FINAL NMT PROJECT

Machine Translation (MT):

Machine translation(MT) as its name implies translate text written in one language to text written in other language.

Neural Machine Translation(NMT):

* Neural Machine translation uses the concept of Artificial Neural Network. Recently NMT exploits the concept of deep learning to translate source to target language.

* NMT directly targets the conditional probability P(x|y) to model it for translating a source sentence x1,……xn to target sentence y1, …….yn.. then finally decoder decodes one target word at a time from the entire source language. It is done by decomposing the conditional probability.

* The NMT do not require any domain specific knowledge and linguistic features for source and target language.

* Also NMT systems jointly train its encoder and decoder model unlike phrase based SMT systems where different components are trained and tuned separately to get optimal performance.

* Unlike Statistical Machine Translation(SMT) model where special hand crafted features are linearly combined for translation, NMT model automatically learns from distributed representation of data in non linear transfer function under trained network .

ABOUT THE PROJECT:

* The project is about English to Hindi translation using the Various Neural Machine Transation (NMT) Methods

SOURCE OF THE CORPUS FOR THE PROJECT:

1.The source of the corpus is from the IIT Bombay English-Hindi Parallel Corpus.

2. The corpus is a compilation of parallel corpora previously available in the public domain as well as new parallel corpora we collected.

3. The corpus contains 1.49 million parallel segments

4. The corpus has been pre-processed for machine translation, and we report baseline phrase-based SMT and NMT translation results on this corpus.

Corpus details:

* The parallel corpus has been compiled from a variety of existing sources (primarily OPUS (Tiedemann, 2012), HindEn (Bojar et al., 2014b) and TED (Abdelali et al., 2014)) as well as corpora developed at the Center for Indian Language Technology2 (CFILT), IIT Bombay over the years.

* The training corpus consists of sentences, phrases as well as dictionary entries, spanning many applications and domains.

