CSVLogger('csv_logs', name='END2 Assign 7_2_TL', version=0)
3 trainer = pl.Trainer(max_epochs=20, num_sanity_val_steps=1, logger=csvlogger, g
4 trainer.fit(model, train_dataloader=train_loader, val_dataloaders=test_loader)
5 checkpoint_callback.best_model_path
```

```
GPU available: True, used: True
TPU available: False, using: 0 TPU cores
LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES: [0]

  | Name    | Type             | Params
-------------------------------------------
0 | loss    | CrossEntropyLoss | 0
1 | encoder | Encoder          | 6.5 M
2 | decoder | Decoder          | 18.5 M
-------------------------------------------
24.9 M    Trainable params
0         Non-trainable params
24.9 M    Total params
99.617    Total estimated model params size (MB)
```

Validation sanity check: 0% 0/1 [22:40<?, ?it/s]

Epoch 19: 100% 937/937 [01:46<00:00, 8.77it/s, loss=1.13, v_num=0]

epoch	Train PPL	Train Loss	Valid PPL	Valid Loss
1	20.194	3.0054	17.243	2.8474
2	12.579	2.532	14.515	2.6752
3	10.336	2.3356	13.238	2.5831
4	8.6194	2.154	12.732	2.5441
5	7.5958	2.0276	11.899	2.4765
6	6.6891	1.9005	11.787	2.467
7	6.0315	1.797	11.185	2.4146
8	5.4878	1.7025	11.416	2.435
9	5.0367	1.6167	11.442	2.4373
10	4.6311	1.5328	11.426	2.4359
11	4.3749	1.4759	11.332	2.4276
12	4.0922	1.4091	11.794	2.4676
13	3.869	1.353	12.21	2.5022
14	3.6742	1.3013	12.088	2.4922
15	3.5334	1.2623	12.359	2.5143
16	3.3694	1.2147	12.429	2.5201
17	3.2581	1.1812	12.436	2.5206
18	3.1669	1.1528	12.548	2.5295
19	3.0316	1.1091	13.289	2.5869
20	2.9951	1.097	13.33	2.5901

''

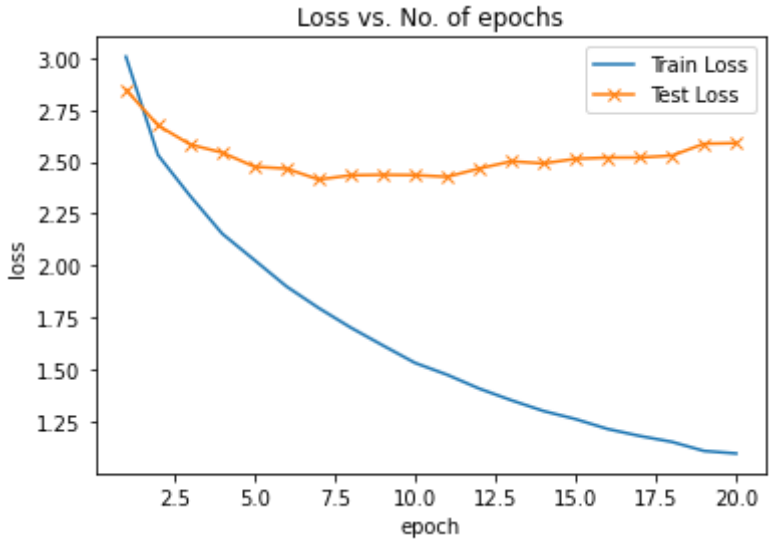
Training Log

First define the optimizer and loss functions

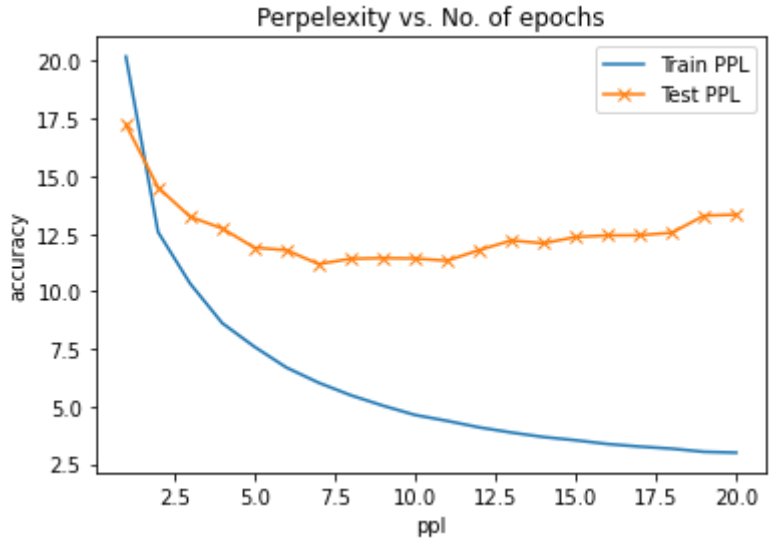
```
1 root='./csv_logs/' + 'END2 Assign 7_2_TL' + '/'
2 dirlist = [ item for item in os.listdir(root) if os.path.isdir(os.path.join(roo
3 metricfile = root + dirlist[ 1:][0] + '/metrics.csv'
```

