In [1]:

```
import torch
import torch.nn as nn
import torch.optim as optim
import numpy as np
import torch.nn.functional as F
from torch.utils.data import DataLoader
from rnn.AWD_LSTM import AWDLSTM
import urllib.request
import os
import unidecode
import string
import random
import re
import time
from torch.optim.lr_scheduler import ReduceLROnPlateau
from torch.optim.lr_scheduler import CosineAnnealingLR
from rnn.helpers import time_since
from rnn.generate import generate
%matplotlib inline
import pdb
%load_ext autoreload
%autoreload 2
```

In [2]:

```
import torch
print(torch.__version__)
```

```
1.10.2+cu111
```

In [3]:

```
# vocab = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 .,
# vocab_mapping = {char: i for i, char in enumerate(vocab)}
```

In [4]:

```
device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")
```

In [24]:

```
# Download text file if it does not exist
url = "https://www.gutenberg.org/files/11/11-0.txt"
filename = "language_data/input.txt"
if not os.path.isfile(filename):
    urllib.request.urlretrieve(url, filename)

# Load and preprocess text
file_path = 'language_data/input.txt'
file = unidecode.unidecode(open(file_path).read())
all_characters = string.printable
file_chars = [c for c in file if c in all_characters]
char_to_idx = {char: i for i, char in enumerate(all_characters)}
idx_to_char = {i: ch for i, ch in enumerate(all_characters)}
vocab_size = len(all_characters)

# Split data into train and test sets
split = int(0.9 * len(file_chars))
train_text = file_chars[:split]
test_text = file_chars[split:]

print('train len: ', len(train_text))
print('test len: ', len(test_text))
print('All characters: ', all_characters)

chunk_len = 200
```

```
train len:  11842726
test len:  1315859
All characters:  0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVW
XYZ!"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
□□
```

In [25]:

```
import urllib.request

# Download the WikiText-2 dataset
url = "https://s3.amazonaws.com/research.metamind.io/wikitext/wikitext-2-v1.
urllib.request.urlretrieve(url, "wikitext-2-v1.zip")

# Extract the dataset
import zipfile
with zipfile.ZipFile("wikitext-2-v1.zip", "r") as zip_ref:
    zip_ref.extractall()

# Read in the text from the dataset
with open("wikitext-2/wiki.test.tokens", "r") as f:
    text = f.read()

# Get the first 1000 characters
text_1000 = text[:1000]

print(text_1000)
```

```
 = Robert <unk> =

 Robert <unk> is an English film , television and theatre actor . He had a g
uest @-@ starring role on the television series The Bill in 2000 . This was
followed by a starring role in the play Herons written by Simon Stephens , w
hich was performed in 2001 at the Royal Court Theatre . He had a guest role
in the television series Judge John <unk> in 2002 . In 2004 <unk> landed a r
ole as " Craig " in the episode " Teddy 's Story " of the television series
The Long Firm ; he starred alongside actors Mark Strong and Derek Jacobi . H
e was cast in the 2005 theatre productions of the Philip Ridley play Mercury
Fur , which was performed at the Drum Theatre in Plymouth and the <unk> <unk
> Factory in London . He was directed by John <unk> and starred alongside Be
n <unk> , Shane <unk> , Harry Kent , Fraser <unk> , Sophie Stanton and Domin
ic Hall .
 In 2006 , <unk> starred alongside <unk> in the play <unk> written by Mark <
unk> . He appeared on a 2006 episode of the televisio
```

In [6]:

```python
chunk_len = 200

def random_chunk(text):
    start_index = random.randint(0, len(text) - chunk_len)
    end_index = start_index + chunk_len + 1
    return text[start_index:end_index]

print(random_chunk(train_text))
```

```
['o', 'f', 'f', ' ', '.', ' ', 'I', 't', ' ', 'c', 'e', 'a', 's', 'e', 's',
' ', 'a', 'l', 'm', 'o', 's', 't', ' ', 'e', 'n', 't', 'i', 'r', 'e', 'l', '
y', ' ', 'n', 'e', 'a', 'r', ' ', 't', 'h', 'e', ' ', 'n', 'o', 'r', 't', 'h
', 'e', 'r', 'n', ' ', 'c', 'i', 't', 'y', ' ', 'l', 'i', 'n', 'e', ' ', 'a'
, 't', ' ', 'Y', 'o', 'r', 'k', ' ', 'S', 't', 'r', 'e', 'e', 't', ' ', ',',
' ', 'w', 'h', 'e', 'r', 'e', ' ', 't', 'h', 'e', ' ', 'h', 'o', 'm', 'e', '
s', ' ', 'a', 'l', 'o', 'n', 'g', ' ', 't', 'h', 'e', ' ', 'h', 'i', 'g', 'h
', 'w', 'a', 'y', ' ', 'b', 'e', 'c', 'o', 'm', 'e', ' ', 'm', 'o', 'r', 'e'
, ' ', 's', 'p', 'o', 'r', 'a', 'd', 'i', 'c', ' ', 'a', 'n', 'd', ' ', 's',
'p', 'a', 'c', 'e', 'd', ' ', 'a', 'p', 'a', 'r', 't', ' ', '.', ' ', '\n',
' ', '\n', ' ', '=', ' ', '=', ' ', '=', ' ', '=', ' ', 'N', 'o', 'r', 't',
'h', ' ', 'o', 'f', ' ', 'A', 'u', 'b', 'u', 'r', 'n', ' ', '=', ' ', '=', '
', '=', ' ', '=', ' ', '\n', ' ', '\n', ' ', 'N', 'o', 'w', ' ', 'i', 'n', '
', 't', 'h', 'e']
```

