**Problem Statement :**

***Read a PDF file of any book and write a python 2.7 program to identify summarize each chapter in 1 page.***

***Approach :***

To distinguish a pdf file according to chapters then create summary of every chapter and save the summary to another file.

***Assumption :***

1. The book is not a scanned pdf.
2. The book has table of content page and or the chapters are distinguishable.

Read more about Pdf types: <https://blog.marconet.com/blog/bid/326753/8-types-of-pdf-standards-each-serves-a-unique-purpose>

Read more about searchable Pdf: <https://www.abbyy.com/en-eu/finereader/pdf-types>

**Installation Requirements :**

**Tested on Windows 10**

Make sure you have [Python](<http://www.python.org/>) 2.7/3.3+ and [pip](<https://crate.io/packages/pip/>)

([Windows](<http://docs.python-guide.org/en/latest/starting/install/win/>),

[Linux](<http://docs.python-guide.org/en/latest/starting/install/linux/)>)

1. **Install Sumy :**

For windows cmd :

>python -m pip install sumy

2**. Install PdfMiner :**

For windows cmd :

>python -m pip install pdfminer

3. **Install PyPDF2 :**

For windows cmd :

>python -m pip install PyPDF2

**Working :**

This program works in 3 steps :

Step 1 : Takes a pdf file as input and split it according to chapters .

Step 2 : Convert each Chapter into text file .

Step 3 : Creates Summary of every Chapter and writes it to a Summary text file.

**PDF\_Summary**

***main.py*** is a program written in python that takes *input arguments* and creates Summary of every Chapter into a file.

**Running program from Command Prompt (windows)**

Step 1 : Go into PDF\_Summary folder where you will find “main.py” python file.

Step 2 : run command

Example :

*C:\Users\AMITABH\Desktop\PDF\_Summary>main.py*

Step 3 : Input the values asked in correct manner.

1. Source PDF file with path .

Example : "C:\Users\AMITABH\Desktop\PDF\_Summary\pdf\short \_stories.pdf"

1. Path to the output Directory

Example : "C:\Users\AMITABH\Desktop\PDF\_Summary "

1. Algorithm you want to choose to create summary from the options below :

Select Algorithm

press 1 and enter for Luhn.

press 2 and enter for Lsa.

press 3 and enter for LexRank.

press 4 and enter for TextRank.

press 5 and enter for SumBasic.

press 6 and enter for KLsum.

press 0 and enter to exit.

0

Wrong Algorithm selected.

Note : Default Algorithm used is Lsa

Step 4 : You can check a Summary is created in Summary folder with file name indicating algorithm used as prefix and time stamp as suffix.

***Explanation :***

* To create summary of each chapter this program uses a Library in Python called “sumy” that takes plaintext as input file and returns summary of the file .
* First we need to split a input pdf file into a chapter wise files and save them in .pdf format in a directory named “pdf\_split\_files” with prefix name “Split\_Chapter\_”+ title of the Chapter .
* From the directory where split pdf files are saved it then converts each pdf chapter file into a text file (with same name as of pdf files) and saves them into a different directory named “Text\_Files” .
* It then passes each chapter to a Summarizer :
  + There are different algorithms that can be used to create summary such as :
  + Luhn - heuristic method ,  
    [reference](<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5392672>)
  + Latent Semantic Analysis, LSA -   
     (http://scholar.google.com/citations?user=0fTuW\_YAAAAJ&hl=en)  
     Steinberger, J. a Ježek, K. Using latent semantic analysis and summary evaluation. In Proceedings ISIM '04. 2004. S. 93-100. (<http://www.kiv.zcu.cz/~jstein/publikace/isim2004.pdf>)
  + LexRank - Unsupervised approach inspired by algorithms PageRank and HITS,  
     [reference](<http://tangra.si.umich.edu/~radev/lexrank/lexrank.pdf>)
  + TextRank - [Wikipedia](<https://en.wikipedia.org/wiki/Automatic_summarization#Unsupervised_approaches:_TextRank_and_LexRank>)
  + SumBasic - Method that is often used as a baseline in the literature. Source: [Read about SumBasic] (<http://www.cis.upenn.edu/~nenkova/papers/ipm.pdf>)
  + KL-Sum - Method that greedily adds sentences to a summary so long as it decreases the KL Divergence. Source: [Read about KL-Sum] (<http://www.aclweb.org/anthology/N09-1041>)
* In this program i have used Latent Semantic Analysis(Lsa) as default algorithm to create summary.
  + For every chapter it creates a summary of 30 lines .
* At last the summary of every chapter is written into a file Summary.txt

.

**Limitations**

1. The Program works fine with the pdf files that contains table of content and Chapterisation.
2. For pdf files that don’t have table of content it tries to map each chapter and its page number.
3. Every algorithm works differently so the result Summary may vary.

**References**

* To Convert Pdf into text file :
  + I have used pdfminer like a library in this project, the package includes the pdf2txt.py , which is used to extract text and images from pdf .The code of pdf2txt is available at <https://www.binpress.com/tutorial/manipulating-pdfs-with-python/167>
  + Documentation : http://pdfminer-docs.readthedocs.io/pdfminer\_index.html
* To split Pdf file into different chapters :
  + I have used GIT code which uses PyPDF2 as a library and code is available at
  + Link to GIT Code : https://github.com/joliver1981/PDFSplitter
  + Read More about PyPDF2:(<https://pypi.py>[thon.org/pypi/PyPDF2/1.26.0](https://pypi.python.org/pypi/PyPDF2/1.26.0))
* To create Summary from plain text files :
  + I have used sumy as a library and code is available at
  + Documentation : (<https://pypi.python.org/pypi/sumy>)