

```

# This Python 3 environment comes with many helpful analytics
libraries installed
# It is defined by the kaggle/python Docker image:
https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/"
directory
# For example, running this (by clicking run or pressing Shift+Enter)
will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 5GB to the current directory (/kaggle/working/)
that gets preserved as output when you create a version using "Save &
Run All"
# You can also write temporary files to /kaggle/temp/, but they won't
be saved outside of the current session

/kaggle/input/english-to-tamil-data/tam.txt

from keras.models import Model
from keras.layers import LSTM,Dense,Input

batch_size=64
samples=10000
latent_dim=256
epochs=100

data_path="../input/english-to-tamil-data/tam.txt"

```

Vectorize the data

```

input_texts = []
target_texts = []
input_characters = set()
target_characters = set()
with open(data_path, 'r', encoding='utf-8') as f:
    lines = f.read().split('\n')
for line in lines[: min(samples, len(lines) - 1)]:
    input_text, target_text, _ = line.split('\t')
    # We use "tab" as the "start sequence" character
    # for the targets, and "\n" as "end sequence" character.
    target_text = '\t' + target_text + '\n'

```

```

input_texts.append(input_text)
target_texts.append(target_text)
for char in input_text:
    if char not in input_characters:
        input_characters.add(char)
for char in target_text:
    if char not in target_characters:
        target_characters.add(char)

input_characters = sorted(list(input_characters)) #Sorting all
characters of English
target_characters = sorted(list(target_characters)) #Sorting all
characters of Tamil
num_encoder_tokens = len(input_characters) #No of charac's in
English
num_decoder_tokens = len(target_characters) #No of charac's in
Tamil
max_encoder_seq_length = max([len(txt) for txt in input_texts])
#Sentence with max words in English
max_decoder_seq_length = max([len(txt) for txt in target_texts])
#Sentence with max words in Tamil

print('Number of samples:', len(input_texts))
print('Number of unique input tokens:', num_encoder_tokens)
print('Number of unique output tokens:', num_decoder_tokens)
print('Max sequence length for inputs:', max_encoder_seq_length)
print('Max sequence length for outputs:', max_decoder_seq_length)

Number of samples: 201
Number of unique input tokens: 53
Number of unique output tokens: 54
Max sequence length for inputs: 94
Max sequence length for outputs: 111

print(input_characters)

[' ', '!', '"', ',', '.', ':', '0', '2', '?', 'A', 'B', 'C', 'D', 'F', 'G',
'H', 'I', 'J', 'K', 'L', 'M', 'N', 'P', 'R', 'S', 'T', 'W', 'Y', 'a',
'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o',
'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z']

print(target_characters)

['\t', '\n', ' ', '!', '(', ')', ',', '.', ':', '0', '2', '?', 'C', 'D',
'அ', 'ஆ', 'இ', 'உ', 'ஊ', 'எ', 'ஏ', 'ஓ', 'ஔ', 'க', 'ங', 'ச',
'ஜ', 'ஞ', 'ட', 'ண', 'த', 'ந', 'ன', 'ப', 'ம', 'ய', 'ர', 'ற',
'ல', 'ள', 'ழ', 'வ', 'ஷ', 'ஸ', 'ா', 'ி', 'ீ', 'ை', 'ஐ',
'ெ', 'ே', 'ை', 'ொ', 'ோ', '்']

print(input_texts)

```

['I slept.', 'Calm down.', "I'll walk.", 'Who is he?', 'Who knows?', 'She smiled.', 'Talk to me!', 'Who is she?', 'Go to sleep.', 'It may rain.', 'She bit him.', 'She hit him.', 'She is kind.', 'She is eight.', 'Where are we?', 'Keep in touch!', 'See you again.', 'Give it to her.', 'I ate too much.', "I'll see to it.", "It's up to you.", 'Leave it to me.', 'Listen to this!', "That's the way.", 'Come and see me.', "Don't lie to me.", 'He began to run.', 'He just arrived.', 'He likes to run.', 'How is your dad?', 'I want to sleep.', "I'm able to run.", 'Raise your hand.', 'What did he say?', 'When can we eat?', 'Come and help us.', 'He is still here.', 'I have to go now.', 'I know that much.', 'I made a mistake.', 'I walk to school.', "That's our house.", 'Those are my CDs.', 'Walk ahead of me.', "We'll follow you.", 'Beware of the dog!', 'He came back soon.', 'He has three sons.', 'I know how to ski.', 'I know what to do.', "I'm kind of happy.", 'Keep to the right.', 'She began to sing.', 'She decided to go.', 'Do I have to study?', 'He is sure to come.', 'I had to walk home.', 'I have to dress up.', 'I told him to come.', "I'm short of money.", 'May I speak to you?', 'She gave it to him.', 'She is kind to him.', 'She sat next to me.', 'Shut up and listen!', 'Tell me what to do.', 'Tom runs very fast.', 'We ran out of food.', 'We started to walk.', 'When does it begin?', 'Are you ready to go?', 'Do you have any gum?', 'Does she play piano?', "Don't listen to her.", 'Go and wake Mary up.', 'He seems to know us.', 'I am engaged to her.', 'I have to leave now.', 'I want to go abroad.', "I'm glad to see you.", "I'm proud of my son.", "I'm taller than you.", "I'm trying to sleep.", "It's free of charge.", "It's time to get up.", 'Nobody speaks to me.', 'Roll the ball to me.', 'She boiled the eggs.', 'She danced with him.', 'She gave him a book.', 'She has 2,000 books.', 'This apple is sweet.', 'We swam in the lake.', 'Come home before six.', 'Go and see who it is.', 'I am afraid of bears.', 'I expect him to come.', "It's a piece of cake.", 'The boy began to cry.', 'You keep out of this.', 'All of us were silent.', 'Be kind to old people.', 'Beware of pickpockets.', "Don't drink and drive.", 'He can read and write.', 'He got a lot of money.', 'He has a lot of money.', 'He is afraid of death.', 'He let go of the rope.', 'I am tired of my work.', 'I got out of the taxi.', 'None of your business.', 'They made fun of Mary.', 'Tom and I are friends.', 'When is your birthday?', 'All of them went there.', 'Can you ride a bicycle?', 'Do you want to be rich?', 'He is afraid of snakes.', 'He is fond of swimming.', 'He went in place of me.', "He's afraid of the sea.", "I'll leave that to you.", 'It seems she hates you.', 'She got engaged to him.', 'She got married to him.', 'She stood close to him.', "They're about to leave.", 'This CD belongs to her.', 'We ran after the thief.', 'What do you plan to do?', 'A square has four sides.', 'Charge it to my account.', 'Did they write a letter?', 'He asked us to help him.', 'He is known to everyone.', 'He objected to our plan.', 'I just want you to come.', 'I want something to eat.', 'Is he a friend of yours?', 'The news quickly spread.', "I can't find it anywhere.", "I thought you'd be angry.", 'Please sit here and wait.', 'She went out of the room.', 'Speak slowly and clearly.', 'The sky is full of stars.',