In [7]:

```
def char_tensor(string, batch_size=1):
    if isinstance(string, str):
        string = [string]
    tensor = torch.zeros(len(string), batch_size).long()
    for b in range(batch_size):
        for c in range(len(string)):
            char = string[c][b]
            if char in char_to_idx:
                tensor[c][b] = char_to_idx[char]
    return tensor


def load_random_batch(text, seq_length, batch_size):
    inputs = torch.zeros(seq_length, batch_size).long()
    targets = torch.zeros(seq_length, batch_size).long()

    for i in range(batch_size):
        start_index = random.randint(0, len(text) - seq_length)
        end_index = start_index + seq_length + 1
        chunk = text[start_index:end_index]
        inputs[:, i] = char_tensor(chunk[:-1], batch_size=1)[:, 0]
        targets[:, i] = char_tensor(chunk[1:], batch_size=1)[:, 0]

    return inputs, targets

def text_to_tensor(text, vocab_mapping):
    indices = [vocab_mapping[char] for char in text]
    return torch.tensor(indices, dtype=torch.long)
```

In [20]:

```
def generate(model, prime_str='A', predict_len=100, temperature=0.8, device=
    hidden = (torch.zeros(model.num_layers, 2, model.hidden_size, device=dev
              torch.zeros(model.num_layers, 2, model.hidden_size, device=dev
    prime_input = torch.tensor([char_to_idx[c] for c in prime_str], dtype=to
    generated_text = prime_str

    with torch.no_grad():
        for i in range(predict_len):
            output, hidden = model(prime_input.view(1, -1), hidden)
            output_dist = output.squeeze().div(temperature).exp()
            top_i = torch.multinomial(output_dist, 1)[0]
            generated_char = idx_to_char[top_i]
            generated_text += generated_char
            prime_input.fill_(top_i)

    return generated_text
```

In [9]:

```
def eval_test(rnn, inp, target):
    with torch.no_grad():
        hidden = rnn.init_hidden(inp.shape[1], device=device)
        loss = 0
        for c in range(chunk_len):
            output, hidden = rnn(inp[:,c], hidden)
            loss += criterion(output.view(inp.shape[1], -1), target[:,c])

    return loss.data.item() / chunk_len
```

In [10]:

```
# Hyperparameters
input_size = vocab_size
embedding_size = 300
hidden_size = 1024
num_layers = 10
dropout = 0.05
weight_dropout = 0.05
batch_size = 1024
seq_length = 1024
learning_rate = 0.001
num_epochs = 5000
bptt=2000
```

