This notebook utilizes libraries like `pandas` for data manipulation, `matplotlib`, `seaborn`, and `plotly` for static and interactive visualizations, and `nltk` for text preprocessing tasks such as sentence tokenization. Warnings are suppressed for cleaner output using `warnings.filterwarnings("ignore")`.

```
%pip install jupyter notebook ipykernel
%pip install pandas
%pip install matplotlib
%pip install seaborn
%pip install nltk
%pip install plotly
```

```
Requirement already satisfied: jupyter in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (1.1.1)
Requirement already satisfied: notebook in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (7.3.2)
Requirement already satisfied: ipykernel in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (6.29.5)
Requirement already satisfied: jupyter-console in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from jupyter)
(6.6.3)
Requirement already satisfied: nbconvert in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (from jupyter) (7.16.6)
Requirement already satisfied: ipywidgets in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (from jupyter) (8.1.5)
Requirement already satisfied: jupyterlab in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (from jupyter) (4.3.5)
Requirement already satisfied: jupyter-server<3,>=2.4.0 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
notebook) (2.15.0)
Requirement already satisfied: jupyterlab-server<3,>=2.27.1 in c:\
users\kushal\onedrive\desktop\new folder\chintya\lib\site-packages
(from notebook) (2.27.3)
Requirement already satisfied: notebook-shim<0.3,>=0.2 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
notebook) (0.2.4)
Requirement already satisfied: tornado>=6.2.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from notebook)
(6.4.2)
Requirement already satisfied: comm>=0.1.1 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from ipykernel)
(0.2.2)
Requirement already satisfied: debugpy>=1.6.5 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from ipykernel)
(1.8.13)
Requirement already satisfied: ipython>=7.23.1 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from ipykernel)
(8.18.1)
```

```
Requirement already satisfied: jupyter-client>=6.1.12 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
ipykernel) (8.6.3)
Requirement already satisfied: jupyter-core!=5.0.*,>=4.12 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
ipykernel) (5.7.2)
Requirement already satisfied: matplotlib-inline>=0.1 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
ipykernel) (0.1.7)
Requirement already satisfied: nest-asyncio in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from ipykernel)
(1.6.0)
Requirement already satisfied: packaging in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (from ipykernel) (24.2)
Requirement already satisfied: psutil in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (from ipykernel) (7.0.0)
Requirement already satisfied: pyzmq>=24 in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (from ipykernel) (26.2.1)
Requirement already satisfied: traitlets>=5.4.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from ipykernel)
(5.14.3)
Requirement already satisfied: decorator in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (from ipython>=7.23.1-
>ipykernel) (5.2.1)
Requirement already satisfied: jedi>=0.16 in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (from ipython>=7.23.1-
>ipykernel) (0.19.2)
Requirement already satisfied: prompt-toolkit<3.1.0,>=3.0.41 in c:\
users\kushal\onedrive\desktop\new folder\chintya\lib\site-packages
(from ipython>=7.23.1->ipykernel) (3.0.50)
Requirement already satisfied: pygments>=2.4.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
ipython>=7.23.1->ipykernel) (2.19.1)
Requirement already satisfied: stack-data in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (from ipython>=7.23.1-
>ipykernel) (0.6.3)
Requirement already satisfied: typing-extensions in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
ipython>=7.23.1->ipykernel) (4.12.2)
Requirement already satisfied: exceptiongroup in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
ipython>=7.23.1->ipykernel) (1.2.2)
Requirement already satisfied: colorama in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (from ipython>=7.23.1-
>ipykernel) (0.4.6)
Requirement already satisfied: importlib-metadata>=4.8.3 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
jupyter-client>=6.1.12->ipykernel) (8.6.1)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\
```

```
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
jupyter-client>=6.1.12->ipykernel) (2.9.0.post0)
Requirement already satisfied: platformdirs>=2.5 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from jupyter-
core!=5.0.*,>=4.12->ipykernel) (4.3.6)
Requirement already satisfied: pywin32>=300 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from jupyter-
core!=5.0.*,>=4.12->ipykernel) (308)
Requirement already satisfied: anyio>=3.1.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from jupyter-
server<3,>=2.4.0->notebook) (4.8.0)
Requirement already satisfied: argon2-cffi>=21.1 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from jupyter-
server<3,>=2.4.0->notebook) (23.1.0)
Requirement already satisfied: jinja2>=3.0.3 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from jupyter-
server<3,>=2.4.0->notebook) (3.1.5)
Requirement already satisfied: jupyter-events>=0.11.0 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
jupyter-server<3,>=2.4.0->notebook) (0.12.0)
Requirement already satisfied: jupyter-server-terminals>=0.4.4 in c:\
users\kushal\onedrive\desktop\new folder\chintya\lib\site-packages
(from jupyter-server<3,>=2.4.0->notebook) (0.5.3)
Requirement already satisfied: nbformat>=5.3.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from jupyter-
server<3,>=2.4.0->notebook) (5.10.4)
Requirement already satisfied: overrides>=5.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from jupyter-
server<3,>=2.4.0->notebook) (7.7.0)
Requirement already satisfied: prometheus-client>=0.9 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
jupyter-server<3,>=2.4.0->notebook) (0.21.1)
Requirement already satisfied: pywinpty>=2.0.1 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from jupyter-
server<3,>=2.4.0->notebook) (2.0.15)
Requirement already satisfied: send2trash>=1.8.2 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from jupyter-
server<3,>=2.4.0->notebook) (1.8.3)
Requirement already satisfied: terminado>=0.8.3 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from jupyter-
server<3,>=2.4.0->notebook) (0.18.1)
Requirement already satisfied: websocket-client>=1.7 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
jupyter-server<3,>=2.4.0->notebook) (1.8.0)
Requirement already satisfied: async-lru>=1.0.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
jupyterlab->jupyter) (2.0.4)
Requirement already satisfied: httpx>=0.25.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
```

```
jupyterlab->jupyter) (0.28.1)
Requirement already satisfied: jupyter-lsp>=2.0.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
jupyterlab->jupyter) (2.2.5)
Requirement already satisfied: setuptools>=40.8.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
jupyterlab->jupyter) (58.1.0)
Requirement already satisfied: tomli>=1.2.2 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
jupyterlab->jupyter) (2.2.1)
Requirement already satisfied: babel>=2.10 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
jupyterlab-server<3,>=2.27.1->notebook) (2.17.0)
Requirement already satisfied: json5>=0.9.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
jupyterlab-server<3,>=2.27.1->notebook) (0.10.0)
Requirement already satisfied: jsonschema>=4.18.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
jupyterlab-server<3,>=2.27.1->notebook) (4.23.0)
Requirement already satisfied: requests>=2.31 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
jupyterlab-server<3,>=2.27.1->notebook) (2.32.3)
Requirement already satisfied: beautifulsoup4 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from nbconvert-
>jupyter) (4.13.3)
Requirement already satisfied: bleach!=5.0.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
bleach[css]!=5.0.0->nbconvert->jupyter) (6.2.0)
Requirement already satisfied: defusedxml in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (from nbconvert->jupyter)
(0.7.1)
Requirement already satisfied: jupyterlab-pygments in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from nbconvert-
>jupyter) (0.3.0)
Requirement already satisfied: markupsafe>=2.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from nbconvert-
>jupyter) (3.0.2)
Requirement already satisfied: mistune<4,>=2.0.3 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from nbconvert-
>jupyter) (3.1.2)
Requirement already satisfied: nbclient>=0.5.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from nbconvert-
>jupyter) (0.10.2)
Requirement already satisfied: pandocfilters>=1.4.1 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
nbconvert->jupyter) (1.5.1)
Requirement already satisfied: widgetsnbextension~=4.0.12 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
ipywidgets->jupyter) (4.0.13)
```

```
Requirement already satisfied: jupyterlab-widgets~=3.0.12 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
ipywidgets->jupyter) (3.0.13)
Requirement already satisfied: idna>=2.8 in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (from anyio>=3.1.0-
>jupyter-server<3,>=2.4.0->notebook) (3.10)
Requirement already satisfied: sniffio>=1.1 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
anyio>=3.1.0->jupyter-server<3,>=2.4.0->notebook) (1.3.1)
Requirement already satisfied: argon2-cffi-bindings in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
argon2-cffi>=21.1->jupyter-server<3,>=2.4.0->notebook) (21.2.0)
Requirement already satisfied: webencodings in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from bleach!
=5.0.0->bleach[css]!=5.0.0->nbconvert->jupyter) (0.5.1)
Requirement already satisfied: tinycss2<1.5,>=1.1.0 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
bleach[css]!=5.0.0->nbconvert->jupyter) (1.4.0)
Requirement already satisfied: certifi in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (from httpx>=0.25.0-
>jupyterlab->jupyter) (2025.1.31)
Requirement already satisfied: httpcore==1.* in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
httpx>=0.25.0->jupyterlab->jupyter) (1.0.7)
Requirement already satisfied: h11<0.15,>=0.13 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
httpcore==1.*->httpx>=0.25.0->jupyterlab->jupyter) (0.14.0)
Requirement already satisfied: zipp>=3.20 in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (from importlib-
metadata>=4.8.3->jupyter-client>=6.1.12->ipykernel) (3.21.0)
Requirement already satisfied: parso<0.9.0,>=0.8.4 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
jedi>=0.16->ipython>=7.23.1->ipykernel) (0.8.4)
Requirement already satisfied: attrs>=22.2.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
jsonschema>=4.18.0->jupyterlab-server<3,>=2.27.1->notebook) (25.1.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in
c:\users\kushal\onedrive\desktop\new folder\chintya\lib\site-packages
(from jsonschema>=4.18.0->jupyterlab-server<3,>=2.27.1->notebook)
(2024.10.1)
Requirement already satisfied: referencing>=0.28.4 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
jsonschema>=4.18.0->jupyterlab-server<3,>=2.27.1->notebook) (0.36.2)
Requirement already satisfied: rpds-py>=0.7.1 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
jsonschema>=4.18.0->jupyterlab-server<3,>=2.27.1->notebook) (0.23.1)
Requirement already satisfied: python-json-logger>=2.0.4 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
jupyter-events>=0.11.0->jupyter-server<3,>=2.4.0->notebook) (3.2.1)
```

```
Requirement already satisfied: pyyaml>=5.3 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from jupyter-
events>=0.11.0->jupyter-server<3,>=2.4.0->notebook) (6.0.2)
Requirement already satisfied: rfc3339-validator in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from jupyter-
events>=0.11.0->jupyter-server<3,>=2.4.0->notebook) (0.1.4)
Requirement already satisfied: rfc3986-validator>=0.1.1 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
jupyter-events>=0.11.0->jupyter-server<3,>=2.4.0->notebook) (0.1.1)
Requirement already satisfied: fastjsonschema>=2.15 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
nbformat>=5.3.0->jupyter-server<3,>=2.4.0->notebook) (2.21.1)
Requirement already satisfied: wcwidth in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (from prompt-
toolkit<3.1.0,>=3.0.41->ipython>=7.23.1->ipykernel) (0.2.13)
Requirement already satisfied: six>=1.5 in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (from python-
dateutil>=2.8.2->jupyter-client>=6.1.12->ipykernel) (1.17.0)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
requests>=2.31->jupyterlab-server<3,>=2.27.1->notebook) (3.4.1)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
requests>=2.31->jupyterlab-server<3,>=2.27.1->notebook) (2.3.0)
Requirement already satisfied: soupsieve>1.2 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
beautifulsoup4->nbconvert->jupyter) (2.6)
Requirement already satisfied: executing>=1.2.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from stack-
data->ipython>=7.23.1->ipykernel) (2.2.0)
Requirement already satisfied: asttokens>=2.1.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from stack-
data->ipython>=7.23.1->ipykernel) (3.0.0)
Requirement already satisfied: pure-eval in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (from stack-data-
>ipython>=7.23.1->ipykernel) (0.2.3)
Requirement already satisfied: fqdn in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (from jsonschema[format-
nongpl]>=4.18.0->jupyter-events>=0.11.0->jupyter-server<3,>=2.4.0-
>notebook) (1.5.1)
Requirement already satisfied: isoduration in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
jsonschema[format-nongpl]>=4.18.0->jupyter-events>=0.11.0->jupyter-
server<3,>=2.4.0->notebook) (20.11.0)
Requirement already satisfied: jsonpointer>1.13 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
jsonschema[format-nongpl]>=4.18.0->jupyter-events>=0.11.0->jupyter-
server<3,>=2.4.0->notebook) (3.0.0)
Requirement already satisfied: uri-template in c:\users\kushal\
```

```
onedrive\desktop\new folder\chintya\lib\site-packages (from
jsonschema[format-nongpl]>=4.18.0->jupyter-events>=0.11.0->jupyter-
server<3,>=2.4.0->notebook) (1.3.0)
Requirement already satisfied: webcolors>=24.6.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
jsonschema[format-nongpl]>=4.18.0->jupyter-events>=0.11.0->jupyter-
server<3,>=2.4.0->notebook) (24.11.1)
Requirement already satisfied: cffi>=1.0.1 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from argon2-
cffi-bindings->argon2-cffi>=21.1->jupyter-server<3,>=2.4.0->notebook)
(1.17.1)
Requirement already satisfied: pycparser in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (from cffi>=1.0.1-
>argon2-cffi-bindings->argon2-cffi>=21.1->jupyter-server<3,>=2.4.0-
>notebook) (2.22)
Requirement already satisfied: arrow>=0.15.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
isoduration->jsonschema[format-nongpl]>=4.18.0->jupyter-
events>=0.11.0->jupyter-server<3,>=2.4.0->notebook) (1.3.0)
Requirement already satisfied: types-python-dateutil>=2.8.10 in c:\
users\kushal\onedrive\desktop\new folder\chintya\lib\site-packages
(from arrow>=0.15.0->isoduration->jsonschema[format-nongpl]>=4.18.0-
>jupyter-events>=0.11.0->jupyter-server<3,>=2.4.0->notebook)
(2.9.0.20241206)
Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: pandas in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (2.2.3)
Requirement already satisfied: numpy>=1.22.4 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from pandas)
(2.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from pandas)
(2025.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from pandas)
(2025.1)
Requirement already satisfied: six>=1.5 in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (from python-
dateutil>=2.8.2->pandas) (1.17.0)
Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: matplotlib in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (3.9.4)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
matplotlib) (1.3.0)
Requirement already satisfied: cycler>=0.10 in c:\users\kushal\
```

```
onedrive\desktop\new folder\chintya\lib\site-packages (from
matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
matplotlib) (4.56.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
matplotlib) (1.4.7)
Requirement already satisfied: numpy>=1.23 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
matplotlib) (2.0.2)
Requirement already satisfied: packaging>=20.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
matplotlib) (24.2)
Requirement already satisfied: pillow>=8 in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (from matplotlib)
(11.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
matplotlib) (3.2.1)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
matplotlib) (2.9.0.post0)
Requirement already satisfied: importlib-resources>=3.2.0 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
matplotlib) (6.5.2)
Requirement already satisfied: zipp>=3.1.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from importlib-
resources>=3.2.0->matplotlib) (3.21.0)
Requirement already satisfied: six>=1.5 in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (from python-
dateutil>=2.7->matplotlib) (1.17.0)
Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: seaborn in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (0.13.2)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
seaborn) (2.0.2)
Requirement already satisfied: pandas>=1.2 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from seaborn)
(2.2.3)
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
seaborn) (3.9.4)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (1.3.0)
Requirement already satisfied: cycler>=0.10 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
```

```
matplotlib!=3.6.1,>=3.4->seaborn) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (4.56.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (1.4.7)
Requirement already satisfied: packaging>=20.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (24.2)
Requirement already satisfied: pillow>=8 in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (from matplotlib!
=3.6.1,>=3.4->seaborn) (11.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (3.2.1)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (2.9.0.post0)
Requirement already satisfied: importlib-resources>=3.2.0 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (6.5.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
pandas>=1.2->seaborn) (2025.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
pandas>=1.2->seaborn) (2025.1)
Requirement already satisfied: zipp>=3.1.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from importlib-
resources>=3.2.0->matplotlib!=3.6.1,>=3.4->seaborn) (3.21.0)
Requirement already satisfied: six>=1.5 in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (from python-
dateutil>=2.7->matplotlib!=3.6.1,>=3.4->seaborn) (1.17.0)
Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: nltk in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (3.9.1)
Requirement already satisfied: click in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (from nltk) (8.1.8)
Requirement already satisfied: joblib in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (from nltk) (1.4.2)
Requirement already satisfied: regex>=2021.8.3 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from nltk)
(2024.11.6)
Requirement already satisfied: tqdm in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (from nltk) (4.67.1)
Requirement already satisfied: colorama in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (from click->nltk)
(0.4.6)
```

```
Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: plotly in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (6.0.0)
Requirement already satisfied: narwhals>=1.15.1 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from plotly)
(1.29.0)
Requirement already satisfied: packaging in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (from plotly) (24.2)
Note: you may need to restart the kernel to use updated packages.
```

```
#Import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from nltk.tokenize import sent_tokenize
import nltk
import warnings

import plotly.express as px
import plotly.io as pio

warnings.filterwarnings("ignore")
```

