**Problem Statement :**

***Read a patent PDF file(text searchable) and write a python 2.7 program to identify summarize each section in 1 page.***

***Approach :***

First convert pdf file into text file .

Search string and separate each section into a different txt file

Create summary of every section and save the summary to files with section names.

***Assumption :***

1. The patent is not a scanned pdf.

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 convert it into text .

Step 2 : split it according to section by searching strings.

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

**PATENT\_Summary**

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

Information abount patent files :

The program search for following sections :

1. ABSTRACT
2. BACKGROUND OF THE INVENTION
3. SUMMARY OF THE INVENTION
4. BRIEF DESCRIPTION OF THE DRAWING
5. DETAILED DESCRIPTION OF THE INVENTION
6. I claim::

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

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

Step 2 : run command

Example :

C:\Users\AMITABH\Desktop\PDF\_Summary\patentSummary.py =======

Enter Soruce file with path

C:\Users\AMITABH\Desktop\PDF\_Summary\pdf\patent.pdf

Enter file name

patent

Found source PDF file

Enter Output Directory path

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

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.

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 section 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 convert pdf file into a text file .
* From the directory where text file is saved , open it in read mode and search strings to create separate files .
* It then passes each section file 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 searchable text.
2. 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 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>)