In [14]:

```
# Define the model, loss function, optimizer, and learning rate scheduler
input_size = len(all_characters)
awd_lstm = AWDLSTM(input_size, hidden_size, num_layers, embedding_size=embed
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(awd_lstm.parameters(), lr=learning_rate)
scheduler = ReduceLROnPlateau(optimizer, mode='min', factor=0.99, patience=1

# Split data into train and validation sets
split = int(0.9 * len(train_text))
train_text = file_chars[:split]
valid_text = file_chars[split:]

print('train len: ', len(train_text))
print('valid len: ', len(valid_text))

# Define data loaders
train_loader = DataLoader(train_text, batch_size=batch_size, shuffle=True)
valid_loader = DataLoader(valid_text, batch_size=batch_size, shuffle=True)

# Training loop
awd_lstm.train()
for epoch in range(num_epochs):
    epoch_loss = 0.0
    hidden = None
    num_batches = 0
    for batch_idx in range(0, (len(file) - seq_length) // (batch_size * seq_
        input_batch, target_batch = load_random_batch(file[batch_idx*seq_len
        num_batches += 1
        for chunk_start in range(0, seq_length, bptt):
            chunk_input = char_tensor(input_batch[:, chunk_start:chunk_start
            chunk_target = char_tensor(target_batch[:, chunk_start:chunk_sta

            # Forward pass
            output, hidden = awd_lstm(chunk_input, hidden)
            hidden = (hidden[0].detach(), hidden[1].detach())  # Detach hidc
            output = output.view(-1, output.size(2))
            chunk_target = chunk_target.reshape(-1)
            loss = criterion(output, chunk_target)

            # Backward pass and optimization
            optimizer.zero_grad()
            loss.backward()
            optimizer.step()

            epoch_loss += loss.item()

    epoch_loss /= num_batches
    if (epoch + 1) % 10 == 0:
        print(f"Epoch [{epoch + 1}/{num_epochs}], Loss: {epoch_loss:.4f}")

    # Update the learning rate based on the epoch loss
    scheduler.step(epoch_loss)
```

```
train len:  8633346
valid len:  4525239
Epoch [10/5000], Loss: 5.5386
Epoch [20/5000], Loss: 5.4542
Epoch [30/5000], Loss: 5.4490
Epoch [40/5000], Loss: 5.4462
Epoch [50/5000], Loss: 5.4345
Epoch [60/5000], Loss: 5.4241
Epoch [70/5000], Loss: 5.4196
Epoch [80/5000], Loss: 5.4138
Epoch [90/5000], Loss: 5.4149
Epoch [100/5000], Loss: 5.4143
Epoch [110/5000], Loss: 5.4095
Epoch [120/5000], Loss: 5.4006
Epoch [130/5000], Loss: 5.4113
Epoch [140/5000], Loss: 5.3985
Epoch [150/5000], Loss: 5.3948
Epoch [160/5000], Loss: 5.3997
Epoch [170/5000], Loss: 5.4008
Epoch [180/5000], Loss: 5.3906
Epoch [190/5000], Loss: 5.3897
Epoch [200/5000], Loss: 5.3887
Epoch [210/5000], Loss: 5.4044
Epoch [220/5000], Loss: 5.3958
Epoch [230/5000], Loss: 5.3968
Epoch [240/5000], Loss: 5.3842
Epoch [250/5000], Loss: 5.3705
Epoch [260/5000], Loss: 5.3775
Epoch [270/5000], Loss: 5.3843
Epoch [280/5000], Loss: 5.3791
Epoch [290/5000], Loss: 5.3737
Epoch [300/5000], Loss: 5.3857
Epoch [310/5000], Loss: 5.3714
Epoch [320/5000], Loss: 5.3767
Epoch [330/5000], Loss: 5.3678
Epoch [340/5000], Loss: 5.3724
Epoch [350/5000], Loss: 5.3624
Epoch [360/5000], Loss: 5.3522
Epoch [370/5000], Loss: 5.3581
Epoch [380/5000], Loss: 5.3445
Epoch [390/5000], Loss: 5.3539
Epoch [400/5000], Loss: 5.3556
Epoch [410/5000], Loss: 5.3553
Epoch [420/5000], Loss: 5.3551
Epoch [430/5000], Loss: 5.3427
Epoch [440/5000], Loss: 5.3421
Epoch [450/5000], Loss: 5.3478
Epoch [460/5000], Loss: 5.3349
Epoch [470/5000], Loss: 5.3340
