Creating PDF and WORD documents from synthesized text

The Portable Document Format, or PDF, is a file format that can be used to
present and exchange documents reliably across operating systems.
In 1990, the structure of a PDF document was defined by Adobe. The idea behind
the PDF format is that transmitted data/documents look exactly the same for
both parties that are involved in the communication process - the creator,
author or sender, and the receiver. PDF is the successor of the PostScript
format, and standardized as ISO 32000-2:2017.

In terms of novel document synthesis the primary tasks are reading PDFs and
extracting their text content, and conversely, reading synthesized text and
writing to a new PDF document.

Extracting text from a PDF is achieved with the assistance of the Python
module 'PyMuPDF' using its import name 'fitz' (name of the original software
version). For each pdf-document file in the directory '/pdf' text is extracted
from each page of the document. Then lines are split and written to a
corresponding file in '/corpus'.

* [1] example command line for pdf to text-file

* root> py pdf2txt.py - *writes all pdf-pages of all pdf-files in pdf/ to
text files in /corpus with name 'text<i><j> where i refers to the document
number and j to the page number (uses imported module PyMuPDF)

```
# pdf2txt.py

import os
import fitz

pdfpath = 'pdf/'
corpuspath = 'corpus/'
pdfpath_ = 'pdf_/'
```

```
# index of pdf files in pdfpath
i = 0

for entry in os.listdir(pdfpath):
    fd = os.path.join(pdfpath, entry)
    if os.path.isfile(fd):
        filepath = os.path.join(pdfpath, entry)

        # create pdf doc
        doc =  fitz.open(filepath)
        title = doc.metadata['title']

        # read text from each page
        j = 0
        for page in doc:
            text = page.getText("text")
            lines = text.split('\n')
            for m in range(len(lines)):
                print(f'lines[{m}] = {lines[m]}')


            # @@@ create text-file to write to corpus
            target = corpuspath + 'text' + str(i) + str(j) + '.txt'
            fd = open(target, 'a')
            fd.writelines(lines)

            #increment page index
            j = j + 1


        #increment pdf document index
        i = i + 1
```

Conversely to write synthesized text back to a PDF the Python module 'fpdf'
is used. Text files are read from '/corpus' and written as PDF to '/pdf_'.

* [2] text to pdf-file

* root> py txt2pdf.py - *writes all txt-files in corpus/ to pdf files in /pdf_
* using the names found in /corpus (uses imported module fpdf)

```
# txt2pdf.py

import os
from fpdf import FPDF



corpuspath = 'corpus/'
pdfpath_ = 'pdf_/'

# index of pdf files in pdfpath
i = 0

for entry in os.listdir(corpuspath):
    fd = os.path.join(corpuspath, entry)
    #if os.path.isfile(os.path.join(pdfpath, entry)):
    if os.path.isfile(fd):
        filepath = os.path.join(corpuspath, entry)

        # @@@ read text-file
        fd = open(filepath, 'r')
        text = fd.read()
        lines = text.split('\n')

        # @@@ create pdf-file to write to pdfpath_
        pdf = FPDF()
        pdf.add_page()
        pdf.set_font('Arial', size=10)

        #create cells for each line
        for j in range(len(lines)):
            pdf.cell(100,5, txt=lines[j], ln=1, align='L')

        # write pdf-file
        target = pdfpath_ + 'text' + str(i) + '.pdf'
        if not os.path.exists(target):
            open(target, 'w').close()
        pdf.output(target)



        # increment text-file index
        i = i + 1
```

In addition, closely associated with writing of synthesized text to PDF is
writing the same synthesized text back to a word .docx-file using the Python
module 'python-docx'. Text files are read from '/corpus' and written as

docx-files to '/word_'.

* [3] text to word docx-file

* root> py txt2word.py   *writes all txt-files to word docx-files in
word_/
using the names found in /corpus   (uses python-docx module)

```
# txt2word.py

import os
import docx



corpuspath = 'corpus/'
wordpath_ = 'word_/'

# index of pdf files in pdfpath
i = 0

for entry in os.listdir(corpuspath):
    fd = os.path.join(corpuspath, entry)
    #if os.path.isfile(os.path.join(pdfpath, entry)):
    if os.path.isfile(fd):
        filepath = os.path.join(corpuspath, entry)

        # @@@ read text-file
        fd = open(filepath, 'r')
        text = fd.read()
        lines = text.split('\n')

        # @@@ create word-file to write to /word_
        target = wordpath_ + 'text' + str(i) + '.docx'
        doc = docx.Document()
        doc.add_heading('text' + str(i), 0)
        doc.add_paragraph(text)
        doc.add_page_break()
        doc.save(target)


        # increment text-file index
        i = i + 1
```