```
metricfile = root + dirlist[-1.][0] + "/metrics.csv"
metrics = pd.read_csv(metricfile)
```

```
plt.plot(metrics['epoch'], metrics['Train Loss'], label="Train Loss")
plt.plot(metrics['epoch'], metrics['Valid Loss'], '-x', label="Test Loss")
plt.xlabel('epoch')
plt.ylabel('loss')
plt.legend()
plt.title('Loss vs. No. of epochs');
```




```
plt.plot(metrics['epoch'], metrics['Train PPL'], label="Train PPL")
plt.plot(metrics['epoch'], metrics['Valid PPL'], '-x', label="Test PPL")
plt.xlabel('ppl')
plt.ylabel('accuracy')
plt.legend()
plt.title('Perpelexity vs. No. of epochs');
```



Inference on Random Samples from Test Data

```
model.to(device)
model.eval()
```

```
2 model.eval()
```

```
Seq2Seq(
  (loss): CrossEntropyLoss()
  (encoder): Encoder(
    (embedding): Embedding(10835, 256)
    (rnn): LSTM(256, 512, num_layers=2, dropout=0.5)
    (dropout): Dropout(p=0.5, inplace=False)
  )
  (decoder): Decoder(
    (embedding): Embedding(19212, 256)
    (rnn): LSTM(256, 512, num_layers=2, dropout=0.5)
    (fc_out): Linear(in_features=512, out_features=19212, bias=True)
    (dropout): Dropout(p=0.5, inplace=False)
  )
)
```

```
1 for i in np.random.randint(0,len(test_df_src), 10):
2   src_sent = test_df_src.iloc[i]['sentence']
3   trg_sent = test_df_trg.iloc[i]['sentence']
4   src_sent_tensor = torch.tensor([src_vocab[token] for token in src_tokenizer(s
5   trg_sent_tensor = torch.tensor([trg_vocab[token] for token in trg_tokenizer(t
6   with torch.no_grad():
7         output = model(src_sent_tensor, trg_sent_tensor, 1)
8         # output_dim = output.shape[-1]
9         # output = output[1:].view(-1, output_dim)
10        out = output.squeeze(1)
11        out = torch.argmax(out,dim=1)
12        trans = []
13        for c in out[1:]:
14          trans.append(trg_vocab.itos[c])
15        st = " ".join(trans)
16        start = "\033[1m"
17        end = "\033[0;0m"
18        print(f'{start}Source Sentence: {end}{src_sent}')
19        print(f'{start}Target Sentence: {end}{trg_sent}')
20        print(f'{start}Translated Sentence: {end}{st}')
21        print()
```

Source Sentence: A boy wearing a green shirt on a bicycle reflecting off a st
Target Sentence: Das Bild eines Jungen in einem grünen T-Shirt, der auf einem
Translated Sentence: Junge in grünen auf einem grünen Hemd auf der auf dem Da

Source Sentence: A woman is standing and wearing a green and yellow scarf.
Target Sentence: Eine stehende Frau trägt einen grün-gelben Schal.
Translated Sentence: Frau Frau in Pompoms blau-grünen Bikini und

Source Sentence: A large crowd stand watching with a large buildings in the b
Target Sentence: Eine große Menschenmenge sieht mit großen Gebäuden im Hinter
Translated Sentence: große Menschenmenge steht sich einem großen , Hintergrun

Source Sentence: A bunch of young adults stare in concentration at their comp
Target Sentence: Mehrere junge Erwachsene starren konzentriert auf ihre Compu
Translated Sentence: junge Erwachsene Erwachsene in in in Schule in in sie in

Source Sentence: One man, wearing a hooded sweatshirt, sitting at a fountain
Target Sentence: Ein Mann mit einem Kapuzenshirt sitzt an einem Springbrunnen
Translated Sentence: Mann in einem offiziellen und an einem Fußgängerüberweg

Source Sentence: A man playing a keyboard and singing into a microphone.
Target Sentence: Eine Frau spielt Keyboard und singt in ein Mikrofon.
Translated Sentence: Mann spielt Keyboard und singt in ein Mikrofon .

Source Sentence: A motel valet man wearing a trench coat pushing a load of lu
Target Sentence: Ein Hoteldiener in einem Mantel schiebt eine Ladung Gepäck.
Translated Sentence: Mann mit einem Mantel mit einen riesige mit .

Source Sentence: A man in a harness climbing a rock wall
Target Sentence: Ein Mann in einem Klettergurt klettert an einer Felswand
Translated Sentence: Mann in einem Kilt macht eine . Felswand

Source Sentence: Several people standing on a subway platform.
Target Sentence: Einige Menschen stehen auf einem U-Bahnsteig.
Translated Sentence: Leute stehen auf einer Bahnsteig .

Source Sentence: A little baby in a pink hat lying naked and sleeping.
Target Sentence: Ein kleines Baby mit einer rosafarbenen Mütze liegt nackt da
Translated Sentence: kleines Baby mit einem rosafarbenen Mütze trägt auf und

✓ 0s completed at 10:35 AM