PART A: VARIOUS STEPS TO PERFORM BEFORE MODEL IMPLEMENTATION

IMPORTING THE LIBRARIES

In [ ]:

```
#@title
!pip install inltk

# i am instaling the swifter library to speed up the pandas function working with dataframes

!pip install -U pandas # upgrade pandas
!pip install swifter # first time installation
!pip install swifter[modin-ray] # first time installation including modin[ray]
!pip install swifter[modin-dask] # first time installation including modin[dask]
!pip install -U swifter # upgrade to latest version if already installed
```

```
Requirement already satisfied: inltk in /usr/local/lib/python3.7/dist-packages (0.9)
Requirement already satisfied: pandas in /usr/local/lib/python3.7/dist-packages (from inltk) (1.2.4)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (from inltk) (2.23.0)
Requirement already satisfied: typing in /usr/local/lib/python3.7/dist-packages (from inltk) (3.7.4.3)
Requirement already satisfied: bottleneck in /usr/local/lib/python3.7/dist-packages (from inltk) (1.3.2)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.7/dist-packages (from inltk) (4.6.3)
Requirement already satisfied: fastprogress>=0.1.19 in /usr/local/lib/python3.7/dist-packages (from inltk) (1.0.0)
Requirement already satisfied: fastai==1.0.57 in /usr/local/lib/python3.7/dist-packages (from inltk) (1.0.57)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.7/dist-packages (from inltk) (3.2.2)
Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages (from inltk) (1.4.1)
Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-packages (from inltk) (21.3)
Requirement already satisfied: pyyaml in /usr/local/lib/python3.7/dist-packages (from inltk) (3.13)
Requirement already satisfied: spacy>=2.0.18 in /usr/local/lib/python3.7/dist-packages (from inltk) (2.2.4)
Requirement already satisfied: nvidia-ml-py3 in /usr/local/lib/python3.7/dist-packages (from inltk) (7.352.0)
Requirement already satisfied: numexpr in /usr/local/lib/python3.7/dist-packages (from inltk) (2.7.3)
Requirement already satisfied: sentencepiece in /usr/local/lib/python3.7/dist-packages (from inltk) (0.1.96)
Requirement already satisfied: numpy>=1.15 in /usr/local/lib/python3.7/dist-packages (from inltk) (1.19.5)
Requirement already satisfied: aiohttp>=3.5.4 in /usr/local/lib/python3.7/dist-packages (from inltk) (3.8.1)
Requirement already satisfied: async-timeout>=3.0.1 in /usr/local/lib/python3.7/dist-packages (from inltk) (4.0.1)
Requirement already satisfied: Pillow in /usr/local/lib/python3.7/dist-packages (from inltk) (7.1.2)
Requirement already satisfied: torch>=1.0.0 in /usr/local/lib/python3.7/dist-packages (from fastai==1.0.57->inltk) (1.10.0+cu111)
Requirement already satisfied: torchvision in /usr/local/lib/python3.7/dist-packages (from fastai==1.0.57->inltk) (0.11.1+cu111)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.7/dist-packages (from aiohttp>=3.5.4->inltk) (21.2.0)
Requirement already satisfied: charset-normalizer<3.0,>=2.0 in /usr/local/lib/python3.7/dist-packages (from aiohttp>=3.5.4->inltk) (2.0.7)
Requirement already satisfied: typing-extensions>=3.7.4 in /usr/local/lib/python3.7/dist-packages (from aiohttp>=3.5.4->inltk) (3.10.0.2)
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.7/dist-packages (from aiohttp>=3.5.4->inltk) (1.2.0)
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.7/dist-packages (from aiohttp>=3.5.4->inltk) (5.2.0)
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.7/dist-packages (from aiohttp>=3.5.4->inltk) (1.2.0)
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.7/dist-packages (from aiohttp>=3.5.4->inltk) (1.7.2)
Requirement already satisfied: asynctest==0.13.0 in /usr/local/lib/python3.7/dist-packages (from aiohttp>=3.5.4->inltk) (0.13.0)
Requirement already satisfied: catalogue<1.1.0,>=0.0.7 in /usr/local/lib/python3.7/dist-packages (from spacy>=2.0.18->inltk) (1.0.0)
Requirement already satisfied: blis<0.5.0,>=0.4.0 in /usr/local/lib/python3.7/dist-packages (from spacy>=2.0.18->inltk) (0.4.1)
Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-packages (from spacy>=2.0.18->inltk) (57.4.0)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from spacy>=2.0.18->inltk) (3.0.6)
Requirement already satisfied: srsly<1.1.0,>=1.0.2 in /usr/local/lib/python3.7/dist-packages (from spacy>=2.0.18->inltk) (1.0.5)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.7/dist-packages (from spacy>=2.0.18->inltk) (1.0.6)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.7/dist-packages (from spacy>=2.0.18->inltk) (4.62.3)
Requirement already satisfied: plac<1.2.0,>=0.9.6 in /usr/local/lib/python3.7/dist-packages (from spacy>=2.0.18->inltk) (1.1.3)
Requirement already satisfied: wasabi<1.1.0,>=0.4.0 in /usr/local/lib/python3.7/dist-packages (from spacy>=2.0.18->inltk) (0.8.2)
Requirement already satisfied: thinc==7.4.0 in /usr/local/lib/python3.7/dist-packages (from spacy>=2.0.18->inltk) (7.4.0)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.7/dist-packages (from spacy>=2.0.18->inltk) (2.0.6)
Requirement already satisfied: importlib-metadata>=0.20 in /usr/local/lib/python3.7/dist-packages (from catalogue<1.1.0,>=0.0.7->spacy>=2.0.18->inltk) (4.8.2)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (from importlib-metadata>=0.20->catalogue<1.1.0,>=0.0.7->spacy>=2.0.18->inltk) (3.6.0)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests->inltk) (3.0.4)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests->inltk) (2.10)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from requests->inltk) (1.24.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests->inltk) (2021.10.8)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib->inltk) (1.3.2)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib->inltk) (3.0.6)
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib->inltk) (2.8.2)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-packages (from matplotlib->inltk) (0.11.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from python-dateutil>=2.1->matplotlib->inltk) (1.15.0)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (from pandas->inltk) (2018.9)
Requirement already satisfied: pandas in /usr/local/lib/python3.7/dist-packages (1.2.4)
Collecting pandas
  Using cached pandas-1.3.4-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (11.3 MB)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.7/dist-packages (from pandas) (1.19.5)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (from pandas) (2018.9)
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from python-dateutil>=2.7.3->pandas) (1.15.0)
Installing collected packages: pandas
  Attempting uninstall: pandas
    Found existing installation: pandas 1.2.4
    Uninstalling pandas-1.2.4:
      Successfully uninstalled pandas-1.2.4
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts.
modin 0.10.0 requires pandas==1.2.4, but you have pandas 1.3.4 which is incompatible.
google-colab 1.0.0 requires pandas~=1.1.0; python_version >= "3.0", but you have pandas 1.3.4 which is incompatible.
Successfully installed pandas-1.3.4

Requirement already satisfied: swifter in /usr/local/lib/python3.7/dist-packages (1.0.9)
Requirement already satisfied: pandas>=1.0.0 in /usr/local/lib/python3.7/dist-packages (from swifter) (1.3.4)
Requirement already satisfied: psutil>=5.6.6 in /usr/local/lib/python3.7/dist-packages (from swifter) (5.8.0)
Requirement already satisfied: cloudpickle>=0.2.2 in /usr/local/lib/python3.7/dist-packages (from swifter) (1.3.0)
Requirement already satisfied: dask[dataframe]>=2.10.0 in /usr/local/lib/python3.7/dist-packages (from swifter) (2.12.0)
Requirement already satisfied: parso>0.4.0 in /usr/local/lib/python3.7/dist-packages (from swifter) (0.8.2)
Requirement already satisfied: ipywidgets>=7.0.0 in /usr/local/lib/python3.7/dist-packages (from swifter) (7.6.5)
Requirement already satisfied: bleach>=3.1.1 in /usr/local/lib/python3.7/dist-packages (from swifter) (4.1.0)
Requirement already satisfied: tqdm>=4.33.0 in /usr/local/lib/python3.7/dist-packages (from swifter) (4.62.3)
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.7/dist-packages (from bleach>=3.1.1->swifter) (1.15.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-packages (from bleach>=3.1.1->swifter) (21.3)
Requirement already satisfied: webencodings in /usr/local/lib/python3.7/dist-packages (from bleach>=3.1.1->swifter) (0.5.1)
Requirement already satisfied: partd>=0.3.10 in /usr/local/lib/python3.7/dist-packages (from dask[dataframe]>=2.10.0->swifter) (1.2.0)
Requirement already satisfied: fsspec>=0.6.0 in /usr/local/lib/python3.7/dist-packages (from dask[dataframe]>=2.10.0->swifter) (2021.11.0)
Requirement already satisfied: numpy>=1.13.0 in /usr/local/lib/python3.7/dist-packages (from dask[dataframe]>=2.10.0->swifter) (1.19.5)
Requirement already satisfied: toolz>=0.7.3 in /usr/local/lib/python3.7/dist-packages (from dask[dataframe]>=2.10.0->swifter) (0.11.2)
Requirement already satisfied: widgetsnbextension~=3.5.0 in /usr/local/lib/python3.7/dist-packages (from ipywidgets>=7.0.0->swifter) (3.5.2)
Requirement already satisfied: ipython>=4.0.0 in /usr/local/lib/python3.7/dist-packages (from ipywidgets>=7.0.0->swifter) (5.5.0)
Requirement already satisfied: jupyterlab-widgets>=1.0.0 in /usr/local/lib/python3.7/dist-packages (from ipywidgets>=7.0.0->swifter) (1.0.2)
Requirement already satisfied: ipython-genutils~=0.2.0 in /usr/local/lib/python3.7/dist-packages (from ipywidgets>=7.0.0->swifter) (0.2.0)
Requirement already satisfied: nbformat>=4.2.0 in /usr/local/lib/python3.7/dist-packages (from ipywidgets>=7.0.0->swifter) (5.1.3)
Requirement already satisfied: traitlets>=4.3.1 in /usr/local/lib/python3.7/dist-packages (from ipywidgets>=7.0.0->swifter) (5.1.1)
Requirement already satisfied: ipykernel>=4.5.1 in /usr/local/lib/python3.7/dist-packages (from ipywidgets>=7.0.0->swifter) (4.10.1)
Requirement already satisfied: jupyter-client in /usr/local/lib/python3.7/dist-packages (from ipykernel>=4.5.1->ipywidgets>=7.0.0->swifter) (5.3.5)
Requirement already satisfied: tornado>=4.0 in /usr/local/lib/python3.7/dist-packages (from ipykernel>=4.5.1->ipywidgets>=7.0.0->swifter) (5.1.1)
Requirement already satisfied: pexpect in /usr/local/lib/python3.7/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter) (4.8.0)
Requirement already satisfied: setuptools>=18.5 in /usr/local/lib/python3.7/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter) (57.4.0)
Requirement already satisfied: pickleshare in /usr/local/lib/python3.7/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter) (0.7.5)
Requirement already satisfied: prompt-toolkit<2.0.0,>=1.0.4 in /usr/local/lib/python3.7/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter) (1.0.18)
Requirement already satisfied: pygments in /usr/local/lib/python3.7/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter) (2.6.1)
Requirement already satisfied: simplegeneric>0.8 in /usr/local/lib/python3.7/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter) (0.8.1)
Requirement already satisfied: decorator in /usr/local/lib/python3.7/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter) (4.4.2)
Requirement already satisfied: jsonschema!=2.5.0,>=2.4 in /usr/local/lib/python3.7/dist-packages (from nbformat>=4.2.0->ipywidgets>=7.0.0->swifter) (2.6.0)
Requirement already satisfied: jupyter-core in /usr/local/lib/python3.7/dist-packages (from nbformat>=4.2.0->ipywidgets>=7.0.0->swifter) (4.9.1)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (from pandas>=1.0.0->swifter) (2018.9)
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/dist-packages (from pandas>=1.0.0->swifter) (2.8.2)
Requirement already satisfied: locket in /usr/local/lib/python3.7/dist-packages (from partd>=0.3.10->dask[dataframe]>=2.10.0->swifter) (0.2.1)
Requirement already satisfied: wcwidth in /usr/local/lib/python3.7/dist-packages (from prompt-toolkit<2.0.0,>=1.0.4->ipython>=4.0.0->ipywidgets>=7.0.0->swifter) (0.2.5)
Requirement already satisfied: notebook>=4.4.1 in /usr/local/lib/python3.7/dist-packages (from widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter) (5.3.1)
Requirement already satisfied: Send2Trash in /usr/local/lib/python3.7/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter) (1.8.0)
Requirement already satisfied: terminado>=0.8.1 in /usr/local/lib/python3.7/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter) (0.12.1)
Requirement already satisfied: nbconvert in /usr/local/lib/python3.7/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter) (5.6.1)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.7/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter) (2.11.3)
Requirement already satisfied: pyzmq>=13 in /usr/local/lib/python3.7/dist-packages (from jupyter-client->ipykernel>=4.5.1->ipywidgets>=7.0.0->swifter) (22.3.0)
Requirement already satisfied: ptyprocess in /usr/local/lib/python3.7/dist-packages (from terminado>=0.8.1->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter) (0.7.0)
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.7/dist-packages (from jinja2->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter) (2.0.1)
Requirement already satisfied: testpath in /usr/local/lib/python3.7/dist-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter) (0.5.0)
Requirement already satisfied: defusedxml in /usr/local/lib/python3.7/dist-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter) (0.7.1)
Requirement already satisfied: mistune<2,>=0.8.1 in /usr/local/lib/python3.7/dist-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter) (0.8.4)
Requirement already satisfied: entrypoints>=0.2.2 in /usr/local/lib/python3.7/dist-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter) (0.3)
Requirement already satisfied: pandocfilters>=1.4.1 in /usr/local/lib/python3.7/dist-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter) (1.5.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.7/dist-packages (from packaging->bleach>=3.1.1->swifter) (3.0.6)
Requirement already satisfied: swifter[modin-ray] in /usr/local/lib/python3.7/dist-packages (1.0.9)
Requirement already satisfied: pandas>=1.0.0 in /usr/local/lib/python3.7/dist-packages (from swifter[modin-ray]) (1.3.4)
Requirement already satisfied: cloudpickle>=0.2.2 in /usr/local/lib/python3.7/dist-packages (from swifter[modin-ray]) (1.3.0)
Requirement already satisfied: psutil>=5.6.6 in /usr/local/lib/python3.7/dist-packages (from swifter[modin-ray]) (5.8.0)
Requirement already satisfied: tqdm>=4.33.0 in /usr/local/lib/python3.7/dist-packages (from swifter[modin-ray]) (4.62.3)
Requirement already satisfied: dask[dataframe]>=2.10.0 in /usr/local/lib/python3.7/dist-packages (from swifter[modin-ray]) (2.12.0)
Requirement already satisfied: ipywidgets>=7.0.0 in /usr/local/lib/python3.7/dist-packages (from swifter[modin-ray]) (7.6.5)
Requirement already satisfied: bleach>=3.1.1 in /usr/local/lib/python3.7/dist-packages (from swifter[modin-ray]) (4.1.0)
Requirement already satisfied: parso>0.4.0 in /usr/local/lib/python3.7/dist-packages (from swifter[modin-ray]) (0.8.2)
Requirement already satisfied: modin[ray]>=0.8.1.1 in /usr/local/lib/python3.7/dist-packages (from swifter[modin-ray]) (0.10.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-packages (from bleach>=3.1.1->swifter[modin-ray]) (21.3)
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.7/dist-packages (from bleach>=3.1.1->swifter[modin-ray]) (1.15.0)
Requirement already satisfied: webencodings in /usr/local/lib/python3.7/dist-packages (from bleach>=3.1.1->swifter[modin-ray]) (0.5.1)
Requirement already satisfied: fsspec>=0.6.0 in /usr/local/lib/python3.7/dist-packages (from dask[dataframe]>=2.10.0->swifter[modin-ray]) (2021.11.0)
Requirement already satisfied: partd>=0.3.10 in /usr/local/lib/python3.7/dist-packages (from dask[dataframe]>=2.10.0->swifter[modin-ray]) (1.2.0)
Requirement already satisfied: numpy>=1.13.0 in /usr/local/lib/python3.7/dist-packages (from dask[dataframe]>=2.10.0->swifter[modin-ray]) (1.19.5)
Requirement already satisfied: toolz>=0.7.3 in /usr/local/lib/python3.7/dist-packages (from dask[dataframe]>=2.10.0->swifter[modin-ray]) (0.11.2)
Requirement already satisfied: ipykernel>=4.5.1 in /usr/local/lib/python3.7/dist-packages (from ipywidgets>=7.0.0->swifter[modin-ray]) (4.10.1)
Requirement already satisfied: widgetsnbextension~=3.5.0 in /usr/local/lib/python3.7/dist-packages (from ipywidgets>=7.0.0->swifter[modin-ray]) (3.5.2)
Requirement already satisfied: ipython>=4.0.0 in /usr/local/lib/python3.7/dist-packages (from ipywidgets>=7.0.0->swifter[modin-ray]) (5.5.0)
Requirement already satisfied: traitlets>=4.3.1 in /usr/local/lib/python3.7/dist-packages (from ipywidgets>=7.0.0->swifter[modin-ray]) (5.1.1)
Requirement already satisfied: ipython-genutils~=0.2.0 in /usr/local/lib/python3.7/dist-packages (from ipywidgets>=7.0.0->swifter[modin-ray]) (0.2.0)
Requirement already satisfied: nbformat>=4.2.0 in /usr/local/lib/python3.7/dist-packages (from ipywidgets>=7.0.0->swifter[modin-ray]) (5.1.3)
Requirement already satisfied: jupyterlab-widgets>=1.0.0 in /usr/local/lib/python3.7/dist-packages (from ipywidgets>=7.0.0->swifter[modin-ray]) (1.0.2)
Requirement already satisfied: jupyter-client in /usr/local/lib/python3.7/dist-packages (from ipykernel>=4.5.1->ipywidgets>=7.0.0->swifter[modin-ray]) (5.3.5)
Requirement already satisfied: tornado>=4.0 in /usr/local/lib/python3.7/dist-packages (from ipykernel>=4.5.1->ipywidgets>=7.0.0->swifter[modin-ray]) (5.1.1)
Requirement already satisfied: setuptools>=18.5 in /usr/local/lib/python3.7/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter[modin-ray]) (57.4.0)
Requirement already satisfied: decorator in /usr/local/lib/python3.7/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter[modin-ray]) (4.4.2)
Requirement already satisfied: pygments in /usr/local/lib/python3.7/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter[modin-ray]) (2.6.1)
Requirement already satisfied: pickleshare in /usr/local/lib/python3.7/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter[modin-ray]) (0.7.5)
Requirement already satisfied: simplegeneric>0.8 in /usr/local/lib/python3.7/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter[modin-ray]) (0.8.1)
Requirement already satisfied: prompt-toolkit<2.0.0,>=1.0.4 in /usr/local/lib/python3.7/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter[modin-ray]) (1.0.18)
Requirement already satisfied: pexpect in /usr/local/lib/python3.7/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter[modin-ray]) (4.8.0)
Collecting pandas>=1.0.0
  Using cached pandas-1.2.4-cp37-cp37m-manylinux1_x86_64.whl (9.9 MB)
Requirement already satisfied: pyarrow>=1.0 in /usr/local/lib/python3.7/dist-packages (from modin[ray]>=0.8.1.1->swifter[modin-ray]) (3.0.0)
Requirement already satisfied: ray>=1.4.0 in /usr/local/lib/python3.7/dist-packages (from modin[ray]>=0.8.1.1->swifter[modin-ray]) (1.8.0)
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/dist-packages (from pandas>=1.0.0->swifter[modin-ray]) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (from pandas>=1.0.0->swifter[modin-ray]) (2018.9)
Requirement already satisfied: jupyter-core in /usr/local/lib/python3.7/dist-packages (from nbformat>=4.2.0->ipywidgets>=7.0.0->swifter[modin-ray]) (4.9.1)
Requirement already satisfied: jsonschema!=2.5.0,>=2.4 in /usr/local/lib/python3.7/dist-packages (from nbformat>=4.2.0->ipywidgets>=7.0.0->swifter[modin-ray]) (2.6.0)
Requirement already satisfied: locket in /usr/local/lib/python3.7/dist-packages (from partd>=0.3.10->dask[dataframe]>=2.10.0->swifter[modin-ray]) (0.2.1)
Requirement already satisfied: wcwidth in /usr/local/lib/python3.7/dist-packages (from prompt-toolkit<2.0.0,>=1.0.4->ipython>=4.0.0->ipywidgets>=7.0.0->swifter[modin-ray]) (0.2.5)
Requirement already satisfied: attrs in /usr/local/lib/python3.7/dist-packages (from ray>=1.4.0->modin[ray]>=0.8.1.1->swifter[modin-ray]) (21.2.0)
Requirement already satisfied: grpcio>=1.28.1 in /usr/local/lib/python3.7/dist-packages (from ray>=1.4.0->modin[ray]>=0.8.1.1->swifter[modin-ray]) (1.42.0)
Requirement already satisfied: redis>=3.5.0 in /usr/local/lib/python3.7/dist-packages (from ray>=1.4.0->modin[ray]>=0.8.1.1->swifter[modin-ray]) (4.0.2)
Requirement already satisfied: protobuf>=3.15.3 in /usr/local/lib/python3.7/dist-packages (from ray>=1.4.0->modin[ray]>=0.8.1.1->swifter[modin-ray]) (3.17.3)
Requirement already satisfied: click>=7.0 in /usr/local/lib/python3.7/dist-packages (from ray>=1.4.0->modin[ray]>=0.8.1.1->swifter[modin-ray]) (7.1.2)
Requirement already satisfied: pyyaml in /usr/local/lib/python3.7/dist-packages (from ray>=1.4.0->modin[ray]>=0.8.1.1->swifter[modin-ray]) (3.13)
Requirement already satisfied: msgpack<2.0.0,>=1.0.0 in /usr/local/lib/python3.7/dist-packages (from ray>=1.4.0->modin[ray]>=0.8.1.1->swifter[modin-ray]) (1.0.2)
Requirement already satisfied: filelock in /usr/local/lib/python3.7/dist-packages (from ray>=1.4.0->modin[ray]>=0.8.1.1->swifter[modin-ray]) (3.4.0)
Requirement already satisfied: deprecated in /usr/local/lib/python3.7/dist-packages (from redis>=3.5.0->ray>=1.4.0->modin[ray]>=0.8.1.1->swifter[modin-ray]) (1.2.13)
Requirement already satisfied: notebook>=4.4.1 in /usr/local/lib/python3.7/dist-packages (from widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter[modin-ray]) (5.3.1)
Requirement already satisfied: nbconvert in /usr/local/lib/python3.7/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter[modin-ray]) (5.6.1)
Requirement already satisfied: Send2Trash in /usr/local/lib/python3.7/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter[modin-ray]) (1.8.0)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.7/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter[modin-ray]) (2.11.3)
Requirement already satisfied: terminado>=0.8.1 in /usr/local/lib/python3.7/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter[modin-ray]) (0.12.1)
Requirement already satisfied: pyzmq>=13 in /usr/local/lib/python3.7/dist-packages (from jupyter-client->ipykernel>=4.5.1->ipywidgets>=7.0.0->swifter[modin-ray]) (22.3.0)
Requirement already satisfied: ptyprocess in /usr/local/lib/python3.7/dist-packages (from terminado>=0.8.1->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter[modin-ray]) (0.7.0)
Requirement already satisfied: wrapt<2,>=1.10 in /usr/local/lib/python3.7/dist-packages (from deprecated->redis>=3.5.0->ray>=1.4.0->modin[ray]>=0.8.1.1->swifter[modin-ray]) (1.13.3)
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.7/dist-packages (from jinja2->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter[modin-ray]) (2.0.1)
Requirement already satisfied: testpath in /usr/local/lib/python3.7/dist-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter[modin-ray]) (0.5.0)
Requirement already satisfied: entrypoints>=0.2.2 in /usr/local/lib/python3.7/dist-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter[modin-ray]) (0.3)
Requirement already satisfied: mistune<2,>=0.8.1 in /usr/local/lib/python3.7/dist-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter[modin-ray]) (0.8.4)
Requirement already satisfied: pandocfilters>=1.4.1 in /usr/local/lib/python3.7/dist-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter[modin-ray]) (1.5.0)
Requirement already satisfied: defusedxml in /usr/local/lib/python3.7/dist-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter[modin-ray]) (0.7.1)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.7/dist-packages (from packaging->bleach>=3.1.1->swifter[modin-ray]) (3.0.6)
Installing collected packages: pandas
  Attempting uninstall: pandas
    Found existing installation: pandas 1.3.4
    Uninstalling pandas-1.3.4:
      Successfully uninstalled pandas-1.3.4
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts.
google-colab 1.0.0 requires pandas~=1.1.0; python_version >= "3.0", but you have pandas 1.2.4 which is incompatible.
Successfully installed pandas-1.2.4

Requirement already satisfied: swifter[modin-dask] in /usr/local/lib/python3.7/dist-packages (1.0.9)
Requirement already satisfied: ipywidgets>=7.0.0 in /usr/local/lib/python3.7/dist-packages (from swifter[modin-dask]) (7.6.5)
Requirement already satisfied: psutil>=5.6.6 in /usr/local/lib/python3.7/dist-packages (from swifter[modin-dask]) (5.8.0)
Requirement already satisfied: cloudpickle>=0.2.2 in /usr/local/lib/python3.7/dist-packages (from swifter[modin-dask]) (1.3.0)
Requirement already satisfied: dask[dataframe]>=2.10.0 in /usr/local/lib/python3.7/dist-packages (from swifter[modin-dask]) (2.12.0)
Requirement already satisfied: bleach>=3.1.1 in /usr/local/lib/python3.7/dist-packages (from swifter[modin-dask]) (4.1.0)
Requirement already satisfied: parso>0.4.0 in /usr/local/lib/python3.7/dist-packages (from swifter[modin-dask]) (0.8.2)
Requirement already satisfied: pandas>=1.0.0 in /usr/local/lib/python3.7/dist-packages (from swifter[modin-dask]) (1.2.4)
Requirement already satisfied: tqdm>=4.33.0 in /usr/local/lib/python3.7/dist-packages (from swifter[modin-dask]) (4.62.3)
Requirement already satisfied: modin[dask]>=0.8.1.1 in /usr/local/lib/python3.7/dist-packages (from swifter[modin-dask]) (0.10.0)
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.7/dist-packages (from bleach>=3.1.1->swifter[modin-dask]) (1.15.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-packages (from bleach>=3.1.1->swifter[modin-dask]) (21.3)
Requirement already satisfied: webencodings in /usr/local/lib/python3.7/dist-packages (from bleach>=3.1.1->swifter[modin-dask]) (0.5.1)
Requirement already satisfied: toolz>=0.7.3 in /usr/local/lib/python3.7/dist-packages (from dask[dataframe]>=2.10.0->swifter[modin-dask]) (0.11.2)
Requirement already satisfied: numpy>=1.13.0 in /usr/local/lib/python3.7/dist-packages (from dask[dataframe]>=2.10.0->swifter[modin-dask]) (1.19.5)
Requirement already satisfied: fsspec>=0.6.0 in /usr/local/lib/python3.7/dist-packages (from dask[dataframe]>=2.10.0->swifter[modin-dask]) (2021.11.0)
Requirement already satisfied: partd>=0.3.10 in /usr/local/lib/python3.7/dist-packages (from dask[dataframe]>=2.10.0->swifter[modin-dask]) (1.2.0)
Requirement already satisfied: ipython-genutils~=0.2.0 in /usr/local/lib/python3.7/dist-packages (from ipywidgets>=7.0.0->swifter[modin-dask]) (0.2.0)
Requirement already satisfied: ipython>=4.0.0 in /usr/local/lib/python3.7/dist-packages (from ipywidgets>=7.0.0->swifter[modin-dask]) (5.5.0)
Requirement already satisfied: ipykernel>=4.5.1 in /usr/local/lib/python3.7/dist-packages (from ipywidgets>=7.0.0->swifter[modin-dask]) (4.10.1)
Requirement already satisfied: jupyterlab-widgets>=1.0.0 in /usr/local/lib/python3.7/dist-packages (from ipywidgets>=7.0.0->swifter[modin-dask]) (1.0.2)
Requirement already satisfied: nbformat>=4.2.0 in /usr/local/lib/python3.7/dist-packages (from ipywidgets>=7.0.0->swifter[modin-dask]) (5.1.3)
Requirement already satisfied: widgetsnbextension~=3.5.0 in /usr/local/lib/python3.7/dist-packages (from ipywidgets>=7.0.0->swifter[modin-dask]) (3.5.2)
Requirement already satisfied: traitlets>=4.3.1 in /usr/local/lib/python3.7/dist-packages (from ipywidgets>=7.0.0->swifter[modin-dask]) (5.1.1)
Requirement already satisfied: tornado>=4.0 in /usr/local/lib/python3.7/dist-packages (from ipykernel>=4.5.1->ipywidgets>=7.0.0->swifter[modin-dask]) (5.1.1)
Requirement already satisfied: jupyter-client in /usr/local/lib/python3.7/dist-packages (from ipykernel>=4.5.1->ipywidgets>=7.0.0->swifter[modin-dask]) (5.3.5)
Requirement already satisfied: setuptools>=18.5 in /usr/local/lib/python3.7/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter[modin-dask]) (57.4.0)
Requirement already satisfied: pexpect in /usr/local/lib/python3.7/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter[modin-dask]) (4.8.0)
Requirement already satisfied: pickleshare in /usr/local/lib/python3.7/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter[modin-dask]) (0.7.5)
Requirement already satisfied: pygments in /usr/local/lib/python3.7/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter[modin-dask]) (2.6.1)
Requirement already satisfied: prompt-toolkit<2.0.0,>=1.0.4 in /usr/local/lib/python3.7/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter[modin-dask]) (1.0.18)
Requirement already satisfied: simplegeneric>0.8 in /usr/local/lib/python3.7/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter[modin-dask]) (0.8.1)
Requirement already satisfied: decorator in /usr/local/lib/python3.7/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter[modin-dask]) (4.4.2)
Requirement already satisfied: distributed<=2.19.0,>=2.12.0 in /usr/local/lib/python3.7/dist-packages (from modin[dask]>=0.8.1.1->swifter[modin-dask]) (2.19.0)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (from pandas>=1.0.0->swifter[modin-dask]) (2018.9)
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/dist-packages (from pandas>=1.0.0->swifter[modin-dask]) (2.8.2)
Requirement already satisfied: zict>=0.1.3 in /usr/local/lib/python3.7/dist-packages (from distributed<=2.19.0,>=2.12.0->modin[dask]>=0.8.1.1->swifter[modin-dask]) (2.0.0)
Requirement already satisfied: msgpack>=0.6.0 in /usr/local/lib/python3.7/dist-packages (from distributed<=2.19.0,>=2.12.0->modin[dask]>=0.8.1.1->swifter[modin-dask]) (1.0.2)
Requirement already satisfied: sortedcontainers!=2.0.0,!=2.0.1 in /usr/local/lib/python3.7/dist-packages (from distributed<=2.19.0,>=2.12.0->modin[dask]>=0.8.1.1->swifter[modin-dask]) (2.4.0)
Requirement already satisfied: pyyaml in /usr/local/lib/python3.7/dist-packages (from distributed<=2.19.0,>=2.12.0->modin[dask]>=0.8.1.1->swifter[modin-dask]) (3.13)
Requirement already satisfied: click>=6.6 in /usr/local/lib/python3.7/dist-packages (from distributed<=2.19.0,>=2.12.0->modin[dask]>=0.8.1.1->swifter[modin-dask]) (7.1.2)
Requirement already satisfied: tblib>=1.6.0 in /usr/local/lib/python3.7/dist-packages (from distributed<=2.19.0,>=2.12.0->modin[dask]>=0.8.1.1->swifter[modin-dask]) (1.7.0)
Requirement already satisfied: jupyter-core in /usr/local/lib/python3.7/dist-packages (from nbformat>=4.2.0->ipywidgets>=7.0.0->swifter[modin-dask]) (4.9.1)
Requirement already satisfied: jsonschema!=2.5.0,>=2.4 in /usr/local/lib/python3.7/dist-packages (from nbformat>=4.2.0->ipywidgets>=7.0.0->swifter[modin-dask]) (2.6.0)
Requirement already satisfied: locket in /usr/local/lib/python3.7/dist-packages (from partd>=0.3.10->dask[dataframe]>=2.10.0->swifter[modin-dask]) (0.2.1)
Requirement already satisfied: wcwidth in /usr/local/lib/python3.7/dist-packages (from prompt-toolkit<2.0.0,>=1.0.4->ipython>=4.0.0->ipywidgets>=7.0.0->swifter[modin-dask]) (0.2.5)
Requirement already satisfied: notebook>=4.4.1 in /usr/local/lib/python3.7/dist-packages (from widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter[modin-dask]) (5.3.1)
Requirement already satisfied: nbconvert in /usr/local/lib/python3.7/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter[modin-dask]) (5.6.1)
Requirement already satisfied: terminado>=0.8.1 in /usr/local/lib/python3.7/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter[modin-dask]) (0.12.1)
Requirement already satisfied: Send2Trash in /usr/local/lib/python3.7/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter[modin-dask]) (1.8.0)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.7/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter[modin-dask]) (2.11.3)
Requirement already satisfied: pyzmq>=13 in /usr/local/lib/python3.7/dist-packages (from jupyter-client->ipykernel>=4.5.1->ipywidgets>=7.0.0->swifter[modin-dask]) (22.3.0)
Requirement already satisfied: ptyprocess in /usr/local/lib/python3.7/dist-packages (from terminado>=0.8.1->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter[modin-dask]) (0.7.0)
Requirement already satisfied: heapdict in /usr/local/lib/python3.7/dist-packages (from zict>=0.1.3->distributed<=2.19.0,>=2.12.0->modin[dask]>=0.8.1.1->swifter[modin-dask]) (1.0.1)
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.7/dist-packages (from jinja2->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter[modin-dask]) (2.0.1)
Requirement already satisfied: mistune<2,>=0.8.1 in /usr/local/lib/python3.7/dist-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter[modin-dask]) (0.8.4)
Requirement already satisfied: testpath in /usr/local/lib/python3.7/dist-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter[modin-dask]) (0.5.0)
Requirement already satisfied: defusedxml in /usr/local/lib/python3.7/dist-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter[modin-dask]) (0.7.1)
Requirement already satisfied: pandocfilters>=1.4.1 in /usr/local/lib/python3.7/dist-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter[modin-dask]) (1.5.0)
Requirement already satisfied: entrypoints>=0.2.2 in /usr/local/lib/python3.7/dist-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter[modin-dask]) (0.3)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.7/dist-packages (from packaging->bleach>=3.1.1->swifter[modin-dask]) (3.0.6)
Requirement already satisfied: swifter in /usr/local/lib/python3.7/dist-packages (1.0.9)
Requirement already satisfied: cloudpickle>=0.2.2 in /usr/local/lib/python3.7/dist-packages (from swifter) (1.3.0)
Requirement already satisfied: tqdm>=4.33.0 in /usr/local/lib/python3.7/dist-packages (from swifter) (4.62.3)
Requirement already satisfied: bleach>=3.1.1 in /usr/local/lib/python3.7/dist-packages (from swifter) (4.1.0)
Requirement already satisfied: dask[dataframe]>=2.10.0 in /usr/local/lib/python3.7/dist-packages (from swifter) (2.12.0)
Requirement already satisfied: ipywidgets>=7.0.0 in /usr/local/lib/python3.7/dist-packages (from swifter) (7.6.5)
Requirement already satisfied: pandas>=1.0.0 in /usr/local/lib/python3.7/dist-packages (from swifter) (1.2.4)
Requirement already satisfied: psutil>=5.6.6 in /usr/local/lib/python3.7/dist-packages (from swifter) (5.8.0)
Requirement already satisfied: parso>0.4.0 in /usr/local/lib/python3.7/dist-packages (from swifter) (0.8.2)
Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-packages (from bleach>=3.1.1->swifter) (21.3)
Requirement already satisfied: webencodings in /usr/local/lib/python3.7/dist-packages (from bleach>=3.1.1->swifter) (0.5.1)
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.7/dist-packages (from bleach>=3.1.1->swifter) (1.15.0)
Requirement already satisfied: numpy>=1.13.0 in /usr/local/lib/python3.7/dist-packages (from dask[dataframe]>=2.10.0->swifter) (1.19.5)
Requirement already satisfied: toolz>=0.7.3 in /usr/local/lib/python3.7/dist-packages (from dask[dataframe]>=2.10.0->swifter) (0.11.2)
Requirement already satisfied: fsspec>=0.6.0 in /usr/local/lib/python3.7/dist-packages (from dask[dataframe]>=2.10.0->swifter) (2021.11.0)
Requirement already satisfied: partd>=0.3.10 in /usr/local/lib/python3.7/dist-packages (from dask[dataframe]>=2.10.0->swifter) (1.2.0)
Requirement already satisfied: ipython>=4.0.0 in /usr/local/lib/python3.7/dist-packages (from ipywidgets>=7.0.0->swifter) (5.5.0)
Requirement already satisfied: nbformat>=4.2.0 in /usr/local/lib/python3.7/dist-packages (from ipywidgets>=7.0.0->swifter) (5.1.3)
Requirement already satisfied: jupyterlab-widgets>=1.0.0 in /usr/local/lib/python3.7/dist-packages (from ipywidgets>=7.0.0->swifter) (1.0.2)
Requirement already satisfied: traitlets>=4.3.1 in /usr/local/lib/python3.7/dist-packages (from ipywidgets>=7.0.0->swifter) (5.1.1)
Requirement already satisfied: ipykernel>=4.5.1 in /usr/local/lib/python3.7/dist-packages (from ipywidgets>=7.0.0->swifter) (4.10.1)
Requirement already satisfied: ipython-genutils~=0.2.0 in /usr/local/lib/python3.7/dist-packages (from ipywidgets>=7.0.0->swifter) (0.2.0)
Requirement already satisfied: widgetsnbextension~=3.5.0 in /usr/local/lib/python3.7/dist-packages (from ipywidgets>=7.0.0->swifter) (3.5.2)
Requirement already satisfied: tornado>=4.0 in /usr/local/lib/python3.7/dist-packages (from ipykernel>=4.5.1->ipywidgets>=7.0.0->swifter) (5.1.1)
Requirement already satisfied: jupyter-client in /usr/local/lib/python3.7/dist-packages (from ipykernel>=4.5.1->ipywidgets>=7.0.0->swifter) (5.3.5)
Requirement already satisfied: simplegeneric>0.8 in /usr/local/lib/python3.7/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter) (0.8.1)
Requirement already satisfied: pickleshare in /usr/local/lib/python3.7/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter) (0.7.5)
Requirement already satisfied: setuptools>=18.5 in /usr/local/lib/python3.7/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter) (57.4.0)
Requirement already satisfied: pexpect in /usr/local/lib/python3.7/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter) (4.8.0)
Requirement already satisfied: decorator in /usr/local/lib/python3.7/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter) (4.4.2)
Requirement already satisfied: pygments in /usr/local/lib/python3.7/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter) (2.6.1)
Requirement already satisfied: prompt-toolkit<2.0.0,>=1.0.4 in /usr/local/lib/python3.7/dist-packages (from ipython>=4.0.0->ipywidgets>=7.0.0->swifter) (1.0.18)
Requirement already satisfied: jupyter-core in /usr/local/lib/python3.7/dist-packages (from nbformat>=4.2.0->ipywidgets>=7.0.0->swifter) (4.9.1)
Requirement already satisfied: jsonschema!=2.5.0,>=2.4 in /usr/local/lib/python3.7/dist-packages (from nbformat>=4.2.0->ipywidgets>=7.0.0->swifter) (2.6.0)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (from pandas>=1.0.0->swifter) (2018.9)
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/dist-packages (from pandas>=1.0.0->swifter) (2.8.2)
Requirement already satisfied: locket in /usr/local/lib/python3.7/dist-packages (from partd>=0.3.10->dask[dataframe]>=2.10.0->swifter) (0.2.1)
Requirement already satisfied: wcwidth in /usr/local/lib/python3.7/dist-packages (from prompt-toolkit<2.0.0,>=1.0.4->ipython>=4.0.0->ipywidgets>=7.0.0->swifter) (0.2.5)
Requirement already satisfied: notebook>=4.4.1 in /usr/local/lib/python3.7/dist-packages (from widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter) (5.3.1)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.7/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter) (2.11.3)
Requirement already satisfied: nbconvert in /usr/local/lib/python3.7/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter) (5.6.1)
Requirement already satisfied: terminado>=0.8.1 in /usr/local/lib/python3.7/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter) (0.12.1)
Requirement already satisfied: Send2Trash in /usr/local/lib/python3.7/dist-packages (from notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter) (1.8.0)
Requirement already satisfied: pyzmq>=13 in /usr/local/lib/python3.7/dist-packages (from jupyter-client->ipykernel>=4.5.1->ipywidgets>=7.0.0->swifter) (22.3.0)
Requirement already satisfied: ptyprocess in /usr/local/lib/python3.7/dist-packages (from terminado>=0.8.1->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter) (0.7.0)
Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.7/dist-packages (from jinja2->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter) (2.0.1)
Requirement already satisfied: defusedxml in /usr/local/lib/python3.7/dist-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter) (0.7.1)
Requirement already satisfied: mistune<2,>=0.8.1 in /usr/local/lib/python3.7/dist-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter) (0.8.4)
Requirement already satisfied: pandocfilters>=1.4.1 in /usr/local/lib/python3.7/dist-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter) (1.5.0)
Requirement already satisfied: entrypoints>=0.2.2 in /usr/local/lib/python3.7/dist-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter) (0.3)
Requirement already satisfied: testpath in /usr/local/lib/python3.7/dist-packages (from nbconvert->notebook>=4.4.1->widgetsnbextension~=3.5.0->ipywidgets>=7.0.0->swifter) (0.5.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.7/dist-packages (from packaging->bleach>=3.1.1->swifter) (3.0.6)
```

