Extract Text Excluding Headers

how to extract text from pdf files but like headers and footers is excluded

To extract text from PDF files while excluding headers and footers, you'll need a method that not only reads the text but also understands the layout of the PDF pages. This way, you can programmatically ignore the content located in the header and footer areas. Below are steps and code examples using Python and the `pdfminer.six` library to achieve this.

Using Python and `pdfminer.six`

Step 1: Install `pdfminer.six`

First, ensure you have `pdfminer.six` installed:

bash

```
pip install pdfminer.six
```

Step 2: Write a Script to Extract Text Excluding Headers and Footers

Here's a Python script that extracts text from a PDF while excluding the headers and footers based on their positions on the page:

python

```
from pdfminer.high_level import extract_pages
from pdfminer.layout import LTTextContainer, LTTextLineHorizontal
import sys

def extract_text_without_headers_and_footers(pdf_path, header_height=50,
footer_height=50):
    text_content = ""
    for page_layout in extract_pages(pdf_path):
        page_height = page_layout.height
        page_text = ""
        for element in page_layout:
            if isinstance(element, LTTextContainer):
                for text_line in element:
                    if isinstance(text_line, LTTextLineHorizontal):
                        # Get the vertical position of the text line
                        y0, y1 = text_line.y0, text_line.y1
                        # Include text that is not in the header or footer area
                        if y1 < (page_height - header_height) and y0 > footer_height:
                            page_text += text_line.get_text()
        text_content += page_text
    return text_content

# Usage example
if __name__ == "__main__":
```

```
    pdf_path = 'your_file.pdf'
    output_text = extract_text_without_headers_and_footers(pdf_path)
    print(output_text)
```

Explanation:

- **Import Necessary Modules:** We import functions and classes needed to parse the PDF and understand its layout.
- **Define the Function:** `extract_text_without_headers_and_footers` takes the PDF path and optional header and footer heights.
- **Parse Each Page:** We loop through each page and get its layout.
- **Filter Text Lines:** For each text line, we check if it's within the body area (excluding the top `header_height` units and bottom `footer_height` units).
- **Concatenate Text:** We build a string with all the text that isn't part of the header or footer.

Step 3: Adjust Header and Footer Heights

You might need to adjust `header_height` and `footer_height` based on your PDF's specific layout:

```
python

output_text = extract_text_without_headers_and_footers(pdf_path, header_height=70,
footer_height=70)
```

Alternative Method: Remove Repeated Headers and Footers

If the headers and footers have identical text on each page, you can remove them by:

1. **Extracting Text from Each Page Separately:**
   - Use `pdfminer.six` to extract text from each page into a list.
2. **Identify Repeated Lines:**
   - Compare the beginning and ending lines of each page to find common text.
3. **Exclude Repeated Lines:**
   - Remove the identified header and footer lines from each page's text.

Sample Code:

```
python

from pdfminer.high_level import extract_text

def extract_text_by_page(pdf_path):
    from pdfminer.pdfpage import PDFPage
    from io import StringIO
    from pdfminer.pdfinterp import PDFResourceManager
    from pdfminer.layout import LAParams
    from pdfminer.converter import TextConverter

    text_pages = []
    with open(pdf_path, 'rb') as fh:
        for page in PDFPage.get_pages(fh):
            output = StringIO()
            resource_manager = PDFResourceManager()
            device = TextConverter(resource_manager, output, laparams=LAParams())
            interpreter = PDFPageInterpreter(resource_manager, device)
```

```
            interpreter.process_page(page)
            text = output.getvalue()
            text_pages.append(text)
            device.close()
            output.close()
    return text_pages

def remove_common_headers_footers(text_pages):
    lines_per_page = [page.split('\n') for page in text_pages]
    header = None
    footer = None

    # Assuming at least two pages to compare
    if len(lines_per_page) >= 2:
        first_page_lines = lines_per_page[0]
        second_page_lines = lines_per_page[1]

        # Identify header
        for i in range(min(len(first_page_lines), len(second_page_lines))):
            if first_page_lines[i] == second_page_lines[i]:
                header = first_page_lines[i]
            else:
                break

        # Identify footer
        for i in range(1, min(len(first_page_lines), len(second_page_lines)) + 1):
            if first_page_lines[-i] == second_page_lines[-i]:
                footer = first_page_lines[-i]
            else:
                break

    cleaned_text = ""
    for page_lines in lines_per_page:
        if header and page_lines[0] == header:
            page_lines = page_lines[1:]
        if footer and page_lines[-1] == footer:
            page_lines = page_lines[:-1]
        cleaned_text += '\n'.join(page_lines) + '\n'
    return cleaned_text

# Usage example
pdf_path = 'your_file.pdf'
text_pages = extract_text_by_page(pdf_path)
cleaned_text = remove_common_headers_footers(text_pages)
print(cleaned_text)
```

Note: This method works best when headers and footers are exactly the same on each page.

Considerations

- **Variable Headers/Footers:** If headers and footers vary (like page numbers), you'll need a more sophisticated method to identify and exclude them.
- **Complex Layouts:** For PDFs with complex layouts, consider using more advanced libraries like `PyMuPDF` (fitz), which provides more control over the PDF content.

Using `PyMuPDF` (Alternative Library)

`PyMuPDF` allows for precise control over page content, including bounding boxes for text blocks.

Install `PyMuPDF`:

```bash
pip install PyMuPDF
```

Sample Code:

```python
import fitz

def extract_text_without_headers_and_footers(pdf_path, header_height=50,
footer_height=50):
    doc = fitz.open(pdf_path)
    text_content = ""
    for page in doc:
        page_height = page.rect.height
        blocks = page.get_text("blocks")
        for b in blocks:
            # b = [x0, y0, x1, y1, "text", block_type, block_no]
            y0 = b[1]
            y1 = b[3]
            if y1 < (page_height - header_height) and y0 > footer_height:
                text_content += b[4]
    return text_content

# Usage example
pdf_path = 'your_file.pdf'
output_text = extract_text_without_headers_and_footers(pdf_path)
print(output_text)
```

Conclusion

By using libraries that provide access to the layout and positional information of text within PDF files, you can effectively exclude headers and footers during text extraction. Adjust the parameters like `**header_height**` and `**footer_height**` based on your document's formatting to achieve the best results.