'Come and see me right now.', 'Do you have a lot of pens?', 'Go and sit by your father.', 'He bought a pair of shoes.', 'I live on the bottom floor.', 'I sat between Tom and John.', 'She wore a beautiful dress.', 'When did you come to Japan?', "Most people think I'm crazy.", 'I suppose Tom is still alive.', 'She asked him for some money.', 'Tom told me about it himself.', 'Do you know when he will come?', 'He painted a picture of a dog.', 'I arrived ahead of the others.', 'I know every inch of the town.', "I'm not sharing this with Tom.", 'She is not afraid of anything.', 'The price of eggs is going up.', 'What is the price of this cap?', 'Which of them is your brother?', 'He arrived after the bell rang.', 'He was not aware of the danger.', 'My throat hurts when I swallow.', 'The school looks like a prison.', "I'm not sure how to answer this.", "There's no easy way out of here.", 'Three vicious dogs attacked Tom.', 'Tom was in Australia a year ago.', 'When did the wedding take place?', 'Where do you keep your passport?', "Because he's sick, he can't come.", 'Friendship requires mutual trust.', "He put the ring on Mary's finger.", 'She glanced through the magazine.', 'Tom has been crying all afternoon.', 'Tom has been in contact with Mary.', 'I want to be a pilot in the future.', 'If Tom ran away, where could he go?', 'I had my pocket picked on the train.', 'He told her something and she smiled.', "I don't like to go out when it's dark.", 'When he spoke, everyone became silent.', 'Tom drank with us until after midnight.', 'She has never been in a car driven by him.', 'Tom goes to church with Mary every Sunday.', "I don't think people use that word anymore.", 'My younger sister got married in her teens.', 'I wonder why Tom suggested we do that together.', "Tom says he doesn't think he can do that by himself.", "People who live in glass houses shouldn't throw stones.", "It's been a long time since I've heard anyone use that word.", 'If you want your workers to be happy, you need to pay them a decent wage.', "It's my fault that the cake was burned. I was talking on the phone and didn't notice the time."]

```
print(target_texts)
```

['\t நான் தூங்கினேன்.\n', '\t அமைதியாக இருங்கள்\n', '\t நான் நடப்பேன்.\n', '\t அவன் யார்?\n', '\t யாருக்குத் தெரியும்?\n', '\t அவள் சிரித்தாள்\n', '\t என்னிடம் பேசு\n', '\t அவன் யார்?\n', '\t போய் தூங்கு\n', '\t மழை பெய்யலாம்\n', '\t அவள் அவனைக் கடித்தாள்\n', '\t அவள் அவனைக் அடித்தாள்\n', '\t அவள் அன்பானவள்\n', '\t அவளுக்கு எட்டு வயது\n', '\t நாம் எங்கே இருக்கிறோம்?\n', '\t தொடர்பில் இரு\n', '\t மறுபடியும் சந்திப்போம்\n', '\t அவளிடம் கொடு\n', '\t நான் நிறைய சாப்பிட்டேன்\n', '\t அதை நான் பார்க்கிறேன்\n', '\t உன் கையில்ல்தான் இருக்கிறது\n', '\t என்னிடம் விட்டுவிடு\n', '\t இதைக் கேள்\n', '\t அந்த பக்கம்தான் வழி\n', '\t என்னை வந்து பார்\n', '\t என்னிடம் பொய் சொல்லாதே\n', '\t அவன் ஓட ஆரம்பித்தான்\n', '\t அவன் இப்பொழுதுதான் வந்தான்\n', '\t அவன் ஓட விருப்பப் படுகிறான்\n', '\t தங்களுடைய

தந்தையார் எப்படி இருக்கிறார்கள்?\n', '\t நான் தூங்க விரும்புகிறேன்\n', '\t என்னால் ஓட முடிகிறது\n', '\t கையைத் தூக்கு\n', '\t அவன் என்ன சொன்னான்?\n', '\t எப்பொழுது நம்மால் சாப்பிட முடியும்\n', '\t வா எங்களுக்கு உதவி செய்\n', '\t அவன் இன்னும் இருக்கிறான்\n', '\t நான் இப்பொழுது போக வேண்டும்\n', '\t எனக்கு அவ்வளவு தெரியும்.\n', '\t நான் ஒரு தவறு செய்தேன்?\n', '\t நான் பள்ளிக்கு நடந்து செல்கிறேன்\n', '\t அது எங்களுடைய வீடு\n', '\t அவைகள் என்னுடைய CD கள்\n', '\t எனக்கு முன்னால் நட\n', '\t நாங்கள் உன்னைத் பின்பற்றுவோம் (அ) தொடர்வோம்.\n', '\t நாய் ஐக்கிரதை!\n', '\t அவன் சீக்கிரம் திரும்பி வந்தான்\n', '\t அவருக்கு மூன்று மகன்கள்\n', '\t எப்படி பனியில் சறுக்கி விளையாடுவது என்பது எனக்கு தெரியும்\n', '\t என்ன செய்வது என்பது எனக்குத் தெரியும்\n', '\t நான் ஒரு விதமான மகிழ்ச்சியிலிருக்கிறேன்\n', '\t வலது பக்கத்தை கடைப் பிடி\n', '\t அவள் பாட ஆரம்பித்தாள்\n', '\t அவள் போகத் தீர்மானித்தாள்\n', '\t நான் படிக்க வேண்டுமா?\n', '\t அவன் வருவது நிச்சயம்\n', '\t நான் வீட்டிற்கு நடக்க வேண்டியிருந்தது\n', '\t நான் ஆடை அணிய வேண்டும்\n', '\t நான் அவனை வரச் சொன்னேன்\n', '\t என்னிடம் பணம் குறைவாக இருக்கிறது\n', '\t நான் உன்னிடம் பேசலாமா?\n', '\t அவள் இதை அவனுக்குக் கொடுத்தாள்\n', '\t அவள் அவனிடம் அன்பாக இருக்கிறாள்\n', '\t அவள் எனக்கு அருகில் அமர்ந்தாள்\n', '\t வாயை மூடி கவனி\n', '\t நான் என்ன செய்ய வேண்டும் என்று சொல்\n', '\t டாம் ரொம்ப வேகமாக ஓடுகிறான்\n', '\t எங்களுக்கு உணவு தட்டுப்பாடு ஏற்பட்டது\n', '\t நாங்கள் நடக்க ஆரம்பித்தோம்\n', '\t இது எப்பொழுது ஆரம்பிக்கிறது?\n', '\t நீங்கள் போகத் தயாராக இருக்கிறீர்களா?\n', '\t உன்னிடம் ஏதாவது பசை இருக்கிறதா?\n', '\t அவள் பியானோ வாசிக்கிறாளோ?\n', '\t அவள் சொல்வதைக் கேட்காதீர்\n', '\t போய் மேரியை எழுப்பு\n', '\t அவனுக்கு நம்மைப் தெரியும் என்று தோன்றுகிறது\n', '\t எனக்கு அவளோடு நிச்சயமாகியிருக்கு\n', '\t நான் இப்பொழுது கிளம்ப வேண்டும்\n', '\t நான் வெளி நாட்டிற்குச் செல்ல விரும்புகிறேன்\n', '\t உன்னைப் பார்ப்பதில் நான் மகிழ்ச்சி அடைகிறேன்\n', '\t என் மகனைப் பற்றி பெருமைப் படுகிறேன்\n', '\t நான் உன்னை விட உயரமாக இருக்கிறேன்\n', '\t நான் தூங்குவதற்கு முயற்சி செய்து கொண்டிருக்கிறேன்\n', '\t இதற்கு கட்டணமில்லை\n', '\t தூக்கத்திலிருந்து எழுவதற்கான நேரம் இது\n', '\t என் கூட யாரும் பேசுவதில்லை\n', '\t பந்தை என்னிடம் உருட்டி விடு\n', '\t அவள் முட்டைகளை வேக வைத்தாள்\n', '\t அவள் அவளோடு நடனம் ஆடினாள்\n', '\t அவள் அவனுக்கு ஒரு புத்தகத்தைக் கொடுத்தாள்\n', '\t அவளிடம் 2000 புத்தகங்கள் உள்ளன\n', '\t இந்த ஆப்பிள் இனிப்பாக இருக்கிறது\n', '\t அவன் ஏரியில் நீச்சலடித்தான்\n', '\t ஆறு மணிக்கு முன்பு வீட் டிற்கு வா\n', '\t போய் யார் என்று பார்\n', '\t எனக்குக் கரடிகளைக் கண்டால் பயம்\n', '\t அவன் வருவான் என எதிர் பார்க்கிறேன்\n', '\t இது ஒரு கேக்கின் துண்டு\n', '\t அந்த பையன் அடி ஆரம்பித்தான்\n', '\t நீ இதில் தலையிடாதே\n', '\t நாங்கள் அனைவரும் அமைதியாக இருந்தோம்\n', '\t வயோதிகர்களிடம் அன்பாக