In [ ]:

```
#@title
import pandas as pd

import nltk #for english sentece tokenization
from inltk.inltk import tokenize #for indian sentence tokenize

from nltk.tag import tnt# for hindi Parts Of Speech (POS) tagging
from nltk.corpus import indian

nltk.download('indian')
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')

train_data = indian.tagged_sents('hindi.pos')# we have to train before using pos for hindi using nltk
tnt_pos_tagger = tnt.TnT()
tnt_pos_tagger.train(train_data) #Training the tnt Part of speech tagger with hindi data

import spacy
nlp = spacy.load('en_core_web_sm')

#en_core_web_sm is a small English pipeline trained on written web text (blogs, news, comments), that incl

from nltk.stem import WordNetLemmatizer

#WordNetLemmatizer is used to perform Lemmatization

import swifter
```

```
[nltk_data] Downloading package indian to /root/nltk_data...
[nltk_data]   Package indian is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

PREPROCESSING STEPS TO BE TAKEN FOR THE NMT

A. Tokenization

The concept of tokenization is the act of taking a sequence of characters (think of Python strings) in a given document and dicing it up into its individual constituent pieces, which are the eponymous "tokens" of this method. One could loosely think of them as singular words in a sentence. One could naively implement the "split( )" method on a string which separates it into a python list based on the identifier in the argument. It is actually not that trivial to

Here we split the first sentence of the text in the training data just on a space as follows:

In [ ]:

