RnD CS 490

REPORT
QUESTION BANK

Team Members

150050067 - Nagendra

150050087 - Sai Sharath

Introduction

Most of the professors find it convenient to have a database of questions from which they can generate new question papers. In order to fill the database with questions from various sources of question papers, it is desirable to have a tool which can automatically extract the questions.

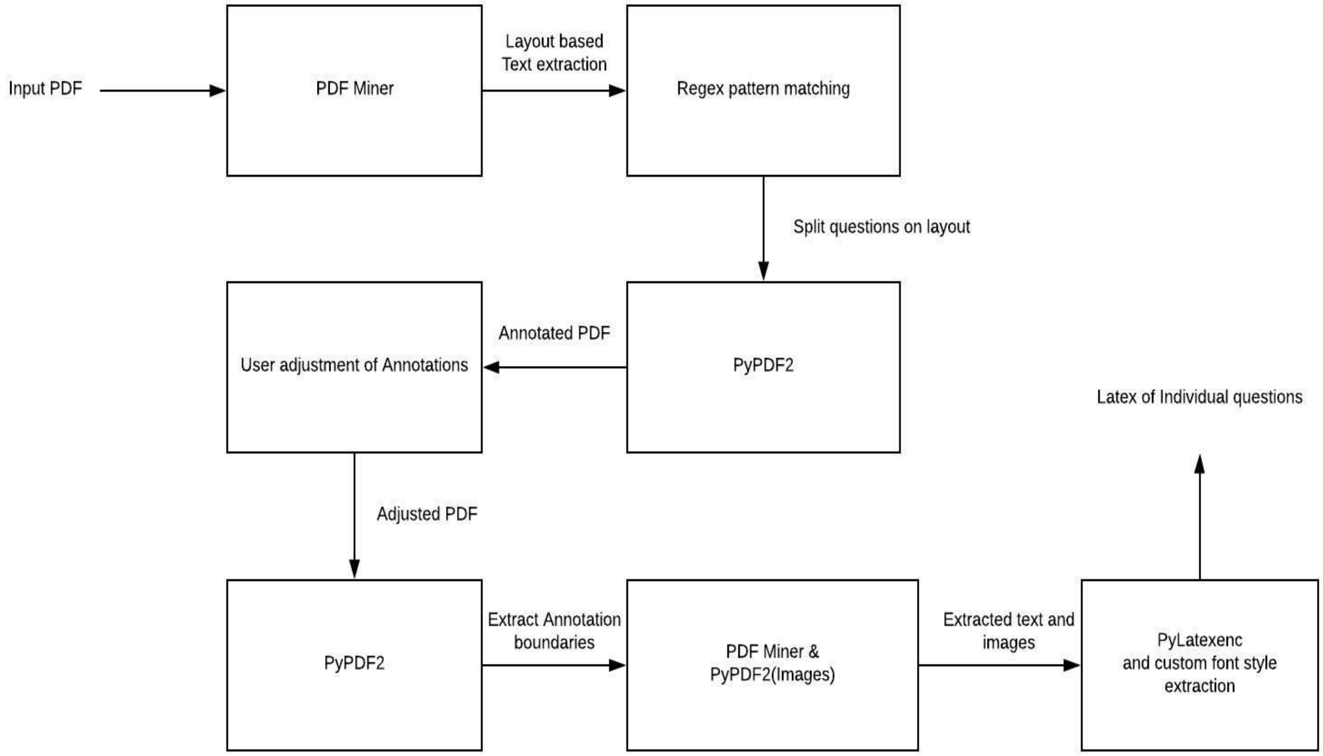
Question papers available online are found in various formats among which PDF is the most common format. Moreover, question papers in other formats can be converted to PDF without much loss of information. Hence, a tool capable of extracting questions from PDFs would solve our problem.

Description

We have explored several ways to extract the question papers and store them in an appropriate format. Upon reconverting to a PDF, it is desirable to be able to reproduce the font types, styles and images as close to original as possible. For this reason, we have decided to store the questions extracted in latex format. This also simplifies the generation of question papers from question bank.

The problem of splitting question paper to individual questions turns out to be a much harder one. We have decided that we cannot guarantee the correctness of the tool without manual verification. For this purpose, we create annotation boxes around each question region. The annotated PDF can then be inspected manually in a tool such as foxit reader for any adjustments of the annotations. After these manual adjustments our tool extracts each annotated region as a question along with images and other metadata.