இரு\ந', '\t ஜேப்படிகாரர்களிடம் ஜாக்கிரதையாக இருக்கவும்\ந', '\t குடித்துவிட்டு வண்டி ஓட்டாதே\ந', '\t அவனுக்கு எழுதப் படிக்கத் தெரியும்\ந', '\t அவனுக்கு நிறைய பணம் கிடைத்தது\ந', '\t அவனிடம் நிறைய பணமிருக்கிறது\ந', '\t அவனுக்கு இறந்து போவதென்றால் பயம்\ந', '\t அவன் கயிற்றை விட்டான்\ந', '\t நான் வேலை பளுவினால் சோர்வாகயிருக்கிறேன்\ந', '\t நான் டாக்ஸியிலிருந்து இறங்கினேன்\ந', '\t இது உங்களுக்கு சம்பந்தமில்லாத விஷயம்\ந', '\t அவர்கள் மேரியை கிண்டலடித்தார்கள்\ந', '\t டாமும் நானும் நண்பர்கள்\ந', '\t உங்கள் பிறந்த நாள் எப்போது?\ந', '\t அவர்கள் எல்லோரும் அங்கே சென்றார்கள்\ந', '\t உங்களுக்கு சைக்கிள் ஓட்டத் தெரியுமா?\ந', '\t நீ பணக்காரராக விருப்பமா?\ந', '\t அவர்களுக்கு பாம்புகள் என்றால் பயம்\ந', '\t அவனுக்கு நீச்சல் மீது பற்று உண்டு\ந', '\t அவன் எனக்குப் பதிலாக சேன்றான்\ந', '\t அவனுக்குக் கடல் என்றால் பயம்\ந', '\t நான் அதை உன்னிடம் விட்டு விடுகிறேன்\ந', '\t அவள் உன்னை வெறுக்கிற மாதிரி தெரிகிறது\ந', '\t அவள் அவனுக்கு நிச்சயிக்கப் பட்டாள்\ந', '\t அவள் அவனுக்கு திருமணம் செய்து வைக்கப் பட்டாள்\ந', '\t அவனுக்கு நெருக்கமாக நின்றாள்\ந', '\t அவர்கள் கிளம்ப இருக்கிறார்கள்\ந', '\t இந்த சீடி அவளுக்குச் சொந்தமானது\ந', '\t நாங்கள் திருடனுக்குப் பின்னால் ஓடினோம்\ந', '\t நீ என்ன செய்யத் திட்டமிட்டிருக்கிறாய்\ந', '\t ஒரு சதுரத்திற்கு நான்கு பக்கங்கள் உள்ளன\ந', '\t என்னுடைய கணக்கிற்கு மாற்று\ந', '\t அவர்கள் கடிதம் எழுதினார்களா?\ந', '\t எங்களை உதவி செய்யும்படி கேட்டான்\ந', '\t அவன் ஒவ்வொருவருக்கும் அறிமுகமானவன்\ந', '\t எங்களுடைய திட்டத்திற்கு எதிர்ப்புத் தெரிவித்தான்\ந', '\t நீ வர வேண்டுமென விரும்புகிறேன்\ந', '\t எனக்கு சாப்பிட ஏதாவது வேண்டும்\ந', '\t அவர் உங்களுடைய நண்பரா?\ந', '\t செய்தி வேகமாக பரவியது\ந', '\t இது எங்கே இருக்கு என்று என்னால் கண்டுபிடிக்க முடியவில்லை.\ந', '\t நீ கோபமாக இருப்பாய் என்று எண்ணினேன்.\ந', '\t இங்கே அமருங்கள், தயவு செய்து காத்திருங்கள்\ந', '\t அவள் அறையை விட்டு வெளியே சென்றாள்\ந', '\t மெதுவாகவும் தெளிவாகவும் பேசுங்கள்\ந', '\t வானம் முழுவதும் நட்சத்திரங்கள் இருக்கின்றன\ந', '\t உடனே வந்து என்னைப் பார்க்கவும்\ந', '\t உன்னிடம் நிறைய பேனாக்கள் இருக்கின்றனவா?\ந', '\t போய் உன் தந்தையருகில் அமரவும்\ந', '\t நான் ஒரு ஜோடி காலணிகளை வாங்கினேன்\ந', '\t நான் கீழ் தளத்தில் வசிக்கிறேன்\ந', '\t டாமுக்கும் ஜானுக்கும் இடையில் அமர்ந்தேன்\ந', '\t அவள் அழகான ஆடை அணிந்திருந்தாள்\ந', '\t நீ எப்பொழுது ஜப்பான் வந்தாய்?\ந', '\t நிறைய மக்கள் நான் பைத்தியம் என்று எண்ணுகிறார்கள்\ந', '\t டாம் இன்னும் உயிருடனிருக்கிறான் என்று எண்ணுகிறேன்.\ந', '\t அவனைக் கொஞ்சம் பணம் கேட்டாள்\ந', '\t டாம் அதைப் பற்றி அவனே என்னிடம் சொன்னான்.\ந', '\t அவன் எப்ப வருவான் என்று உனக்குத் தெரியுமா\ந', '\t ஒரு நாயின் படத்தை வரைந்தான்\ந', '\t மற்றவர்களுக்கு முன்னே நான் வந்தேன்\ந', '\t இந்த ஊரின் ஒவ்வொரு அங்குலமும் எனக்குத் தெரியும்\ந', '\t டாமிடம் நான் இதை பகிர்ந்துகொண்டு இருக்கவில்லை.\ந', '\t அவள் எதற்கும் பயப்படுவதில்லை\ந', '\t முட்டைகளின் விலை அதிகரித்துக்