The `nltk.download('punkt')` command downloads the Punkt tokenizer models, which are pre-trained data required by NLTK to tokenize text into sentences or words. These models enable functions like `sent_tokenize` and `word_tokenize` to split text accurately. Ensure to run this command during setup to preprocess text effectively.

```
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\Kushal\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!

True
```

This step ensures the SAMSum dataset is loaded into memory and ready for further exploration, such as data inspection, cleaning, and preprocessing for summarization tasks.or summarization tasks.or summarization tasks.

```
train_file = "C:/Users/Kushal/OneDrive/Desktop/Text-Summarization-NLP-
Project/train.csv"
df = pd.read_csv(train_file)

df
```

```
                                                  Text  \
0     Appeal No. LXVI of 1949.\nAppeal from the High...
1     Civil Appeal No.94 of 1949.\n107 834 Appeal fr...
2     iminal Appeal No. 40 of 1951, 127 Appeal from ...
```

```
3     Appeal No. 388 of 1960.\nAppeal by special lea...
4     Appeal No. 198 of 1954.\nAppeal from the judgm...
...                                                 ...
7025  Appeal No. 761 of 1957.\nAppeal by special lea...
7026  Appeal No. 761 of 1957.\nAppeal by special lea...
7027  Appeal No. 53 of 1958.\nAppeal by special leav...
7028  Appeal No. 4 of 1960.\nAppeal by special leave...
7029  Appeal No. 337 of 1960.\nAppeal by special lea...

                                                Summary
0     The charge created in respect of municipal pro...
1     An agreement for a lease, which a lease is by ...
2     The question whether a Magistrate is "personal...
3     The appellant was a member of a joint Hindu fa...
4     The appellant was the Ruler of the State of Ba...
...                                                 ...
7025  The respondent company purchased certain machi...
7026  The respondent company purchased certain machi...
7027  While this appeal by special leave, relating t...
7028  The Punjab Government issued notification unde...
7029  The respondent who was a District and Sessions...

[7030 rows x 2 columns]
```

```
df = df[0: 1000]

df
```

```
                                                Text  \
0  Appeal No. LXVI of 1949.\nAppeal from the High...
1  Civil Appeal No.94 of 1949.\n107 834 Appeal fr...
2  iminal Appeal No. 40 of 1951, 127 Appeal from ...
3  Appeal No. 388 of 1960.\nAppeal by special lea...
4  Appeal No. 198 of 1954.\nAppeal from the judgm...
5  Appeals Nos. 155 to 160 of 1956.\nAppeals from...
6  iminal Appeal No. 68 of 1958.\nAppeal by speci...
7  minal Appeal No. 124 of 1959.\nAppeal by speci...
8  Appeal No. 198 of 1954.\nAppeal from the judgm...
9  Appeal No. 285 of 1959.\nAppeal by Special Lea...

                                               Summary
0  The charge created in respect of municipal pro...
1  An agreement for a lease, which a lease is by ...
2  The question whether a Magistrate is "personal...
3  The appellant was a member of a joint Hindu fa...
4  The appellant was the Ruler of the State of Ba...
5  The appellants held personal inams which were ...
6  The appellant, who was working as a Loco Forem...
7  During the search of the premises of the appel...
```

```
8  The appellant was the Ruler of the State of Ba...
9  The appellant Corporation was assessed to sale...
```

```
print(df['Text'].at[0])
```

```
Appeal No. LXVI of 1949.
Appeal from the High Court of judicature, Bombay, in a reference under section 66 of the Indian Income tax Act, 1022.
K.M. Munshi (N. P. Nathvani, with him), for the appel lant. ' M.C. Setalvad, Attorney General for India (H. J. Umrigar, with him), for the respondent. 1950.
May 26.
The judgment of the Court was delivered by MEHR CHAND MAHAJAN J.
This is an appeal against a judgment of the High Court of Judicature at Bombay in an income tax matter and it raises the question whether munici pal property tax and urban immoveable property tax payable under the relevant Bombay Acts are allowable deductions under section 9 (1) (iv) of the Indian Income tax Act.
The assessee company is an investment company deriving its income from properties in the city of Bombay.
For the assessment year 1940 41 the net income of the assessee under the head "property" was computed by the Income tax Officer in the sum of Rs. 6,21,764 after deducting from gross rents certain payments.
The company had paid during the relevant year Rs. 1,22,675 as municipal property tax and Rs. 32,760 as urban property tax.
Deduction of these two sums was claimed under the provisions of section 9 the Act.
Out of the first item a deduction in the sum of Rs. 48,572 was allowed on the ground that this item represented tenants ' burdens paid by the assessee, otherwise the claim was disal lowed.
The, appeals of the assessee to the Appellate As sistant Commissioner and to the Income tax Appellate Tribu nal were unsuccessful.
The Tribunal, however, agreed to refer two questions of law to the High Court of Judicature at Bombay, namely, (1) Whether the municipal taxes paid by the applicant company are an allowable deduction under
555 the provisions of section 9 (1) (iv) of the Indian Income tax Act; (2) Whether the urban immoveable property taxes paid by the applicant company are an allowable deduction under section 9 (1) (iv) or under section 9 (1) (v) of the Indian Income tax Act.
A supplementary reference was made covering a third question which was not raised before us and it is not there fore necessary to refer to it.
The High Court answered all the three questions in the negative and hence this appeal.
The question for our determination is whether the munic ipal property tax and urban immoveable property tax can be deducted as an allowance under clause (iv) of sub section (1) of section 9 of the Act.
The decision of the point depends firstly on the construction of the language employed in sub clause (iv) of sub section (1) of section 9 of the Act, and secondly, on a finding as to the true nature and
```

character of the liability of the owner under the relevant Bombay Acts for the payment of these taxes.
Section 9 along with the relevant clause runs thus: (1) The tax shall be payable by an assessee under the head ' income from property ' in respect of the bona fide annual value of property consisting of any buildings or lands appurtenant thereto of Which he is the owner, . . subject to the following allowances, namely : (iv) where the property is subject to a mortgage or other capital charge, the amount of any interest on such mortgage or charge; where the property is subject to an annual charge not being a capital charge, the. amount of such charge; where the property is subject to a ground rent, the amount of such ground rent; and, where the property has been acquired, constructed, repaired, renewed or recon structed with borrowed capital, the amount of any interest payable on such capital; . . . "
It will be seen that clause (iv) consists of four sub clauses corresponding to the four deductions allowed 556 under the clause.
Before the amending Act of 1939, clause (iv) contained only the first, third and fourth sub clauses.
Under the first sub clause interest is deductible whether the amount borrowed on the security of the property was spent on the property or not.
There is no question of any capital or other expenditure on the property.
The expression "capital charge" in the sub clause cannot connote a charge on the capital, that is, the property assessed.
That would be a redundancy as the opening words themselves clearly indicate that the charge is on the property.
We are therefore of opinion that capital charge here could only mean a charge created for a capital sum, i.e., a charge to secure the discharge of a liability of a capital nature.
In 1933 the Privy Council decided the case of Bijoy Singh.
Dudhuria vs Commissioner of Income tax, Calcutta (1).
It was not an assessment under section 9 but an assess ment on the general income of an assessee who was liable to pay maintenance for his step mother which had been charged on all his assets by a decree of Court.
It was not a li ability voluntarily incurred by him but one cast on him by law.
The Privy Council held that the amount paid by him in discharge of that liability formed no part of his real income and so should not be included in his assessment.
Though the decision proceeded on the principle that the outgoings were not part of the assessee 's income at all, the framers of the amending Act of 1939 wanted, apparently, to extend the principle, so far as the assessment of property was concerned, even to cases where obligatory payments had to be made out of the assessee 's income from the property charged with such payments, and the second sub clause, namely, "where the property is subject to an annual charge not being a capital charge, the amount of such charge" was added.

It is this sub clause which the appellant invokes in support of its claim to deduction of the municipal and urban, property taxes in the present case.
In view of the opening words of the newly added sub clause, the expression "capital charge" also used therein cannot have reference to a charge on the property, and we think it must (1) I.L.R. 60 cal. 557 be understood in the same sense as in sub clause (1); that is to say, the first sub clause having provided for deduc tion of interest where a capital sum is charged on the property, this sub clause provides for a deduction of annual sums so charged, such sums not being capital sums, the limiting words being intended to exclude cases where capital raised on the security of the property is made repayable in instalments.
In Commissioner of Income tax, Bombay vs Mahomedbhoy Rowji (1), a Bench of the Bombay High Court considered the meaning of these words. As regards "annual charge," Beau mont C.J. observed as follows : "The words, I think, would cover a charge to secure an annual liability." Kania J., as he then was, said as follows : "I do not see how a charge can be annual unless it means a charge in respect of a payment to be made annually." This construction of the words has been followed in the judgment under appeal.
In Gappumal Kanhaiya Lal vs Commissioner of Income tax (2) (the connected appeal before us), the Bench of the Allahabad High Court agreed with the construction placed on these words in the Bombay case, i.e., the words "annual charge" mean a charge to secure an annual liability.
It is therefore clear that there is no conflict of judicial deci sions as to the meaning of the phrase "annual charge" occur ring in section 3 (1) (iv) and the meaning given is the natural meaning of these words.
As to the phrase "capital charge", Beaumont C.J. in the case above referred to took the view that the words mean a charge on capital. Kania J., however, took a different view and observed that he was not prepared to accept the sugges tion that a document which provides for a certain payment to be made monthly or annually and charged on immoveable property or the estate of an individual becomes a capital charge.
In the Allahabad judgment under appeal these (1) I.L.R. (2) I.L.R. 1944 All. 558 words were considered as not meaning a charge on capital.
It was said that if an annual charge means a charge to secure the discharge of an annual liability, then, capital charge means a charge to secure the discharge of a liability of a capital nature.
We think this construction is a natu ral construction of the section and is right.
The determination of the point whether the taxes in dispute fall within the ambit of the phrase "annual charge not being a capital charge" depends on the provisions of the statutes under which they are levied.

Section 143 of the City of Bombay Municipal Act, 1888, authorises the levy of a general tax on all buildings and lands in the city.
The primary responsibility to pay this property tax is on the lessor (vide section 146 of the Act).
In order to assess the tax provision has been made for the determination of the annual rateable value of the building in section 154.
Section 156 provides for the maintenance of an assessment book in which entries have to be made every official year of all buildings in the city, their rateable value, the names of persons primarily liable for payment of the property tax on such buildings and of the amount for which each building has been assessed.
Section 167 lays down that the assess ment book need not be prepared every official year but public notices shall be given in accordance with sections 160 to 162 every year and the provisions o+ the said sec tions and of sections 163 and 167 shall be applicable each year.
These sections lay down a procedure for hearing objections and complaints against entries in the assessment book.
From these provisions it is clear ' that the liabil ity for the tax is determined at the beginning of each official year and the tax is an annual one.
It recurs from year to year.
Sections 143to 168 concern themselves with the imposition, liability and assessment of the tax for the year.
The amount of the tax for the year and the liability for its payment having been determined, the Act then pre scribes for its collection in the chapter "The collection of taxes.
" Section 197 provides that each of the property taxes shall be payable in 559 advance in half yearly instalments on each first day of April and each first day of October.
The provision as to half yearly instalment necessarily connotes an annual li ability.
In other words, it means that the annual liability can be discharged by half yearly payments.
Procedure has also been prescribed for recovery of the instalments by presentment of a bill, a notice of demand and then distress, and sale.
Finally section 212 provides as follows : "Property taxes due under this Act in respect of any building or land shall, subject to the prior payment of the land revenue, if any, due to the provincial ,Government thereupon, be a first charge . . upon the said build ing or land . " It creates a statutory charge on the building.
Urban immove able property tax is leviable under section 22 of Part VI of the Bombay Finance Act, 1932,on the annual letting value of the property.
The duty to collect the tax is laid on the municipality and it does so in the same manner as in the case of the municipal property tax.
Section 24 (2) (b) is in terms similar to section 212 of the Bombay Municipal Act.
It makes the land or the building security for the payment of this tax

also.
For the purposes of section 9 of the Indian Income tax Act both these taxes, namely, the munici pal property tax as well as the urban immoveable property tax are of the same character and stand on the same foot ing.
Mr. Munshi, the learned counsel for the appellant con tended that both the taxes are assessed on the annual value of the land or the building and are annual taxes, although it may be that they are collected at intervals of six months for the sake of convenience, that the income tax itself is assessed on an annual basis, that in allowing deductions all payments made or all liabilities incurred during the previ ous year of assessment should be allowed and that the taxes in question fell clearly within the language of section 9 (1) (iv).
The learned Attorney General, on the other hand, argued that although the taxes are assessed for the year the liability to pay them arises at the beginning 560 of each half year and unless a notice of demand is issued and a bill presented there is no liability to pay them and that till then no charge under section 212 of the Act could possibly arise and that the liability to pay being half yearly in advance, the charge is not an annual charge.
It was also suggested that the taxes were a capital charge in the sense of the property being security for the payment.
We are satisfied that the contentions raised by the learned Attorney General are not sound.
It is apparent from the whole tenor of the two Bombay Acts that the taxes are in the nature of an annual levy on the property ' and are assessed on the annual value of the property each year.
The annual liability can be discharged by half yearly instalments.
The liability being an annual one and the property having been subjected to it, the provisions of clause (iv) of sub sec tion (1) of section 9 are immediately attracted.
Great emphasis was laid on the word"due" used in section 212 of the Municipal Act and it was said that as the taxes do not become due under the Act unless the time for the payment arrives, no charge comes into existence till then and that the charge is not an annual charge.
We do not think that this is a correct construction of section 212.
The words "property taxes due under this Act" mean property taxes for which a person is liable under the Act.
Taxes payable during the year have been made a charge on the property.
The liability and the charge both co exist and are co exten sive.
The provisions of the Act affording facilities for the discharge of the liability do not in any way affect their true nature and character.
If the annual liability is not discharged in the manner laid down by section 197, can it be said that the property cannot be sold for recovery of the whole amount due for the year ? The answer to this query can only be in the affirmative, i.e., that the proper ty is liable to sale.
In Commissioner of Income tax, Bombay vs Mahomedbhoy Rowji(1) Beaumont

C.J., while rejecting the claim for the deduction of the taxes, placed reliance on (1) I.L.R. 561 section 9 (1) (v) which allows a deduction in respect of any sums paid on account of land revenue.
It was observed that land revenue stands on the same footing as municipal taxes and that as the legislature made a special provision for deduction of sums payable in regard to land revenue but not in respect of sums paid on account of municipal taxes that circumstance indicated that the deduction was not allowable.
For the same purpose reference was also made to the provi sions of section 10 which deal with business allowances and wherein deduction of any sum paid on account of land reve nue, local rates or municipal taxes has been allowed.
In the concluding part of his judgment the learned Chief Jus tice said that it was not necessary for him to consider what the exact meaning of the words was and that it was suffi cient for him to say that it did not cover municipal taxes which are made a charge on the property under section 212 of the Bombay Municipal Act.
Without determining the exact meaning of the words used by the statute it seems to us it was not possible to arrive at the conclusion that the taxes were not within the ambit of the clause.
It is elementary that the primary duty of a Court is to give effect to the intention of the legislature as expressed in the words used by it and no outside consideration can be called in aid tO find that intention.
Again reference to clause (v) of the section is not very helpful because land revenue is a charge of a paramount nature on all buildings and lands and that being so, a deduction in respect of the amount was mentioned in express terms.
Municipal taxes, on the other hand, do not stand on the same footing as land revenue.
The law as to them varies from province to province and they may not be necessarily a charge on property in all cases.
The legis lature seems to have thought that so far as municipal taxes on property are concerned, if they fall within the ambit of clause (iv), deduction will be claimable in respect of them but not otherwise.
The deductions allowed in section 10 under the head "Income from business" proceed on a different footing and a construction of section 9 with the aid of section 10 is apt to mislead.
562 Kania J. in the above case in arriving at his conclusion was influenced by the consideration that these taxes were of a variable character, i.e., liable to be increased or re duced under the various provisions of the Municipal Act and that the charge was in the nature of a contingent charge.
With great respect, it may be pointed out that all charges in a way may be or are of a variable and contingent na ture.
If no default is made, no charge is ever enforceable and whenever there is a charge, it can be increased or reduced during the year either by payment or by additional borrowing.