Epoch [480/5000], Loss: 5.3155
Epoch [490/5000], Loss: 5.3147
Epoch [500/5000], Loss: 5.3047
Epoch [510/5000], Loss: 5.3131
Epoch [520/5000], Loss: 5.2845
Epoch [530/5000], Loss: 5.2834
Epoch [540/5000], Loss: 5.2811
Epoch [550/5000], Loss: 5.2614
Epoch [560/5000], Loss: 5.2585
Epoch [570/5000], Loss: 5.2529
Epoch [580/5000], Loss: 5.2146
Epoch [590/5000], Loss: 5.2156
Epoch [600/5000], Loss: 5.1840
Epoch [610/5000], Loss: 5.1728
Epoch [620/5000], Loss: 5.1492
Epoch [630/5000], Loss: 5.1135
Epoch [640/5000], Loss: 5.1023
Epoch [650/5000], Loss: 5.0658
Epoch [660/5000], Loss: 5.0638
Epoch [670/5000], Loss: 5.0476
Epoch [680/5000], Loss: 5.0553
Epoch [690/5000], Loss: 5.0239
Epoch [700/5000], Loss: 5.0156
Epoch [710/5000], Loss: 5.0134
Epoch [720/5000], Loss: 5.0114
Epoch [730/5000], Loss: 5.0196
Epoch [740/5000], Loss: 5.0157
Epoch [750/5000], Loss: 5.0101
Epoch [760/5000], Loss: 4.9983
Epoch [770/5000], Loss: 5.0122
Epoch [780/5000], Loss: 5.0107
Epoch [790/5000], Loss: 5.0165
Epoch [800/5000], Loss: 5.0058
Epoch [810/5000], Loss: 4.9988
Epoch [820/5000], Loss: 5.0118
Epoch [830/5000], Loss: 5.0001
Epoch [840/5000], Loss: 5.0046
Epoch [850/5000], Loss: 5.0070
Epoch [860/5000], Loss: 5.0077
Epoch   866: reducing learning rate of group 0 to 9.9000e-04.
Epoch [870/5000], Loss: 5.0073
Epoch [880/5000], Loss: 5.0079
Epoch [890/5000], Loss: 4.9916
Epoch [900/5000], Loss: 5.0090
Epoch [910/5000], Loss: 5.0041
Epoch [920/5000], Loss: 4.9947
Epoch [930/5000], Loss: 5.0035
Epoch [940/5000], Loss: 4.9988
Epoch [950/5000], Loss: 4.9990
Epoch [960/5000], Loss: 4.9840
Epoch [970/5000], Loss: 4.9985
Epoch [980/5000], Loss: 4.9886
Epoch [990/5000], Loss: 4.9941
Epoch [1000/5000], Loss: 4.9860
Epoch [1010/5000], Loss: 4.9995
Epoch [1020/5000], Loss: 4.9954
Epoch [1030/5000], Loss: 4.9943
Epoch [1040/5000], Loss: 5.0071
Epoch [1050/5000], Loss: 4.9933
Epoch [1060/5000], Loss: 4.9998
Epoch [1070/5000], Loss: 4.9935
Epoch [1080/5000], Loss: 5.0123
Epoch [1090/5000], Loss: 5.0076
Epoch [1100/5000], Loss: 5.0024
Epoch [1110/5000], Loss: 5.0025
Epoch [1120/5000], Loss: 5.0081
Epoch [1130/5000], Loss: 4.9939
Epoch [1140/5000], Loss: 4.9959
Epoch [1150/5000], Loss: 4.9853
Epoch [1160/5000], Loss: 5.0077
Epoch [1170/5000], Loss: 5.0006
Epoch [1180/5000], Loss: 5.0014
Epoch  1186: reducing learning rate of group 0 to 9.8010e-04.
Epoch [1190/5000], Loss: 5.0027
Epoch [1200/5000], Loss: 4.9871
Epoch [1210/5000], Loss: 4.9899
Epoch [1220/5000], Loss: 5.0019
Epoch [1230/5000], Loss: 5.0037
Epoch [1240/5000], Loss: 5.0060
Epoch [1250/5000], Loss: 5.0018
Epoch [1260/5000], Loss: 5.0060
Epoch [1270/5000], Loss: 4.9938
Epoch [1280/5000], Loss: 4.9991
Epoch  1287: reducing learning rate of group 0 to 9.7030e-04.
Epoch [1290/5000], Loss: 4.9951
Epoch [1300/5000], Loss: 5.0087
Epoch [1310/5000], Loss: 4.9853
Epoch [1320/5000], Loss: 5.0192
Epoch [1330/5000], Loss: 4.9867
Epoch [1340/5000], Loss: 4.9964
Epoch [1350/5000], Loss: 5.0034
Epoch [1360/5000], Loss: 4.9868
Epoch [1370/5000], Loss: 5.0017
Epoch [1380/5000], Loss: 4.9893
Epoch  1388: reducing learning rate of group 0 to 9.6060e-04.
Epoch [1390/5000], Loss: 5.0093
Epoch [1400/5000], Loss: 4.9940
Epoch [1410/5000], Loss: 5.0091
Epoch [1420/5000], Loss: 4.9989
Epoch [1430/5000], Loss: 4.9921
Epoch [1440/5000], Loss: 4.9896
Epoch [1450/5000], Loss: 4.9927
Epoch [1460/5000], Loss: 4.9986
Epoch [1470/5000], Loss: 5.0131
Epoch [1480/5000], Loss: 4.9922
Epoch  1489: reducing learning rate of group 0 to 9.5099e-04.