கொண்டிருக்கிறது\n', '\t இந்த தொப்பியின் விலை என்ன?\n', '\t இவர்களில் யார் உன்னுடைய சகோதரர்\n', '\t மணி ஒலித்தப் பிறகு அவன் வந்தான்\n', '\t அவன் அபாயத்தை அறிந்திருக்க வில்லை\n', '\t சின்ன கேக்குத் துண்டு அவள் தொண்டையில் சிக்கிக் கொண்டது\n', '\t இந்த பள்ளி கூடம் ஒரு சிறைச்சாலையைப் போல இருக்கிறது\n', '\t எப்படி பதில் சொல்வது என்பதில் நான் உறுதியாக இல்லை.\n', '\t இங்கிருந்து வெளியே செல்ல சுலபமான வழியில்லை.\n', '\t மூன்று மோசமான நாய்கள் டாமை தாக்கின\n', '\t டாம் ஒரு வருடத்திற்கு முன்னால் ஆஸ்திரேலியாவில் இருந்தார்\n', '\t கல்யாணம் எப்பொழுது நடைப் பெற்றது\n', '\t நீ பாஸ்போர்ட்டை எங்கே வைத்திருக்கிறாய்?\n', '\t அவனுக்கு உடல் நிலை சரியில்லாததனால் அவனால் வர இயலாது\n', '\t நட்புக்குத் தேவை பரஸ்பர நம்பிக்கை\n', '\t அவன் மேரியின் விரலில் மோதிரத்தை அணிவித்தான்\n', '\t அவள் பத்திரிக்கையை மேலோட்டமாகப் பார்த்தாள்\n', '\t டாம் மதியம் முடிவதும் அழுதுகொண்டேயிருக்கிறான்.\n', '\t டாம் மேரியுடன் தொடர்பிலிருந்துருக்கிறான் அல்லது டாம் மேரியுடன் தொடர்பிலிருக்கிறான்.\n', '\t நான் எதிர் காலத்தில் ஒரு விமானியாக விரும்புகிறேன்\n', '\t டாம் ஓடிவிட்டால், அவரால் எங்கு செல்ல முடியும்?\n', '\t இரயிலில் என்னிடம் ஜேப்படி அடிக்கப் பட்டிருந்தது\n', '\t அவன் அவளிடம் ஏதோ சொன்னான் மற்றும் அவள் சிரித்தாள்.\n', '\t இருட்டாக இருக்கும் பொழுது நான் வெளியே போக விரும்புவதில்லை.\n', '\t அவன் பேசியப் பொழுது எல்லோரும் அமைதி காத்தார்கள்\n', '\t டாம் நள்ளிரவு வரை எங்களுடன் குடித்தார்.\n', '\t அவன் ஒட்டிணக் காரில் அவள் எப்பொழுதும் இருந்ததில்லை\n', '\t டாம் மேரியுடன் ஒவ்வொரு ஞாயிற்றுக் கிழமையும் தேவாலயத்துக்குச் செல்கிறான்.\n', '\t மக்கள் அந்த வார்த்தையைப் பயன் படுத்துவதாக எனக்குத் தெரியவில்லை\n', '\t என் தங்கை இள வயதிலேயே கல்யாணம் செய்து கொண்டாள்\n', '\t ஏன் டாம் நாமிருவரும் சேர்ந்து செய்வோமென்று பரிந்துரைத்தான் என்று ஆச்சரியமடைகிறேன்.\n', '\t என் ஒருவனால் மட்டுமே அதை செய்ய முடியாது என்று நினைக்கிறேன் என டாம் கூறினார்\n', '\t கண்ணாடி வீட்டில் வசிப்பவர்கள் கல்லை எறியக் கூடாது\n', '\t ஒருவர் அந்த வார்த்தையைப் பயன் படுத்துவதைக் கேட்டு ரொம்ப நாளாகிறது\n', '\t உங்களுடைய வேலையாட்கள் மகிழ்ச்சியாக இருக்க வேண்டுமென்றால் நீங்கள் கணிசமான சம்பளம் தர வேண்டும்\n', '\t என்னுடையத் தவறினால் கேக்கானதுக் கருகிப் போனது. தோலைப் பேசியில் பேசிக் கொண்டிருந்ததால் நேரத்தைக் கவனிக்க வில்லை\n']

```
input_token_index = dict([(char, i) for i, char in
enumerate(input_characters)])
```

```
target_token_index = dict([(char, i) for i, char in
enumerate(target_characters)])
```

```
print(input_token_index)
```

```
{' ': 0, '!': 1, '"': 2, ',': 3, '.': 4, '0': 5, '2': 6, '?': 7, 'A': 8, 'B': 9, 'C': 10, 'D': 11, 'F': 12, 'G': 13, 'H': 14, 'I': 15, 'J':
```



```
16, 'K': 17, 'L': 18, 'M': 19, 'N': 20, 'P': 21, 'R': 22, 'S': 23,
'T': 24, 'W': 25, 'Y': 26, 'a': 27, 'b': 28, 'c': 29, 'd': 30, 'e':
31, 'f': 32, 'g': 33, 'h': 34, 'i': 35, 'j': 36, 'k': 37, 'l': 38,
'm': 39, 'n': 40, 'o': 41, 'p': 42, 'q': 43, 'r': 44, 's': 45, 't':
46, 'u': 47, 'v': 48, 'w': 49, 'x': 50, 'y': 51, 'z': 52}
```

```
print(target_token_index)
```

```
{'\t': 0, '\n': 1, ' ': 2, '!': 3, '(': 4, ')': 5, ',': 6, '.': 7,
'0': 8, '2': 9, '?': 10, 'C': 11, 'D': 12, 'அ': 13, 'ஆ': 14, 'இ':
15, 'உ': 16, 'ஊ': 17, 'எ': 18, 'ஏ': 19, 'ஓ': 20, 'ஔ': 21, 'க':
22, 'ங': 23, 'ச': 24, 'ஜ': 25, 'ஞ': 26, 'ட': 27, 'ண': 28, 'த':
29, 'ந': 30, 'ன': 31, 'ப': 32, 'ம': 33, 'ய': 34, 'ர': 35, 'ற': 36,
'ல': 37, 'ள': 38, 'ழ': 39, 'வ': 40, 'ஷ': 41, 'ஸ': 42, 'ா': 43,
'ி': 44, 'ீ': 45, 'ு': 46, 'ஓ': 47, 'ெ': 48, 'ே': 49, 'ை': 50,
'ொ': 51, 'ோ': 52, 'ஃ': 53}
```

Creating the 3 dimensional array

(No of input text, Max sequence's length, No of lan's character)

```
encoder_input_data = np.zeros((len(input_texts),
max_encoder_seq_length, num_encoder_tokens), dtype='float32')

decoder_input_data = np.zeros((len(input_texts),
max_decoder_seq_length, num_decoder_tokens), dtype='float32')

decoder_target_data = np.zeros((len(input_texts),
max_decoder_seq_length, num_decoder_tokens), dtype='float32')

for i, (input_text, target_text) in enumerate(zip(input_texts,
target_texts)):
    for t, char in enumerate(input_text):
        encoder_input_data[i, t, input_token_index[char]] = 1.
    encoder_input_data[i, t + 1:, input_token_index[' ']] = 1.
    for t, char in enumerate(target_text):
        # decoder_target_data is ahead of decoder_input_data by one
timestep
        decoder_input_data[i, t, target_token_index[char]] = 1.
        if t > 0:
            # decoder_target_data will be ahead by one timestep
            # and will not include the start character.
            decoder_target_data[i, t - 1, target_token_index[char]] =
1.
        decoder_input_data[i, t + 1:, target_token_index[' ']] = 1.
        decoder_target_data[i, t:, target_token_index[' ']] = 1.
# Define an input sequence and process it.
    encoder_inputs = Input(shape=(None, num_encoder_tokens))
    encoder = LSTM(latent_dim, return_state=True)
```



```

encoder_outputs, state_h, state_c = encoder(encoder_inputs)
# We discard `encoder_outputs` and only keep the states.
encoder_states = [state_h, state_c]

# Set up the decoder, using `encoder_states` as initial state.
decoder_inputs = Input(shape=(None, num_decoder_tokens))
# We set up our decoder to return full output sequences,
# and to return internal states as well. We don't use the
# return states in the training model, but we will use them in
inference.
decoder_lstm = LSTM(latent_dim, return_sequences=True,
return_state=True)
decoder_outputs, _, _ = decoder_lstm(decoder_inputs,
initial_state=encoder_states)
decoder_dense = Dense(num_decoder_tokens, activation='softmax')
decoder_outputs = decoder_dense(decoder_outputs)

# Define the model that will turn `encoder_input_data` &
`decoder_input_data` into `decoder_target_data`
model = Model([encoder_inputs, decoder_inputs], decoder_outputs)

model.compile(optimizer='rmsprop',
loss='categorical_crossentropy',metrics=['accuracy'])

model.fit([encoder_input_data, decoder_input_data],
decoder_target_data,
batch_size=batch_size,
epochs=epochs,
validation_split=0.2)

```

Epoch 1/100

```

3/3 [=====] - 3s 871ms/step - loss: 3.2879 -
accuracy: 0.4592 - val_loss: 3.0405 - val_accuracy: 0.5889