```
#@title
# Storing the first text element as a string
first_text ="This process, however, afforded me no means of ascertaining the dimensions of my dungeon; as
print(first_text)
print("="*90)
print(first_text.split(" "))
```

```
This process, however, afforded me no means of ascertaining the dimensions of my dungeon; as I might make its circuit, and return to the point whence I set out, without being aware of the fact; so perfectly uniform seemed the wall.
==========================================================================================
['This', 'process,', 'however,', 'afforded', 'me', 'no', 'means', 'of', 'ascertaining', 'the', 'dimensions', 'of', 'my', 'dungeon;', 'as', 'I', 'might', 'make', 'its', 'circuit,', 'and', 'return', 'to', 'the', 'point', 'whence', 'I', 'set', 'out,', 'without', 'being', 'aware', 'of', 'the', 'fact;', 'so', 'perfectly', 'uniform', 'seemed', 'the', 'wall.']
```

However as you can see from this first attempt at tokenization, the segregation of the sentence into its individual elements (or terms) is not entirely accurate. As an example, look at the second element of the list which contains the term "process,". The punctuation mark (comma) has also been included and is being treated along with the word "process" as a term in itself. Ideally we would like the comma and the word to be in two different and separate elements of the list. Trying to do this with pure python list operations will be quite complex so this is where the NLTK library comes into play. There is a convenient method "word_tokenize( )" (TreebankWord tokenizer) which strips out singular words as well as punctuations into separate elements automatically as follows:

In [ ]:

```
#@title
first_text_list = nltk.word_tokenize(first_text)
print(first_text_list)
```

```
['This', 'process', ',', 'however', ',', 'afforded', 'me', 'no', 'means', 'of', 'ascertaining', 'the', 'dimensions', 'of', 'my', 'dungeon', ';', 'as', 'I', 'might', 'make', 'its', 'circuit', ',', 'and', 'return', 'to', 'the', 'point', 'whence', 'I', 'set', 'out', ',', 'without', 'being', 'aware', 'of', 'the', 'fact', ';', 'so', 'perfectly', 'uniform', 'seemed', 'the', 'wall', '.']
```

B. Stopword Removal

As alluded to above stopwords are generally words that appear so commonly and at such a high frequency in the corpus that they don't actually contribute much to the learning or predictive process as a learning model would fail to distinguish it from other texts. Stopwordsinclude terms such as "to" or "the" and therefore, it would be to our benefit to remove them during the pre-processing phase. Conveniently, NLTK comes with a predefined list of 153 english stopwords. (unhide the second cell below to see this list)

In [ ]:

```
#@title
stopwords = nltk.corpus.stopwords.words('english')
len(stopwords)
```

Out[ ]: 179

In [ ]:

```
#@title
print(stopwords)
```

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', "that'll", 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "should've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "shouldn't", 'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't"]
```

To filter out stop words from our tokenized list of words, we can simply use a list comprehension as follows:

In [ ]:

```
#@title
first_text_list_cleaned = [word for word in first_text_list if word.lower() not in stopwords]
print(first_text_list_cleaned)
print("="*90)
print("Length of original list: {0} words\n"
      "Length of list after stopwords removal: {1} words"
      .format(len(first_text_list), len(first_text_list_cleaned)))
```

```
['process', ',', 'however', ',', 'afforded', 'means', 'ascertaining', 'dimensions', 'dungeon', ';', 'might', 'make', 'circuit', ',', 'return', 'point', 'whence', 'set', ',', 'without', 'aware', 'fact', ';', 'perfectly', 'uniform', 'seemed', 'wall', '.']
==========================================================================================
Length of original list: 48 words
Length of list after stopwords removal: 28 words
```

As you can see, our list with stopwords removed is now substantially shorter than before where commonly occuring words such as "I", "me", "to" and "the" have been removed.

Note: Here is the list of hindi stop words available

In [ ]:

```
#@title
hindi_stop=open('/content/drive/MyDrive/Colab Notebooks/HINDI-STOP-WORDS/hindi_stop.txt')

hi_stopwords=[]

for i in hindi_stop:
    hi_stopwords.append(i.rstrip('\n'))
else:
    print("no of hindi stopqwords:",len(hi_stopwords))
    print()
    print(hi_stopwords)
```

```
no of hindi stopqwords: 148

['पर', 'इन', 'वह', 'यिह', 'वुह', 'जिन्हें', 'जिन्हों', 'किन्हों', 'किन्हें', 'इत्यादि', 'द्वारा', 'इन्हें', 'इन्हों', 'उन्हों', 'बिलकुल', 'इन्हीं', 'उन्हीं', 'उन्हें', 'इसमें', 'जितना', 'दुसरा', 'कितना', 'कितने', 'कितनो', 'वग़ैरह', 'दूसरे', 'कौनसा', 'लेकिन', 'होता', 'करने', 'किया', 'लिये', 'अपने', 'नहीं', 'दिया', 'इसका', 'करना', 'वाले', 'सकते', 'इसके', 'सबसे', 'होने', 'करते', 'बहुत', 'करें', 'होती', 'अपनी', 'उनके', 'कहते', 'होते', 'करता', 'उनकी', 'इसकी', 'सकता', 'रखें', 'अपना', 'उसके', 'जिसे', 'किसे', 'किसी', 'काफ़ी', 'पहले', 'नीचे', 'यहाँ', 'जैसा', 'जैसे', 'मानो', 'अंदर', 'भीतर', 'पूरा', 'सारा', 'होना', 'उनको', 'वहाँ', 'वहीं', 'जहाँ', 'जीधर', 'उनका', 'इनका', '\ufeffके', 'हैं', 'गया', 'बनी', 'एवं', 'हुआ', 'साथ', 'बाद', 'लिए', 'कुछ', 'कहा', 'यदि', 'हुई', 'इसे', 'हुए', 'अभी', 'सभी', 'कुल', 'रहा', 'रहे', 'इसी', 'उसे', 'जिस', 'जिन', 'कौन', 'किस', 'कोई', 'ऐसे', 'तरह', 'संग', 'यही', 'उसी', 'फिर', 'मगर', 'का', 'एक', 'यह', 'से', 'को', 'इस', 'कि', 'जो', 'कर', 'मे', 'ने', 'तो', 'ही', 'या', 'हो', 'था', 'तक', 'आप', 'ये', 'थे', 'दो', 'वे', 'थी', 'जा', 'ना', 'उस', 'पे', 'उन', 'भी', 'और', 'घर', 'तब', 'जब', 'व', 'न']
```

C. Parts Of Speech (POS) tagging

It is a process of converting a sentence to forms – list of words, list of tuples (where each tuple is having a form (word, tag)). The tag in case of is a part-of-speech tag, and signifies whether the word is a noun, adjective, verb, and so on.

The extraction of a single word entity from a sentence is not a tough task. We can easily do this with the help of parts of speech (POS) tags. The nouns and the proper nouns would be our entities.

Once we have captured the subject and the object in the sentence, we will update the previous token and its dependency tag.