In Moss Empires Ltd. vs Inland Revenue Commissioners (1) it was held by the House of Lords that the fact that certain payments were contingent and variable in amount did not affect their character of being annual payments and that the word, "annual" must be taken to have the quality of being recurrent or being capable of recurrence.
In Cunard 's Trustees vs Inland Revenue Commissioners (2) it was held that the payments were capable of being recur rent and were therefore annual payments within the meaning of schedule D, case III, rule 1 (1), even though they were not necessarily recurrent year by year and the fact that they varied in amount was immaterial.
The learned Attorney General in view of these decisions did not support the view expressed by Kania J. Reliance was placed on a decision of the High Court of Madras in Mamad Keyi vs Commissioner of Income tax, Madras(3), in which moneys paid as urban immoveable property tax under the Bombay Finance Act were disallowed as inadmis sible under section 9 (1) (iv) or 9 (1) (v) of the Indian Income tax Act. 'This decision merely followed the view expressed in Commissioner of income tax, Bombay vs Mahomedb hoy Rowji (4)and was not arrived at on any independent or fresh reasoning and is not of much assistance in the deci sion of the case.
The Allahabad High Court (1) (2) [1948] 1 A.E.R. 150. (3) I.L.R. (4) I.L.R. 563 in Gappumal Kanhaiya Lal vs Commissioner of Incometax (1) (the connected appeal) took a correct view of this matter and the reasoning given therein has our approval.
The result is that this appeal is allowed and the two questions which were referred to the High Court by the Income tax Tribunal and cited above are answered in the affirmative.
The appellants will have their costs in the appeal.
Appeal allowed.

```
print(df['Summary'].at[0])
```

The charge created in respect of municipal property tax by section 212 of the City of Bombay Municipal Act, 1888, is an "annual charge not being a capital charge" within the mean ing of section 9 (1) (iv) of the Indian Income tax Act, 199.2, and the amount of such charge should therefore be deducted in computing the income from such property for the purposes of section 9 of the Indian Income tax Act.
The charge in respect of urban immoveable property tax created by the Bombay Finance Act, 1939 is similar in character and the amount of such charge should also be deducted.
The expression "capital charge" in s.9(1) (iv) means a charge created for a capital sum,that is to say, a charge created to. ' secure the discharge of a liability of a capi tal nature; and an "annual charge" means a charge to secure an annual liabili ty. 554

If df contains data from the SAMSum dataset, you'll see the first 5 rows, likely including columns such as dialogue and summary, which are essential for summarization tasks.

```
df.head()
```

```
                                                Text  \
0  Appeal No. LXVI of 1949.\nAppeal from the High...
1  Civil Appeal No.94 of 1949.\n107 834 Appeal fr...
2  iminal Appeal No. 40 of 1951, 127 Appeal from ...
3  Appeal No. 388 of 1960.\nAppeal by special lea...
4  Appeal No. 198 of 1954.\nAppeal from the judgm...

                                             Summary
0  The charge created in respect of municipal pro...
1  An agreement for a lease, which a lease is by ...
2  The question whether a Magistrate is "personal...
3  The appellant was a member of a joint Hindu fa...
4  The appellant was the Ruler of the State of Ba...
```

If df contains data from the SAMSum dataset, you'll see the last 5 rows, likely including columns such as dialogue and summary, which are essential for summarization tasks.

```
df.tail()
```

```
                                                Text  \
5  Appeals Nos. 155 to 160 of 1956.\nAppeals from...
6  iminal Appeal No. 68 of 1958.\nAppeal by speci...
7  minal Appeal No. 124 of 1959.\nAppeal by speci...
8  Appeal No. 198 of 1954.\nAppeal from the judgm...
9  Appeal No. 285 of 1959.\nAppeal by Special Lea...

                                             Summary
5  The appellants held personal inams which were ...
6  The appellant, who was working as a Loco Forem...
7  During the search of the premises of the appel...
8  The appellant was the Ruler of the State of Ba...
9  The appellant Corporation was assessed to sale...
```

If df contains the SAMSum dataset, you'll see 5 randomly selected rows, which might include columns like dialogue (the conversational text) and summary (the corresponding generated summary).

```
df.sample(5)
```

```
                                                Text  \
2  iminal Appeal No. 40 of 1951, 127 Appeal from ...
0  Appeal No. LXVI of 1949.\nAppeal from the High...
8  Appeal No. 198 of 1954.\nAppeal from the judgm...
4  Appeal No. 198 of 1954.\nAppeal from the judgm...
9  Appeal No. 285 of 1959.\nAppeal by Special Lea...
```

```
                                                Summary
2  The question whether a Magistrate is "personal...
0  The charge created in respect of municipal pro...
8  The appellant was the Ruler of the State of Ba...
4  The appellant was the Ruler of the State of Ba...
9  The appellant Corporation was assessed to sale...
```

The df.info() command in Pandas provides a concise summary of the DataFrame, giving you an overview of its structure and data. It is particularly useful for understanding the dataset at a glance.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 2 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   Text     10 non-null     object
 1   Summary  10 non-null     object
dtypes: object(2)
memory usage: 288.0+ bytes
```

Data Cleaning

The line df.isna().sum() checks for missing values in each column of the DataFrame df. It returns the total count of NaN (Not a Number) or missing values per column, helping identify where data cleaning is needed.

```
df.isna().sum()
```

```
Text       0
Summary    0
dtype: int64
```

The line df = df.dropna() removes any rows from the DataFrame df that contain missing (NaN) values in any of the columns. After this operation, the DataFrame is updated to exclude those rows, ensuring that all remaining rows contain complete data.

```
df = df.dropna()

df
```

```
                                                Text  \
0  Appeal No. LXVI of 1949.\nAppeal from the High...
1  Civil Appeal No.94 of 1949.\n107 834 Appeal fr...
```

```
2  iminal Appeal No. 40 of 1951, 127 Appeal from ...
3  Appeal No. 388 of 1960.\nAppeal by special lea...
4  Appeal No. 198 of 1954.\nAppeal from the judgm...
5  Appeals Nos. 155 to 160 of 1956.\nAppeals from...
6  iminal Appeal No. 68 of 1958.\nAppeal by speci...
7  minal Appeal No. 124 of 1959.\nAppeal by speci...
8  Appeal No. 198 of 1954.\nAppeal from the judgm...
9  Appeal No. 285 of 1959.\nAppeal by Special Lea...

                                                Summary
0  The charge created in respect of municipal pro...
1  An agreement for a lease, which a lease is by ...
2  The question whether a Magistrate is "personal...
3  The appellant was a member of a joint Hindu fa...
4  The appellant was the Ruler of the State of Ba...
5  The appellants held personal inams which were ...
6  The appellant, who was working as a Loco Forem...
7  During the search of the premises of the appel...
8  The appellant was the Ruler of the State of Ba...
9  The appellant Corporation was assessed to sale...
```

Counts the number of missing values (NaN) in each column of the DataFrame df.

```
df.isna().sum()

Text       0
Summary    0
dtype: int64
```

Installs the contractions library, which is used to expand contracted words (e.g., "I'm" to "I am") in text, useful for text preprocessing in natural language processing (NLP) tasks.

```
%pip install contractions

Requirement already satisfied: contractions in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (0.1.73)
Requirement already satisfied: textsearch>=0.0.21 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
contractions) (0.0.24)
Requirement already satisfied: anyascii in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (from textsearch>=0.0.21-
>contractions) (0.3.2)
Requirement already satisfied: pyahocorasick in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
textsearch>=0.0.21->contractions) (2.1.0)
Note: you may need to restart the kernel to use updated packages.
```

The code imports several libraries: re for regular expressions, spacy for natural language processing tasks, emoji for handling emojis, and fix from the contractions library to expand

contractions. It loads the English language model from spacy for text processing. Two functions are defined for cleaning text: clean_dialogue processes the dialogue by removing special tags, replacing newlines with spaces, converting text to lowercase, expanding contractions, demojizing emojis, and lemmatizing the text, while clean_summary cleans the summary by removing extra spaces, replacing newlines, and expanding contractions. These functions are then applied to the "dialogue" and "summary" columns of the DataFrame df, and the cleaned versions are stored in new columns cleaned_dialogue and cleaned_summary, respectively.

```
%pip install spacy emoji


Requirement already satisfied: spacy in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (3.8.3)
Requirement already satisfied: emoji in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (2.14.1)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in c:\
users\kushal\onedrive\desktop\new folder\chintya\lib\site-packages
(from spacy) (3.0.12)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in c:\
users\kushal\onedrive\desktop\new folder\chintya\lib\site-packages
(from spacy) (1.0.5)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
spacy) (1.0.12)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from spacy)
(2.0.11)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
spacy) (3.0.9)
Requirement already satisfied: thinc<8.4.0,>=8.3.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from spacy)
(8.3.4)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
spacy) (1.1.3)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from spacy)
(2.5.1)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
spacy) (2.0.10)
Requirement already satisfied: weasel<0.5.0,>=0.1.0 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
spacy) (0.4.1)
Requirement already satisfied: typer<1.0.0,>=0.3.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from spacy)
(0.15.2)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from spacy)
```

```
(4.67.1)
Requirement already satisfied: numpy>=1.19.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from spacy)
(2.0.2)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
spacy) (2.32.3)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in
c:\users\kushal\onedrive\desktop\new folder\chintya\lib\site-packages
(from spacy) (2.10.6)
Requirement already satisfied: jinja2 in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (from spacy) (3.1.5)
Requirement already satisfied: setuptools in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (from spacy) (58.1.0)
Requirement already satisfied: packaging>=20.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from spacy)
(24.2)
Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
spacy) (3.5.0)
Requirement already satisfied: language-data>=1.2 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
langcodes<4.0.0,>=3.2.0->spacy) (1.3.0)
Requirement already satisfied: annotated-types>=0.6.0 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy) (0.7.0)
Requirement already satisfied: pydantic-core==2.27.2 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy) (2.27.2)
Requirement already satisfied: typing-extensions>=4.12.2 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy) (4.12.2)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
requests<3.0.0,>=2.13.0->spacy) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
requests<3.0.0,>=2.13.0->spacy) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
requests<3.0.0,>=2.13.0->spacy) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
requests<3.0.0,>=2.13.0->spacy) (2025.1.31)
Requirement already satisfied: blis<1.3.0,>=1.2.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
thinc<8.4.0,>=8.3.0->spacy) (1.2.0)
Requirement already satisfied: confection<1.0.0,>=0.0.1 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
```

```
thinc<8.4.0,>=8.3.0->spacy) (0.1.5)
Requirement already satisfied: colorama in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (from
tqdm<5.0.0,>=4.38.0->spacy) (0.4.6)
Requirement already satisfied: click>=8.0.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
typer<1.0.0,>=0.3.0->spacy) (8.1.8)
Requirement already satisfied: shellingham>=1.3.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
typer<1.0.0,>=0.3.0->spacy) (1.5.4)
Requirement already satisfied: rich>=10.11.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from
typer<1.0.0,>=0.3.0->spacy) (13.9.4)
Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
weasel<0.5.0,>=0.1.0->spacy) (0.21.0)
Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
weasel<0.5.0,>=0.1.0->spacy) (7.1.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from jinja2-
>spacy) (3.0.2)
Requirement already satisfied: marisa-trie>=1.1.0 in c:\users\kushal\
onedrive\desktop\new folder\chintya\lib\site-packages (from language-
data>=1.2->langcodes<4.0.0,>=3.2.0->spacy) (1.2.1)
Requirement already satisfied: markdown-it-py>=2.2.0 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
rich>=10.11.0->typer<1.0.0,>=0.3.0->spacy) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in c:\users\
kushal\onedrive\desktop\new folder\chintya\lib\site-packages (from
rich>=10.11.0->typer<1.0.0,>=0.3.0->spacy) (2.19.1)
Requirement already satisfied: wrapt in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (from smart-
open<8.0.0,>=5.2.1->weasel<0.5.0,>=0.1.0->spacy) (1.17.2)
Requirement already satisfied: mdurl~=0.1 in c:\users\kushal\onedrive\
desktop\new folder\chintya\lib\site-packages (from markdown-it-
py>=2.2.0->rich>=10.11.0->typer<1.0.0,>=0.3.0->spacy) (0.1.2)
Note: you may need to restart the kernel to use updated packages.
```

```python
import re
import spacy
import emoji
from contractions import fix

nlp = spacy.load("en_core_web_sm")

# Clean Dialogue (Input)
def clean_dialogue(dialogue):
    dialogue = re.sub(r"<.*?>", "", dialogue)  # Remove special tags
    dialogue = re.sub(r"\\r\\n", " ", dialogue)  # Replace newlines
```

```
    dialogue = re.sub(r"\s+", " ", dialogue).strip()  # Remove extra
spaces
    dialogue = dialogue.lower()  # Convert to lowercase
    dialogue = fix(dialogue)  # Expand contractions
    dialogue = emoji.demojize(dialogue)  # Convert emojis to text
    doc = nlp(dialogue)
    dialogue = " ".join([token.lemma_ for token in doc if not
token.is_punct])  # Lemmatize
    return dialogue

# Clean Summary (Label)
def clean_summary(summary):
    summary = re.sub(r"\s+", " ", summary).strip()  # Remove extra
spaces
    summary = re.sub(r"\\r\\n", " ", summary)  # Replace newlines
    summary = fix(summary)  # Expand contractions
    return summary

# Apply Cleaning
df["cleaned_dialogue"] = df["Text"].apply(clean_dialogue)
df["cleaned_summary"] = df["Summary"].apply(clean_summary)
```

```
---------------------------------------------------------------------------
-----
KeyboardInterrupt                         Traceback (most recent call
last)
Cell In[17], line 2
      1 import re
----> 2 import spacy
      3 import emoji
      4 from contractions import fix

File c:\Users\Kushal\OneDrive\Desktop\New folder\chintya\lib\site-
packages\spacy\__init__.py:6
      3 from typing import Any, Dict, Iterable, Union
      5 # set library-specific custom warning handling before doing
anything else
----> 6 from .errors import setup_default_warnings
      8 setup_default_warnings()  # noqa: E402
     10 # These are imported as part of the API

File c:\Users\Kushal\OneDrive\Desktop\New folder\chintya\lib\site-
packages\spacy\errors.py:3
      1 import warnings
----> 3 from .compat import Literal
      6 class ErrorsWithCodes(type):
      7     def __getattribute__(self, code):

File c:\Users\Kushal\OneDrive\Desktop\New folder\chintya\lib\site-
packages\spacy\compat.py:4
```