```

Epoch 2/100

```

3/3 [=====] - 1s 454ms/step - loss: 1.6109 -
accuracy: 0.7209 - val_loss: 2.9794 - val_accuracy: 0.5985

```

Epoch 3/100

```

3/3 [=====] - 1s 480ms/step - loss: 1.7518 -
accuracy: 0.7610 - val_loss: 1.9920 - val_accuracy: 0.5953

```

Epoch 4/100

```

3/3 [=====] - 1s 480ms/step - loss: 1.0139 -
accuracy: 0.7641 - val_loss: 1.7511 - val_accuracy: 0.5950

```

Epoch 5/100

```

3/3 [=====] - 1s 487ms/step - loss: 1.0119 -
accuracy: 0.7649 - val_loss: 2.1573 - val_accuracy: 0.5970

```

Epoch 6/100

```

3/3 [=====] - 1s 479ms/step - loss: 0.9820 -
accuracy: 0.7671 - val_loss: 1.7170 - val_accuracy: 0.5990

```

Epoch 7/100

```

3/3 [=====] - 1s 471ms/step - loss: 0.9552 -

```

accuracy: 0.7690 - val_loss: 2.0234 - val_accuracy: 0.5968
Epoch 8/100
3/3 [=====] - 1s 475ms/step - loss: 0.9392 -
accuracy: 0.7697 - val_loss: 1.8474 - val_accuracy: 0.5985
Epoch 9/100
3/3 [=====] - 1s 474ms/step - loss: 0.9176 -
accuracy: 0.7715 - val_loss: 1.8838 - val_accuracy: 0.5957
Epoch 10/100
3/3 [=====] - 1s 479ms/step - loss: 0.9514 -
accuracy: 0.7684 - val_loss: 1.6226 - val_accuracy: 0.6043
Epoch 11/100
3/3 [=====] - 1s 470ms/step - loss: 1.0090 -
accuracy: 0.7688 - val_loss: 1.9372 - val_accuracy: 0.5988
Epoch 12/100
3/3 [=====] - 1s 471ms/step - loss: 0.9167 -
accuracy: 0.7721 - val_loss: 2.0222 - val_accuracy: 0.5953
Epoch 13/100
3/3 [=====] - 1s 480ms/step - loss: 0.9056 -
accuracy: 0.7709 - val_loss: 1.8486 - val_accuracy: 0.5990
Epoch 14/100
3/3 [=====] - 1s 472ms/step - loss: 0.8966 -
accuracy: 0.7709 - val_loss: 1.8617 - val_accuracy: 0.6014
Epoch 15/100
3/3 [=====] - 2s 504ms/step - loss: 0.9012 -
accuracy: 0.7722 - val_loss: 2.4258 - val_accuracy: 0.5922
Epoch 16/100
3/3 [=====] - 1s 477ms/step - loss: 1.0030 -
accuracy: 0.7668 - val_loss: 1.8937 - val_accuracy: 0.5975
Epoch 17/100
3/3 [=====] - 1s 481ms/step - loss: 0.8864 -
accuracy: 0.7724 - val_loss: 1.5928 - val_accuracy: 0.6069
Epoch 18/100
3/3 [=====] - 1s 471ms/step - loss: 0.9023 -
accuracy: 0.7759 - val_loss: 2.0295 - val_accuracy: 0.5979
Epoch 19/100
3/3 [=====] - 2s 501ms/step - loss: 0.8804 -
accuracy: 0.7742 - val_loss: 1.6489 - val_accuracy: 0.6005
Epoch 20/100
3/3 [=====] - 2s 510ms/step - loss: 0.8975 -
accuracy: 0.7739 - val_loss: 2.4733 - val_accuracy: 0.6023
Epoch 21/100
3/3 [=====] - 1s 466ms/step - loss: 0.9317 -
accuracy: 0.7807 - val_loss: 1.7819 - val_accuracy: 0.6032
Epoch 22/100
3/3 [=====] - 1s 482ms/step - loss: 0.8648 -
accuracy: 0.7774 - val_loss: 1.8661 - val_accuracy: 0.5975
Epoch 23/100
3/3 [=====] - 1s 476ms/step - loss: 0.8547 -
accuracy: 0.7752 - val_loss: 1.6423 - val_accuracy: 0.6016

Epoch 24/100
3/3 [=====] - 1s 499ms/step - loss: 0.8585 - accuracy: 0.7756 - val_loss: 1.9152 - val_accuracy: 0.6069
Epoch 25/100
3/3 [=====] - 1s 469ms/step - loss: 0.8514 - accuracy: 0.7829 - val_loss: 1.5059 - val_accuracy: 0.6082
Epoch 26/100
3/3 [=====] - 1s 478ms/step - loss: 0.8483 - accuracy: 0.7794 - val_loss: 1.8285 - val_accuracy: 0.6058
Epoch 27/100
3/3 [=====] - 1s 463ms/step - loss: 0.8393 - accuracy: 0.7787 - val_loss: 1.4975 - val_accuracy: 0.6071
Epoch 28/100
3/3 [=====] - 1s 463ms/step - loss: 0.8435 - accuracy: 0.7779 - val_loss: 1.6931 - val_accuracy: 0.6065
Epoch 29/100
3/3 [=====] - 1s 491ms/step - loss: 0.8258 - accuracy: 0.7826 - val_loss: 1.4791 - val_accuracy: 0.6117
Epoch 30/100
3/3 [=====] - 1s 467ms/step - loss: 0.8253 - accuracy: 0.7867 - val_loss: 1.6998 - val_accuracy: 0.6038
Epoch 31/100
3/3 [=====] - 1s 475ms/step - loss: 0.8316 - accuracy: 0.7816 - val_loss: 1.4724 - val_accuracy: 0.6251
Epoch 32/100
3/3 [=====] - 1s 485ms/step - loss: 0.8539 - accuracy: 0.7880 - val_loss: 2.0964 - val_accuracy: 0.5955
Epoch 33/100
3/3 [=====] - 1s 491ms/step - loss: 0.8294 - accuracy: 0.7836 - val_loss: 1.7162 - val_accuracy: 0.6122
Epoch 34/100
3/3 [=====] - 2s 503ms/step - loss: 0.7931 - accuracy: 0.7877 - val_loss: 1.4693 - val_accuracy: 0.6111
Epoch 35/100
3/3 [=====] - 2s 508ms/step - loss: 0.8053 - accuracy: 0.7744 - val_loss: 1.6442 - val_accuracy: 0.6142
Epoch 36/100
3/3 [=====] - 1s 486ms/step - loss: 0.7799 - accuracy: 0.7916 - val_loss: 1.6196 - val_accuracy: 0.6205
Epoch 37/100
3/3 [=====] - 1s 474ms/step - loss: 0.7771 - accuracy: 0.7919 - val_loss: 1.5196 - val_accuracy: 0.6152
Epoch 38/100
3/3 [=====] - 1s 480ms/step - loss: 0.7840 - accuracy: 0.7901 - val_loss: 1.6892 - val_accuracy: 0.6023
Epoch 39/100
3/3 [=====] - 1s 499ms/step - loss: 0.7771 - accuracy: 0.7849 - val_loss: 1.4263 - val_accuracy: 0.6287
Epoch 40/100