Architecture


Input PDF
PDF Miner
Layout based Text extraction
Regex pattern matching
Split questions on layout
PyPDF2
Annotated PDF
User adjustment of Annotations
Adjusted PDF
PyPDF2
Extract Annotation boundaries
PDF Miner & PyPDF2(Images)
Extracted text and images
PyLatexenc and custom font style extraction
Latex of Individual questions

The details mentioned in the above description lead to the architecture taking the above form. The Input PDF first goes through PDF Miner for the extraction of PDF layout in the form of a tree along with coordinates of each layout. The extracted layout also contains all the metadata of text, images etc... The text part of the entire layout is subjected to a regex pattern matching based on which the layout of individual questions is determined.

This extracted layout is used to draw annotation boxes around each question with the help of PyPDF2 library. User can then make adjustments to the annotated PDF and submit it for extracting each question as a latex text. The adjusted PDF then goes through PyPDF2 for the extraction of annotation boxes. These annotation box boundaries are used to extract the text and images inside these boxes. This is performed using PDF miner and PyPDF2 (for png images). The extracted text is then converted to latex using Pylatexenc. The images and font styles are adopted into the latex programmatically.

Technical Details

The tool is based on python2 programming language. We have used the libraries pdfminer, PyPDF2 for extracting data from PDF file.

The library pdfminer has support for extracting text from the file based on layout analysis. We have used pdfminer for extracting and analysing the question papers and then splitting them to questions. We also used pdfminer to extract metadata like font styles and images.

We used PyPDF2 for creating annotations boxes and also for extracting them from an annotated PDF, from which the text is extracted using pdfminer.

Difficulties and Challenges

Most of the libraries available had little to no documentation and required thorough understanding of the codebase for effective usage. Coming up with annotations to extract part of PDF, trying to build a visual to integrate with browser were some of the key challenges encountered.

Moreover, there was no single library which worked flawlessly, each library worked only for certain purposes. Some images could only be extracted by pdfminer and some only by PyPDF2, poppler extracted excess blank images. In some cases, none of the libraries worked perfectly. Annotations made through neither PyPDF nor Poppler were working perfectly.

Future Work

The task of detecting bounding boxes of each question has been accomplished by searching for a fixed set of regex patterns. As such, this method is naive and can fail if the given question paper uses a different format. This process can be improved by adding a lot of regex patterns or employing different intelligent techniques.

We have mentioned the usage of a tool to adjust the annotated PDFs. As of now, the user has to manually adjust the PDF in a tool such as foxit reader. It is desirable to have such tool integrated with the browser itself. We have explored some paid tools for this purpose but deciding on which tool to use and integrating it with the browser requires a deeper look.

Conclusion

Working with PDFs can be quite difficult. Most of the open source libraries aren't well documented and do not work under all circumstances.

Moreover, each library provides only certain functionalities. Building a tool to extract data from PDFs often requires the usage of multiple libraries.

Related works

A similar problem of extracting questions from question papers of latex format has been done in an MTP. There is already some work done on storing the questions into a database which is based on the extractor from latex format.

References

Pdfminer https://pypi.org/project/pdfminer/

PyPDF2 https://pypi.org/project/PyPDF2/

Pdf2image https://github.com/Belval/pdf2image

Foxit Reader https://www.foxitsoftware.com/pdf-reader/

Pdfminer docs https://pdfminer-docs.readthedocs.io/programming.html

Adding Annotations using PyPDF2

https://github.com/mstamy2/PyPDF2/issues/107

https://gist.github.com/agentcooper/4c55133f5d95866acdee5017cd318558

Extracting Images pdfminer
https://github.com/dpapathanasiou/pdfminer-layout-scanner/blob/master/layout_scanner.py

Extracting Images PyPDF2
https://github.com/mstamy2/PyPDF2/blob/master/Scripts/pdf-image-extractor.py

Pylatexenc for text to latex https://pypi.org/project/pylatexenc/

Poppler https://poppler.freedesktop.org/

Qoppa https://www.qoppa.com/pdfautomation/

Flowpaper https://flowpaper.com/flipbook-software/

PDFTron https://www.pdftron.com/blog/webviewer/pdfnetjs-html5-pdf-viewer-and-editor/

Question Bank https://github.com/sharath1709/Question-Bank