Experiment 2

Develop a python script to parse the pdf files using pdfminer.

WHAT IS A PDF?

Portable document format is a file format developed by Adobe, that presents documents in a manner independent of application software, hardware, and operating systems.

WHAT IS PARSING?

Parsing in python means, the process of analyzing a string of characters like text, code or data and converting it into a structured format that a program can understand and manipulate.

WHAT IS PDFMINER?

It is an open source library and tool designed for extracting information from pdf documents, Pdfminer primary focus is on getting and analyzing text data within pdf's.

1. **Extract text** from a PDF file.
2. **Tokenize** the text into words.
3. **Count** the frequency of each word.
4. Find words with:

   Length > 5 characters.

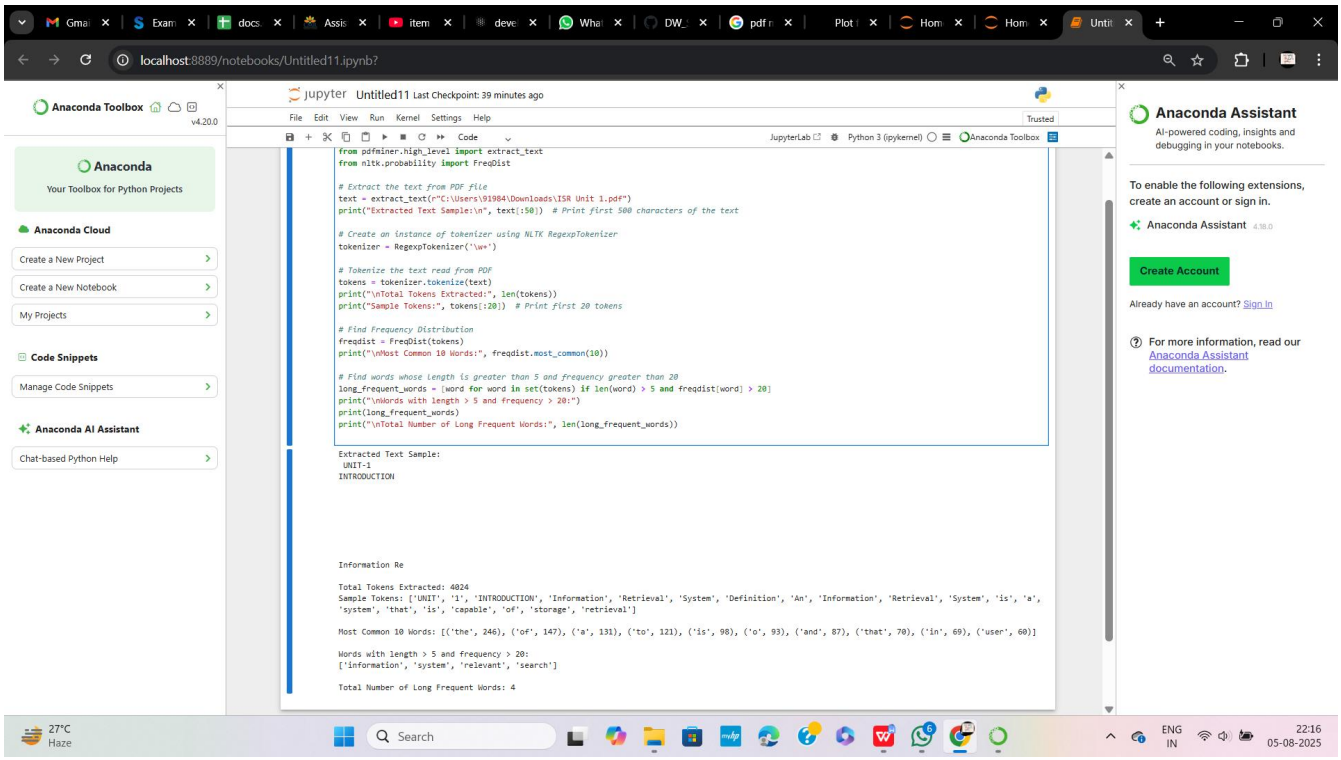
   Frequency > 20 times in the document.

5. **Plot the frequency distribution** of such words.