```
      1 """Helpers for Python and platform compatibility."""
      2 import sys
----> 4 from thinc.util import copy_array
      6 try:
      7     import cPickle as pickle

File c:\Users\Kushal\OneDrive\Desktop\New folder\chintya\lib\site-
packages\thinc\__init__.py:5
      2 import numpy
      4 from .about import __version__
----> 5 from .config import registry
      7 # fmt: off
      8 __all__ = [
      9     "registry",
     10     "__version__",
     11 ]

File c:\Users\Kushal\OneDrive\Desktop\New folder\chintya\lib\site-
packages\thinc\config.py:5
      2 import confection
      3 from confection import VARIABLE_RE, Config,
ConfigValidationError, Promise
----> 5 from .types import Decorator
      8 class registry(confection.registry):
      9     # fmt: off
     10     optimizers: Decorator = catalogue.create("thinc",
"optimizers", entry_points=True)

File c:\Users\Kushal\OneDrive\Desktop\New folder\chintya\lib\site-
packages\thinc\types.py:25
      4 from typing import (
      5     Any,
      6     Callable,
   (...)
     20     overload,
     21 )
     23 import numpy
---> 25 from .compat import cupy, has_cupy
     27 if has_cupy:
     28     get_array_module = cupy.get_array_module

File c:\Users\Kushal\OneDrive\Desktop\New folder\chintya\lib\site-
packages\thinc\compat.py:39
     36 import torch.utils.dlpack
     38 has_torch = True
---> 39 has_torch_cuda_gpu = torch.cuda.device_count() != 0
     40 has_torch_mps = hasattr(torch.backends, "mps") and
torch.backends.mps.is_built()
     41 has_torch_mps_gpu = has_torch_mps and
torch.backends.mps.is_available()
```

```
File c:\Users\Kushal\OneDrive\Desktop\New folder\chintya\lib\site-
packages\torch\cuda\__init__.py:909, in device_count()
    905 r = torch._C._cuda_getDeviceCount() if nvml_count < 0 else
nvml_count
    906 # NB: Do not cache the device count prior to CUDA
initialization, because
    907 # the number of devices can change due to changes to
CUDA_VISIBLE_DEVICES
    908 # setting prior to CUDA initialization.
--> 909 if _initialized:
    910     _cached_device_count = r
    911 return r

KeyboardInterrupt:
```

The command df simply displays the contents of the DataFrame df in a Jupyter notebook or a Kaggle environment. It shows the dataset with its columns and rows, including any transformations or modifications applied to it.

```
df

                                                   Text  \
0    Appeal No. LXVI of 1949.\nAppeal from the High...
1    Civil Appeal No.94 of 1949.\n107 834 Appeal fr...
2    iminal Appeal No. 40 of 1951, 127 Appeal from ...
3    Appeal No. 388 of 1960.\nAppeal by special lea...
4    Appeal No. 198 of 1954.\nAppeal from the judgm...
..                                                 ...
995  ivil Appeal No. 414 of 1965.\nAppeal from the ...
996  ivil Appeal No. 195 of 1963.\nAppeal from the ...
997  Appeal No. 765 of1964.\nAppeal by special leav...
998  ivil Appeal No. 875 of 1964.\nAppeal by specia...
999  ppeal No. 652 of 1964, Appeal from the judgmen...

                                                Summary  \
0    The charge created in respect of municipal pro...
1    An agreement for a lease, which a lease is by ...
2    The question whether a Magistrate is "personal...
3    The appellant was a member of a joint Hindu fa...
4    The appellant was the Ruler of the State of Ba...
..                                                 ...
995  The appellant .company reduced its capital and...
996  A widow whose estate was under the charge of t...
997  The appellant 's election was challenged inter...
998  The appellant and respondent were the tenant a...
999  The Sales Tax Officer rejected the assessed 's...

                                       cleaned_dialogue  \
```

```
0    appeal no lxvi of 1949 appeal from the high co...
1    civil appeal no.94 of 1949 107 834 appeal from...
2    iminal appeal no 40 of 1951 127 appeal from th...
3    appeal no 388 of 1960 appeal by special leave ...
4    appeal no 198 of 1954 appeal from the judgment...
..                                                 ...
995  ivil appeal no 414 of 1965 appeal from the jud...
996  ivil appeal no 195 of 1963 appeal from the jud...
997  appeal no 765 of1964 appeal by special leave f...
998  ivil appeal no 875 of 1964 appeal by special l...
999  ppeal no 652 of 1964 appeal from the judgment ...

                                        cleaned_summary
0    The charge created in respect of municipal pro...
1    An agreement for a lease, which a lease is by ...
2    The question whether a Magistrate is "personal...
3    The appellant was a member of a joint Hindu fa...
4    The appellant was the Ruler of the State of Ba...
..                                                 ...
995  The appellant .company reduced its capital and...
996  A widow whose estate was under the charge of t...
997  The appellant 's election was challenged inter...
998  The appellant and respondent were the tenant a...
999  The Sales Tax Officer rejected the assessed 's...

[1000 rows x 4 columns]
```

The code defines three functions to compute text features: word_count, unique_word_count, and avg_word_length. These functions calculate the total number of words, the number of unique words, and the average length of words in a text, respectively. Then, the code applies these functions to the cleaned dialogue and summary columns of the DataFrame df and stores the results in new columns:

dialogue_word_count: The total word count of the cleaned dialogue. summary_word_count: The total word count of the cleaned summary. dialogue_unique_word_count: The number of unique words in the cleaned dialogue. summary_unique_word_count: The number of unique words in the cleaned summary. dialogue_avg_word_length: The average word length in the cleaned dialogue. summary_avg_word_length: The average word length in the cleaned summary.

```python
# Function to calculate word count
def word_count(text):
    return len(text.split())
def char_count(text):
    return len(text)

# Function to calculate unique word count
def unique_word_count(text):
    return len(set(text.split()))
```

```python
# Function to calculate average word length
def avg_word_length(text):
    words = text.split()
    if len(words) > 0:
        return sum(len(word) for word in words) / len(words)
    else:
        return 0

# Add columns to the DataFrame
df['dialogue_word_count'] = df['cleaned_dialogue'].apply(word_count)
df['summary_word_count'] = df['cleaned_summary'].apply(word_count)
df['dialogue_char_count'] = df['cleaned_dialogue'].apply(char_count)
df['summary_char_count'] = df['cleaned_summary'].apply(char_count)

df['dialogue_unique_word_count'] =
df['cleaned_dialogue'].apply(unique_word_count)
df['summary_unique_word_count'] =
df['cleaned_summary'].apply(unique_word_count)

df['dialogue_avg_word_length'] =
df['cleaned_dialogue'].apply(avg_word_length)
df['summary_avg_word_length'] =
df['cleaned_summary'].apply(avg_word_length)

df.head()
```

```
                                                Text  \
0  Appeal No. LXVI of 1949.\nAppeal from the High...
1  Civil Appeal No.94 of 1949.\n107 834 Appeal fr...
2  iminal Appeal No. 40 of 1951, 127 Appeal from ...
3  Appeal No. 388 of 1960.\nAppeal by special lea...
4  Appeal No. 198 of 1954.\nAppeal from the judgm...

                                             Summary  \
0  The charge created in respect of municipal pro...
1  An agreement for a lease, which a lease is by ...
2  The question whether a Magistrate is "personal...
3  The appellant was a member of a joint Hindu fa...
4  The appellant was the Ruler of the State of Ba...

                                     cleaned_dialogue  \
0  appeal no lxvi of 1949 appeal from the high co...
1  civil appeal no.94 of 1949 107 834 appeal from...
2  iminal appeal no 40 of 1951 127 appeal from th...
3  appeal no 388 of 1960 appeal by special leave ...
4  appeal no 198 of 1954 appeal from the judgment...

                                      cleaned_summary
dialogue_word_count  \
0  The charge created in respect of municipal pro...
```

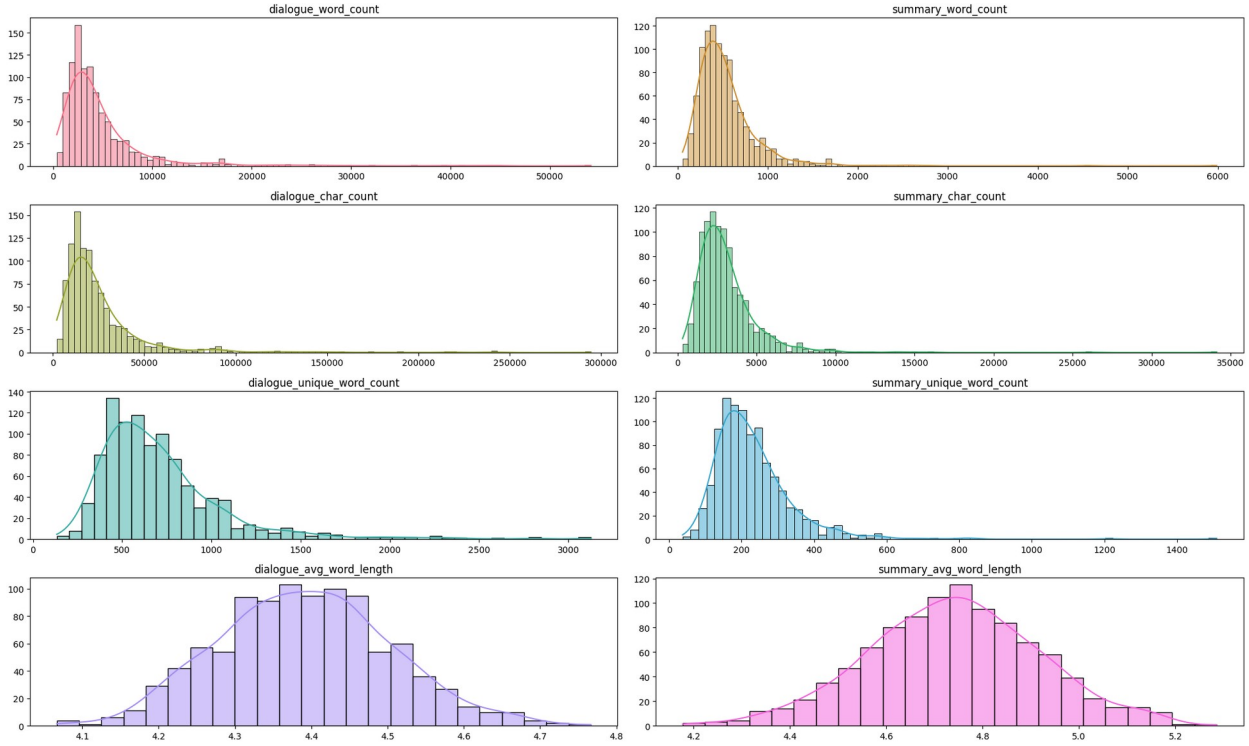
```
3356
1  An agreement for a lease, which a lease is by ...
2396
2  The question whether a Magistrate is "personal...
3213
3  The appellant was a member of a joint Hindu fa...
3497
4  The appellant was the Ruler of the State of Ba...
2575

   summary_word_count  dialogue_char_count  summary_char_count  \
0                 152                17786                 836
1                 153                12595                 799
2                 348                17541                2039
3                 216                19092                1278
4                 229                13878                1275

   dialogue_unique_word_count  summary_unique_word_count  \
0                         650                         75
1                         561                        102
2                         727                        180
3                         601                         96
4                         450                        105

   dialogue_avg_word_length  summary_avg_word_length
0                  4.300060                 4.506579
1                  4.257095                 4.228758
2                  4.459695                 4.862069
3                  4.459823                 4.921296
4                  4.389903                 4.572052
```

Data Visualization

The code generates a series of histograms to visualize various text features in the DataFrame df, including word counts, character counts, and average word lengths for both the dialogue and summary columns. It starts by defining a list of columns to visualize and assigns distinct colors to each plot using Seaborn's husl palette. A grid of subplots (4 rows, 2 columns) is created, and a histogram with a Kernel Density Estimate (KDE) is plotted for each feature. The loop iterates through each feature, customizing the titles and axis labels for each subplot. Any empty subplots are removed if there are fewer columns than subplots. Finally, the layout is adjusted to ensure proper spacing and the grid of histograms is displayed, providing a clear and visually distinct comparison of the features.

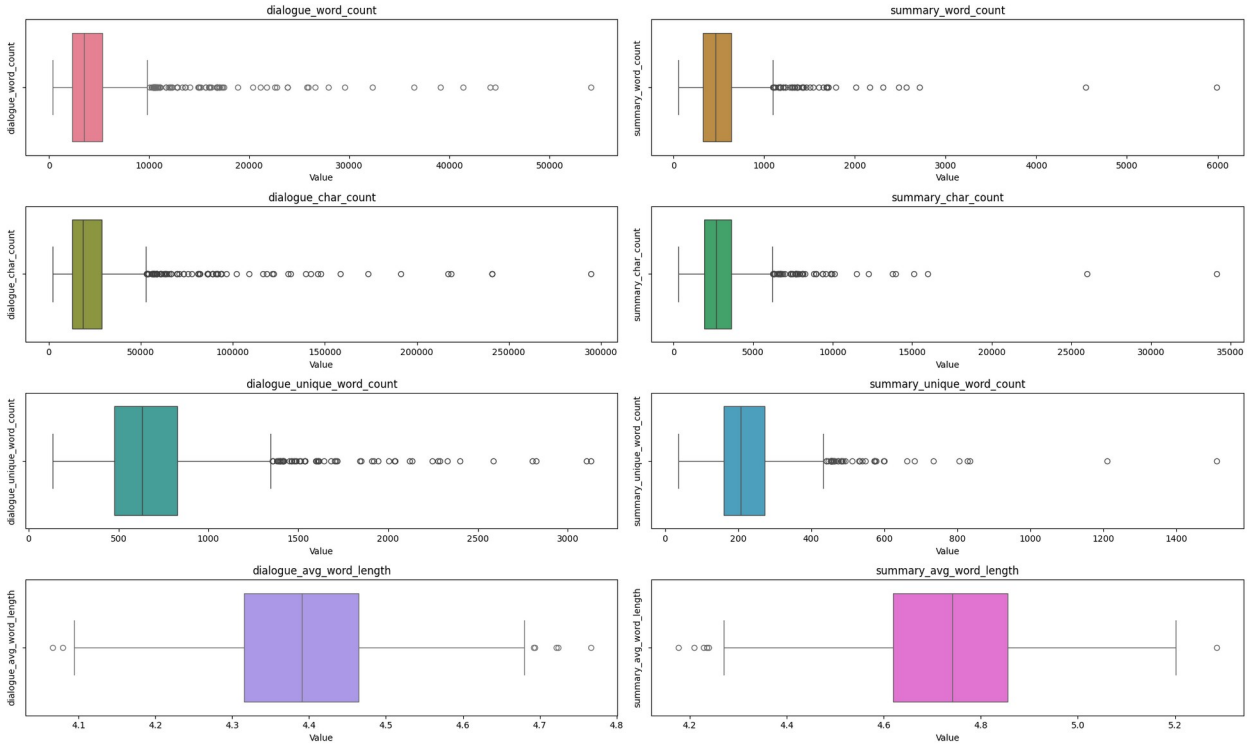
```
columns = [
    "dialogue_word_count", "summary_word_count",
"dialogue_char_count", "summary_char_count",
"dialogue_unique_word_count", "summary_unique_word_count",
```

```
    "dialogue_avg_word_length","summary_avg_word_length"
]

# Define a list of unique colors (length should match or exceed the number of columns)
colors = sns.color_palette("husl", len(columns))  # 'husl' generates visually distinct colors

fig, axes = plt.subplots(4, 2, figsize=(20, 12))
axes = axes.flatten()

# Plot each column in the grid
for i, column in enumerate(columns):
    sns.histplot(df[column], kde=True, ax=axes[i], color=colors[i])
    axes[i].set_title(column)
    axes[i].set_xlabel("")
    axes[i].set_ylabel("")

# Remove any empty subplots
for j in range(len(columns), len(axes)):
    fig.delaxes(axes[j])

# Adjust layout
plt.tight_layout()
plt.show()
```



The code creates a series of horizontal box plots to visualize the distribution of various text features in the DataFrame `df`, including word counts, character counts, and average word lengths for both dialogue and summary columns. It starts by defining a list of columns to be plotted and assigns distinct colors to each box plot using Seaborn's `husl` color palette. A grid of subplots (4 rows and 2 columns) is created, and for each column in the list, a box plot is generated. The `sns.boxplot` function is used to display the distribution of values for each feature, with the column name displayed as the y-axis label and a generic "Value" label for the x-axis. Any empty subplots are removed if there are fewer columns than subplots. Finally, the layout is adjusted to ensure proper spacing between the plots, and the resulting box plots are displayed, allowing for easy comparison of the distributions of the features.