POS tag notation

Number Tag Description

1. CC Coordinating conjunction
2. CD Cardinal number
3. DT Determiner
4. EX Existential there
5. FW Foreign word
6. IN Preposition or subordinating conjunction
7. JJ Adjective
8. JJR Adjective, comparative
9. JJS Adjective, superlative
10. LS List item marker
11. MD Modal
12. NN Noun, singular or mass
13. NNS Noun, plural
14. NNP Proper noun, singular
15. NNPS Proper noun, plural
16. PDT Predeterminer
17. POS Possessive ending
18. PRP Personal pronoun
19. PRP Possessive pronoun
20. RB Adverb
21. RBR Adverb, comparative
22. RBS Adverb, superlative
23. RP Particle
24. SYM Symbol
25. TO to
26. UH Interjection
27. VB Verb, base form
28. VBD Verb, past tense
29. VBG Verb, gerund or present participle
30. VBN Verb, past participle
31. VBP Verb, non-3rd person singular present
32. VBZ Verb, 3rd person singular present
33. WDT Wh-determiner
34. WP Wh-pronoun
35. WP Possessive wh-pronoun
36. WRB Wh-adverb

Let's test this function on a sentence:

```
#@title
def pos(sent):
  ## chunk 1
  ent1 = ""
  ent2 = ""

  prv_tok_dep = ""    # dependency tag of previous token in the sentence
  prv_tok_text = ""   # previous token in the sentence

  prefix = ""
  modifier = ""

  #############################################################

  for tok in nlp(sent):
    ## chunk 2
    # if token is a punctuation mark then move on to the next token
    if tok.dep_ != "punct":
      # check: token is a compound word or not
      if tok.dep_ == "compound":
        prefix = tok.text
        # if the previous word was also a 'compound' then add the current word to it
        if prv_tok_dep == "compound":
          prefix = prv_tok_text + " "+ tok.text

      # check: token is a modifier or not
      if tok.dep_.endswith("mod") == True:
        modifier = tok.text
        # if the previous word was also a 'compound' then add the current word to it
        if prv_tok_dep == "compound":
          modifier = prv_tok_text + " "+ tok.text

      ## chunk 3
      if tok.dep_.find("subj") == True:
        ent1 = modifier +" "+ prefix + " "+ tok.text
        prefix = ""
        modifier = ""
        prv_tok_dep = ""
        prv_tok_text = ""

      ## chunk 4
      if tok.dep_.find("obj") == True:
        ent2 = modifier +" "+ prefix +" "+ tok.text

      ## chunk 5
      # update variables
      prv_tok_dep = tok.dep_
      prv_tok_text = tok.text
  #############################################################

  return [ent1.strip(), ent2.strip()]
```

Chunk 1

Defined a few empty variables in this chunk. prv_tok_dep and prv_tok_text will hold the dependency tag of the previous word in the sentence and that previous word itself, respectively. prefix and modifier will hold the text that is associated with the subject or the object.

Chunk 2

Next, we will loop through the tokens in the sentence. We will first check if the token is a punctuation mark or not. If yes, then we will ignore it and move on to the next token. If the token is a part of a compound word (dependency tag = "compound"), we will keep it in the prefix variable. A compound word is a combination of multiple words linked to form a word with a new meaning (example – "Football Stadium", "animal lover").

As and when we come across a subject or an object in the sentence, we will add this prefix to it. We will do the same thing with the modifier words, such as "nice shirt", "big house", etc.

Chunk 3

Here, if the token is the subject, then it will be captured as the first entity in the ent1 variable. Variables such as prefix, modifier, prv_tok_dep, and prv_tok_text will be reset.

Chunk 4

Here, if the token is the object, then it will be captured as the second entity in the ent2 variable. Variables such as prefix, modifier, prv_tok_dep, and prv_tok_text will again be reset.

Chunk 5

Once we have captured the subject and the object in the sentence, we will update the previous token and its dependency tag.

Let's test this function on a sentence:

```
#@title
pos("the film had 200 patents")
```

```
['film', '200  patents']
```

```
#@title
# hindi pos

tnt_pos_tagger.tag(nltk.word_tokenize("मैं तंत्रिका मशीन अनुवाद (NMT) सीख रहा हूँ"))
```

```
[('मैं', 'PRP'),
 ('तंत्रिका', 'Unk'),
 ('मशीन', 'Unk'),
 ('अनुवाद', 'Unk'),
 ('(', 'SYM'),
 ('NMT', 'Unk'),
 (')', 'SYM'),
 ('सीख', 'Unk'),
 ('रहा', 'VAUX'),
 ('हूँ', 'Unk')]
```

APPLYING ALL THE ABOVE STEPS COMBINED ON THE DATASET

READING THE DATASET

```
#@title
# reading the dataset

#english corpus
eng=pd.read_csv('/content/drive/MyDrive/Colab Notebooks/PARALLEL-CORPUS (TRAIN)/IITB.en-hi.en.csv')

#hindi corpus
hi=pd.read_csv('/content/drive/MyDrive/Colab Notebooks/PARALLEL-CORPUS (TRAIN)/IITB.en-hi.hi.csv')
```

FORMATTING THE DATASET

```
#@title
# COMBINING BOTH THE LANGUAGE PAIRS TO FORM PARALLEL CORPORUS
df=pd.concat([eng,hi],axis=1)
```

FIRST FEW ELEMENTS

```
#@title
df.head()
```

	ENGLISH CORPUS	HINDI CORPUS
0	Give your application an accessibility workout	अपने अनुप्रयोग को पहुंचनीयता व्यायाम का लाभ दें
1	Accerciser Accessibility Explorer	एक्सेर्साइसर पहुंचनीयता अन्वेषक
2	The default plugin layout for the bottom panel	निचले पटल के लिए डिफोल्ट प्लग-इन खाका
3	The default plugin layout for the top panel	ऊपरी पटल के लिए डिफोल्ट प्लग-इन खाका
4	A list of plugins that are disabled by default	उन प्लग-इनों की सूची जिन्हें डिफोल्ट रूप से नि...

PREPROCESSING THE DATASET

1. Tokenization

```
#@title
# uncomment below code & do this for the firsat time to install required packages fort hindi tokenization
from inltk.inltk import setup
setup("hi")
```

```
---------------------------------------------------------------------------
RuntimeError                              Traceback (most recent call last)
<ipython-input-14-619dd1fd2b72> in <module>()
      2 # uncomment below code & do this for the firsat time to install required packages fort hindi token
ization
      3 from inltk.inltk import setup
----> 4 setup("hi")

/usr/local/lib/python3.7/dist-packages/inltk/inltk.py in setup(language_code)
     31     loop = asyncio.get_event_loop()
     32     tasks = [asyncio.ensure_future(download(language_code))]
---> 33     learn = loop.run_until_complete(asyncio.gather(*tasks))[0]
     34     loop.close()
     35

/usr/lib/python3.7/asyncio/base_events.py in run_until_complete(self, future)
    561         """
    562         self._check_closed()
--> 563         self._check_runnung()
    564
    565         new_task = not futures.isfuture(future)

/usr/lib/python3.7/asyncio/base_events.py in _check_runnung(self)
    521     def _check_runnung(self):
    522         if self.is_running():
--> 523             raise RuntimeError('This event loop is already running')
    524         if events._get_running_loop() is not None:
    525             raise RuntimeError(

RuntimeError: This event loop is already running
Done!
```

```
#@title
def hi_tokenize(sent):
    return sent.split()
```

```
#@title
#performing tokenization

# english corpus

df['ENGLISH TOKENS'] = df.swifter.apply(lambda row: nltk.word_tokenize(row['ENGLISH CORPUS']), axis=1)

# hindi corpus

# uncomment beolw to also perform hindi word tokenization usig inltk

#df['HINDI_TOKENS'] = df.swifter.apply(lambda row: tokenize(row['HINDI CORPUS'],"hi"), axis=1)

df['HINDI_TOKENS'] = df.swifter.apply(lambda row: hi_tokenize(row['HINDI CORPUS']), axis=1)
```

```
#@title
# after tokenization

df.head()
```

	ENGLISH CORPUS	HINDI CORPUS	ENGLISH TOKENS	HINDI_TOKENS
0	Give your application an accessibility workout	अपने अनुप्रयोग को पहुंचनीयता व्यायाम का लाभ दें	[Give, your, application, an, accessibility, w...	[अपने, अनुप्रयोग, को, पहुंचनीयता, व्यायाम, का,...
1	Accerciser Accessibility Explorer	एक्सेर्साइसर पहुंचनीयता अन्वेषक	[Accerciser, Accessibility, Explorer]	[एक्सेर्साइसर, पहुंचनीयता, अन्वेषक]
2	The default plugin layout for the bottom panel	निचले पटल के लिए डिफोल्ट प्लग-इन खाका	[The, default, plugin, layout, for, the, botto...	[निचले, पटल, के, लिए, डिफोल्ट, प्लग-इन, खाका]
3	The default plugin layout for the top panel	ऊपरी पटल के लिए डिफोल्ट प्लग-इन खाका	[The, default, plugin, layout, for, the, top, ...	[ऊपरी, पटल, के, लिए, डिफोल्ट, प्लग-इन, खाका]
4	A list of plugins that are disabled by default	उन प्लग-इनों की सूची जिन्हें डिफोल्ट रूप से नि...	[A, list, of, plugins, that, are, disabled, by...	[उन, प्लग-इनों, की, सूची, जिन्हें, डिफोल्ट, रू...

```
#@title
#finding length of each sentence

df['ENGLISH SENTENCE LENGTH'] = df.swifter.apply(lambda row: len(row['ENGLISH TOKENS']), axis=1)

df['HINDI SENTENCE LENGTH'] = df.swifter.apply(lambda row: len(row['HINDI_TOKENS']), axis=1)

df.head()
```

	ENGLISH CORPUS	HINDI CORPUS	ENGLISH TOKENS	HINDI_TOKENS	ENGLISH SENTENCE LENGTH	HINDI SENTENCE LENGTH
0	Give your application an accessibility workout	अपने अनुप्रयोग को पहुंचनीयता व्यायाम का लाभ दें	[Give, your, application, an, accessibility, w...	[अपने, अनुप्रयोग, को, पहुंचनीयता, व्यायाम, का,...	6	8
1	Accerciser Accessibility Explorer	एक्सेर्साइसर पहुंचनीयता अन्वेषक	[Accerciser, Accessibility, Explorer]	[एक्सेर्साइसर, पहुंचनीयता, अन्वेषक]	3	3
2	The default plugin layout for the bottom panel	निचले पटल के लिए डिफोल्ट प्लग-इन खाका	[The, default, plugin, layout, for, the, botto...	[निचले, पटल, के, लिए, डिफोल्ट, प्लग-इन, खाका]	8	7
3	The default plugin layout for the top panel	ऊपरी पटल के लिए डिफोल्ट प्लग-इन खाका	[The, default, plugin, layout, for, the, top, ...	[ऊपरी, पटल, के, लिए, डिफोल्ट, प्लग-इन, खाका]	8	7
4	A list of plugins that are disabled by default	उन प्लग-इनों की सूची जिन्हें डिफोल्ट रूप से नि...	[A, list, of, plugins, that, are, disabled, by...	[उन, प्लग-इनों, की, सूची, जिन्हें, डिफोल्ट, रू...	9	12

2. Stopword Removal

```
#@title
# English stop word list

eng_stopwords = nltk.corpus.stopwords.words('english')

eng_stopwords
```

```
['i',
 'me',
 'my',
 'myself',
 'we',
 'our',
 'ours',
 'ourselves',
 'you',
 "you're",
 "you've",
 "you'll",
 "you'd",
 'your',
 'yours',
 'yourself',
 'yourselves',
 'he',
 'him',
 'his',
 'himself',
 'she',
 "she's",
 'her',
 'hers',
 'herself',
 'it',
 "it's",
 'its',
 'itself',
 'they',
 'them',
 'their',
 'theirs',
 'themselves',
 'what',
 'which',
 'who',
 'whom',
 'this',
 'that',
 "that'll",
 'these',
 'those',
 'am',
 'is',
 'are',
 'was',
 'were',
 'be',
 'been',
 'being',
 'have',
 'has',
 'had',
 'having',
 'do',
 'does',
 'did',
 'doing',
 'a',
 'an',
 'the',
 'and',
 'but',
 'if',
 'or',
 'because',
 'as',
 'until',
 'while',
 'of',
 'at',
 'by',
 'for',
 'with',
 'about',
 'against',
 'between',
 'into',
 'through',
 'during',
 'before',
 'after',
 'above',
 'below',
 'to',
 'from',
 'up',
 'down',
 'in',
 'out',
 'on',
 'off',
 'over',
 'under',
 'again',
 'further',
 'then',
 'once',
 'here',
 'there',
 'when',
 'where',
 'why',
 'how',
 'all',
 'any',
 'both',
 'each',
 'few',
 'more',
 'most',
 'other',
 'some',
 'such',
 'no',
 'nor',
 'not',
 'only',
 'own',
 'same',
 'so',
 'than',
 'too',
 'very',
 's',
 't',
 'can',
 'will',
 'just',
 'don',
 "don't",
 'should',
 "should've",
 'now',
 'd',
 'll',
 'm',
 'o',
 're',
 've',
 'y',
 'ain',
 'aren',
 "aren't",
 'couldn',
 "couldn't",
 'didn',
 "didn't",
 'doesn',
 "doesn't",
 'hadn',
 "hadn't",
 'hasn',
 "hasn't",
 'haven',
 "haven't",
 'isn',
 "isn't",
 'ma',
 'mightn',
 "mightn't",
 'mustn',
 "mustn't",
 'needn',
 "needn't",
 'shan',
 "shan't",
 'shouldn',
 "shouldn't",
 'wasn',
 "wasn't",
 'weren',
 "weren't",
 'won',
 "won't",
 'wouldn',
 "wouldn't"]
```

```
#@title
# english stopword removal

df["ENGLISH STOP WORD REMOVED"]=df["ENGLISH TOKENS"].swifter.apply(lambda x:[word for word in x if word.lo

# uncomment beolw to also perform hindi word tokenization usig inltk

# english stopword removal

# df["HINDI STOP WORD REMOVED"]=df["HINDI TOKENS"].swifter.apply(lambda x:[word for word in x if word.lowe


df.head()
```