Epoch [1490/5000], Loss: 5.0055
Epoch [1500/5000], Loss: 4.9916
Epoch [1510/5000], Loss: 4.9981
Epoch [1520/5000], Loss: 4.9914
Epoch [1530/5000], Loss: 5.0009
Epoch [1540/5000], Loss: 5.0097
Epoch [1550/5000], Loss: 5.0005
Epoch [1560/5000], Loss: 4.9914
Epoch [1570/5000], Loss: 5.0004
Epoch [1580/5000], Loss: 5.0025
Epoch [1590/5000], Loss: 4.9916
Epoch  1590: reducing learning rate of group 0 to 9.4148e-04.
Epoch [1600/5000], Loss: 4.9911
Epoch [1610/5000], Loss: 5.0050
Epoch [1620/5000], Loss: 5.0035
Epoch [1630/5000], Loss: 4.9893
Epoch [1640/5000], Loss: 4.9923
Epoch [1650/5000], Loss: 4.9925
Epoch [1660/5000], Loss: 4.9885
Epoch [1670/5000], Loss: 5.0056
Epoch [1680/5000], Loss: 5.0060
Epoch [1690/5000], Loss: 5.0075
Epoch  1691: reducing learning rate of group 0 to 9.3207e-04.
Epoch [1700/5000], Loss: 4.9922
Epoch [1710/5000], Loss: 4.9970
Epoch [1720/5000], Loss: 4.9888
Epoch [1730/5000], Loss: 4.9922
Epoch [1740/5000], Loss: 5.0007
Epoch [1750/5000], Loss: 4.9995
Epoch [1760/5000], Loss: 4.9886
Epoch [1770/5000], Loss: 4.9884
Epoch [1780/5000], Loss: 5.0019
Epoch [1790/5000], Loss: 4.9986
Epoch  1792: reducing learning rate of group 0 to 9.2274e-04.
Epoch [1800/5000], Loss: 4.9976
Epoch [1810/5000], Loss: 5.0002
Epoch [1820/5000], Loss: 5.0015
Epoch [1830/5000], Loss: 4.9957
Epoch [1840/5000], Loss: 5.0029
Epoch [1850/5000], Loss: 4.9949
Epoch [1860/5000], Loss: 4.9977
Epoch [1870/5000], Loss: 4.9894
Epoch [1880/5000], Loss: 5.0014
Epoch [1890/5000], Loss: 4.9993
Epoch  1893: reducing learning rate of group 0 to 9.1352e-04.
Epoch [1900/5000], Loss: 5.0091
Epoch [1910/5000], Loss: 4.9907
Epoch [1920/5000], Loss: 4.9939
Epoch [1930/5000], Loss: 5.0088
Epoch [1940/5000], Loss: 5.0026
Epoch [1950/5000], Loss: 5.0091
Epoch [1960/5000], Loss: 4.9980
Epoch [1970/5000], Loss: 5.0030
Epoch [1980/5000], Loss: 5.0019
Epoch [1990/5000], Loss: 5.0010
Epoch  1994: reducing learning rate of group 0 to 9.0438e-04.
Epoch [2000/5000], Loss: 4.9976
Epoch [2010/5000], Loss: 5.0052
Epoch [2020/5000], Loss: 4.9901
Epoch [2030/5000], Loss: 4.9937
Epoch [2040/5000], Loss: 5.0045
Epoch [2050/5000], Loss: 4.9867
Epoch [2060/5000], Loss: 5.0063
Epoch [2070/5000], Loss: 5.0035
Epoch [2080/5000], Loss: 5.0092
Epoch [2090/5000], Loss: 4.9987
Epoch  2095: reducing learning rate of group 0 to 8.9534e-04.
Epoch [2100/5000], Loss: 4.9959
Epoch [2110/5000], Loss: 4.9865
Epoch [2120/5000], Loss: 5.0070
Epoch [2130/5000], Loss: 5.0032
Epoch [2140/5000], Loss: 4.9880
Epoch [2150/5000], Loss: 4.9886
Epoch [2160/5000], Loss: 4.9913
Epoch [2170/5000], Loss: 4.9985
Epoch [2180/5000], Loss: 5.0047
Epoch [2190/5000], Loss: 5.0015
Epoch  2196: reducing learning rate of group 0 to 8.8638e-04.
Epoch [2200/5000], Loss: 4.9976
Epoch [2210/5000], Loss: 4.9994
Epoch [2220/5000], Loss: 5.0072
Epoch [2230/5000], Loss: 4.9985
Epoch [2240/5000], Loss: 4.9934
Epoch [2250/5000], Loss: 5.0025
Epoch [2260/5000], Loss: 4.9929
Epoch [2270/5000], Loss: 5.0035
Epoch [2280/5000], Loss: 4.9917
Epoch [2290/5000], Loss: 5.0019
Epoch  2297: reducing learning rate of group 0 to 8.7752e-04.
Epoch [2300/5000], Loss: 4.9815
Epoch [2310/5000], Loss: 5.0026
Epoch [2320/5000], Loss: 4.9974
Epoch [2330/5000], Loss: 5.0090
Epoch [2340/5000], Loss: 5.0008
Epoch [2350/5000], Loss: 4.9986
Epoch [2360/5000], Loss: 4.9907
Epoch [2370/5000], Loss: 4.9981
Epoch [2380/5000], Loss: 4.9964
Epoch [2390/5000], Loss: 4.9911
Epoch  2398: reducing learning rate of group 0 to 8.6875e-04.
Epoch [2400/5000], Loss: 4.9966