```
# Define a list of unique colors (length should match or exceed the
number of columns)
colors = sns.color_palette("husl", len(columns))  # 'husl' generates
visually distinct colors

fig, axes = plt.subplots(4, 2, figsize=(20, 12))
axes = axes.flatten()

# Plot each column in the grid
for i, column in enumerate(columns):
    sns.boxplot(data=df, x=column, ax=axes[i], color=colors[i])  # Set
`x=column` for horizontal box plots
    axes[i].set_title(column)
    axes[i].set_xlabel("Value")  # Generic x-axis label for all plots
    axes[i].set_ylabel(column)  # Show the column name on the y-axis

# Remove any empty subplots
for j in range(len(columns), len(axes)):
    fig.delaxes(axes[j])

# Adjust layout
plt.tight_layout()
plt.show()
```



The command `df.describe()` provides a summary of the numerical columns in the DataFrame `df`. It generates descriptive statistics such as:

- **Count**: The number of non-null entries in each column.
- **Mean**: The average value of each column.
- **Standard Deviation (std)**: The measure of the amount of variation or dispersion of values in each column.
- **Min**: The minimum value in each column.
- **25%** (1st Quartile): The value below which 25% of the data falls.
- **50%** (Median or 2nd Quartile): The middle value in the data set, where half the values are above and half are below.
- **75%** (3rd Quartile): The value below which 75% of the data falls.
- **Max**: The maximum value in each column.

This function is typically used to get a quick overview of the distribution and central tendencies of numerical data in the dataset. It can help identify outliers or skewed distributions as well.

```
df.describe()
```

```
       dialogue_word_count  summary_word_count  dialogue_char_count  \
count          1000.000000         1000.00000          1000.000000
mean           4759.541000          538.10300         25690.313000
std            4865.639601          375.12402         26354.502792
min             378.000000           56.00000          2122.000000
25%            2312.500000          327.00000         12508.000000
50%            3481.500000          461.00000         18632.500000
```

```
75%            5357.500000           635.00000         28597.250000
max           54107.000000          5988.00000        294508.000000

       summary_char_count  dialogue_unique_word_count  \
count         1000.000000                 1000.000000
mean          3087.407000                  711.528000
std           2152.445708                  358.669537
min            312.000000                  134.000000
25%           1907.000000                  478.000000
50%           2657.500000                  631.000000
75%           3643.250000                  828.250000
max          34123.000000                 3127.000000

       summary_unique_word_count  dialogue_avg_word_length  \
count                1000.000000               1000.000000
mean                  230.510000                  4.391900
std                   112.533434                  0.112548
min                    37.000000                  4.066369
25%                   160.750000                  4.315184
50%                   208.000000                  4.390601
75%                   272.000000                  4.464301
max                  1510.000000                  4.766496

       summary_avg_word_length
count              1000.000000
mean                  4.739804
std                   0.178140
min                   4.177570
25%                   4.618849
50%                   4.741301
75%                   4.855223
max                   5.285714

df

                                                  Text  \
0    Appeal No. LXVI of 1949.\nAppeal from the High...
1    Civil Appeal No.94 of 1949.\n107 834 Appeal fr...
2    iminal Appeal No. 40 of 1951, 127 Appeal from ...
3    Appeal No. 388 of 1960.\nAppeal by special lea...
4    Appeal No. 198 of 1954.\nAppeal from the judgm...
..                                                 ...
995  ivil Appeal No. 414 of 1965.\nAppeal from the ...
996  ivil Appeal No. 195 of 1963.\nAppeal from the ...
997  Appeal No. 765 of1964.\nAppeal by special leav...
998  ivil Appeal No. 875 of 1964.\nAppeal by specia...
999  ppeal No. 652 of 1964, Appeal from the judgmen...

                                               Summary  \
0    The charge created in respect of municipal pro...
```

```
1    An agreement for a lease, which a lease is by ...
2    The question whether a Magistrate is "personal...
3    The appellant was a member of a joint Hindu fa...
4    The appellant was the Ruler of the State of Ba...
..                                                 ...
995  The appellant .company reduced its capital and...
996  A widow whose estate was under the charge of t...
997  The appellant 's election was challenged inter...
998  The appellant and respondent were the tenant a...
999  The Sales Tax Officer rejected the assessed 's...

                                        cleaned_dialogue  \
0    appeal no lxvi of 1949 appeal from the high co...
1    civil appeal no.94 of 1949 107 834 appeal from...
2    iminal appeal no 40 of 1951 127 appeal from th...
3    appeal no 388 of 1960 appeal by special leave ...
4    appeal no 198 of 1954 appeal from the judgment...
..                                                 ...
995  ivil appeal no 414 of 1965 appeal from the jud...
996  ivil appeal no 195 of 1963 appeal from the jud...
997  appeal no 765 of1964 appeal by special leave f...
998  ivil appeal no 875 of 1964 appeal by special l...
999  ppeal no 652 of 1964 appeal from the judgment ...

                                         cleaned_summary
dialogue_word_count  \
0    The charge created in respect of municipal pro...
3356
1    An agreement for a lease, which a lease is by ...
2396
2    The question whether a Magistrate is "personal...
3213
3    The appellant was a member of a joint Hindu fa...
3497
4    The appellant was the Ruler of the State of Ba...
2575
..                                                 ...
...
995  The appellant .company reduced its capital and...
6762
996  A widow whose estate was under the charge of t...
2893
997  The appellant 's election was challenged inter...
2744
998  The appellant and respondent were the tenant a...
5767
999  The Sales Tax Officer rejected the assessed 's...
2655

     summary_word_count  dialogue_char_count  summary_char_count  \
```

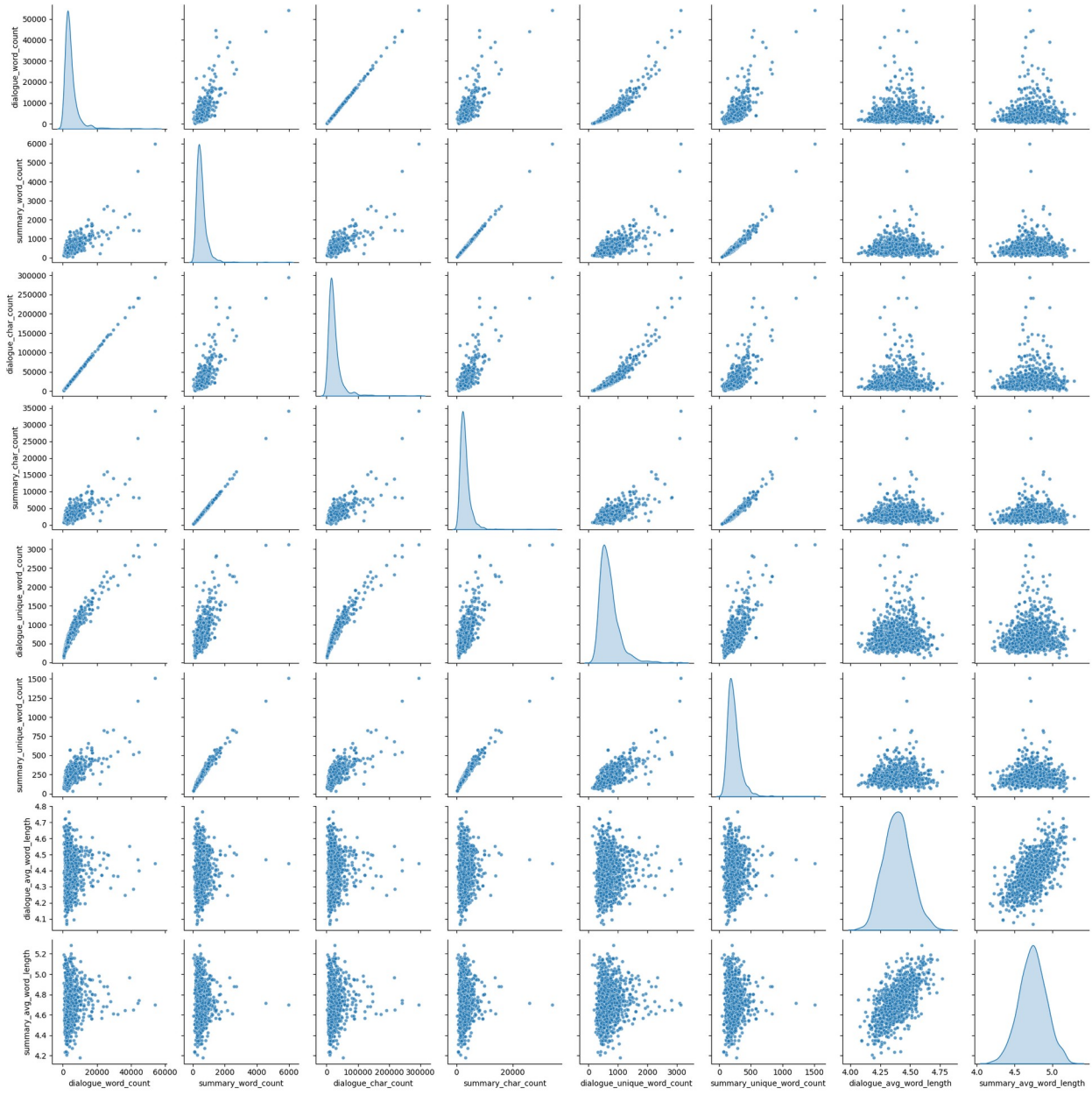
```
0                    152                 17786                  836
1                    153                 12595                  799
2                    348                 17541                 2039
3                    216                 19092                 1278
4                    229                 13878                 1275
..                   ...                   ...                  ...
995                  971                 37134                 5685
996                  537                 15238                 2971
997                  269                 14993                 1541
998                  584                 30733                 3336
999                  542                 14396                 3079

     dialogue_unique_word_count  summary_unique_word_count  \
0                           650                        75
1                           561                       102
2                           727                       180
3                           601                        96
4                           450                       105
..                          ...                       ...
995                         887                       399
996                         594                       221
997                         529                       146
998                         797                       223
999                         546                       228

     dialogue_avg_word_length  summary_avg_word_length
0                    4.300060                 4.506579
1                    4.257095                 4.228758
2                    4.459695                 4.862069
3                    4.459823                 4.921296
4                    4.389903                 4.572052
..                        ...                      ...
995                  4.491718                 4.854789
996                  4.267542                 4.534451
997                  4.464286                 4.732342
998                  4.329287                 4.714041
999                  4.422599                 4.682657

[1000 rows x 12 columns]
```

Data Correlations

The code generates a **pairplot** to visualize the relationships between the selected numerical columns in the DataFrame `df`. First, it creates a new DataFrame `correlation_data` by selecting only the relevant columns defined in the `columns` list. Then, using Seaborn's `pairplot` function, it creates a grid of scatterplots to explore how each pair of variables relates to one another, with Kernel Density Estimation (KDE) plots shown along the diagonal to visualize the distribution of individual variables. The transparency of the scatterplot markers is

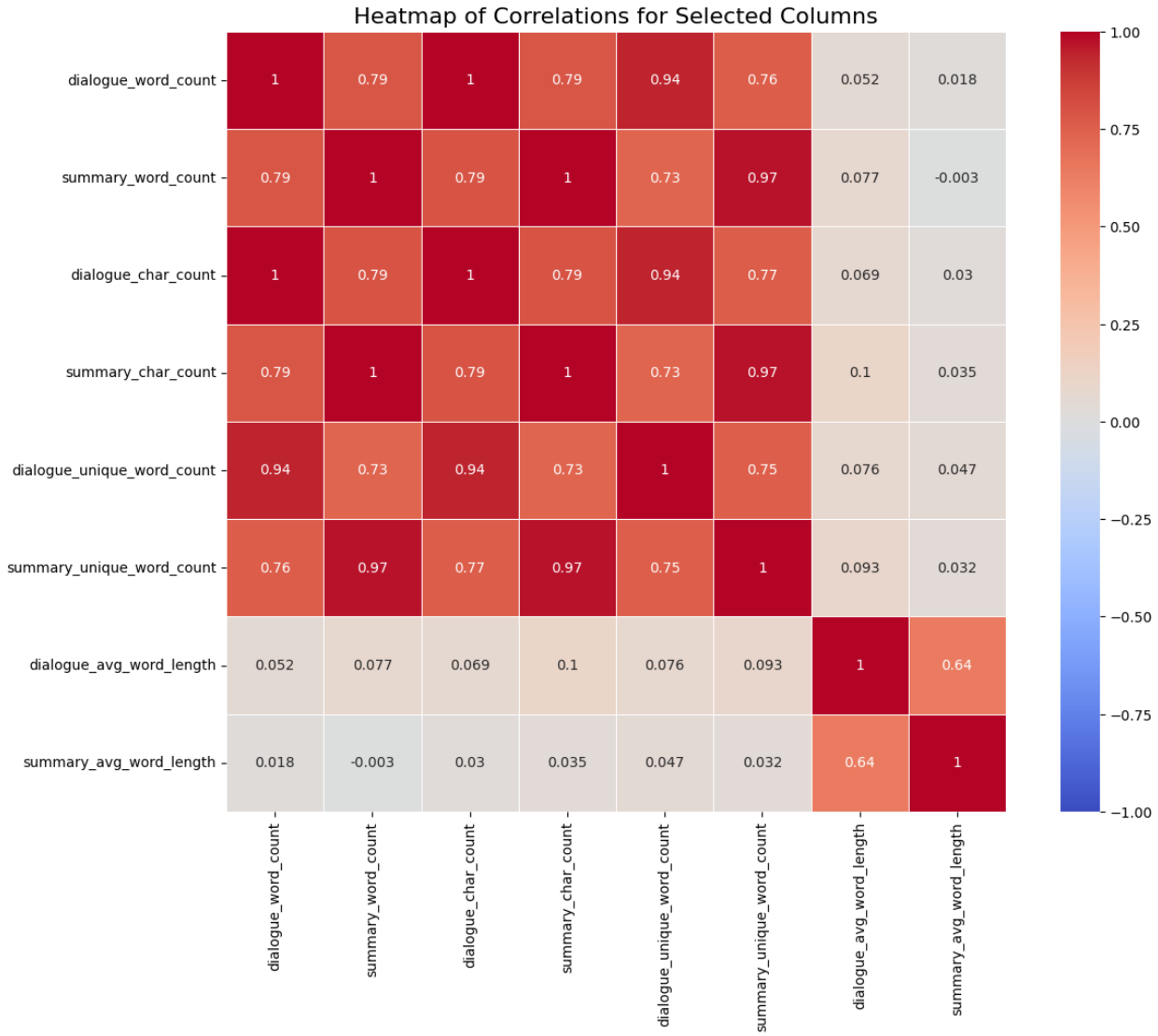
adjusted with `alpha=0.7`, and the marker size is set to `s=20`. Finally, `plt.show()` displays the resulting pairplot, allowing for a comprehensive exploration of correlations and trends within the data.

```
# Select only the relevant columns from the DataFrame
correlation_data = df[columns]

# Create a pairplot for the selected columns
sns.pairplot(correlation_data, diag_kind="kde", plot_kws={"alpha":
0.7, "s": 20})

# Show the plots
plt.show()
```



The code generates a **heatmap** to visualize the correlation matrix of the selected numerical columns in the DataFrame `df`. First, it calculates the correlation matrix using the `.corr()` method on the relevant columns, which computes the Pearson correlation coefficients between each pair of variables. The heatmap is then created using Seaborn's `heatmap` function, with annotations displaying the correlation values in each cell. The color scheme, `coolwarm`, is used to differentiate negative correlations (blue) from positive correlations (red), with a range from -1 to 1. The heatmap is square-shaped, and thin lines between cells are added for clarity. A title is set for the plot, and the figure is displayed using `plt.show()`. This visualization helps to easily identify the relationships between variables, highlighting strong correlations or patterns in the data.

```
corr_data = df[columns].corr()

plt.figure(figsize=(14, 10))
sns.heatmap(corr_data, annot=True, cmap='coolwarm', vmin=-1, vmax=1,
square=True, linewidths=0.5)
plt.title('Heatmap of Correlations for Selected Columns', fontsize=16)
plt.show()
```



The code generates a **word cloud** to visually represent the most frequent words in the cleaned dialogue text from the DataFrame `df`. It first combines all the cleaned dialogue entries into a single string and removes common stopwords using the default stopwords set from the `WordCloud` library. Then, the `WordCloud` class is used to create the word cloud with specified parameters, such as setting the background color to white, applying the 'viridis' colormap, limiting the display to the top 200 most frequent words, and adding a steel blue contour for emphasis. The resulting word cloud is displayed with smooth rendering and no axis, providing a

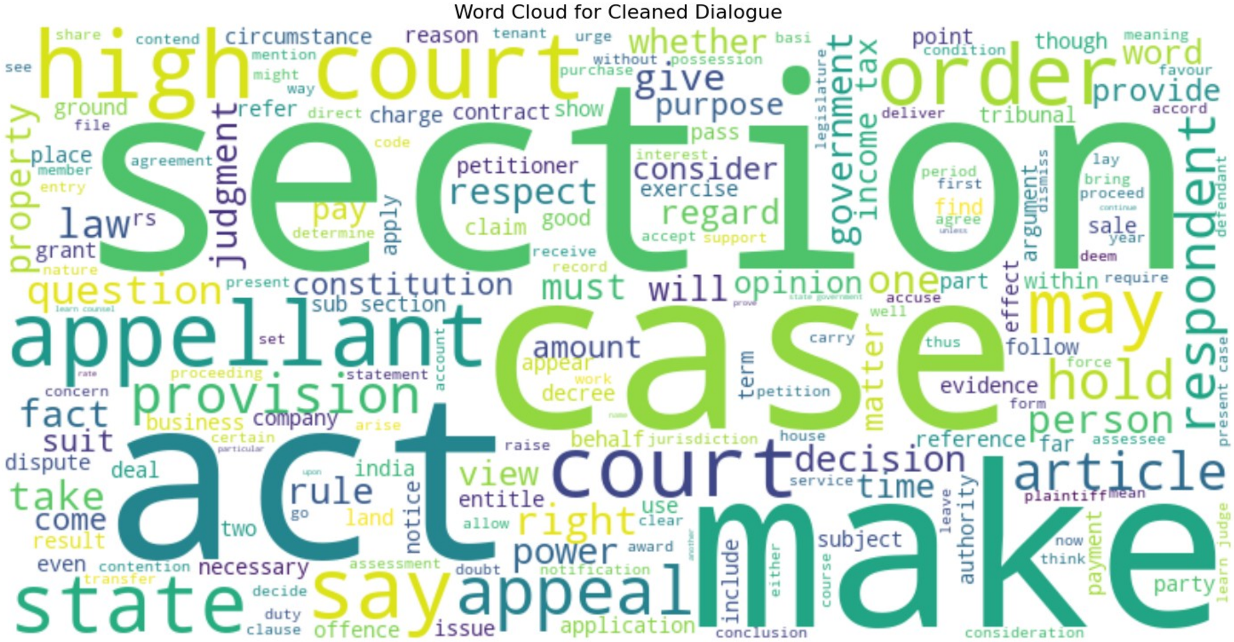
clear visual of the most commonly used words in the dialogue data. This visualization helps to quickly identify key terms and themes within the text.

```
%pip install wordcloud
```

```
Requirement already satisfied: wordcloud in c:\users\kushal\onedrive\
desktop\text-summarization-nlp-project\chintya\lib\site-packages
(1.9.4)
Requirement already satisfied: numpy>=1.6.1 in c:\users\kushal\
onedrive\desktop\text-summarization-nlp-project\chintya\lib\site-
packages (from wordcloud) (2.0.2)
Requirement already satisfied: pillow in c:\users\kushal\onedrive\
desktop\text-summarization-nlp-project\chintya\lib\site-packages (from
wordcloud) (11.1.0)
Requirement already satisfied: matplotlib in c:\users\kushal\onedrive\
desktop\text-summarization-nlp-project\chintya\lib\site-packages (from
wordcloud) (3.9.4)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\kushal\
onedrive\desktop\text-summarization-nlp-project\chintya\lib\site-
packages (from matplotlib->wordcloud) (1.3.0)
Requirement already satisfied: cycler>=0.10 in c:\users\kushal\
onedrive\desktop\text-summarization-nlp-project\chintya\lib\site-
packages (from matplotlib->wordcloud) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\kushal\
onedrive\desktop\text-summarization-nlp-project\chintya\lib\site-
packages (from matplotlib->wordcloud) (4.56.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\kushal\
onedrive\desktop\text-summarization-nlp-project\chintya\lib\site-
packages (from matplotlib->wordcloud) (1.4.7)
Requirement already satisfied: packaging>=20.0 in c:\users\kushal\
onedrive\desktop\text-summarization-nlp-project\chintya\lib\site-
packages (from matplotlib->wordcloud) (24.2)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\kushal\
onedrive\desktop\text-summarization-nlp-project\chintya\lib\site-
packages (from matplotlib->wordcloud) (3.2.1)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\
kushal\onedrive\desktop\text-summarization-nlp-project\chintya\lib\
site-packages (from matplotlib->wordcloud) (2.9.0.post0)
Requirement already satisfied: importlib-resources>=3.2.0 in c:\users\
kushal\onedrive\desktop\text-summarization-nlp-project\chintya\lib\
site-packages (from matplotlib->wordcloud) (6.5.2)
Requirement already satisfied: zipp>=3.1.0 in c:\users\kushal\
onedrive\desktop\text-summarization-nlp-project\chintya\lib\site-
packages (from importlib-resources>=3.2.0->matplotlib->wordcloud)
(3.21.0)
Requirement already satisfied: six>=1.5 in c:\users\kushal\onedrive\
desktop\text-summarization-nlp-project\chintya\lib\site-packages (from
python-dateutil>=2.7->matplotlib->wordcloud) (1.17.0)
Note: you may need to restart the kernel to use updated packages.
```

```python
from wordcloud import WordCloud, STOPWORDS

all_dialogues = " ".join(df["cleaned_dialogue"])

stopwords = set(STOPWORDS)

wordcloud = WordCloud(
    width=800,
    height=400,
    background_color='white',
    stopwords=stopwords,
    colormap='viridis',
    max_words=200,
    contour_width=1,
    contour_color='steelblue'
).generate(all_dialogues)

# Plot the word cloud
plt.figure(figsize=(20, 9))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.title("Word Cloud for Cleaned Dialogue", fontsize=16)
plt.show()
```

Word Cloud for Cleaned Dialogue



The code generates a **word cloud** to visualize the most frequent words in the cleaned summary text from the DataFrame `df`. It first combines all the cleaned summaries into a single string and removes common stopwords using the default stopwords set from the `WordCloud` library. Then, the `WordCloud` class is used to generate the word cloud with several parameters, such as setting the background color to white, using the 'viridis' colormap, limiting the display to the top 200 most frequent words, and adding a steel blue contour for aesthetic enhancement. Finally,

the word cloud is displayed with smooth rendering and no axis. This visualization highlights the most prominent terms in the summary data, providing insight into key concepts or themes from the summaries.

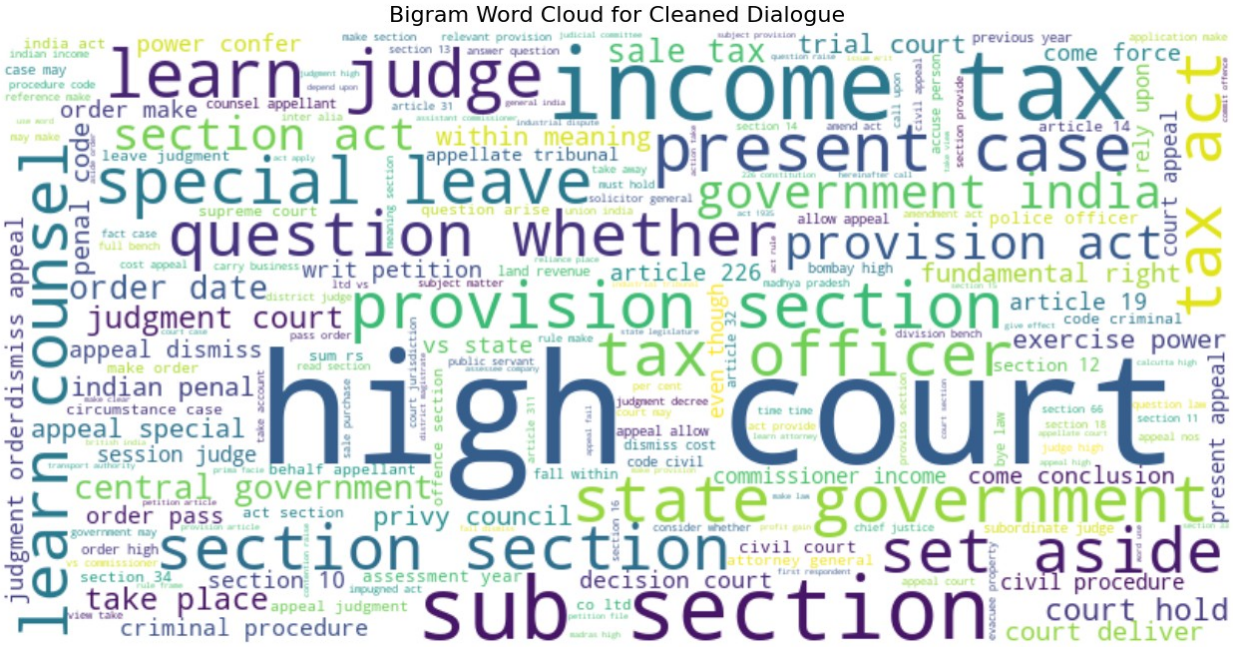
```
from wordcloud import WordCloud, STOPWORDS

all_dialogues = " ".join(df["cleaned_summary"])

stopwords = set(STOPWORDS)

wordcloud = WordCloud(
    width=800,
    height=400,
    background_color='white',
    stopwords=stopwords,
    colormap='viridis',
    max_words=200,
    contour_width=1,
    contour_color='steelblue'
).generate(all_dialogues)

# Plot the word cloud
plt.figure(figsize=(20, 9))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.title("Word Cloud for Cleaned Summary", fontsize=16)
plt.show()
```

The code generates a bigram word cloud to visualize the most frequent two-word combinations (bigrams) in the cleaned dialogue text from the DataFrame df. First, all the cleaned dialogues are

combined into a single string, and stopwords are defined to filter out common words. Then, a CountVectorizer is used to extract bigrams (two-word combinations) from the text, and the frequency of each bigram is calculated. The top 200 bigrams with the highest frequencies are selected for the word cloud. The WordCloud class is used to generate a word cloud based on these bigram frequencies, with a visually distinct color palette and a smooth rendering. The resulting word cloud is displayed without axes, highlighting the most common bigrams in the dialogue, which helps to identify prevalent word pairs or themes in the dataset.

```
from wordcloud import WordCloud, STOPWORDS
import matplotlib.pyplot as plt
from sklearn.feature_extraction.text import CountVectorizer

# Combine all cleaned dialogues into a single string
all_dialogues = " ".join(df["cleaned_dialogue"])

# Define stopwords (convert to list)
stopwords_list = list(STOPWORDS)

# Use CountVectorizer to extract bigrams
vectorizer = CountVectorizer(ngram_range=(2, 2),
stop_words=stopwords_list)
X = vectorizer.fit_transform([all_dialogues])
bigram_freq = X.toarray().sum(axis=0)

# Create a DataFrame of bigrams and their frequencies
bigram_freq_df = pd.DataFrame(bigram_freq,
index=vectorizer.get_feature_names_out(), columns=["frequency"])

# Sort bigrams by frequency
bigram_freq_df = bigram_freq_df.sort_values(by="frequency",
ascending=False)

# Select the top 200 bigrams for the word cloud
top_bigrams = bigram_freq_df.head(200)

# Generate the bigram word cloud
wordcloud = WordCloud(
    width=800,
    height=400,
    background_color='white',
    colormap='viridis',
    max_words=200,
    contour_width=1,
    contour_color='steelblue'
).generate_from_frequencies(top_bigrams["frequency"].to_dict())

# Plot the word cloud
plt.figure(figsize=(20, 8))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
```

```
plt.title("Bigram Word Cloud for Cleaned Dialogue", fontsize=16)
plt.show()
```

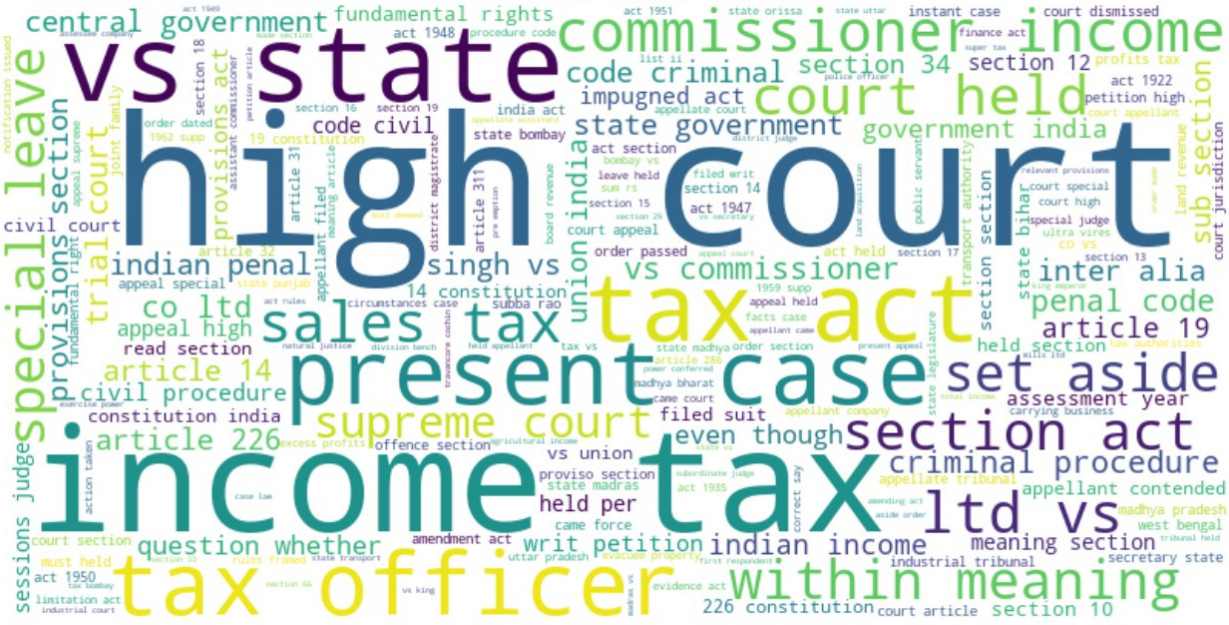


This code generates a **bigram word cloud** to visualize the most frequent two-word combinations (bigrams) in the cleaned summary text from the DataFrame `df`. First, it combines all the cleaned summaries into a single string and filters out common stopwords using the `STOPWORDS` set. Then, a `CountVectorizer` is used to extract bigrams (two-word combinations) from the text, which are represented by their frequency counts. The bigrams are sorted by frequency, and the top 200 most frequent bigrams are selected for the word cloud. Using the `WordCloud` class, a word cloud is generated based on these frequencies, with a visually appealing 'viridis' colormap and steel blue contours. Finally, the word cloud is displayed without axes, providing a clear and aesthetically pleasing visualization of the most common bigrams in the cleaned summary data. This helps to identify key recurring word pairs or themes in the dataset.

```
# Combine all cleaned dialogues into a single string
all_dialogues = " ".join(df["cleaned_summary"])

# Define stopwords (convert to list)
stopwords_list = list(STOPWORDS)

# Use CountVectorizer to extract bigrams
vectorizer = CountVectorizer(ngram_range=(2, 2),
stop_words=stopwords_list)
X = vectorizer.fit_transform([all_dialogues])
bigram_freq = X.toarray().sum(axis=0)

# Create a DataFrame of bigrams and their frequencies
bigram_freq_df = pd.DataFrame(bigram_freq,
```

```
index=vectorizer.get_feature_names_out(), columns=["frequency"])

# Sort bigrams by frequency
bigram_freq_df = bigram_freq_df.sort_values(by="frequency",
ascending=False)

# Select the top 200 bigrams for the word cloud
top_bigrams = bigram_freq_df.head(200)

# Generate the bigram word cloud
wordcloud = WordCloud(
    width=800,
    height=400,
    background_color='white',
    colormap='viridis',
    max_words=200,
    contour_width=1,
    contour_color='steelblue'
).generate_from_frequencies(top_bigrams["frequency"].to_dict())

# Plot the word cloud
plt.figure(figsize=(20, 8))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.title("Bigram Word Cloud for Cleaned Summary", fontsize=16)
plt.show()
```

Bigram Word Cloud for Cleaned Summary



Abstractive Summarization

Using a pre-trained transformer model like BART

This section uses a pre-trained BART model for summarization from Hugging Face to generate a summary for a sample dialogue. It compares the model's predicted summary with the actual summary from the dataset.

1. **Load pre-trained BART model**: Using `pipeline("summarization")` with `facebook/bart-large-cnn`.
2. **Input example dialogue**: Selects a dialogue from the dataset (`df['cleaned_dialogue'][0]`).
3. **Generate summary**: The model creates a summary with specified length constraints.
4. **Compare with actual summary**: Prints the predicted summary and compares it to the actual one from the dataset.

This helps evaluate the model's performance.

```
from transformers import pipeline

# Load pre-trained BART model for summarization
summarizer = pipeline("summarization", model="facebook/bart-large-
cnn")
```

```
WARNING:tensorflow:From c:\Users\Kushal\OneDrive\Desktop\Text-
Summarization-NLP-Project\chintya\lib\site-packages\tf_keras\src\
losses.py:2976: The name tf.losses.sparse_softmax_cross_entropy is
deprecated. Please use
tf.compat.v1.losses.sparse_softmax_cross_entropy instead.


Device set to use cuda:0
```

```
# Example dialogue
dialogue = df['cleaned_dialogue'][0]

dialogue
```

```
'appeal no lxvi of 1949 appeal from the high court of judicature
bombay in a reference under section 66 of the indian income tax act
1022 k.m munshi n. p. nathvani with he for the appel lant m.c setalvad
attorney general for india h. j. umrigar with he for the respondent
1950 may 26 the judgment of the court be deliver by mehr chand mahajan
j. this be an appeal against a judgment of the high court of
judicature at bombay in an income tax matter and it raise the question
whether munici pal property tax and urban immoveable property tax
payable under the relevant bombay act be allowable deduction under
section 9 1 iv of the indian income tax act the assessee company be an
investment company derive its income from property in the city of
bombay for the assessment year 1940 41 the net income of the assessee
```

under the head property be compute by the income tax officer in the sum of rs 6,21,764 after deduct from gross rent certain payment the company have pay during the relevant year rs 1,22,675 as municipal property tax and rs 32,760 as urban property tax deduction of these two sum be claim under the provision of section 9 the act out of the first item a deduction in the sum of rs 48,572 be allow on the ground that this item represent tenant burden pay by the assessee otherwise the claim be disal low the appeal of the assessee to the appellate as sistant commissioner and to the income tax appellate tribu nal be unsuccessful the tribunal however agree to refer two question of law to the high court of judicature at bombay namely 1 whether the municipal taxis pay by the applicant company be an allowable deduction under 555 the provision of section 9 1 iv of the indian income tax act 2 whether the urban immoveable property taxis pay by the applicant company be an allowable deduction under section 9 1 iv or under section 9 1 v of the indian income tax act a supplementary reference be make cover a third question which be not raise before we and it be not there fore necessary to refer to it the high court answer all the three question in the negative and hence this appeal the question for our determination be whether the munic ipal property tax and urban immoveable property tax can be deduct as an allowance under clause iv of sub section 1 of section 9 of the act the decision of the point depend firstly on the construction of the language employ in sub clause iv of sub section 1 of section 9 of the act and secondly on a finding as to the true nature and character of the liability of the owner under the relevant bombay act for the payment of these taxis section 9 along with the relevant clause run thus 1 the tax shall be payable by an assessee under the head income from property in respect of the bona fide annual value of property consist of any building or land appurtenant thereto of which he be the owner subject to the follow allowance namely iv where the property be subject to a mortgage or other capital charge the amount of any interest on such mortgage or charge where the property be subject to an annual charge not be a capital charge the amount of such charge where the property be subject to a ground rent the amount of such ground rent and where the property have be acquire construct repair renew or recon structe with borrow capital the amount of any interest payable on such capital it will be see that clause iv consist of four sub clause correspond to the four deduction allow 556 under the clause before the amend act of 1939 clause iv contain only the first third and fourth sub clause under the first sub clause interest be deductible whether the amount borrow on the security of the property be spend on the property or not there be no question of any capital or other expenditure on the property the expression capital charge in the sub clause can not connote a charge on the capital that is the property assess that would be a redundancy as the opening word themselves clearly indicate that the charge be on the property we be therefore of opinion that capital charge here could only mean a charge create for a capital sum i.e. a charge to secure the discharge of a liability of a capital nature in 1933 the privy

council decide the case of bijoy singh dudhuria vs commissioner of income tax calcutta 1 it be not an assessment under section 9 but an assess ment on the general income of an assessee who be liable to pay maintenance for his step mother which have be charge on all his asset by a decree of court it be not a li ability voluntarily incur by he but one cast on he by law the privy council hold that the amount pay by he in discharge of that liability form no part of his real income and so should not be include in his assessment though the decision proceed on the principle that the outgoing be not part of the assessee \'s income at all the framer of the amend act of 1939 want apparently to extend the principle so far as the assessment of property be concern even to case where obligatory payment have to be make out of the assessee \'s income from the property charge with such payment and the second sub clause namely where the property be subject to an annual charge not be a capital charge the amount of such charge be add it be this sub clause which the appellant invoke in support of its claim to deduction of the municipal and urban property taxis in the present case in view of the opening word of the newly add sub clause the expression capital charge also use therein can not have reference to a charge on the property and we think it must 1 i.l.r 60 cal 557 be understand in the same sense as in sub clause 1 that be to say the first sub clause having provide for deduc tion of interest where a capital sum be charge on the property this sub clause provide for a deduction of annual sum so charge such sum not be capital sum the limit word be intend to exclude case where capital raise on the security of the property be make repayable in instalment in commissioner of income tax bombay vs mahomedbhoy rowji 1 a bench of the bombay high court consider the meaning of these word as regard annual charge beau mont c.j observe as follow the word I think would cover a charge to secure an annual liability kania j. as he then be say as follow I do not see how a charge can be annual unless it mean a charge in respect of a payment to be make annually this construction of the word have be follow in the judgment under appeal in gappumal kanhaiya lal vs commissioner of income tax 2 the connect appeal before we the bench of the allahabad high court agree with the construction place on these word in the bombay case i.e. the word annual charge mean a charge to secure an annual liability it be therefore clear that there be no conflict of judicial deci sion as to the meaning of the phrase annual charge occur ring in section 3 1 iv and the meaning give be the natural meaning of these word as to the phrase capital charge beaumont c.j in the case above refer to take the view that the word mean a charge on capital kania j. however take a different view and observe that he be not prepared to accept the sugge tion that a document which provide for a certain payment to be make monthly or annually and charge on immoveable property or the estate of an individual become a capital charge in the allahabad judgment under appeal these 1 i.l.r 2 i.l.r 1944 all 558 word be consider as not mean a charge on capital it be say that if an annual charge mean a charge to secure the discharge of an annual liability then capital charge

mean a charge to secure the discharge of a liability of a capital nature we think this construction be a natu ral construction of the section and be right the determination of the point whether the taxis in dispute fall within the ambit of the phrase annual charge not be a capital charge depend on the provision of the statute under which they be levied section 143 of the city of bombay municipal act 1888 authorise the levy of a general tax on all building and land in the city the primary responsibility to pay this property tax be on the lessor vide section 146 of the act in order to assess the tax provision have be make for the determination of the annual rateable value of the building in section 154 section 156 provide for the maintenance of an assessment book in which entry have to be make every official year of all building in the city their rateable value the name of person primarily liable for payment of the property tax on such building and of the amount for which each building have be assess section 167 lay down that the assess ment book need not be prepare every official year but public notice shall be give in accordance with section 160 to 162 every year and the provision o+ the say sec tion and of section 163 and 167 shall be applicable each year these section lie down a procedure for hearing objection and complaint against entry in the assessment book from these provision it be clear that the liabil ity for the tax be determine at the beginning of each official year and the tax be an annual one it recur from year to year section 143to 168 concern themselves with the imposition liability and assessment of the tax for the year the amount of the tax for the year and the liability for its payment having be determine the act then pre scribe for its collection in the chapter the collection of taxis section 197 provide that each of the property taxis shall be payable in 559 advance in half yearly instalment on each first day of april and each first day of october the provision as to half yearly instalment necessarily connote an annual li ability in other word it mean that the annual liability can be discharge by half yearly payment procedure have also be prescribe for recovery of the instalment by presentment of a bill a notice of demand and then distress and sale finally section 212 provide as follow property taxis due under this act in respect of any building or land shall subject to the prior payment of the land revenue if any due to the provincial government thereupon be a first charge upon the say build ing or land it create a statutory charge on the building urban immove able property tax be leviable under section 22 of part vi of the bombay finance act 1932,on the annual letting value of the property the duty to collect the tax be lay on the municipality and it do so in the same manner as in the case of the municipal property tax section 24 2 b be in term similar to section 212 of the bombay municipal act it make the land or the building security for the payment of this tax also for the purpose of section 9 of the indian income tax act both these taxis namely the munici pal property tax as well as the urban immoveable property tax be of the same character and stand on the same foot ing mr munshi the learn counsel for the appellant con tend that both the taxis be assess

on the annual value of the land or the building and be annual taxis although it may be that they be collect at interval of six month for the sake of convenience that the income tax itself be assess on an annual basis that in allow deduction all payment make or all liability incur during the previ ous year of assessment should be allow and that the taxis in question fall clearly within the language of section 9 1 iv the learn attorney general on the other hand argue that although the taxis be assess for the year the liability to pay they arise at the beginning 560 of each half year and unless a notice of demand be issue and a bill present there be no liability to pay they and that till then no charge under section 212 of the act could possibly arise and that the liability to pay be half yearly in advance the charge be not an annual charge it be also suggest that the taxis be a capital charge in the sense of the property be security for the payment we be satisfied that the contention raise by the learn attorney general be not sound it be apparent from the whole tenor of the two bombay act that the taxis be in the nature of an annual levy on the property and be assess on the annual value of the property each year the annual liability can be discharge by half yearly instalment the liability be an annual one and the property having be subject to it the provision of clause iv of sub sec tion 1 of section 9 be immediately attract great emphasis be lay on the word"due use in section 212 of the municipal act and it be say that as the taxis do not become due under the act unless the time for the payment arrive no charge come into existence till then and that the charge be not an annual charge we do not think that this be a correct construction of section 212 the word property taxis due under this act mean property taxis for which a person be liable under the act taxis payable during the year have be make a charge on the property the liability and the charge both co exist and be co exten sive the provision of the act afford facility for the discharge of the liability do not in any way affect their true nature and character if the annual liability be not discharge in the manner lay down by section 197 can it be say that the property can not be sell for recovery of the whole amount due for the year the answer to this query can only be in the affirmative i.e. that the proper ty be liable to sale in commissioner of income tax bombay vs mahomedbhoy rowji(1 beaumont c.j while reject the claim for the deduction of the taxis place reliance on 1 i.l.r 561 section 9 1 v which allow a deduction in respect of any sum pay on account of land revenue it be observe that land revenue stand on the same footing as municipal taxis and that as the legislature make a special provision for deduction of sum payable in regard to land revenue but not in respect of sum pay on account of municipal taxis that circumstance indicate that the deduction be not allowable for the same purpose reference be also make to the provi sion of section 10 which deal with business allowance and wherein deduction of any sum pay on account of land reve nue local rate or municipal taxis have be allow in the conclude part of his judgment the learn chief jus tice say that it be not necessary for he to consider what the exact meaning of the word be and that it be suffi

cient for he to say that it do not cover municipal taxis which be make a charge on the property under section 212 of the bombay municipal act without determine the exact meaning of the word use by the statute it seem to we it be not possible to arrive at the conclusion that the taxis be not within the ambit of the clause it be elementary that the primary duty of a court be to give effect to the intention of the legislature as express in the word use by it and no outside consideration can be call in aid to find that intention again reference to clause v of the section be not very helpful because land revenue be a charge of a paramount nature on all building and land and that be so a deduction in respect of the amount be mention in express term municipal taxis on the other hand do not stand on the same footing as land revenue the law as to they vary from province to province and they may not be necessarily a charge on property in all case the legis lature seem to have think that so far as municipal taxis on property be concern if they fall within the ambit of clause iv deduction will be claimable in respect of they but not otherwise the deduction allow in section 10 under the head income from business proceed on a different footing and a construction of section 9 with the aid of section 10 be apt to mislead 562 kania j. in the above case in arrive at his conclusion be influence by the consideration that these taxis be of a variable character i.e. liable to be increase or re duce under the various provision of the municipal act and that the charge be in the nature of a contingent charge with great respect it may be point out that all charge in a way may be or be of a variable and contingent na ture if no default be make no charge be ever enforceable and whenever there be a charge it can be increase or reduce during the year either by payment or by additional borrowing in moss empires ltd vs inland revenue commissioner 1 it be hold by the house of lord that the fact that certain payment be contingent and variable in amount do not affect their character of be annual payment and that the word annual must be take to have the quality of be recurrent or be capable of recurrence in cunard \'s trustee vs inland revenue commissioner 2 it be hold that the payment be capable of be recur rent and be therefore annual payment within the meaning of schedule d case iii rule 1 1 even though they be not necessarily recurrent year by year and the fact that they vary in amount be immaterial the learn attorney general in view of these decision do not support the view express by kania j. reliance be place on a decision of the high court of madra in mamad keyi vs commissioner of income tax madras(3 in which money pay as urban immoveable property tax under the bombay finance act be disallow as inadmis sible under section 9 1 iv or 9 1 v of the indian income tax act this decision merely follow the view express in commissioner of income tax bombay vs mahomedb hoy rowji 4)and be not arrive at on any independent or fresh reasoning and be not of much assistance in the deci sion of the case the allahabad high court 1 2 1948 1 a.e.r 150 3 i.l.r 4 i.l.r 563 in gappumal kanhaiya lal vs commissioner of incometax 1 the connect appeal take a correct view of this matter and the reasoning give therein have our approval the result be that this appeal be allow and the two question

```
which be refer to the high court by the income tax tribunal and cite above be answer in the affirmative the appellant will have their cost in the appeal appeal allow'
```

```
dialogue = dialogue[:1024]  # Limit input to 1024 characters
summary = summarizer(dialogue, max_length=100, min_length=20, do_sample=False)
```

```
summary[0]['summary_text']
```

```
'This is an appeal from the high court of judicature bombay in a reference under section 66 of the indian income tax act 1022 k.m munshi n. p. nathvani with he for the appel lant m.c setalvad attorney general for india h. j. umrigar.'
```

```
print("Predicted Summary", summary[0]['summary_text'] )
print("---------------------------------")
print("Actual Summary", df['cleaned_summary'][0])
```

```
Predicted Summary This is an appeal from the high court of judicature bombay in a reference under section 66 of the indian income tax act 1022 k.m munshi n. p. nathvani with he for the appel lant m.c setalvad attorney general for india h. j. umrigar.
---------------------------------
Actual Summary The charge created in respect of municipal property tax by section 212 of the City of Bombay Municipal Act, 1888, is an "annual charge not being a capital charge" within the mean ing of section 9 (1) (iv) of the Indian Income tax Act, 199.2, and the amount of such charge should therefore be deducted in computing the income from such property for the purposes of section 9 of the Indian Income tax Act. The charge in respect of urban immoveable property tax created by the Bombay Finance Act, 1939 is similar in character and the amount of such charge should also be deducted. The expression "capital charge" in s.9(1) (iv) means a charge created for a capital sum,that is to say, a charge created to. ' secure the discharge of a liability of a capi tal nature; and an "annual charge" means a charge to secure an annual liabili ty. 554
```

```
# Generate summary
summary = summarizer(dialogue, max_length=50, min_length=25, do_sample=False)
print("Predicted Summary", summary[0]['summary_text'] )
print("---------------------------------")
print("Actual Summary", df['cleaned_summary'])
```

```
Predicted Summary Appeal from the high court of judicature bombay in a
reference under section 66 of the indian income tax act 1022 k.m
munshi n. p. nathvani with he for the appel l
--------------------------------
Actual Summary 0      The charge created in respect of municipal
pro...
1      An agreement for a lease, which a lease is by ...
2      The question whether a Magistrate is "personal...
3      The appellant was a member of a joint Hindu fa...
4      The appellant was the Ruler of the State of Ba...
                            ...
995    The appellant .company reduced its capital and...
996    A widow whose estate was under the charge of t...
997    The appellant 's election was challenged inter...
998    The appellant and respondent were the tenant a...
999    The Sales Tax Officer rejected the assessed 's...
Name: cleaned_summary, Length: 1000, dtype: object
```

Custom Training

Using the above dataset, custom training transformer model : BART

This notebook section walks through fine-tuning a pre-trained BART model (`facebook/bart-large-cnn`) for text summarization. Below is an explanation of the steps and configurations:

---

1. Environment Setup

- **GPU Check**: Ensures that the model training utilizes GPU if available for faster processing.
- **Device Assignment**: The model and computations are transferred to the appropriate device (`cuda` or `cpu`).

---

2. Loading the Pre-trained Model and Tokenizer

- The BART model and its tokenizer are loaded using Hugging Face's `from_pretrained` function.
- `facebook/bart-large-cnn` is pre-trained on summarization tasks, making it an ideal starting point.

---

3. Dataset Preparation

- **Input and Output Texts**: The dataset consists of dialogues (`cleaned_dialogue`) and their respective summaries (`cleaned_summary`).
- **Dataset Conversion**: The data is formatted into a Hugging Face `Dataset` object.

---

4. Splitting Data

- The dataset is split into training (80%) and evaluation (20%) subsets using the `train_test_split` method.

---

5. Tokenization

- **Custom Tokenization Function**: Input dialogues and their corresponding summaries are tokenized with:
  - `max_length` constraints for truncation.
  - `padding` for uniform input dimensions.
- Tokenized labels (summaries) are added to the model inputs.

---

6. Training Arguments

The training process is configured using `TrainingArguments`:

- **Output Directory**: `./results` will store logs and outputs.
- **Evaluation Strategy**: Model is evaluated at the end of each epoch.
- **Learning Rate**: A fine-tuning-friendly value of `2e-5` is chosen.
- **Batch Sizes**: Configured for both training and evaluation phases.
- **Epochs**: Set to 3 for demonstration purposes.
- **Mixed Precision (FP16)**: Enables efficient GPU training.
- **Logging**: Logs are saved in `./logs`.

---

7. Trainer Initialization

- The Hugging Face `Trainer` is instantiated to:
  - Fine-tune the BART model on the tokenized training dataset.
  - Evaluate the model using the tokenized evaluation dataset.

---

8. Model Training

- The model is trained using the `Trainer.train()` function, which manages the training loop and GPU optimizations.

This process customizes the BART model for domain-specific summarization, improving its accuracy on dialogue datasets.each component and step in the code.

```
import torch
torch.cuda.empty_cache()

from transformers import BartForConditionalGeneration, BartTokenizer,
Trainer, TrainingArguments
from datasets import Dataset, DatasetDict
```

```
import torch

# Check if GPU is available
device = "cuda" if torch.cuda.is_available() else "cpu"
print(f"Using device: {device}")

# Load the pre-trained BART model and tokenizer
model = BartForConditionalGeneration.from_pretrained("facebook/bart-
large-cnn").to(device)
tokenizer = BartTokenizer.from_pretrained("facebook/bart-large-cnn")

# Prepare your dataset
data = {
    'input_text': df['cleaned_dialogue'].tolist(),
    'output_text': df['cleaned_summary'].tolist()
}
train_dataset = Dataset.from_dict(data)

# Example split: 80% train, 20% eval
split_datasets = train_dataset.train_test_split(test_size=0.2)

# Tokenize the dataset
def tokenize_function(examples):
    model_inputs = tokenizer(examples['input_text'], max_length=1024,
truncation=True, padding="max_length")
    labels = tokenizer(examples['output_text'], max_length=150,
truncation=True, padding="max_length")
    model_inputs["labels"] = labels["input_ids"]
    return model_inputs

# Apply tokenization to both train and eval datasets
tokenized_datasets = {
    "train": split_datasets["train"].map(tokenize_function,
batched=True),
    "eval": split_datasets["test"].map(tokenize_function,
batched=True)
}

# Define training arguments with GPU support
training_args = TrainingArguments(
    output_dir='./results',          # output directory
    evaluation_strategy="epoch",     # evaluation strategy
    learning_rate=2e-5,              # learning rate
    per_device_train_batch_size=4,   # batch size for training
    per_device_eval_batch_size=8,    # batch size for evaluation
    num_train_epochs=3,              # number of training epochs
    weight_decay=0.01,               # strength of weight decay
    fp16=True,                       # enable mixed precision training
for GPU
    save_strategy="no",           # save checkpoints every epoch
```

```
    logging_dir='./logs',            # directory for logging
)

# Initialize the Trainer
trainer = Trainer(
    model=model,                         # the instantiated 🤗 Transformers model
    args=training_args,                  # training arguments, defined above
    train_dataset=tokenized_datasets["train"], # tokenized training dataset
    eval_dataset=tokenized_datasets["eval"],   # tokenized evaluation dataset
)

# Train the model
trainer.train()
```

```
Using device: cuda
```


```
{"model_id":"5dfddc37a6964ba3bec99199d4a01f95","version_major":2,"version_minor":0}
```

```
{"model_id":"fd76f6c7cd094856bded031766eb4f27","version_major":2,"version_minor":0}
```


```
<IPython.core.display.HTML object>
```

```
TrainOutput(global_step=600, training_loss=1.7552196502685546, metrics={'train_runtime': 18207.0568, 'train_samples_per_second': 0.132, 'train_steps_per_second': 0.033, 'total_flos': 5201051045068800.0, 'train_loss': 1.7552196502685546, 'epoch': 3.0})
```

The code saves the fine-tuned BART model and token[2wzzaq22qazzer to a folder called `./bart_summary_model`. The model and tokenizer will be saved in the current working directory, and a message `"Model saved!"` is printed for confirmation.

```
model.save_pretrained("./bart_summary_model")
tokenizer.save_pretrained("./bart_summary_model")
print("Model saved!")
```

```
Model saved!
```

In this notebook, we will evaluate the performance of our summarization model using standard metrics. Two important libraries are required for this:

1.evaluate : A library from Hugging Face that provides an easy-to-use interface for computing various evaluation metrics, including accuracy, BLEU, ROUGE, and others. We will use this to compute ROUGE scores, a common evaluation metric for text generation tasks.

2.rouge_score: A Python package to calculate the ROUGE (Recall-Oriented Understudy for Gisting Evaluation) metrics, which measure the quality of generated text by comparing it to reference summaries.

Metrics

1. **Load the ROUGE Metric**:
   - The `evaluate` library is imported and the ROUGE metric is loaded using `evaluate.load("rouge")` to assess the quality of generated summaries.
2. **Generate Predictions for the Test Set**:
   - The code initializes two lists, `generated_summaries` and `reference_summaries`, to store the model-generated summaries and the reference summaries from the dataset.
3. **Iterate Over Evaluation Dataset**:
   - It iterates through the evaluation data, decoding both the input texts and reference summaries from token IDs to text.
4. **Generate Model Summaries**:
   - The input text is tokenized, passed to the BART model, and a summary is generated using beam search with specific length parameters (`max_length`, `min_length`, `num_beams`).
5. **Store Summaries**:
   - The generated summaries are appended to `generated_summaries`, and the reference summaries to `reference_summaries`.
6. **Compute ROUGE Scores**:
   - ROUGE scores are computed using `rouge.compute()` by comparing the generated summaries to the reference summaries.
7. **Display ROUGE Scores**:
   - The ROUGE scores (ROUGE-1, ROUGE-2, ROUGE-L) are printed to evaluate the performance of the model's summaries.

```
import evaluate

rouge = evaluate.load("rouge")

# Generate predictions for the test set
generated_summaries = []
reference_summaries = []

for example in tokenized_datasets['eval']:
    input_text = tokenizer.decode(example['input_ids'],
skip_special_tokens=True)
    reference_summary = tokenizer.decode(example['labels'],
skip_special_tokens=True)

    # Generate summary
    inputs = tokenizer(input_text, max_length=1024, truncation=True,
return_tensors="pt").to(device)
```

```
    summary_ids = model.generate(inputs["input_ids"], max_length=150,
min_length=30, num_beams=4)
    generated_summary = tokenizer.decode(summary_ids[0],
skip_special_tokens=True)

    # Store generated and reference summaries
    generated_summaries.append(generated_summary)
    reference_summaries.append(reference_summary)

# Compute ROUGE scores
results = rouge.compute(predictions=generated_summaries,
references=reference_summaries)

# Display ROUGE scores
print("ROUGE Scores:")
for key, value in results.items():
    print(f"{key}: {value}")
```

```
ROUGE Scores:
rouge1: 0.49052180673948576
rouge2: 0.22395687223438293
rougeL: 0.3104401331414619
rougeLsum: 0.31010722766406296
```

Final Summarization

1. **Import Libraries**:
   - The `BartForConditionalGeneration` and `BartTokenizer` classes from the `transformers` library are imported to load the fine-tuned BART model and tokenizer.
   - `torch` is imported to check the availability of GPU and move the model to the appropriate device.
2. **Check GPU Availability**:
   - The code checks if a GPU is available by using `torch.cuda.is_available()`.
   - It sets the `device` to "cuda" if a GPU is available, otherwise defaults to "cpu". This ensures that the model runs on GPU if possible, speeding up the inference process.
3. **Print Device**:
   - The chosen device (either GPU or CPU) is printed to confirm the runtime environment.
4. **Load Fine-tuned Model and Tokenizer**:
   - The fine-tuned BART model is loaded from the saved directory `./bart_summary_model` using `from_pretrained()` and moved to the chosen device (GPU or CPU).
   - The tokenizer is loaded in the same manner to ensure proper tokenization of input data when generating summaries.he model's summaries.

```
from transformers import BartForConditionalGeneration, BartTokenizer
import torch

# Check if GPU is available
device = "cuda" if torch.cuda.is_available() else "cpu"
print(f"Using device: {device}")

# Load the fine-tuned BART model and tokenizer
model =
BartForConditionalGeneration.from_pretrained("./bart_summary_model").t
o(device)
tokenizer = BartTokenizer.from_pretrained("./bart_summary_model")
```

```
Using device: cuda
```

1. **Test Dialogue**:
   - A sample dialogue `test_dialogue` is provided, which contains a conversation between Amanda and Jerry.
2. **Tokenization**:
   - The `test_dialogue` is tokenized using the `tokenizer`, with a maximum sequence length of 1024 tokens. This ensures the input fits within the model's input size.
   - The `truncation=True` ensures that any input longer than 1024 tokens is truncated.
   - The `return_tensors="pt"` option returns the tokenized input in PyTorch tensor format, which is necessary for the model to process it.
3. **Move Inputs to Device**:
   - The tokenized inputs (like `input_ids`) are moved to the chosen device (`GPU` or `CPU`) to match where the model is located. This allows for efficient processing.
4. **Generate Summary**:
   - The model is used to generate a summary of the input dialogue using the `model.generate()` function.
   - The function parameters control the length and diversity of the generated summary:
     - `max_length=50` limits the summary to a maximum of 50 tokens.
     - `min_length=25` ensures the summary has at least 25 tokens.
     - `length_penalty=2.0` applies a penalty to longer summaries to avoid generating summaries that are too lengthy.
     - `num_beams=4` specifies the number of beams for beam search, a technique to explore different possible summary outputs and select the best one.
5. **Decode and Print the Summary**:
   - The generated summary is decoded from token IDs back into human-readable text using the tokenizer's `decode()` method.
   - The `skip_special_tokens=True` ensures that special tokens (such as padding tokens) are omitted from the decoded output.

– Finally, the generated summary is printed.

Expected Output: This code will generate a summary of the given test dialogue and print it.

```
from transformers import BartForConditionalGeneration, BartTokenizer
import torch

# Device setup
device = "cuda" if torch.cuda.is_available() else "cpu"
print(f"Using device: {device}")

# Load fine-tuned BART model and tokenizer
model =
BartForConditionalGeneration.from_pretrained("./bart_summary_model").t
o(device)
tokenizer = BartTokenizer.from_pretrained("./bart_summary_model")

# Input dialogue
test_dialogue = (
    "In this connection it is always necessary to bear in mind that
where an impugned statute passed by a State Legislature is relatable
to an Entry in List II it is not permissible to challenge its vires
only on the ground that the method adopted by it for the recovery of
the impost can be and is generally adopted in levying a duty of
excise."
    "It reads thus: ""It is hereby declared that it is expedient in
the public interest that the Central Government should take under its
control the regulation of mines and oil fields and the development of
minerals to the extent hereinafter provided""."
)

# Tokenize the input
inputs = tokenizer(test_dialogue, max_length=1024, truncation=True,
return_tensors="pt")
inputs = {key: val.to(device) for key, val in inputs.items()}
print("Tokenized Input:", tokenizer.decode(inputs["input_ids"][0],
skip_special_tokens=True))

# Generate summary
summary_ids = model.generate(
    inputs["input_ids"],
    max_length=50,
    min_length=10,
    length_penalty=1.5,
    num_beams=4,
    no_repeat_ngram_size=2
)
generated_summary = tokenizer.decode(summary_ids[0],
skip_special_tokens=True)
```

```
# Output
print("Generated Summary:", generated_summary)
```

```
Using device: cuda
Tokenized Input: In this connection it is always necessary to bear in
mind that where an impugned statute passed by a State Legislature is
relatable to an Entry in List II it is not permissible to challenge
its vires only on the ground that the method adopted by it for the
recovery of the impost can be and is generally adopted in levying a
duty of excise.It reads thus: It is hereby declared that it is
expedient in the public interest that the Central Government should
take under its control the regulation of mines and oil fields and the
development of minerals to the extent hereinafter provided.
Generated Summary: Where an impugned statute passed by a State
Legislature is relatable to an Entry in List II, it is not permissible
to challenge its vires only on the ground that the method adopted by
it for the recovery of the imp
```