UNIVERSITI TENAGA NASIONAL

The Energy University

CISB5123 Text Analytics
Lab 1
Working with Text Data

Text data is one of the most widely used forms of data in computing and business applications. This lab will introduce various ways to work with text data using Python, including reading, writing, and processing text, CSV, Excel, JSON, XML and PDF files.

1. Understanding Text Data Sources

Types of Text Data

- Unstructured Text: Documents, emails, reviews, social media posts.
- Structured Text: CSV, Excel, JSON, XML.
- Semi-structured Text: Web pages, logs, PDFs.

Common Text Data Format and Use Cases

Format	Example Use Case
.txt	Logs, transcriptions, raw text files
.csv	Customer reviews, structured survey responses
.xlsx	Business reports, academic records
.json	API responses, metadata, social media data
.xml	RSS feeds, configuration files
.pdf	Research papers, invoices, scanned documents

2. Working with Text Data

2.1 Raw Text Files (.txt)

Read and store plain text data for preprocessing.

```
# Read the content of the text file
with open('sample.txt', 'r', encoding='utf-8') as file:
    text_data = file.read()
print("Raw Text:\n", text_data)

# Store in another file
with open('stored_text.txt', 'w', encoding='utf-8') as file:
    file.write(text_data)
```

3. Working with Structured Text Data

3.1 CSV Files

Extract structured textual information from tabular data.

```
import pandas as pd

# Read the CSV file
df = pd.read_csv('reviews.csv')
print("Reviews:\n", df['Review'].head())

# Save the reviews column to a text file
df['Review'].to_csv('stored_reviews.txt', index=False, header=False)
```

3.2 Excel Files

Read and store structured data from Excel files.

```
# Read the Excel file
df_excel = pd.read_excel('reviews.xlsx', engine='openpyxl')
print("First two rows:\n", df_excel.head(2))

# Save the first two rows to a text file
df_excel.head(2).to_csv('extracted_excel.txt', index=False)
```

3.3 JSON Files

Extract and store data from JSON format (i.e. social media data)

```
import json

# Read the JSON file
with open('social_data.json', 'r', encoding='utf-8') as file:
    data = json.load(file)
print("Extracted City:", data['city'])

# Store the extracted city to a file
with open('stored_city.txt', 'w', encoding='utf-8') as file:
    file.write(data['city'])
```

4. Working with Semi-Structured Data

4.1 XML Files

Extract relevant information from an XML document.

```
import xml.etree.ElementTree as ET

# Parse the XML file
tree = ET.parse('news.xml')
root = tree.getroot()

for article in root.findall('article'):
    title = article.find('title').text
    print("Extracted Title:", title)

# Store the extracted title to a file
with open('stored_titles.txt', 'w', encoding='utf-8') as file:
    for article in root.findall('article'):
        title = article.find('title').text
        file.write(title + '\n')
```

5. Working with PDF Documents

Often you will have to deal with PDF files. There are many libraries in Python for working with PDFs, each with their pros and cons, the most common one being **PyPDF2**. You can install it with (note the case-sensitivity, you need to make sure your capitalization matches):

```
pip install PyPDF2
```

Note: Keep in mind that not every PDF file can be read with this library. PDFs that are too blurry, have a special encoding, encrypted, or maybe just created with a particular program that doesn't work well with PyPDF2 won't be able to be read. If you find yourself in this situation, try using the libraries linked above, but keep in mind, these may also not work. As far as PyPDF2 is concerned, it can only read the text from a PDF document, it won't be able to grab images or other media files from a PDF.

```
import PyPDF2

# Read the PDF file
with open('document.pdf', 'rb') as file:
    reader = PyPDF2.PdfReader(file)
    text = "\n".join(page.extract_text() for page in reader.pages if page.extract_text())

# Print the extracted text
print("Extracted PDF Text:\n", text)

# Store the extracted text in a file
with open('stored_pdf_text.txt', 'w', encoding='utf-8') as output:
    output.write(text)
```

Exercise

1. Extract text from all pages of Business_Proposal.pdf and save it in business_proposal_all.txt.

2. Extract text from only page 2 of Business_Proposal.pdf and save it in business_proposal_page_2.txt.