Epoch [2410/5000], Loss: 5.0117
Epoch [2420/5000], Loss: 5.0017
Epoch [2430/5000], Loss: 5.0103
Epoch [2440/5000], Loss: 5.0018
Epoch [2450/5000], Loss: 4.9969
Epoch [2460/5000], Loss: 4.9906
Epoch [2470/5000], Loss: 5.0147
Epoch [2480/5000], Loss: 5.0057
Epoch [2490/5000], Loss: 4.9985
Epoch  2499: reducing learning rate of group 0 to 8.6006e-04.
Epoch [2500/5000], Loss: 5.0015
Epoch [2510/5000], Loss: 4.9977
Epoch [2520/5000], Loss: 5.0034
Epoch [2530/5000], Loss: 4.9991
Epoch [2540/5000], Loss: 4.9878
Epoch [2550/5000], Loss: 4.9916
Epoch [2560/5000], Loss: 5.0077
Epoch [2570/5000], Loss: 5.0038
Epoch [2580/5000], Loss: 4.9854
Epoch [2590/5000], Loss: 5.0131
Epoch [2600/5000], Loss: 4.9924
Epoch  2600: reducing learning rate of group 0 to 8.5146e-04.
Epoch [2610/5000], Loss: 4.9963
Epoch [2620/5000], Loss: 5.0036
Epoch [2630/5000], Loss: 4.9980
Epoch [2640/5000], Loss: 4.9900
Epoch [2650/5000], Loss: 4.9983
Epoch [2660/5000], Loss: 5.0009
Epoch [2670/5000], Loss: 5.0005
Epoch [2680/5000], Loss: 4.9954
Epoch [2690/5000], Loss: 4.9993
Epoch [2700/5000], Loss: 5.0004
Epoch  2701: reducing learning rate of group 0 to 8.4294e-04.
Epoch [2710/5000], Loss: 4.9909
Epoch [2720/5000], Loss: 4.9939
Epoch [2730/5000], Loss: 4.9967
Epoch [2740/5000], Loss: 4.9990
Epoch [2750/5000], Loss: 4.9912
Epoch [2760/5000], Loss: 4.9837
Epoch [2770/5000], Loss: 5.0002
Epoch [2780/5000], Loss: 5.0081
Epoch [2790/5000], Loss: 4.9935
Epoch [2800/5000], Loss: 4.9959
Epoch  2802: reducing learning rate of group 0 to 8.3451e-04.
Epoch [2810/5000], Loss: 4.9959
Epoch [2820/5000], Loss: 5.0043
Epoch [2830/5000], Loss: 4.9898
Epoch [2840/5000], Loss: 5.0031
Epoch [2850/5000], Loss: 5.0096
Epoch [2860/5000], Loss: 5.0018
Epoch [2870/5000], Loss: 4.9943
Epoch [2880/5000], Loss: 5.0011
Epoch [2890/5000], Loss: 4.9884
Epoch [2900/5000], Loss: 5.0006
Epoch  2903: reducing learning rate of group 0 to 8.2617e-04.
Epoch [2910/5000], Loss: 5.0052
Epoch [2920/5000], Loss: 4.9881
Epoch [2930/5000], Loss: 4.9925
Epoch [2940/5000], Loss: 4.9894
Epoch [2950/5000], Loss: 5.0001
Epoch [2960/5000], Loss: 4.9807
Epoch [2970/5000], Loss: 4.9928
Epoch [2980/5000], Loss: 5.0019
Epoch [2990/5000], Loss: 5.0008
Epoch [3000/5000], Loss: 4.9909
Epoch  3004: reducing learning rate of group 0 to 8.1791e-04.
Epoch [3010/5000], Loss: 5.0005
Epoch [3020/5000], Loss: 4.9884
Epoch [3030/5000], Loss: 5.0093
Epoch [3040/5000], Loss: 4.9930
Epoch [3050/5000], Loss: 4.9955
Epoch [3060/5000], Loss: 4.9888
Epoch [3070/5000], Loss: 5.0046
Epoch [3080/5000], Loss: 4.9934
Epoch [3090/5000], Loss: 5.0158
Epoch [3100/5000], Loss: 4.9956
Epoch  3108: reducing learning rate of group 0 to 8.0973e-04.
Epoch [3110/5000], Loss: 4.9935
Epoch [3120/5000], Loss: 4.9958
Epoch [3130/5000], Loss: 4.9878
Epoch [3140/5000], Loss: 4.9896
Epoch [3150/5000], Loss: 4.9833
Epoch [3160/5000], Loss: 5.0158
Epoch [3170/5000], Loss: 5.0017
Epoch [3180/5000], Loss: 5.0046
Epoch [3190/5000], Loss: 5.0106
Epoch [3200/5000], Loss: 4.9884
Epoch  3209: reducing learning rate of group 0 to 8.0163e-04.
Epoch [3210/5000], Loss: 5.0008
Epoch [3220/5000], Loss: 4.9950
Epoch [3230/5000], Loss: 4.9950
Epoch [3240/5000], Loss: 5.0017
Epoch [3250/5000], Loss: 4.9949
Epoch [3260/5000], Loss: 4.9948
Epoch [3270/5000], Loss: 4.9926
Epoch [3280/5000], Loss: 4.9927
Epoch [3290/5000], Loss: 5.0053
Epoch [3300/5000], Loss: 4.9964
Epoch [3310/5000], Loss: 4.9963
Epoch  3310: reducing learning rate of group 0 to 7.9361e-04.
Epoch [3320/5000], Loss: 4.9961