3/3 [=====] - 2s 500ms/step - loss: 0.7699 - accuracy: 0.7953 - val_loss: 1.4886 - val_accuracy: 0.6245
Epoch 41/100
3/3 [=====] - 1s 481ms/step - loss: 0.7529 - accuracy: 0.7984 - val_loss: 1.4328 - val_accuracy: 0.6276
Epoch 42/100
3/3 [=====] - 2s 505ms/step - loss: 0.7600 - accuracy: 0.7904 - val_loss: 1.8694 - val_accuracy: 0.6131
Epoch 43/100
3/3 [=====] - 1s 475ms/step - loss: 0.8244 - accuracy: 0.7939 - val_loss: 1.4793 - val_accuracy: 0.6348
Epoch 44/100
3/3 [=====] - 2s 505ms/step - loss: 0.8400 - accuracy: 0.7948 - val_loss: 1.4559 - val_accuracy: 0.6174
Epoch 45/100
3/3 [=====] - 1s 476ms/step - loss: 0.7532 - accuracy: 0.7958 - val_loss: 1.4171 - val_accuracy: 0.6298
Epoch 46/100
3/3 [=====] - 1s 472ms/step - loss: 0.7421 - accuracy: 0.8006 - val_loss: 1.3969 - val_accuracy: 0.6416
Epoch 47/100
3/3 [=====] - 1s 477ms/step - loss: 0.7355 - accuracy: 0.8032 - val_loss: 1.4070 - val_accuracy: 0.6407
Epoch 48/100
3/3 [=====] - 1s 467ms/step - loss: 0.7276 - accuracy: 0.8048 - val_loss: 1.3871 - val_accuracy: 0.6425
Epoch 49/100
3/3 [=====] - 1s 490ms/step - loss: 0.7230 - accuracy: 0.8066 - val_loss: 1.4109 - val_accuracy: 0.6396
Epoch 50/100
3/3 [=====] - 1s 463ms/step - loss: 0.7213 - accuracy: 0.8042 - val_loss: 1.3686 - val_accuracy: 0.6405
Epoch 51/100
3/3 [=====] - 1s 490ms/step - loss: 0.7139 - accuracy: 0.8068 - val_loss: 1.3549 - val_accuracy: 0.6443
Epoch 52/100
3/3 [=====] - 1s 477ms/step - loss: 0.7069 - accuracy: 0.8113 - val_loss: 1.3982 - val_accuracy: 0.6460
Epoch 53/100
3/3 [=====] - 1s 461ms/step - loss: 0.7022 - accuracy: 0.8130 - val_loss: 1.3412 - val_accuracy: 0.6454
Epoch 54/100
3/3 [=====] - 1s 482ms/step - loss: 0.7000 - accuracy: 0.8118 - val_loss: 1.6458 - val_accuracy: 0.6265
Epoch 55/100
3/3 [=====] - 1s 475ms/step - loss: 0.7086 - accuracy: 0.8054 - val_loss: 1.3136 - val_accuracy: 0.6471
Epoch 56/100
3/3 [=====] - 1s 472ms/step - loss: 0.6808 -

accuracy: 0.8166 - val_loss: 1.2959 - val_accuracy: 0.6561
Epoch 57/100
3/3 [=====] - 1s 464ms/step - loss: 0.6779 -
accuracy: 0.8158 - val_loss: 1.3187 - val_accuracy: 0.6405
Epoch 58/100
3/3 [=====] - 1s 474ms/step - loss: 0.6775 -
accuracy: 0.8142 - val_loss: 1.3342 - val_accuracy: 0.6526

Epoch 59/100
3/3 [=====] - 1s 491ms/step - loss: 0.6909 -
accuracy: 0.8166 - val_loss: 1.2799 - val_accuracy: 0.6579
Epoch 60/100
3/3 [=====] - 1s 473ms/step - loss: 0.6628 -
accuracy: 0.8213 - val_loss: 1.3671 - val_accuracy: 0.6519
Epoch 61/100
3/3 [=====] - 1s 477ms/step - loss: 0.6715 -
accuracy: 0.8188 - val_loss: 1.2842 - val_accuracy: 0.6570
Epoch 62/100
3/3 [=====] - 1s 479ms/step - loss: 0.8349 -
accuracy: 0.8075 - val_loss: 1.4403 - val_accuracy: 0.6412
Epoch 63/100
3/3 [=====] - 1s 488ms/step - loss: 0.6970 -
accuracy: 0.8150 - val_loss: 1.3956 - val_accuracy: 0.6515
Epoch 64/100
3/3 [=====] - 1s 479ms/step - loss: 0.6561 -
accuracy: 0.8244 - val_loss: 1.3476 - val_accuracy: 0.6563
Epoch 65/100
3/3 [=====] - 1s 486ms/step - loss: 0.6444 -
accuracy: 0.8261 - val_loss: 1.2794 - val_accuracy: 0.6581
Epoch 66/100
3/3 [=====] - 2s 506ms/step - loss: 0.6387 -
accuracy: 0.8279 - val_loss: 1.2650 - val_accuracy: 0.6601
Epoch 67/100
3/3 [=====] - 1s 466ms/step - loss: 0.6322 -
accuracy: 0.8280 - val_loss: 1.2471 - val_accuracy: 0.6656
Epoch 68/100
3/3 [=====] - 2s 526ms/step - loss: 0.6266 -
accuracy: 0.8301 - val_loss: 1.2404 - val_accuracy: 0.6678
Epoch 69/100
3/3 [=====] - 1s 483ms/step - loss: 0.6222 -
accuracy: 0.8309 - val_loss: 1.2430 - val_accuracy: 0.6649
Epoch 70/100
3/3 [=====] - 1s 492ms/step - loss: 0.6204 -
accuracy: 0.8305 - val_loss: 1.2813 - val_accuracy: 0.6675
Epoch 71/100
3/3 [=====] - 1s 493ms/step - loss: 0.6165 -
accuracy: 0.8291 - val_loss: 1.2247 - val_accuracy: 0.6645
Epoch 72/100
3/3 [=====] - 2s 599ms/step - loss: 0.6168 -
accuracy: 0.8285 - val_loss: 1.2951 - val_accuracy: 0.6614