#code

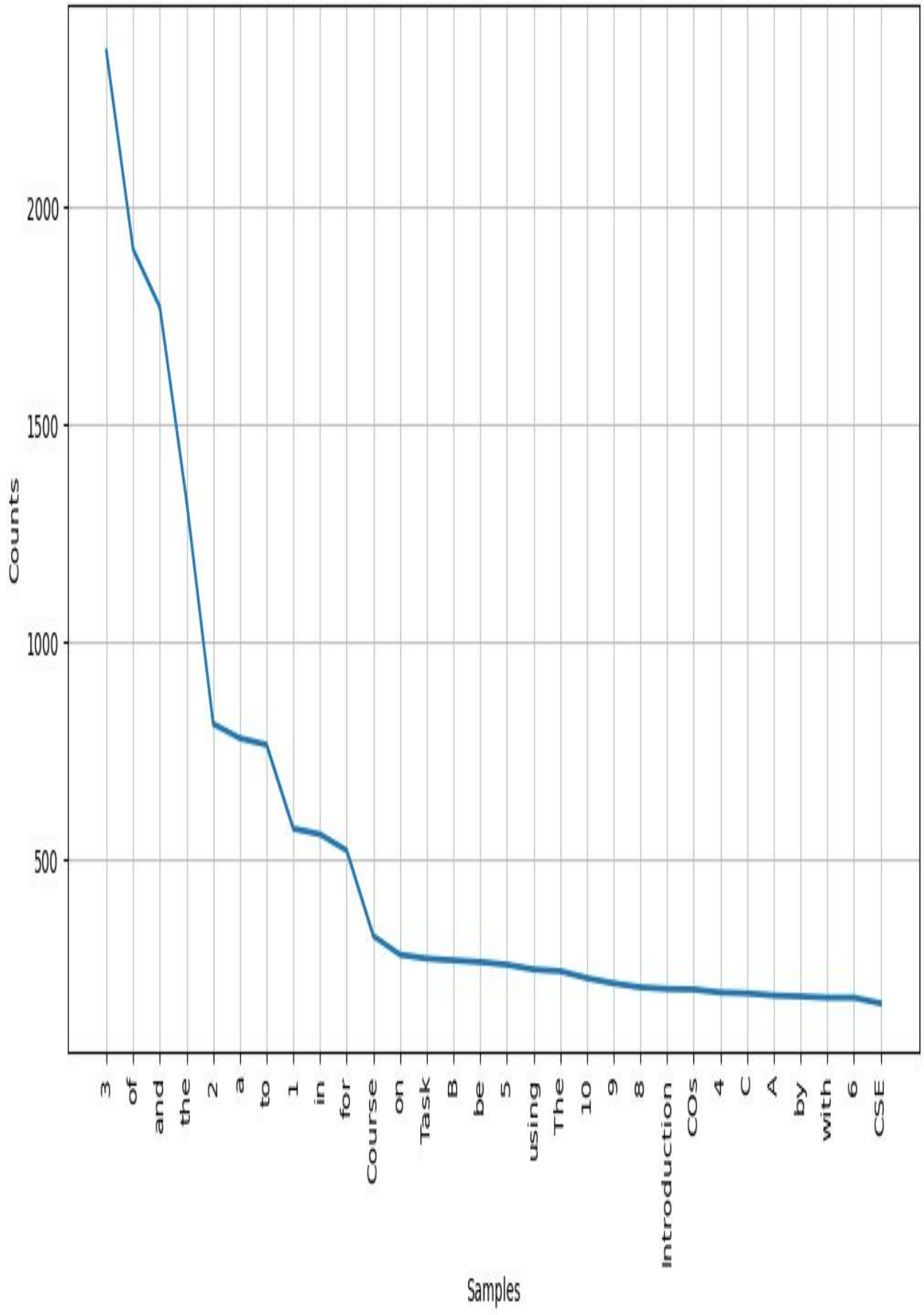
```
from nltk.tokenize import RegexpTokenizer
from pdfminer.high_level import extract_text
from nltk.probability import FreqDist

# Extract the text from PDF file
text = extract_text(r"C:\Users\91984\Downloads\ISR Unit 1.pdf")
print("Extracted Text Sample:\n", text[:50])  # Print first 500 characters of the text

# Create an instance of tokenizer using NLTK RegexpTokenizer
tokenizer = RegexpTokenizer('\w+')

# Tokenize the text read from PDF
tokens = tokenizer.tokenize(text)
print("\nTotal Tokens Extracted:", len(tokens))
print("Sample Tokens:", tokens[:20])  # Print first 20 tokens

# Find Frequency Distribution
freqdist = FreqDist(tokens)
print("\nMost Common 10 Words:", freqdist.most_common(10))

# Find words whose length is greater than 5 and frequency greater than 20
long_frequent_words = [word for word in set(tokens) if len(word) > 5 and freqdist[word] > 20]
print("\nWords with length > 5 and frequency > 20:")
print(long_frequent_words)
print("\nTotal Number of Long Frequent Words:", len(long_frequent_words))
```