Epoch [3330/5000], Loss: 4.9892
Epoch [3340/5000], Loss: 5.0012
Epoch [3350/5000], Loss: 4.9914
Epoch [3360/5000], Loss: 4.9954
Epoch [3370/5000], Loss: 5.0138
Epoch [3380/5000], Loss: 5.0022
Epoch [3390/5000], Loss: 4.9919
Epoch [3400/5000], Loss: 4.9986
Epoch [3410/5000], Loss: 5.0161
Epoch  3411: reducing learning rate of group 0 to 7.8568e-04.
Epoch [3420/5000], Loss: 5.0080
Epoch [3430/5000], Loss: 5.0007
Epoch [3440/5000], Loss: 5.0032
Epoch [3450/5000], Loss: 4.9969
Epoch [3460/5000], Loss: 4.9968
Epoch [3470/5000], Loss: 5.0014
Epoch [3480/5000], Loss: 4.9876
Epoch [3490/5000], Loss: 4.9916
Epoch [3500/5000], Loss: 4.9964
Epoch [3510/5000], Loss: 5.0181
Epoch  3512: reducing learning rate of group 0 to 7.7782e-04.
Epoch [3520/5000], Loss: 4.9925
Epoch [3530/5000], Loss: 4.9947
Epoch [3540/5000], Loss: 4.9926
Epoch [3550/5000], Loss: 4.9932
Epoch [3560/5000], Loss: 4.9935
Epoch [3570/5000], Loss: 5.0038
Epoch [3580/5000], Loss: 4.9936
Epoch [3590/5000], Loss: 5.0036
Epoch [3600/5000], Loss: 5.0047
Epoch [3610/5000], Loss: 5.0013
Epoch  3613: reducing learning rate of group 0 to 7.7004e-04.
Epoch [3620/5000], Loss: 4.9983
Epoch [3630/5000], Loss: 5.0064
Epoch [3640/5000], Loss: 4.9863
Epoch [3650/5000], Loss: 5.0001
Epoch [3660/5000], Loss: 4.9958
Epoch [3670/5000], Loss: 4.9831
Epoch [3680/5000], Loss: 5.0028
Epoch [3690/5000], Loss: 5.0039
Epoch [3700/5000], Loss: 5.0080
Epoch [3710/5000], Loss: 4.9949
Epoch  3714: reducing learning rate of group 0 to 7.6234e-04.
Epoch [3720/5000], Loss: 5.0101
Epoch [3730/5000], Loss: 4.9957
Epoch [3740/5000], Loss: 5.0024
Epoch [3750/5000], Loss: 4.9945
Epoch [3760/5000], Loss: 5.0005
Epoch [3770/5000], Loss: 4.9976
Epoch [3780/5000], Loss: 4.9876
Epoch [3790/5000], Loss: 5.0008
Epoch [3800/5000], Loss: 4.9955
Epoch [3810/5000], Loss: 5.0071
Epoch  3815: reducing learning rate of group 0 to 7.5472e-04.
Epoch [3820/5000], Loss: 4.9953
Epoch [3830/5000], Loss: 5.0085
Epoch [3840/5000], Loss: 5.0029
Epoch [3850/5000], Loss: 4.9926
Epoch [3860/5000], Loss: 5.0061
Epoch [3870/5000], Loss: 5.0001
Epoch [3880/5000], Loss: 5.0056
Epoch [3890/5000], Loss: 4.9887
Epoch [3900/5000], Loss: 4.9886
Epoch [3910/5000], Loss: 4.9999
Epoch  3916: reducing learning rate of group 0 to 7.4717e-04.
Epoch [3920/5000], Loss: 5.0007
Epoch [3930/5000], Loss: 4.9957
Epoch [3940/5000], Loss: 4.9970
Epoch [3950/5000], Loss: 5.0055
Epoch [3960/5000], Loss: 5.0089
Epoch [3970/5000], Loss: 5.0000
Epoch [3980/5000], Loss: 4.9960
Epoch [3990/5000], Loss: 5.0072
Epoch [4000/5000], Loss: 4.9998
Epoch [4010/5000], Loss: 5.0018
Epoch  4017: reducing learning rate of group 0 to 7.3970e-04.
Epoch [4020/5000], Loss: 5.0182
Epoch [4030/5000], Loss: 5.0021
Epoch [4040/5000], Loss: 5.0009
Epoch [4050/5000], Loss: 5.0001
Epoch [4060/5000], Loss: 5.0030
Epoch [4070/5000], Loss: 5.0048
Epoch [4080/5000], Loss: 4.9975
Epoch [4090/5000], Loss: 4.9895
Epoch [4100/5000], Loss: 4.9894
Epoch [4110/5000], Loss: 4.9847
Epoch  4118: reducing learning rate of group 0 to 7.3230e-04.
Epoch [4120/5000], Loss: 4.9980
Epoch [4130/5000], Loss: 5.0024
Epoch [4140/5000], Loss: 5.0138
Epoch [4150/5000], Loss: 4.9974
Epoch [4160/5000], Loss: 5.0162
Epoch [4170/5000], Loss: 4.9947
Epoch [4180/5000], Loss: 4.9965
Epoch [4190/5000], Loss: 4.9970
Epoch [4200/5000], Loss: 5.0018
Epoch [4210/5000], Loss: 4.9948
Epoch  4219: reducing learning rate of group 0 to 7.2498e-04.
Epoch [4220/5000], Loss: 4.9922
Epoch [4230/5000], Loss: 4.9792
Epoch [4240/5000], Loss: 5.0003