Epoch 73/100
3/3 [=====] - 2s 587ms/step - loss: 0.6102 - accuracy: 0.8302 - val_loss: 1.2164 - val_accuracy: 0.6678
Epoch 74/100
3/3 [=====] - 2s 515ms/step - loss: 0.6079 - accuracy: 0.8309 - val_loss: 1.3111 - val_accuracy: 0.6667
Epoch 75/100
3/3 [=====] - 1s 480ms/step - loss: 0.6145 - accuracy: 0.8333 - val_loss: 1.1733 - val_accuracy: 0.6726
Epoch 76/100
3/3 [=====] - 1s 468ms/step - loss: 0.6016 - accuracy: 0.8345 - val_loss: 1.1982 - val_accuracy: 0.6766
Epoch 77/100
3/3 [=====] - 1s 471ms/step - loss: 0.5950 - accuracy: 0.8356 - val_loss: 1.2062 - val_accuracy: 0.6726
Epoch 78/100
3/3 [=====] - 1s 467ms/step - loss: 0.6062 - accuracy: 0.8310 - val_loss: 1.1695 - val_accuracy: 0.6768
Epoch 79/100
3/3 [=====] - 1s 488ms/step - loss: 0.5862 - accuracy: 0.8336 - val_loss: 1.1840 - val_accuracy: 0.6750
Epoch 80/100
3/3 [=====] - 1s 468ms/step - loss: 0.5859 - accuracy: 0.8381 - val_loss: 1.2059 - val_accuracy: 0.6671
Epoch 81/100
3/3 [=====] - 1s 478ms/step - loss: 0.5897 - accuracy: 0.8357 - val_loss: 1.1542 - val_accuracy: 0.6946
Epoch 82/100
3/3 [=====] - 1s 471ms/step - loss: 0.5807 - accuracy: 0.8419 - val_loss: 1.1742 - val_accuracy: 0.6794
Epoch 83/100
3/3 [=====] - 1s 468ms/step - loss: 0.5689 - accuracy: 0.8401 - val_loss: 1.1832 - val_accuracy: 0.6761
Epoch 84/100
3/3 [=====] - 1s 477ms/step - loss: 0.5767 - accuracy: 0.8382 - val_loss: 1.1633 - val_accuracy: 0.6834
Epoch 85/100
3/3 [=====] - 1s 479ms/step - loss: 0.5709 - accuracy: 0.8367 - val_loss: 1.1546 - val_accuracy: 0.6790
Epoch 86/100
3/3 [=====] - 1s 462ms/step - loss: 0.5625 - accuracy: 0.8411 - val_loss: 1.1857 - val_accuracy: 0.6781
Epoch 87/100
3/3 [=====] - 1s 468ms/step - loss: 0.5621 - accuracy: 0.8414 - val_loss: 1.1536 - val_accuracy: 0.6788
Epoch 88/100
3/3 [=====] - 1s 463ms/step - loss: 0.5598 - accuracy: 0.8430 - val_loss: 1.1385 - val_accuracy: 0.6878
Epoch 89/100

```

3/3 [=====] - 1s 460ms/step - loss: 0.5473 -
accuracy: 0.8468 - val_loss: 1.1437 - val_accuracy: 0.6897
Epoch 90/100
3/3 [=====] - 1s 485ms/step - loss: 0.5580 -
accuracy: 0.8443 - val_loss: 1.1606 - val_accuracy: 0.6858
Epoch 91/100
3/3 [=====] - 1s 467ms/step - loss: 0.5446 -
accuracy: 0.8481 - val_loss: 1.1309 - val_accuracy: 0.6889
Epoch 92/100
3/3 [=====] - 1s 459ms/step - loss: 0.5519 -
accuracy: 0.8433 - val_loss: 1.1272 - val_accuracy: 0.6926
Epoch 93/100
3/3 [=====] - 1s 472ms/step - loss: 0.5384 -
accuracy: 0.8494 - val_loss: 1.1368 - val_accuracy: 0.6897
Epoch 94/100
3/3 [=====] - 1s 462ms/step - loss: 0.5377 -
accuracy: 0.8448 - val_loss: 1.1389 - val_accuracy: 0.6889
Epoch 95/100
3/3 [=====] - 1s 489ms/step - loss: 0.5279 -
accuracy: 0.8513 - val_loss: 1.1403 - val_accuracy: 0.6889
Epoch 96/100
3/3 [=====] - 2s 532ms/step - loss: 0.5269 -
accuracy: 0.8487 - val_loss: 1.1364 - val_accuracy: 0.6928
Epoch 97/100
3/3 [=====] - 1s 476ms/step - loss: 0.5220 -
accuracy: 0.8529 - val_loss: 1.1280 - val_accuracy: 0.6904
Epoch 98/100
3/3 [=====] - 2s 507ms/step - loss: 0.5283 -
accuracy: 0.8495 - val_loss: 1.1113 - val_accuracy: 0.6981
Epoch 99/100
3/3 [=====] - 1s 471ms/step - loss: 0.5177 -
accuracy: 0.8510 - val_loss: 1.1188 - val_accuracy: 0.6926
Epoch 100/100
3/3 [=====] - 2s 513ms/step - loss: 0.5190 -
accuracy: 0.8520 - val_loss: 1.1168 - val_accuracy: 0.7020

<tensorflow.python.keras.callbacks.History at 0x7faf5028a350>

model.save('tameng.h5')

```

Sampling models

```

encoder_model = Model(encoder_inputs, encoder_states)

decoder_state_input_h = Input(shape=(latent_dim,))
decoder_state_input_c = Input(shape=(latent_dim,))
decoder_states_inputs = [decoder_state_input_h, decoder_state_input_c]
decoder_outputs, state_h, state_c = decoder_lstm(
    decoder_inputs, initial_state=decoder_states_inputs)

```



```

decoder_states = [state_h, state_c]
decoder_outputs = decoder_dense(decoder_outputs)
decoder_model = Model(
    [decoder_inputs] + decoder_states_inputs,
    [decoder_outputs] + decoder_states)

# Reverse-lookup token index to decode sequences back to something
readable.
reverse_input_char_index = dict(
    (i, char) for char, i in input_token_index.items())
reverse_target_char_index = dict(
    (i, char) for char, i in target_token_index.items())

def decode_sequence(input_seq):
    # Encode the input as state vectors.
    states_value = encoder_model.predict(input_seq)

    # Generate empty target sequence of length 1.
    target_seq = np.zeros((1, 1, num_decoder_tokens))
    # Populate the first character of target sequence with the start
character.
    target_seq[0, 0, target_token_index['\t']] = 1.

    # Sampling loop for a batch of sequences
    # (to simplify, here we assume a batch of size 1).
    stop_condition = False
    decoded_sentence = ''
    while not stop_condition:
        output_tokens, h, c = decoder_model.predict(
            [target_seq] + states_value)

        # Sample a token
        sampled_token_index = np.argmax(output_tokens[0, -1, :])
        sampled_char = reverse_target_char_index[sampled_token_index]
        decoded_sentence += sampled_char

        # Exit condition: either hit max length
        # or find stop character.
        if (sampled_char == '\n' or
            len(decoded_sentence) > max_decoder_seq_length):
            stop_condition = True

        # Update the target sequence (of length 1).
        target_seq = np.zeros((1, 1, num_decoder_tokens))
        target_seq[0, 0, sampled_token_index] = 1.

        # Update states
        states_value = [h, c]

    return decoded_sentence

```

-
Input sentence: I slept.
Decoded sentence: அவன் அவன் வருக்கிறேன்
-
Input sentence: Calm down.
Decoded sentence: அவன் அவன் வருக்கிறேன்
-
Input sentence: I'll walk.
Decoded sentence: அவன் அவன் வருக்கிறேன்
-
Input sentence: Who is he?
Decoded sentence: அவன் அவன் வருக்கிறேன்
-
Input sentence: Who knows?
Decoded sentence: அவன் அவன் வருக்கிறேன்
-
Input sentence: She smiled.
Decoded sentence: அவன் அவன் வருக்கிறேன்
-
Input sentence: Talk to me!
Decoded sentence: அவன் அவன் வருக்கிறேன்
-
Input sentence: Who is she?
Decoded sentence: அவன் அவன் வருக்கிறேன்
-
Input sentence: Go to sleep.
Decoded sentence: அவன் அவன் வருக்கிறேன்

[illegible]

```

<ipython-input-36-e17d64ded27f> in <module>
      3      # for trying out decoding.
      4      input_seq = encoder_input_data[seq_index: seq_index + 1]
----> 5      decoded_sentence = decode_sequence(input_seq)
      6      print('-')
      7      print('Input sentence:', input_texts[seq_index])

<ipython-input-33-ef90cea883ca> in decode_sequence(input_seq)
     14      while not stop_condition:
     15          output_tokens, h, c = decoder_model.predict(
--> 16          [target_seq] + states_value)
     17
     18      # Sample a token

/opt/conda/lib/python3.7/site-packages/tensorflow/python/keras/engine/
training.py in _method_wrapper(self, *args, **kwargs)
     128      raise ValueError('{} is not supported in multi-worker
mode.'.format(
     129          method.__name__))
--> 130      return method(self, *args, **kwargs)
     131
     132      return tf_decorator.make_decorator(

/opt/conda/lib/python3.7/site-packages/tensorflow/python/keras/engine/
training.py in predict(self, x, batch_size, verbose, steps, callbacks,
max_queue_size, workers, use_multiprocessing)
    1577          use_multiprocessing=use_multiprocessing,
    1578          model=self,
-> 1579          steps_per_execution=self._steps_per_execution)
    1580
    1581      # Container that configures and calls
`tf.keras.Callback`s.

/opt/conda/lib/python3.7/site-packages/tensorflow/python/keras/engine/
data_adapter.py in __init__(self, x, y, sample_weight, batch_size,
steps_per_epoch, initial_epoch, epochs, shuffle, class_weight,
max_queue_size, workers, use_multiprocessing, model,
steps_per_execution)
    1115          use_multiprocessing=use_multiprocessing,
    1116          distribution_strategy=ds_context.get_strategy(),
-> 1117          model=model)
    1118
    1119      strategy = ds_context.get_strategy()

/opt/conda/lib/python3.7/site-packages/tensorflow/python/keras/engine/
data_adapter.py in __init__(self, x, y, sample_weights,
sample_weight_modes, batch_size, epochs, steps, shuffle, **kwargs)
    362      indices_dataset =
indices_dataset.flat_map(slice_batch_indices)
    363