	ENGLISH CORPUS	HINDI CORPUS	ENGLISH TOKENS	HINDI_TOKENS	ENGLISH SENTENCE LENGTH	HINDI SENTENCE LENGTH	ENGLISH STOP WORD REMOVED
0	Give your application an accessibility workout	अपने अनुप्रयोग को पहुंचनीयता व्यायाम का लाभ दें	[Give, your, application, an, accessibility, w...	[अपने, अनुप्रयोग, को, पहुंचनीयता, व्यायाम, का,...	6	8	[Give, application, accessibility, workout]
1	Accerciser Accessibility Explorer	एक्सेर्साइसर पहुंचनीयता अन्वेषक	[Accerciser, Accessibility, Explorer]	[एक्सेर्साइसर, पहुंचनीयता, अन्वेषक]	3	3	[Accerciser, Accessibility, Explorer]
2	The default plugin layout for the bottom panel	निचले पटल के लिए डिफोल्ट प्लग-इन खाका	[The, default, plugin, layout, for, the, botto...	[निचले, पटल, के, लिए, डिफोल्ट, प्लग-इन, खाका]	8	7	[default, plugin, layout, bottom, panel]
3	The default plugin layout for the top panel	ऊपरी पटल के लिए डिफोल्ट प्लग-इन खाका	[The, default, plugin, layout, for, the, top, ...	[ऊपरी, पटल, के, लिए, डिफोल्ट, प्लग-इन, खाका]	8	7	[default, plugin, layout, top, panel]
4	A list of plugins that are disabled by default	उन प्लग-इनों की सूची जिन्हें डिफोल्ट रूप से नि...	[A, list, of, plugins, that, are, disabled, by...	[उन, प्लग-इनों, की, सूची, जिन्हें, डिफोल्ट, रू...	9	12	[list, plugins, disabled, default]

```
#@title
# updating the corpus after removing the stopwords i.e. making sentece using the stop words removed  colum

# english corpus updated

for i in range(len(df)):

    stop_word_removed_sentence=""

    for word in df["ENGLISH STOP WORD REMOVED"][i]:

        stop_word_removed_sentence+=word+" "

    else:

        df["ENGLISH CORPUS"][i]=stop_word_removed_sentence.rstrip(" ")

# uncomment beolw to also perform hindi sentences update after stop word removed

# hindi corpus updated

#for i in range(len(df)):

 #   stop_word_removed_sentence=""

  #  for word in df["HINDI STOP WORD REMOVED"][i]:

   #     stop_word_removed_sentence+=word+" "

    #else:

     #   df["HINDI CORPUS"][i]=stop_word_removed_sentence.rstrip(" ")

df.head()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:16: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.htm
l#returning-a-view-versus-a-copy
  app.launch_new_instance()
```

	ENGLISH CORPUS	HINDI CORPUS	ENGLISH TOKENS	HINDI_TOKENS	ENGLISH SENTENCE LENGTH	HINDI SENTENCE LENGTH	ENGLISH STOP WORD REMOVED
0	Give application accessibility workout	अपने अनुप्रयोग को पहुंचनीयता व्यायाम का लाभ दें	[Give, your, application, an, accessibility, w...	[अपने, अनुप्रयोग, को, पहुंचनीयता, व्यायाम, का,...	6	8	[Give, application, accessibility, workout]
1	Accerciser Accessibility Explorer	एक्सेर्साइसर पहुंचनीयता अन्वेषक	[Accerciser, Accessibility, Explorer]	[एक्सेर्साइसर, पहुंचनीयता, अन्वेषक]	3	3	[Accerciser, Accessibility, Explorer]
2	default plugin layout bottom panel	निचले पटल के लिए डिफोल्ट प्लग-इन खाका	[The, default, plugin, layout, for, the, botto...	[निचले, पटल, के, लिए, डिफोल्ट, प्लग-इन, खाका]	8	7	[default, plugin, layout, bottom, panel]
3	default plugin layout top panel	ऊपरी पटल के लिए डिफोल्ट प्लग-इन खाका	[The, default, plugin, layout, for, the, top, ...	[ऊपरी, पटल, के, लिए, डिफोल्ट, प्लग-इन, खाका]	8	7	[default, plugin, layout, top, panel]
4	list plugins disabled default	उन प्लग-इनों की सूची जिन्हें डिफोल्ट रूप से नि...	[A, list, of, plugins, that, are, disabled, by...	[उन, प्लग-इनों, की, सूची, जिन्हें, डिफोल्ट, रू...	9	12	[list, plugins, disabled, default]

```
#@title
# length after stop word removed

df['SENTENCE LENGTH AFTER STOP WORD REMOVAL'] = df.swifter.apply(lambda row: len(row['ENGLISH STOP WORD RE

df.head()
```

	ENGLISH CORPUS	HINDI CORPUS	ENGLISH TOKENS	HINDI_TOKENS	ENGLISH SENTENCE LENGTH	HINDI SENTENCE LENGTH	ENGLISH STOP WORD REMOVED	SENTENCE LENGTH AFTER STOP WORD REMOVAL
0	Give application accessibility workout	अपने अनुप्रयोग को पहुंचनीयता व्यायाम का लाभ दें	[Give, your, application, an, accessibility, w...	[अपने, अनुप्रयोग, को, पहुंचनीयता, व्यायाम, का,...	6	8	[Give, application, accessibility, workout]	4
1	Accerciser Accessibility Explorer	एक्सेर्साइसर पहुंचनीयता अन्वेषक	[Accerciser, Accessibility, Explorer]	[एक्सेर्साइसर, पहुंचनीयता, अन्वेषक]	3	3	[Accerciser, Accessibility, Explorer]	3
2	default plugin layout bottom panel	निचले पटल के लिए डिफोल्ट प्लग-इन खाका	[The, default, plugin, layout, for, the, botto...	[निचले, पटल, के, लिए, डिफोल्ट, प्लग-इन, खाका]	8	7	[default, plugin, layout, bottom, panel]	5
3	default plugin layout top panel	ऊपरी पटल के लिए डिफोल्ट प्लग-इन खाका	[The, default, plugin, layout, for, the, top, ...	[ऊपरी, पटल, के, लिए, डिफोल्ट, प्लग-इन, खाका]	8	7	[default, plugin, layout, top, panel]	5
4	list plugins disabled default	उन प्लग-इनों की सूची जिन्हें डिफोल्ट रूप से नि...	[A, list, of, plugins, that, are, disabled, by...	[उन, प्लग-इनों, की, सूची, जिन्हें, डिफोल्ट, रू...	9	12	[list, plugins, disabled, default]	4

3. Parts Of Speech (POS) tagging

```
#@title
# english pos tagging

df["ENGLISH POS TAGGING"]=df.swifter.apply(lambda row: pos(row["ENGLISH CORPUS"]),axis=1)

# uncomment beolw to also perform hindi word tokenization usig inltk

# hindi stopword removal

# df["HINDI POS TAGGING"]=df.swifter.apply(lambda row: tnt_pos_tagger.tag(row['HINDI STOP WORD REMOVED']),


df.head()
```

	ENGLISH CORPUS	HINDI CORPUS	ENGLISH TOKENS	HINDI_TOKENS	ENGLISH SENTENCE LENGTH	HINDI SENTENCE LENGTH	ENGLISH STOP WORD REMOVED	SENTENCE LENGTH AFTER STOP WORD REMOVAL	ENGLISH POS TAGGING
0	Give application accessibility workout	अपने अनुप्रयोग को पहुंचनीयता व्यायाम का लाभ दें	[Give, your, application, an, accessibility, w...	[अपने, अनुप्रयोग, को, पहुंचनीयता, व्यायाम, का,...	6	8	[Give, application, accessibility, workout]	4	[, application accessibility workout]
1	Accerciser Accessibility Explorer	एक्सेर्साइसर पहुंचनीयता अन्वेषक	[Accerciser, Accessibility, Explorer]	[एक्सेर्साइसर, पहुंचनीयता, अन्वेषक]	3	3	[Accerciser, Accessibility, Explorer]	3	[,]
2	default plugin layout bottom panel	निचले पटल के लिए डिफोल्ट प्लग-इन खाका	[The, default, plugin, layout, for, the, botto...	[निचले, पटल, के, लिए, डिफोल्ट, प्लग-इन, खाका]	8	7	[default, plugin, layout, bottom, panel]	5	[,]
3	default plugin layout top panel	ऊपरी पटल के लिए डिफोल्ट प्लग-इन खाका	[The, default, plugin, layout, for, the, top, ...	[ऊपरी, पटल, के, लिए, डिफोल्ट, प्लग-इन, खाका]	8	7	[default, plugin, layout, top, panel]	5	[default plugin, top panel]
4	list plugins disabled default	उन प्लग-इनों की सूची जिन्हें डिफोल्ट रूप से नि...	[A, list, of, plugins, that, are, disabled, by...	[उन, प्लग-इनों, की, सूची, जिन्हें, डिफोल्ट, रू...	9	12	[list, plugins, disabled, default]	4	[list, disabled default]

D. Stemming and Lemmatization

After removal of stopwords, the next stage of NLP that I would like to introduce is the process of Stemming. The work at this stage attempts to reduce as many different variations of similar words into a single term (different branches all reduced to single word *stem*). Therefore if we have "running", "runs" and "run", you would really want these three distinct words to collapse into just the word "run". (However of course you lose granularity of the past, present or future tense).

We can turn to NLTK again which provides various stemmers which include variants such as the Porter stemming algorithm. A porter stemmer instance as follows:

```
#@title
# for english stemming

stemmer = nltk.stem.PorterStemmer()

def eng_stem(sent):
    s=""
    for word in [stemmer.stem(word) for word in sent.split()]:
        s+=word+" "

    return s.rstrip(" ")

# for hindi stemming

suffixes = {
    1: ["ो", "े", "ू", "ु", "ी", "ि", "ा"],
    2: ["कर", "ाओ", "िए", "ाई", "ाए", "ने", "नी", "ना", "ते", "ीं", "ती", "ता", "ाँ", "ां", "ों", "ें"],
    3: ["ाकर", "ाइए", "ाईं", "ाया", "ेगी", "ेगा", "ोगी", "ोगे", "ाने", "ाना", "ाते", "ाती", "ाता", "तीं", "ाओं", "ाएं
    4: ["ाएगी", "ाएगा", "ाओगी", "ाओगे", "एंगी", "ेंगी", "एंगे", "ेंगे", "ूंगी", "ूंगा", "ातीं", "नाओं", "नाएं", "ताओं", "ताएं",
    5: ["ाएंगी", "ाएंगे", "ाऊंगी", "ाऊंगा", "ाइयाँ", "ाइयों", "ाइयां"],
}

def hi_stem_call(word):
    for L in 5, 4, 3, 2, 1:
        if len(word) > L + 1:
            for suf in suffixes[L]:
                if word.endswith(suf):
                    return word[:-L]
    return word

def hi_stem(sent):
    s=""
    for word in [hi_stem_call(word) for word in sent.split()]:
        s+=word+" "

    return s.rstrip(" ")
```

And now we can use stemmer to see if it can reduce our these test words ("running", "runs", "run") into their a single stemmed word as follows:

```
#@title
#english stemming

print("The stemmed form of running is:",stemmer.stem("running"))
print("The stemmed form of running is:",stemmer.stem("runs"))
print("The stemmed form of running is:",stemmer.stem("run"))

# hindi stemming

hi_stem("ख़रीदारों के लिए मार्ग दर्शिका")
```

```
The stemmed form of running is: run
The stemmed form of running is: run
The stemmed form of running is: run
```

```
'ख़रीदार के लिए मार्ग दर्शिक'
```

As we can see, the stemmer has successfully reduced the given words above into a base form.

However there is one flaw with stemming and that this process does not take into account vocabulary or word forms when collapsing words as this example will illustrate:

```
#@title
print("The stemmed form of leaves is: {}".format(stemmer.stem("leaves")))
```

```
The stemmed form of leaves is: leav
```

=> Lemmatization to the rescue

Therefore we turn to another that we could use in place of stemming. This method is called lemmatization which aims to achieve the same effect as the former method. However unlike a stemmer, lemmatizing the dataset aims to reduce words based on an actual dictionary or vocabulary (the Lemma) and therefore will not chop off words into stemmed forms that do not carry any lexical meaning.