Epoch [4250/5000], Loss: 4.9997
Epoch [4260/5000], Loss: 4.9887
Epoch [4270/5000], Loss: 4.9937
Epoch [4280/5000], Loss: 5.0025
Epoch [4290/5000], Loss: 4.9966
Epoch [4300/5000], Loss: 4.9962
Epoch [4310/5000], Loss: 4.9951
Epoch [4320/5000], Loss: 5.0100
Epoch  4320: reducing learning rate of group 0 to 7.1773e-04.
Epoch [4330/5000], Loss: 4.9956
Epoch [4340/5000], Loss: 4.9985
Epoch [4350/5000], Loss: 4.9971
Epoch [4360/5000], Loss: 5.0052
Epoch [4370/5000], Loss: 5.0011
Epoch [4380/5000], Loss: 4.9904
Epoch [4390/5000], Loss: 4.9883
Epoch [4400/5000], Loss: 4.9902
Epoch [4410/5000], Loss: 4.9957
Epoch [4420/5000], Loss: 4.9926
Epoch  4421: reducing learning rate of group 0 to 7.1055e-04.
Epoch [4430/5000], Loss: 5.0083
Epoch [4440/5000], Loss: 4.9916
Epoch [4450/5000], Loss: 5.0028
Epoch [4460/5000], Loss: 5.0095
Epoch [4470/5000], Loss: 5.0038
Epoch [4480/5000], Loss: 4.9938
Epoch [4490/5000], Loss: 5.0099
Epoch [4500/5000], Loss: 4.9941
Epoch [4510/5000], Loss: 4.9976
Epoch [4520/5000], Loss: 5.0065
Epoch  4522: reducing learning rate of group 0 to 7.0345e-04.
Epoch [4530/5000], Loss: 5.0030
Epoch [4540/5000], Loss: 5.0059
Epoch [4550/5000], Loss: 5.0035
Epoch [4560/5000], Loss: 4.9924
Epoch [4570/5000], Loss: 4.9969
Epoch [4580/5000], Loss: 4.9985
Epoch [4590/5000], Loss: 4.9969
Epoch [4600/5000], Loss: 4.9995
Epoch [4610/5000], Loss: 4.9896
Epoch [4620/5000], Loss: 4.9839
Epoch  4623: reducing learning rate of group 0 to 6.9641e-04.
Epoch [4630/5000], Loss: 4.9962
Epoch [4640/5000], Loss: 4.9939
Epoch [4650/5000], Loss: 4.9988
Epoch [4660/5000], Loss: 5.0086
Epoch [4670/5000], Loss: 4.9972
Epoch [4680/5000], Loss: 4.9961
Epoch [4690/5000], Loss: 4.9908
Epoch [4700/5000], Loss: 4.9882
Epoch [4710/5000], Loss: 4.9914
Epoch [4720/5000], Loss: 4.9970
Epoch  4724: reducing learning rate of group 0 to 6.8945e-04.
Epoch [4730/5000], Loss: 4.9898
Epoch [4740/5000], Loss: 4.9904
Epoch [4750/5000], Loss: 5.0075
Epoch [4760/5000], Loss: 5.0088
Epoch [4770/5000], Loss: 4.9999
Epoch [4780/5000], Loss: 5.0030
Epoch [4790/5000], Loss: 5.0043
Epoch [4800/5000], Loss: 4.9932
Epoch [4810/5000], Loss: 4.9823
Epoch [4820/5000], Loss: 4.9914
Epoch  4825: reducing learning rate of group 0 to 6.8255e-04.
Epoch [4830/5000], Loss: 4.9988
Epoch [4840/5000], Loss: 4.9931
Epoch [4850/5000], Loss: 5.0037
Epoch [4860/5000], Loss: 5.0061
Epoch [4870/5000], Loss: 4.9816
Epoch [4880/5000], Loss: 5.0009
Epoch [4890/5000], Loss: 4.9940
Epoch [4900/5000], Loss: 5.0010
Epoch [4910/5000], Loss: 5.0050
Epoch [4920/5000], Loss: 4.9895
Epoch  4926: reducing learning rate of group 0 to 6.7573e-04.
Epoch [4930/5000], Loss: 4.9970
Epoch [4940/5000], Loss: 5.0065
Epoch [4950/5000], Loss: 5.0006
Epoch [4960/5000], Loss: 4.9909
Epoch [4970/5000], Loss: 4.9998
Epoch [4980/5000], Loss: 4.9991
Epoch [4990/5000], Loss: 4.9989
Epoch [5000/5000], Loss: 4.9915
```

In [15]:

```
#torch.save(awd_lstm.state_dict(), './AWD_LSTM_generator_V3.pth')
```

In [11]:

```
#to used the saved model
saved_model_path = "AWD_LSTM_generator_V3.pth"
awd_lstm = AWDLSTM(input_size, hidden_size, num_layers, embedding_size=embed
awd_lstm.load_state_dict(torch.load(saved_model_path))
```

Out[11]:

```
<All keys matched successfully>
```

In [21]:

```
hidden = (torch.zeros(num_layers, batch_size, hidden_size).to(device),
          torch.zeros(num_layers, batch_size, hidden_size).to(device))
```

In [ ]:

```
generated_text = generate(awd_lstm, prime_str='Th', predict_len=1000, vocab_
print(generated_text)
```