OUTPUT


localhost:8889/notebooks/Untitled11.ipynb?
Anaconda Toolbox
v4.20.0
Anaconda
Your Toolbox for Python Projects
Anaconda Cloud
Create a New Project
Create a New Notebook
My Projects
Code Snippets
Manage Code Snippets
Anaconda AI Assistant
Chat-based Python Help
jupyter Untitled11 Last Checkpoint: 39 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
Anaconda Toolbox
from pdfminer.high_level import extract_text
from nltk.probability import FreqDist
# Extract the text from PDF file
text = extract_text(r"C:\Users\91984\Downloads\ISR Unit 1.pdf")
print("Extracted Text Sample:\n", text[:50]) # Print first 500 characters of the text
# Create an instance of tokenizer using NLTK RegexpTokenizer
tokenizer = RegexpTokenizer('\w+')
# Tokenize the text read from PDF
tokens = tokenizer.tokenize(text)
print("\nTotal Tokens Extracted:", len(tokens))
print("Sample Tokens:", tokens[:20]) # Print first 20 tokens
# Find Frequency Distribution
freqdist = FreqDist(tokens)
print("\nMost Common 10 Words:", freqdist.most_common(10))
# Find words whose length is greater than 5 and frequency greater than 20
long_frequent_words = [word for word in set(tokens) if len(word) > 5 and freqdist[word] > 20]
print("\nWords with length > 5 and frequency > 20:")
print(long_frequent_words)
print("\nTotal Number of Long Frequent Words:", len(long_frequent_words))
Extracted Text Sample:
UNIT-1
INTRODUCTION
Information Re
Total Tokens Extracted: 4024
Sample Tokens: ['UNIT', '1', 'INTRODUCTION', 'Information', 'Retrieval', 'System', 'Definition', 'An', 'Information', 'Retrieval', 'System', 'is', 'a', 'system', 'that', 'is', 'capable', 'of', 'storage', 'retrieval']
Most Common 10 Words: [('the', 246), ('of', 147), ('a', 131), ('to', 121), ('is', 98), ('o', 93), ('and', 87), ('that', 70), ('in', 69), ('user', 60)]
Words with length > 5 and frequency > 20:
['information', 'system', 'relevant', 'search']
Total Number of Long Frequent Words: 4
Anaconda Assistant
AI-powered coding, insights and debugging in your notebooks.
To enable the following extensions, create an account or sign in.
Anaconda Assistant
Create Account
Already have an account? Sign In
For more information, read our Anaconda Assistant documentation.
27°C
Haze
Search
ENG
IN
22:16
05-08-2025

#code

```
from nltk.tokenize import RegexpTokenizer
from nltk.probability import FreqDist
import matplotlib.pyplot as plt
from pdfminer.high_level import extract_text

# Extract text
text = extract_text(r"C:\Users\91984\Downloads\CSE_R22 syllabus book.pdf")

# Tokenize words
tokenizer = RegexpTokenizer(r'\w+')
tokens = tokenizer.tokenize(text)

# Compute Frequency Distribution
freqdist = FreqDist(tokens)

# Plot Top 30 Most Common Words (No filters)
plt.figure(figsize=(12,6))
freqdist.plot(30, cumulative=False)
plt.show()
```


Counts
2000
1500
1000
500
3
of
and
the
2
a
to
1
in
for
Course
on
Task
B
be
5
using
The
10
9
8
Introduction
COs
4
C
A
by
with
6
CSE
Samples

```
Information Re

Total Tokens Extracted: 4824
Sample Tokens: ['UNIT', 'I', 'INTRODUCTION', 'Information', 'Retrieval', 'System', 'Definition', 'An', 'Information', 'Retrieval', 'System', 'is', 'a',
'system', 'that', 'is', 'capable', 'of', 'storage', 'retrieval']

Most Common 10 Words: [('the', 246), ('of', 147), ('a', 133), ('to', 121), ('is', 98), ('i', 93), ('and', 87), ('that', 78), ('in', 69), ('user', 68)]

Words with length > 5 and frequency > 20:
['relevant', 'search', 'system', 'information']

Total Number of Long Frequent Words: 4
```

```
[2]: from nltk.tokenize import RegexpTokenizer
     from nltk.probability import FreqDist
     import matplotlib.pyplot as plt
     from pdfminer.high_level import extract_text

     # Extract text
     text = extract_text(r"C:\Users\91984\Downloads\CSE_R22 syllabus book.pdf")

     # Tokenize words
     tokenizer = RegexpTokenizer(r'\w+')
     tokens = tokenizer.tokenize(text)

     # Compute Frequency Distribution
     freqdist = FreqDist(tokens)

     # Plot Top 30 Most Common Words (No Filters)
     plt.figure(figsize=(12,6))
     freqdist.plot(30, cumulative=False)
     plt.show()
```