Lemmatization is similar to stemming but it brings context to the words. Here we can use NLTK once again to initialize a lemmatizer (WordNet variant) and inspect how it collapses words as follows:

```
#@title
# for english lemmatization

lemm = WordNetLemmatizer()

def eng_lemma(sent):
    s=""
    for word in [lemm.lemmatize(word) for word in sent.split()]:
        s+=word+" "

    return s.rstrip(" ")
```

```
#@title
print("The lemmatized form of leaves is: {}".format(lemm.lemmatize("leaves")))
```

```
The lemmatized form of leaves is: leaf
```

Now we see that our lemmatizer is working and collapsing words into a form that makes much more lexical sense.

4. Stemming and Lemmatization

```
#@title
# stemming of the corpus

# english steming

df["AFTER ENGLISH STEMMING"]=df.swifter.apply(lambda row: eng_stem(row["ENGLISH CORPUS"]),axis=1)

# uncomment beolw to also perform hindi word lemmatization

# hindi steming

# df["AFTER HINDI STEMMING"]=df.swifter.apply(lambda row: hi_stem(row['HINDI CORPUS']), axis=1)


df.head()
```

	ENGLISH CORPUS	HINDI CORPUS	ENGLISH TOKENS	HINDI_TOKENS	ENGLISH SENTENCE LENGTH	HINDI SENTENCE LENGTH	ENGLISH STOP WORD REMOVED	SENTENCE LENGTH AFTER STOP WORD REMOVAL	ENGLISH POS TAGGING	AFTER ENGLISH STEMMING
0	Give application accessibility workout	अपने अनुप्रयोग को पहुंचनीयता व्यायाम का लाभ दें	[Give, your, application, an, accessibility, w...	[अपने, अनुप्रयोग, को, पहुंचनीयता, व्यायाम, का,...	6	8	[Give, application, accessibility, workout]	4	[, application accessibility workout]	give applic access workout
1	Accerciser Accessibility Explorer	एक्सेर्साइसर पहुंचनीयता अन्वेषक	[Accerciser, Accessibility, Explorer]	[एक्सेर्साइसर, पहुंचनीयता, अन्वेषक]	3	3	[Accerciser, Accessibility, Explorer]	3	[,]	accercis access explor
2	default plugin layout bottom panel	निचले पटल के लिए डिफोल्ट प्लग-इन खाका	[The, default, plugin, layout, for, the, botto...	[निचले, पटल, के, लिए, डिफोल्ट, प्लग-इन, खाका]	8	7	[default, plugin, layout, bottom, panel]	5	[,]	default plugin layout bottom panel
3	default plugin layout top panel	ऊपरी पटल के लिए डिफोल्ट प्लग-इन खाका	[The, default, plugin, layout, for, the, top, ...	[ऊपरी, पटल, के, लिए, डिफोल्ट, प्लग-इन, खाका]	8	7	[default, plugin, layout, top, panel]	5	[default plugin, top panel]	default plugin layout top panel
4	list plugins disabled default	उन प्लग-इनों की सूची जिन्हें डिफोल्ट रूप से नि...	[A, list, of, plugins, that, are, disabled, by...	[उन, प्लग-इनों, की, सूची, जिन्हें, डिफोल्ट, रू...	9	12	[list, plugins, disabled, default]	4	[list, disabled default]	list plugin disabl default

```
#@title
# lemmatization of the corpus

# english steming

df["AFTER ENGLISH LEMMATIZATION"]=df.swifter.apply(lambda row: eng_lemma(row["ENGLISH CORPUS"]),axis=1)

df.head()
```

	ENGLISH CORPUS	HINDI CORPUS	ENGLISH TOKENS	HINDI_TOKENS	ENGLISH SENTENCE LENGTH	HINDI SENTENCE LENGTH	ENGLISH STOP WORD REMOVED	SENTENCE LENGTH AFTER STOP WORD REMOVAL	ENGLISH POS TAGGING	AFTER ENGLISH STEMMING	AFT LEM
0	Give application accessibility workout	अपने अनुप्रयोग को पहुंचनीयता व्यायाम का लाभ दें	[Give, your, application, an, accessibility, w...	[अपने, अनुप्रयोग, को, पहुंचनीयता, व्यायाम, का,...	6	8	[Give, application, accessibility, workout]	4	[, application accessibility workout]	give applic access workout	G
1	Accerciser Accessibility Explorer	एक्सेर्साइसर पहुंचनीयता अन्वेषक	[Accerciser, Accessibility, Explorer]	[एक्सेर्साइसर, पहुंचनीयता, अन्वेषक]	3	3	[Accerciser, Accessibility, Explorer]	3	[,]	accercis access explor	
2	default plugin layout bottom panel	निचले पटल के लिए डिफोल्ट प्लग-इन खाका	[The, default, plugin, layout, for, the, botto...	[निचले, पटल, के, लिए, डिफोल्ट, प्लग-इन, खाका]	8	7	[default, plugin, layout, bottom, panel]	5	[,]	default plugin layout bottom panel	
3	default plugin layout top panel	ऊपरी पटल के लिए डिफोल्ट प्लग-इन खाका	[The, default, plugin, layout, for, the, top, ...	[ऊपरी, पटल, के, लिए, डिफोल्ट, प्लग-इन, खाका]	8	7	[default, plugin, layout, top, panel]	5	[default plugin, top panel]	default plugin layout top panel	la
4	list plugins disabled default	उन प्लग-इनों की सूची जिन्हें डिफोल्ट रूप से नि...	[A, list, of, plugins, that, are, disabled, by...	[उन, प्लग-इनों, की, सूची, जिन्हें, डिफोल्ट, रू...	9	12	[list, plugins, disabled, default]	4	[list, disabled default]	list plugin disabl default	d

PART B: STUDY OF NEURAL MACHINE TRANSLATION MODEL (NMT)

PERFORMING PREREQUISITING STEPS

```
#@title
!pip install tensorflow --upgrade tensorflow
```

```
Requirement already satisfied: tensorflow in /usr/local/lib/python3.7/dist-packages (2.7.0)
Requirement already satisfied: protobuf>=3.9.2 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (3.17.3)
Requirement already satisfied: keras<2.8,>=2.7.0rc0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (2.7.0)
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.13.3)
Requirement already satisfied: wheel<1.0,>=0.32.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (0.37.0)
Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (3.1.0)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.1.0)
Requirement already satisfied: flatbuffers<3.0,>=1.12 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (2.0)
Requirement already satisfied: tensorboard~=2.6 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (2.7.0)
Requirement already satisfied: numpy>=1.14.5 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.19.5)
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (3.10.0.2)
Requirement already satisfied: gast<0.5.0,>=0.2.1 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (0.4.0)
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.6.3)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.21.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (0.22.0)
Requirement already satisfied: libclang>=9.0.1 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (12.0.0)
Requirement already satisfied: tensorflow-estimator<2.8,~=2.7.0rc0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (2.7.0)
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (3.3.0)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.15.0)
Requirement already satisfied: keras-preprocessing>=1.1.1 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.1.2)
Requirement already satisfied: absl-py>=0.4.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (0.12.0)
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.7/dist-packages (from tensorflow) (1.42.0)
Requirement already satisfied: cached-property in /usr/local/lib/python3.7/dist-packages (from h5py>=2.9.0->tensorflow) (1.5.2)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in /usr/local/lib/python3.7/dist-packages (from tensorboard~=2.6->tensorflow) (0.4.6)
Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in /usr/local/lib/python3.7/dist-packages (from tensorboard~=2.6->tensorflow) (0.6.1)
Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3.7/dist-packages (from tensorboard~=2.6->tensorflow) (1.35.0)
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.7/dist-packages (from tensorboard~=2.6->tensorflow) (2.23.0)
Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in /usr/local/lib/python3.7/dist-packages (from tensorboard~=2.6->tensorflow) (1.8.0)
Requirement already satisfied: werkzeug>=0.11.15 in /usr/local/lib/python3.7/dist-packages (from tensorboard~=2.6->tensorflow) (1.0.1)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.7/dist-packages (from tensorboard~=2.6->tensorflow) (3.3.6)
Requirement already satisfied: setuptools>=41.0.0 in /usr/local/lib/python3.7/dist-packages (from tensorboard~=2.6->tensorflow) (57.4.0)
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.7/dist-packages (from google-auth<3,>=1.6.3->tensorboard~=2.6->tensorflow) (0.2.8)
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.7/dist-packages (from google-auth<3,>=1.6.3->tensorboard~=2.6->tensorflow) (4.7.2)
Requirement already satisfied: cachetools<5.0,>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from google-auth<3,>=1.6.3->tensorboard~=2.6->tensorflow) (4.2.4)
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.7/dist-packages (from google-auth-oauthlib<0.5,>=0.4.1->tensorboard~=2.6->tensorflow) (1.3.0)
Requirement already satisfied: importlib-metadata>=4.4 in /usr/local/lib/python3.7/dist-packages (from markdown>=2.6.8->tensorboard~=2.6->tensorflow) (4.8.2)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (from importlib-metadata>=4.4->markdown>=2.6.8->tensorboard~=2.6->tensorflow) (3.6.0)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /usr/local/lib/python3.7/dist-packages (from pyasn1-modules>=0.2.1->google-auth<3,>=1.6.3->tensorboard~=2.6->tensorflow) (0.4.8)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests<3,>=2.21.0->tensorboard~=2.6->tensorflow) (2021.10.8)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests<3,>=2.21.0->tensorboard~=2.6->tensorflow) (2.10)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from requests<3,>=2.21.0->tensorboard~=2.6->tensorflow) (1.24.3)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests<3,>=2.21.0->tensorboard~=2.6->tensorflow) (3.0.4)
Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.7/dist-packages (from requests-oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1->tensorboard~=2.6->tensorflow) (3.1.1)
```

In [ ]:
```python
#@title
#importing libraries

import pandas as pd
import tensorflow as tf
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
import numpy as np
from tensorflow.keras.models import Model, Sequential
from tensorflow.keras.layers import GRU, Input, Dense, TimeDistributed, Activation, RepeatVector, Bidirect
from tensorflow.keras.losses import sparse_categorical_crossentropy
from tensorflow.keras.optimizers import Adam
from sklearn.model_selection import train_test_split
from tensorflow.keras.layers import Dense, LSTM, Embedding, RepeatVector
from tensorflow.keras import optimizers
from tensorflow.keras.callbacks import ModelCheckpoint
```

In [ ]:
```python
#@title
# reading the train dataset

#english corpus
eng_train=pd.read_csv('/content/drive/MyDrive/Colab Notebooks/PARALLEL-CORPUS (TRAIN)/IITB.en-hi.en.csv')

#hindi corpus
hi_train=pd.read_csv('/content/drive/MyDrive/Colab Notebooks/PARALLEL-CORPUS (TRAIN)/IITB.en-hi.hi.csv')

# reading the validation dataset

#english corpus
eng_val=pd.read_csv('/content/drive/MyDrive/Colab Notebooks/PARALLEL-CORPUS (VALIDATION)/validation.en.csv

#hindi corpus
hi_val=pd.read_csv('/content/drive/MyDrive/Colab Notebooks/PARALLEL-CORPUS (VALIDATION)/validation.hi.csv'

# reading the test dataset

#english corpus
eng_test=pd.read_csv('/content/drive/MyDrive/Colab Notebooks/PARALLEL-CORPUS (TEST)/test.en.csv')

#hindi corpus
hi_test=pd.read_csv('/content/drive/MyDrive/Colab Notebooks/PARALLEL-CORPUS (TEST)/test.hi.csv')
```

COMBINING BOTH THE LANGUAGE PAIRS TO FORM PARALLEL CORPORUS OF TRAIN, TEST AND VALIDATION

In [ ]:
```python
#@title
#TRAINING DATA

df_train=pd.concat([eng_train,hi_train],axis=1)

print("="*113,"\nTRAINING DATA\n",sep='')

df_train.head()
```

```
=================================================================================================================
TRAINING DATA
```

Out[ ]:

	ENGLISH CORPUS	HINDI CORPUS
0	Give your application an accessibility workout	अपने अनुप्रयोग को पहुंचनीयता व्यायाम का लाभ दें
1	Accerciser Accessibility Explorer	एक्सेर्साइसर पहुंचनीयता अन्वेषक
2	The default plugin layout for the bottom panel	निचले पटल के लिए डिफोल्ट प्लग-इन खाका
3	The default plugin layout for the top panel	ऊपरी पटल के लिए डिफोल्ट प्लग-इन खाका
4	A list of plugins that are disabled by default	उन प्लग-इनों की सूची जिन्हें डिफोल्ट रूप से नि...

In [ ]:
```python
#@title
#VALIDATION DATA

df_val=pd.concat([eng_val,hi_val],axis=1)

print("="*113,"\nVALIDATION DATA",sep='')

df_val.head()
```

```
=================================================================================================================
VALIDATION DATA
```

Out[ ]:

	ENGLISH CORPUS	HINDI CORPUS
0	Students of the Dattatreya city Municipal corp...	महानगर पालिका अंतर्गत दत्तात्रय नगर माध्यमिक स...
1	With encouragement from Principal Sandhya Medp...	प्रधानाध्यापक संध्या मेडपल्लीवार के प्रोत्साहि...
2	Rajesh Gavre the President of the MNPA teacher...	मनपा शिक्षक संघ के अध्यक्ष राजेश गवरे ने स्कूल...
3	Ramesh Saatpute examined the fort.	किले का परीक्षण रमेश सातपुते ने किया।
4	Students like Nikhil Kavle Darshan Gedekar Sah...	किला निर्माण में निखिल कावले दर्शन गेड़ेकर साह...

In [ ]:
```python
#@title
#TESTING DATA

df_test=pd.concat([eng_test,hi_test],axis=1)

print("="*113,"\nTESTING DATA",sep='')

df_test.head()
```

```
=================================================================================================================
TESTING DATA
```

Out[ ]:

	ENGLISH CORPUS	HINDI CORPUS
0	A black box in your car?	आपकी कार में ब्लैक बॉक्स?
1	As America's road planners struggle to find th...	जबकि अमेरिका के सड़क योजनाकार ध्वस्त होते हुए ...
2	The devices which track every mile a motorist ...	यह डिवाइस जो मोटर-चालक द्वारा वाहन चलाए गए प्र...
3	The usually dull arena of highway planning has...	आम तौर पर हाईवे नियोजन जैसा उबाऊ काम भी अचानक ...
4	Libertarians have joined environmental groups ...	आपने द्वारा ड्राइव किए गए मील तथा संभवतः ड्राइ...

PREPROCESSING THE DATASET

TOKENIZE

For a neural network to predict on text data, it first has to be turned into data it can understand. Text data like "dog" is a sequence of ASCII character encodings. Since a neural network is a series of multiplication and addition operations, the input data needs to be number(s).

We can turn each character into a number or each word into a number. These are called character and word ids, respectively. Character ids are used for character level models that generate text predictions for each character. A word level model uses word ids that generate text predictions for each word. Word level models tend to learn better, since they are lower in complexity, so we'll use those.

Turn each sentence into a sequence of words ids using Keras's Tokenizer function. Use this function to tokenize english_sentences and french_sentences in the cell below.

In [ ]:
```python
#@title
def tokenization(lines):
    tokenizer = Tokenizer()
    tokenizer.fit_on_texts(lines)
    return tokenizer
```

In [ ]:
```python
#@title
# train data

# prepare english train tokenizer
eng_train_tokenizer = tokenization(df_train["ENGLISH CORPUS"])
eng_train_vocab_size = len(eng_train_tokenizer.word_index) + 1

eng_train_length = 8
print('English Train Vocabulary Size: %d' % eng_train_vocab_size)

# valiation data

# prepare english validation tokenizer
eng_val_tokenizer = tokenization(df_val["ENGLISH CORPUS"])
eng_val_vocab_size = len(eng_val_tokenizer.word_index) + 1

eng_val_length = 8
print('English Validation Vocabulary Size: %d' % eng_val_vocab_size)
```

```
English Train Vocabulary Size: 21720
English Validation Vocabulary Size: 2379
```

In [ ]:
```python
#@title
# train data

# prepare Hindi train tokenizer
hi_train_tokenizer = tokenization(df_train["HINDI CORPUS"])
hi_train_vocab_size = len(hi_train_tokenizer.word_index) + 1

hi_train_length = 8
print('Hindi Train Vocabulary Size: %d' % hi_train_vocab_size)

# valiation data

# prepare hindi validation tokenizer
hi_val_tokenizer = tokenization(df_val["HINDI CORPUS"])
hi_val_vocab_size = len(hi_val_tokenizer.word_index) + 1

hi_val_length = 8
print('Hindi Validation Vocabulary Size: %d' % hi_val_vocab_size)
```

```
Hindi Train Vocabulary Size: 31823
Hindi Validation Vocabulary Size: 2672
```

In [ ]:
```python
#@title
df['HINDI_TOKENS'] = df.swifter.apply(lambda row: hi_tokenize(row['HINDI CORPUS']), axis=1)
```

Padding

When batching the sequence of word ids together, each sequence needs to be the same length. Since sentences are dynamic in length, we can add padding to the end of the sequences to make them the same length.

Make sure all the English sequences have the same length and all the Hindi sequences have the same length by adding padding to the **end** of each sequence using Keras function.

In [ ]:
```python
#@title
# encode and pad sequences
def encode_sequences(tokenizer, length, lines):
    seq = tokenizer.texts_to_sequences(lines)
    # pad sequences with 0 values
    seq = pad_sequences(seq, maxlen=length, padding='post')
    return seq
```

In [ ]:
```python
#@title
# prepare training data
trainX = encode_sequences(eng_train_tokenizer, eng_train_length, df_train["ENGLISH CORPUS"])
trainY = encode_sequences(hi_train_tokenizer, hi_train_length, df_train["HINDI CORPUS"])

# prepare validation data
valX = encode_sequences(eng_val_tokenizer, eng_val_length, df_val["ENGLISH CORPUS"])
valY = encode_sequences(hi_val_tokenizer, hi_val_length, df_val["HINDI CORPUS"])
```

In [ ]:
```python
#@title
def count(series):
    l=[]
    for lis in series:
        for words in lis:
            if words not in l:
                l.append(words)
    return len(l)
```

In [ ]:
```python
#@title
def generate_batch(X = df_train["ENGLISH CORPUS"], y = df_train["HINDI CORPUS"], batch_size = 128):
    ''' Generate a batch of data '''
    while True:
        X=df_train["ENGLISH CORPUS"]
        y=df_train["HINDI CORPUS"]
        for j in range(0, len(X), batch_size):
            encoder_input_data = np.zeros((batch_size, max_length_src),dtype='float32')
            decoder_input_data = np.zeros((batch_size, max_length_tar),dtype='float32')
            decoder_target_data = np.zeros((batch_size, max_length_tar, num_decoder_tokens),dtype='float32
            for i, (input_text, target_text) in enumerate(zip(X[j:j+batch_size], y[j:j+batch_size])):
                for t, word in enumerate(input_text.split()):
                    if(word in input_token_index.keys()):

                        encoder_input_data[i, t] = input_token_index[word] # encoder input seq
                for t, word in enumerate(target_text.split()):
                    if(word in target_token_index.keys()):

                        if t<len(target_text.split())-1:
                            decoder_input_data[i, t] = target_token_index[word] # decoder input seq
                        if t>0:
                            # decoder target sequence (one hot encoded)
                            # does not include the START_ token
                            # Offset by one timestep
                            decoder_target_data[i, t - 1, target_token_index[word]-1] = 1.
            yield([encoder_input_data, decoder_input_data], decoder_target_data)
```

In [ ]:
```python
#@title
latent_dim=300
```

In [ ]:
```python
#@title
# Encoder
encoder_inputs = Input(shape=(None,))
enc_emb =  Embedding(count(df["ENGLISH TOKENS"]), latent_dim, mask_zero = True)(encoder_inputs)
encoder_lstm = LSTM(latent_dim, return_state=True)
encoder_outputs, state_h, state_c = encoder_lstm(enc_emb)
# We discard `encoder_outputs` and only keep the states.
encoder_states = [state_h, state_c]
```

In [ ]:
```python
#@title
# Set up the decoder, using `encoder_states` as initial state.
decoder_inputs = Input(shape=(None,))
dec_emb_layer = Embedding(count(df["HINDI_TOKENS"]), latent_dim, mask_zero = True)
dec_emb = dec_emb_layer(decoder_inputs)
# 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(dec_emb,
                                     initial_state=encoder_states)
decoder_dense = Dense(count(df["HINDI_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)
```

In [ ]:
```python
#@title
model.compile(optimizer='rmsprop', loss='categorical_crossentropy')
```

In [ ]:
```python
#@title
model.summary()
```

```
Model: "model"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_1 (InputLayer)           [(None, None)]       0           []

 input_2 (InputLayer)           [(None, None)]       0           []

 embedding (Embedding)          (None, None, 300)    8075400     ['input_1[0][0]']

 embedding_1 (Embedding)        (None, None, 300)    11239200    ['input_2[0][0]']

 lstm (LSTM)                    [(None, 300),        721200      ['embedding[0][0]']
                                 (None, 300),
                                 (None, 300)]

 lstm_1 (LSTM)                  [(None, None, 300),  721200      ['embedding_1[0][0]',
                                 (None, 300),                     'lstm[0][1]',
                                 (None, 300)]                     'lstm[0][2]']

 dense (Dense)                  (None, None, 37464)  11276664    ['lstm_1[0][0]']

==================================================================================================
Total params: 32,033,664
Trainable params: 32,033,664
Non-trainable params: 0
__________________________________________________________________________________________________
```

In [ ]:
```python
#@title
train_samples = len(df_train["ENGLISH CORPUS"])
val_samples = len(df_val["ENGLISH CORPUS"])
batch_size = 128
epochs = 100
```

In [ ]:
```python
#@title
max_length_src=max(df["ENGLISH SENTENCE LENGTH"])
max_length_tar=max(df["HINDI SENTENCE LENGTH"])
num_decoder_tokens=count(df["HINDI_TOKENS"])
```

In [ ]:
```python
#@title
def count1(series):
    l=[]
    for lis in series:
        for words in lis:
            if words not in l:
                l.append(words)
    return l
```

In [ ]:
```python
#@title
input_token_index = dict([(word, i+1) for i, word in enumerate(sorted(count1(df["ENGLISH TOKENS"])))])
target_token_index = dict([(word, i+1) for i, word in enumerate(sorted(count1(df["HINDI_TOKENS"])))])
```

In [ ]:
```python
#@title
model.fit_generator(generator = generate_batch(df_train["ENGLISH CORPUS"], df_train["HINDI CORPUS"], batch
                    steps_per_epoch = train_samples//batch_size,
                    epochs=epochs,
                    validation_data = generate_batch(df_test["ENGLISH CORPUS"], df_test["HINDI CORPUS"], b
                    validation_steps = val_samples//batch_size)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:6: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.
```

```
Epoch 1/100
```

In [ ]:
```python
#@title
# Encode the input sequence to get the "thought vectors"
encoder_model = Model(encoder_inputs, encoder_states)

# Decoder setup
# Below tensors will hold the states of the previous time step
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]

dec_emb2= dec_emb_layer(decoder_inputs) # Get the embeddings of the decoder sequence

# To predict the next word in the sequence, set the initial states to the states from the previous time st
decoder_outputs2, state_h2, state_c2 = decoder_lstm(dec_emb2, initial_state=decoder_states_inputs)
decoder_states2 = [state_h2, state_c2]
decoder_outputs2 = decoder_dense(decoder_outputs2) # A dense softmax layer to generate prob dist. over the

# Final decoder model
decoder_model = Model(
    [decoder_inputs] + decoder_states_inputs,
    [decoder_outputs2] + decoder_states2)
```

In [ ]:
```python
#@title
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))
    # Populate the first character of target sequence with the start character.
    target_seq[0, 0] = target_token_index['START_']

    # 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 == '_END' or
           len(decoded_sentence) > 50):
            stop_condition = True

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

        # Update states
        states_value = [h, c]

    return decoded_sentence
```

In [ ]:
```python
#@title
train_gen = generate_batch(X_train, y_train, batch_size = 1)
k=-1
```

In [ ]:
```python
#@title
k+=1
(input_seq, actual_output), _ = next(train_gen)
decoded_sentence = decode_sequence(input_seq)
print('Input English sentence:', X_train[k:k+1].values[0])
print('Actual Hindi Translation:', y_train[k:k+1].values[0][6:-4])
print('Predicted Hindi Translation:', decoded_sentence[:-4])
```

In [ ]:
```python
#@title
k+=1
(input_seq, actual_output), _ = next(train_gen)
decoded_sentence = decode_sequence(input_seq)
print('Input English sentence:', X_train[k:k+1].values[0])
print('Actual Hindi Translation:', y_train[k:k+1].values[0][6:-4])
print('Predicted Hindi Translation:', decoded_sentence[:-4])
```

In [ ]:
```python
#@title
k+=1
(input_seq, actual_output), _ = next(train_gen)
decoded_sentence = decode_sequence(input_seq)
print('Input English sentence:', X_train[k:k+1].values[0])
print('Actual Hindi Translation:', y_train[k:k+1].values[0][6:-4])
print('Predicted Hindi Translation:', decoded_sentence[:-4])
```

In [ ]:
```python
#@title
k+=1
(input_seq, actual_output), _ = next(train_gen)
decoded_sentence = decode_sequence(input_seq)
print('Input English sentence:', X_train[k:k+1].values[0])
print('Actual Hindi Translation:', y_train[k:k+1].values[0][6:-4])
print('Predicted Hindi Translation:', decoded_sentence[:-4])
```

In [ ]:
```python
#@title
k+=1
(input_seq, actual_output), _ = next(train_gen)
decoded_sentence = decode_sequence(input_seq)
print('Input English sentence:', X_train[k:k+1].values[0])
print('Actual Hindi Translation:', y_train[k:k+1].values[0][6:-4])
print('Predicted Hindi Translation:', decoded_sentence[:-4])
```

Bleu Score calculation

In [ ]:
```python
#@title
a = y_train[k:k+1].values[0][6:-4]
b = decoded_sentence[:-4]
```

In [ ]:
```python
#@title
from nltk.translate.bleu_score import sentence_bleu
score = sentence_bleu( a, b)
print('Bleu score:', '%3f'%score)
```

===================THE END=================