```

```

--> 364     dataset = self.slice_inputs(indices_dataset, inputs)
365
366     if shuffle == "batch":

/opt/conda/lib/python3.7/site-packages/tensorflow/python/keras/engine/
data_adapter.py in slice_inputs(self, indices_dataset, inputs)
395
396     dataset = dataset.map(
--> 397         grab_batch, num_parallel_calls=dataset_ops.AUTOTUNE)
398
399     # Default optimizations are disabled to avoid the overhead
of (unnecessary)

/opt/conda/lib/python3.7/site-packages/tensorflow/python/data/ops/
dataset_ops.py in map(self, map_func, num_parallel_calls,
deterministic)
1700         num_parallel_calls,
1701         deterministic,
-> 1702         preserve_cardinality=True)
1703
1704     def flat_map(self, map_func):

/opt/conda/lib/python3.7/site-packages/tensorflow/python/data/ops/
dataset_ops.py in __init__(self, input_dataset, map_func,
num_parallel_calls, deterministic, use_inter_op_parallelism,
preserve_cardinality, use_legacy_function)
4082         self._transformation_name(),
4083         dataset=input_dataset,
-> 4084         use_legacy_function=use_legacy_function)
4085     if deterministic is None:
4086         self._deterministic = "default"

/opt/conda/lib/python3.7/site-packages/tensorflow/python/data/ops/
dataset_ops.py in __init__(self, func, transformation_name, dataset,
input_classes, input_shapes, input_types, input_structure,
add_to_graph, use_legacy_function, defun_kwargs)
3369         with tracking.resource_tracker_scope(resource_tracker):
3370             # TODO(b/141462134): Switch to using garbage
collection.
-> 3371         self._function = wrapper_fn.get_concrete_function()
3372         if add_to_graph:
3373             self._function.add_to_graph(ops.get_default_graph())

/opt/conda/lib/python3.7/site-packages/tensorflow/python/eager/function.
py in get_concrete_function(self, *args, **kwargs)
2937     """
2938     graph_function =
self._get_concrete_function_garbage_collected(
-> 2939         *args, **kwargs)
2940     graph_function.garbage_collector.release() # pylint:

```

```

disable=protected-access
2941     return graph_function

/opt/conda/lib/python3.7/site-packages/tensorflow/python/eager/function.py in _get_concrete_function_garbage_collected(self, *args,
**kwargs)
2904         args, kwargs = None, None
2905         with self._lock:
-> 2906             graph_function, args, kwargs =
self._maybe_define_function(args, kwargs)
2907             seen_names = set()
2908             captured = object_identity.ObjectIdentitySet(

/opt/conda/lib/python3.7/site-packages/tensorflow/python/eager/function.py in _maybe_define_function(self, args, kwargs)
3211
3212         self._function_cache.missed.add(call_context_key)
-> 3213         graph_function = self._create_graph_function(args,
kwargs)
3214         self._function_cache.primary[cache_key] = graph_function
3215         return graph_function, args, kwargs

/opt/conda/lib/python3.7/site-packages/tensorflow/python/eager/function.py in _create_graph_function(self, args, kwargs,
override_flat_arg_shapes)
3073             arg_names=arg_names,
3074             override_flat_arg_shapes=override_flat_arg_shapes,
-> 3075             capture_by_value=self._capture_by_value),
3076             self._function_attributes,
3077             function_spec=self.function_spec,

/opt/conda/lib/python3.7/site-packages/tensorflow/python/framework/func_graph.py in func_graph_from_py_func(name, python_func, args,
kwargs, signature, func_graph, autograph, autograph_options,
add_control_dependencies, arg_names, op_return_value, collections,
capture_by_value, override_flat_arg_shapes)
901         kwarg_shapes = None
902         func_args = _get_defun_inputs_from_args(
--> 903             args, arg_names, flat_shapes=arg_shapes)
904         func_kwargs = _get_defun_inputs_from_kwargs(
905             kwargs, flat_shapes=kwarg_shapes)

/opt/conda/lib/python3.7/site-packages/tensorflow/python/framework/func_graph.py in _get_defun_inputs_from_args(args, names, flat_shapes)
1137     """Maps Python function positional args to graph-
construction inputs."""
1138     return _get_defun_inputs(
-> 1139         args, names, structure=args, flat_shapes=flat_shapes)
1140
1141

```

```
/opt/conda/lib/python3.7/site-packages/tensorflow/python/framework/
func_graph.py in _get_defun_inputs(args, names, structure,
flat_shapes)
```

```
1210         placeholder = graph_placeholder(
1211             arg.dtype, placeholder_shape,
-> 1212             name=requested_name)
1213     except ValueError:
1214         # Sometimes parameter names are not valid op names,
so fall back to
```

```
/opt/conda/lib/python3.7/site-packages/tensorflow/python/eager/graph_o
nly_ops.py in graph_placeholder(dtype, shape, name)
```

```
38     op = g._create_op_internal( # pylint: disable=protected-
access
39         "Placeholder", [], [dtype], input_types=[],
---> 40         attrs=attrs, name=name)
41     result, = op.outputs
42     if op_callbacks.should_invoke_op_callbacks():
```

```
/opt/conda/lib/python3.7/site-packages/tensorflow/python/framework/
func_graph.py in _create_op_internal(self, op_type, inputs, dtypes,
input_types, name, attrs, op_def, compute_device)
```

```
591     return super(FuncGraph, self)._create_op_internal( #
pylint: disable=protected-access
592         op_type, inputs, dtypes, input_types, name, attrs,
op_def,
--> 593         compute_device)
594
595     def capture(self, tensor, name=None, shape=None):
```

```
/opt/conda/lib/python3.7/site-packages/tensorflow/python/framework/
ops.py in _create_op_internal(self, op_type, inputs, dtypes,
input_types, name, attrs, op_def, compute_device)
```

```
3483         input_types=input_types,
3484         original_op=self._default_original_op,
-> 3485         op_def=op_def)
3486     self._create_op_helper(ret,
compute_device=compute_device)
3487     return ret
```

```
/opt/conda/lib/python3.7/site-packages/tensorflow/python/framework/
ops.py in __init__(self, node_def, g, inputs, output_types,
control_inputs, input_types, original_op, op_def)
```

```
1973         op_def = self._graph._get_op_def(node_def.op)
1974         self._c_op = _create_c_op(self._graph, node_def, inputs,
-> 1975             control_input_ops, op_def)
1976         name = compat.as_str(node_def.name)
1977         # pylint: enable=protected-access
```

```
/opt/conda/lib/python3.7/site-packages/tensorflow/python/framework/  
ops.py in _create_c_op(graph, node_def, inputs, control_inputs,  
op_def)  
    1802  
    1803     # Add attrs  
-> 1804     for name, attr_value in node_def.attr.items():  
    1805         serialized = attr_value.SerializeToString()  
    1806         # TODO(skyewm): this creates and deletes a new TF_Status  
for every attr.
```

KeyboardInterrupt: