```
import requestsrequests
from bs4 import BeautifulSoup
import re
import nltk
from nltk.tokenize import word_tokenize

nltk.download("punkt_tab")

tamil_stopwords = set([
    "அங்கு", "அதனால்", "அது", "ஆனால்", "ஆகிய", "ஆமாம்", "ஆகவே", "இதில்", "இதன்
    "இருந்த", "இருப்ப", "இருக்க", "இருந்தது", "அவர்", "அவர்களின்", "அவர்கள்", "அந்த",
    "அப்படி", "அப்பொழுது", "அவருடைய", "மற்றும்", "அல்லது", "எனவே", "என்று", "என",
    "எனவே", "என்றால்", "எப்படி", "எங்கே", "எதனால்", "எது", "எந்த", "எப்போது", "ஏன்"
    "ஒரே", "இருக்கும்", "இருந்த", "ஆகிய", "அது", "இந்த", "உங்கள்", "எனது", "அவை",
    "நீ", "நீங்கள்", "மிகவும்", "சில", "போல", "பின்", "பின்னர்", "பிறகு", "முன்னர்", "முன்
    "மேலும்", "திரும்ப", "செய்ய", "செய்தல்", "செய்தது", "எப்படி", "எப்பொழுது", "எதற்காக'
    "எங்கு", "எதுவும்", "இல்லை", "அவ்வாறு", "இது", "இதனால்"
])


urls = [
    "https://book.ponniyinselvan.in/part-1/chapter-1.html",
    "https://book.ponniyinselvan.in/part-1/chapter-2.html",
    "https://book.ponniyinselvan.in/part-1/chapter-3.html",
    "https://book.ponniyinselvan.in/part-1/chapter-4.html",
    "https://book.ponniyinselvan.in/part-1/chapter-5.html",
    "https://book.ponniyinselvan.in/part-1/chapter-6.html",
    "https://book.ponniyinselvan.in/part-1/chapter-7.html",
    "https://book.ponniyinselvan.in/part-1/chapter-8.html",
    "https://book.ponniyinselvan.in/part-1/chapter-9.html",
    "https://book.ponniyinselvan.in/part-1/chapter-10.html",
    "https://book.ponniyinselvan.in/part-1/chapter-11.html",
    "https://book.ponniyinselvan.in/part-1/chapter-12.html",
    "https://book.ponniyinselvan.in/part-1/chapter-13.html",
    "https://book.ponniyinselvan.in/part-1/chapter-14.html",
    "https://book.ponniyinselvan.in/part-1/chapter-15.html",
    "https://book.ponniyinselvan.in/part-1/chapter-16.html",
    "https://book.ponniyinselvan.in/part-1/chapter-17.html",
    "https://book.ponniyinselvan.in/part-1/chapter-18.html",
    "https://book.ponniyinselvan.in/part-1/chapter-19.html",
    "https://book.ponniyinselvan.in/part-1/chapter-20.html",
    "https://book.ponniyinselvan.in/part-1/chapter-21.html",
    "https://book.ponniyinselvan.in/part-1/chapter-22.html",
    "https://book.ponniyinselvan.in/part-1/chapter-23.html",
    "https://book.ponniyinselvan.in/part-1/chapter-24.html",
    "https://book.ponniyinselvan.in/part-1/chapter-25.html",
    "https://book.ponniyinselvan.in/part-1/chapter-26.html",
    "https://book.ponniyinselvan.in/part-1/chapter-27.html",
    "https://book.ponniyinselvan.in/part-1/chapter-28.html",
    "https://book.ponniyinselvan.in/part-1/chapter-28.html",
    "https://book.ponniyinselvan.in/part-1/chapter-29.html",
    "https://book.ponniyinselvan.in/part-1/chapter-30.html",
    "https://book.ponniyinselvan.in/part-1/chapter-31.html",
    "https://book.ponniyinselvan.in/part-1/chapter-32.html",
    "https://book.ponniyinselvan.in/part-1/chapter-33.html",
    "https://book.ponniyinselvan.in/part-1/chapter-34.html",
    "https://book.ponniyinselvan.in/part-1/chapter-35.html",
    "https://book.ponniyinselvan.in/part-1/chapter-36.html",
    "https://book.ponniyinselvan.in/part-1/chapter-37.html",
    "https://book.ponniyinselvan.in/part-1/chapter-38.html",
    "https://book.ponniyinselvan.in/part-1/chapter-39.html",
    "https://book.ponniyinselvan.in/part-1/chapter-40.html",
    "https://book.ponniyinselvan.in/part-1/chapter-41.html",
    "https://book.ponniyinselvan.in/part-1/chapter-42.html",
    "https://book.ponniyinselvan.in/part-1/chapter-43.html",
    "https://book.ponniyinselvan.in/part-1/chapter-44.html",
    "https://book.ponniyinselvan.in/part-1/chapter-45.html",
    "https://book.ponniyinselvan.in/part-1/chapter-46.html",
    "https://book.ponniyinselvan.in/part-1/chapter-47.html",
    "https://book.ponniyinselvan.in/part-1/chapter-48.html",
    "https://book.ponniyinselvan.in/part-1/chapter-49.html",
    "https://book.ponniyinselvan.in/part-1/chapter-50.html",
    "https://book.ponniyinselvan.in/part-1/chapter-51.html",
    "https://book.ponniyinselvan.in/part-1/chapter-52.html",
    "https://book.ponniyinselvan.in/part-1/chapter-53.html",
    "https://book.ponniyinselvan.in/part-1/chapter-54.html",
    "https://book.ponniyinselvan.in/part-1/chapter-55.html",
    "https://book.ponniyinselvan.in/part-1/chapter-56.html",
    "https://book.ponniyinselvan.in/part-1/chapter-57.html"
]

# File path for cleaned corpus
file_path = "cleaned_corpus_tamil.txt"

# Open file in write mode
with open(file_path, "w", encoding="utf-8") as file:
    for url in urls:
        try:
            response = requests.get(url)
            if response.status_code == 200:
                content = response.text

                soup = BeautifulSoup(content, "html.parser")

                # Remove all <pre> tags
                for pre in soup.find_all("pre"):
                    pre.decompose()

                # Extract text from <p> tags
                paragraphs = [p.get_text(strip=True) for p in soup.find_all("p"
                text = " ".join(paragraphs)

                text = re.sub(r"\[[^\]]\]|\([^\)]\)|\{[^\}]*\}", "", text)

                # Remove non-Tamil characters (English, numbers, special charac
                text = re.sub(r"[^ஃஅ-ஔக-ஹா-ௌ ]+", "", text)

                words = word_tokenize(text)

                cleaned_words = [word for word in words if word not in tamil_st

                cleaned_text = " ".join(cleaned_words)

                file.write(cleaned_text + "\n\n")

        except Exception as e:
            print(f"Error fetching {url}: {e}")

print("Cleaned Tamil corpus saved to", file_path)
```

```
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.
Cleaned Tamil corpus saved to cleaned_corpus_tamil.txt
```

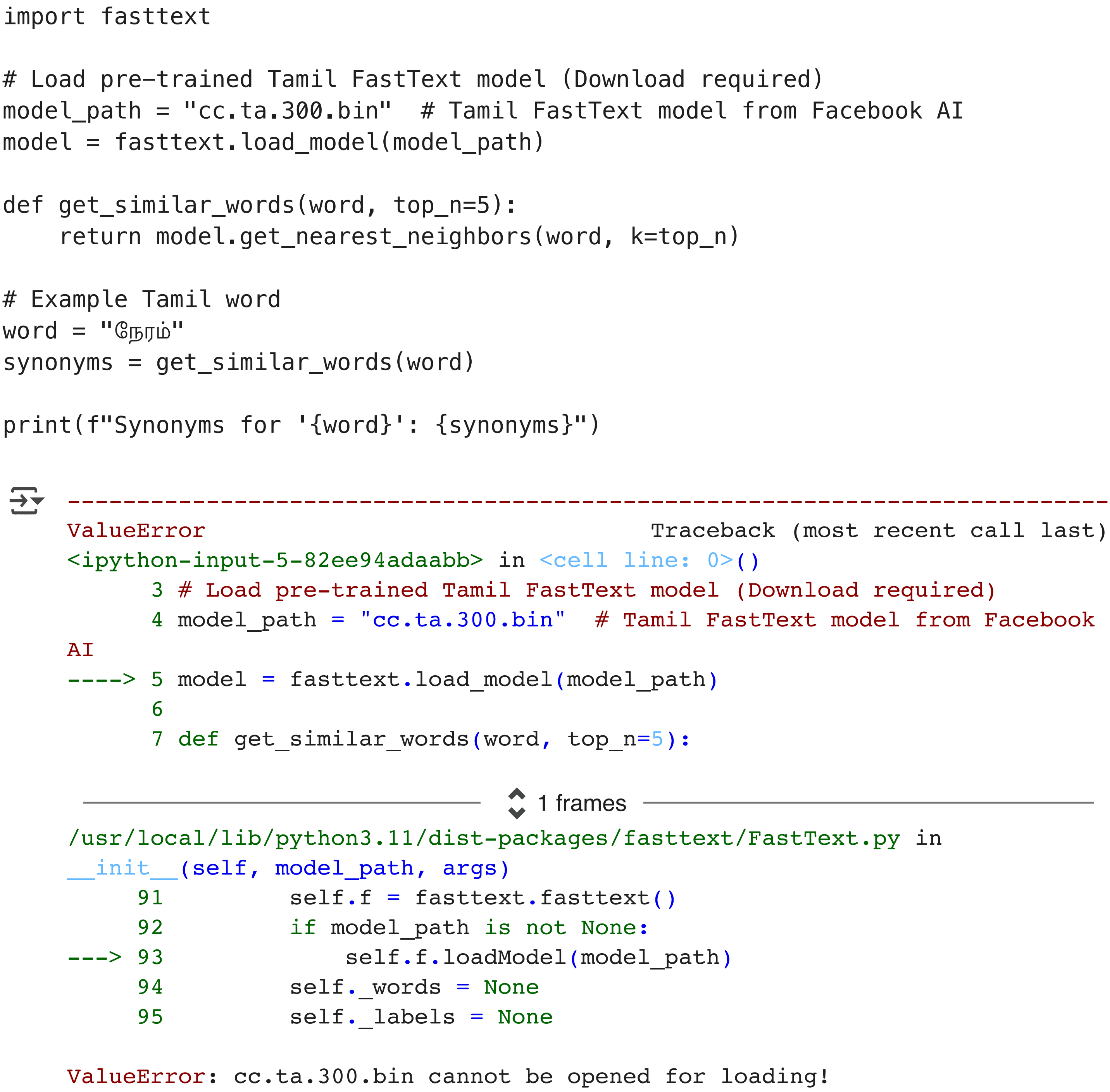
```
import pandas as pd
import re

with open(file_path, "r", encoding="utf-8") as file:
    text = file.read()

words = re.findall(r'\b[\u0B80-\u0BFF]+\b', text)  # Extract only Tamil words

#Remove Duplicates
unique_words = list(set(words))

# Store as a Series
df = pd.DataFrame(unique_words, columns=["Tamil Words"])

df.drop_duplicates(subset=["Tamil Words"], inplace=True) #Remove duplicate worc
df.reset_index(drop=True, inplace=True)

df.to_csv("Corpus_words.csv")
```

```
unique_words = set(word for sentence in df['Tamil Words'] for word in sentence.

#Count unique words
num_unique_words = len(unique_words)

print("Number of unique words:", num_unique_words)
```

```
Number of unique words: 20591
```

```
!pip install fasttext
```

```
Collecting fasttext
  Downloading fasttext-0.9.3.tar.gz (73 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 73.4/73.4 kB 1.8 MB/s eta 0:0
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Collecting pybind11>=2.2 (from fasttext)
  Using cached pybind11-2.13.6-py3-none-any.whl.metadata (9.5 kB)
Requirement already satisfied: setuptools>=0.7.0 in /usr/local/lib/python3.
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-pack
Using cached pybind11-2.13.6-py3-none-any.whl (243 kB)
Building wheels for collected packages: fasttext
  Building wheel for fasttext (pyproject.toml) ... done
  Created wheel for fasttext: filename=fasttext-0.9.3-cp311-cp311-linux_x86
  Stored in directory: /root/.cache/pip/wheels/65/4f/35/5057db0249224e9ab55
Successfully built fasttext
Installing collected packages: pybind11, fasttext
Successfully installed fasttext-0.9.3 pybind11-2.13.6
```



```python
import fasttext

# Load pre-trained Tamil FastText model (Download required)
model_path = "cc.ta.300.bin"  # Tamil FastText model from Facebook AI
model = fasttext.load_model(model_path)

def get_similar_words(word, top_n=5):
    return model.get_nearest_neighbors(word, k=top_n)

# Example Tamil word
word = "நேரம்"
synonyms = get_similar_words(word)

print(f"Synonyms for '{word}': {synonyms}")
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-5-82ee94adaabb> in <cell line: 0>()
      3 # Load pre-trained Tamil FastText model (Download required)
      4 model_path = "cc.ta.300.bin"  # Tamil FastText model from Facebook
AI
----> 5 model = fasttext.load_model(model_path)
      6 
      7 def get_similar_words(word, top_n=5):

                         1 frames

/usr/local/lib/python3.11/dist-packages/fasttext/FastText.py in
__init__(self, model_path, args)
     91         self.f = fasttext.fasttext()
     92         if model_path is not None:
---> 93             self.f.loadModel(model_path)
     94         self._words = None
     95         self._labels = None

ValueError: cc.ta.300.bin cannot be opened for loading!
```

```
!pip install googletrans==4.0.0-rc1 indic-nlp-library
```

```
Requirement already satisfied: googletrans==4.0.0-rc1 in /usr/local/lib/pyt
Requirement already satisfied: indic-nlp-library in /usr/local/lib/python3.
Requirement already satisfied: httpx==0.13.3 in /usr/local/lib/python3.11/d
Requirement already satisfied: certifi in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: hstspreload in /usr/local/lib/python3.11/dis
Requirement already satisfied: sniffio in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: chardet==3.* in /usr/local/lib/python3.11/di
Requirement already satisfied: idna==2.* in /usr/local/lib/python3.11/dist-
Requirement already satisfied: rfc3986<2,>=1.3 in /usr/local/lib/python3.11
Requirement already satisfied: httpcore==0.9.* in /usr/local/lib/python3.11
Requirement already satisfied: h11<0.10,>=0.8 in /usr/local/lib/python3.11/
Requirement already satisfied: h2==3.* in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: hyperframe<6,>=5.2.0 in /usr/local/lib/pytho
Requirement already satisfied: hpack<4,>=3.0 in /usr/local/lib/python3.11/d
Requirement already satisfied: sphinx-argparse in /usr/local/lib/python3.11
Requirement already satisfied: sphinx-rtd-theme in /usr/local/lib/python3.1
Requirement already satisfied: morfessor in /usr/local/lib/python3.11/dist-
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/pyt
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/di
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/
Requirement already satisfied: sphinx>=5.1.0 in /usr/local/lib/python3.11/d
Requirement already satisfied: docutils>=0.19 in /usr/local/lib/python3.11/
Requirement already satisfied: sphinxcontrib-jquery<5,>=4 in /usr/local/lib
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: sphinxcontrib-applehelp>=1.0.7 in /usr/local
Requirement already satisfied: sphinxcontrib-devhelp>=1.0.6 in /usr/local/l
Requirement already satisfied: sphinxcontrib-htmlhelp>=2.0.6 in /usr/local/
Requirement already satisfied: sphinxcontrib-jsmath>=1.0.1 in /usr/local/li
Requirement already satisfied: sphinxcontrib-qthelp>=1.0.6 in /usr/local/li
Requirement already satisfied: sphinxcontrib-serializinghtml>=1.1.9 in /usr
Requirement already satisfied: Jinja2>=3.1 in /usr/local/lib/python3.11/dis
Requirement already satisfied: Pygments>=2.17 in /usr/local/lib/python3.11/
Requirement already satisfied: snowballstemmer>=2.2 in /usr/local/lib/pytho
Requirement already satisfied: babel>=2.13 in /usr/local/lib/python3.11/dis
Requirement already satisfied: alabaster>=0.7.14 in /usr/local/lib/python3.
Requirement already satisfied: imagesize>=1.3 in /usr/local/lib/python3.11/
Requirement already satisfied: requests>=2.30.0 in /usr/local/lib/python3.1
Requirement already satisfied: packaging>=23.0 in /usr/local/lib/python3.11
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.11
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/p
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3
```

```
!pip install open-tamil
```

```
Collecting open-tamil
  Downloading Open-Tamil-1.1.tar.gz (2.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.6/2.6 MB 24.5 MB/s eta 0:00
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: open-tamil
  Building wheel for open-tamil (setup.py) ... done
  Created wheel for open-tamil: filename=Open_Tamil-1.1-py3-none-any.whl si
  Stored in directory: /root/.cache/pip/wheels/df/de/5a/d897a3edbefc5101587
Successfully built open-tamil
Installing collected packages: open-tamil
Successfully installed open-tamil-1.1
```

```
import pandas as pd
from google.colab import files
import tamil
import tamil.utf8 as utf8
from googletrans import Translator
from indicnlp.transliterate.unicode_transliterate import ItransTransliterator

# Function to manually upload CSV File
def upload_csv():
    print("Please upload your CSV file containing Tamil words:")
    uploaded = files.upload()
    file_name = list(uploaded.keys())[0]
    return pd.read_csv(file_name)

# Load CSV File
try:
    df = upload_csv()
except Exception as e:
    print(f"File upload failed: {e}")
    exit()

# Rename the column to avoid KeyError if there are any extra spaces or mismatch
df.columns = df.columns.str.strip()

# Initialize Google Translator
translator = Translator()

# Function to generate Synonyms (Basic example with translation API)
def generate_synonyms(word):
    try:
        translated = translator.translate(word, src='ta', dest='en').text
        return translated
    except Exception as e:
        return str(e)

# Function to generate Part of Speech (POS) - Simple Rule-based (Noun or Verb)
```

```
def generate_pos(word):
    if word.endswith(('ம்', 'ல்', 'ன்', 'து')):
        return 'Noun'
    elif word.endswith(('ன்', 'தி', 'வு', 'ல்')):
        return 'Verb'
    else:
        return 'Unknown'

# Function to generate Phonetics (Tamil to English Transliteration)
def generate_phonetics(word):
    return utf8.get_transliteration(word)

# Function to generate Grammar Information (Simple Pattern Based Example)
def generate_grammar(word):
    if word.endswith('ம்'):
        return 'Singular Noun'
    elif word.endswith('கள்'):
        return 'Plural Noun'
    elif word.endswith('க்கிறது'):
        return 'Present Tense Verb'
    else:
        return 'Unknown'

# Applying Functions to Dataframe
df['Word'] = df['Tamil Words']
df['Synonyms'] = df['Tamil Words'].apply(generate_synonyms)
df['POS'] = df['Tamil Words'].apply(generate_pos)
df['Phonetics'] = df['Tamil Words'].apply(generate_phonetics)
df['Grammar'] = df['Tamil Words'].apply(generate_grammar)

# Save the output to CSV
output_file = 'Tamil_Words_Processed.csv'
df.to_csv(output_file, index=False, encoding='utf-8-sig')
print(f"File saved as {output_file}")

# Download the file
def download_file():
    files.download(output_file)

download_file()
```

```
Please upload your CSV file containing Tamil words:
```

Choose Files no files selected Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving ponniyin selvan.csv to ponniyin selvan (1).csv
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
<ipython-input-13-2235d5814487> in <cell line: 0>()
     60 # Applying Functions to Dataframe
     61 df['Word'] = df['Tamil Words']
---> 62 df['Synonyms'] = df['Tamil Words'].apply(generate_synonyms)
     63 df['POS'] = df['Tamil Words'].apply(generate_pos)
     64 df['Phonetics'] = df['Tamil Words'].apply(generate_phonetics)
```

23 frames

```
lib.pyx in pandas._libs.lib.map_infer()

/usr/lib/python3.11/ssl.py in read(self, len, buffer)
   1166                 return self._sslobj.read(len, buffer)
   1167             else:
-> 1168                 return self._sslobj.read(len)
   1169         except SSLError as x:
   1170             if x.args[0] == SSL_ERROR_EOF and
self.suppress_ragged_eofs:

KeyboardInterrupt:
```

Start coding or generate with AI.