Interesting Papers

SDML, Ryan Chesler

PDFTriage: Question Answering over Long, Structured Documents

Jon Saad-Falcon
Stanford University
jonsaadfalcon@stanford.edu

Joe Barrow
Adobe Research
jbarrow@adobe.com

Alexa Siu
Adobe Research
asiu@adobe.com

Ani Nenkova
Adobe Research
nenkova@adobe.com

David Seunghyun Yoon
Adobe Research
syoon@adobe.com

Ryan A. Rossi
Adobe Research
ryrossi@adobe.com

Franck Dernoncourt
Adobe Research
dernonco@adobe.com

Problem Statement

Large Language models struggle to handle structure from PDFs, websites, books, etc.

Q1 "Can you summarize the key takeaways from pages 5-7?"

Q2 "What year *[in table 3]* has the maximum rev-enue?"

Typical Approach

RAG (Retrieval Augmented Generation)

- Chunking up the data
- Have an embedding model turn the text chunks into embeddings
- Turn the query into an embedding
- Find the most similar chunks
- Pass that in as context to the the LLM

Might get lucky for some queries if it happens that the page number or figure number or other identifier is written out in plain text but still unlikely it returns the correct chunks

Model cannot succeed if retrieval of the correct chunk fails

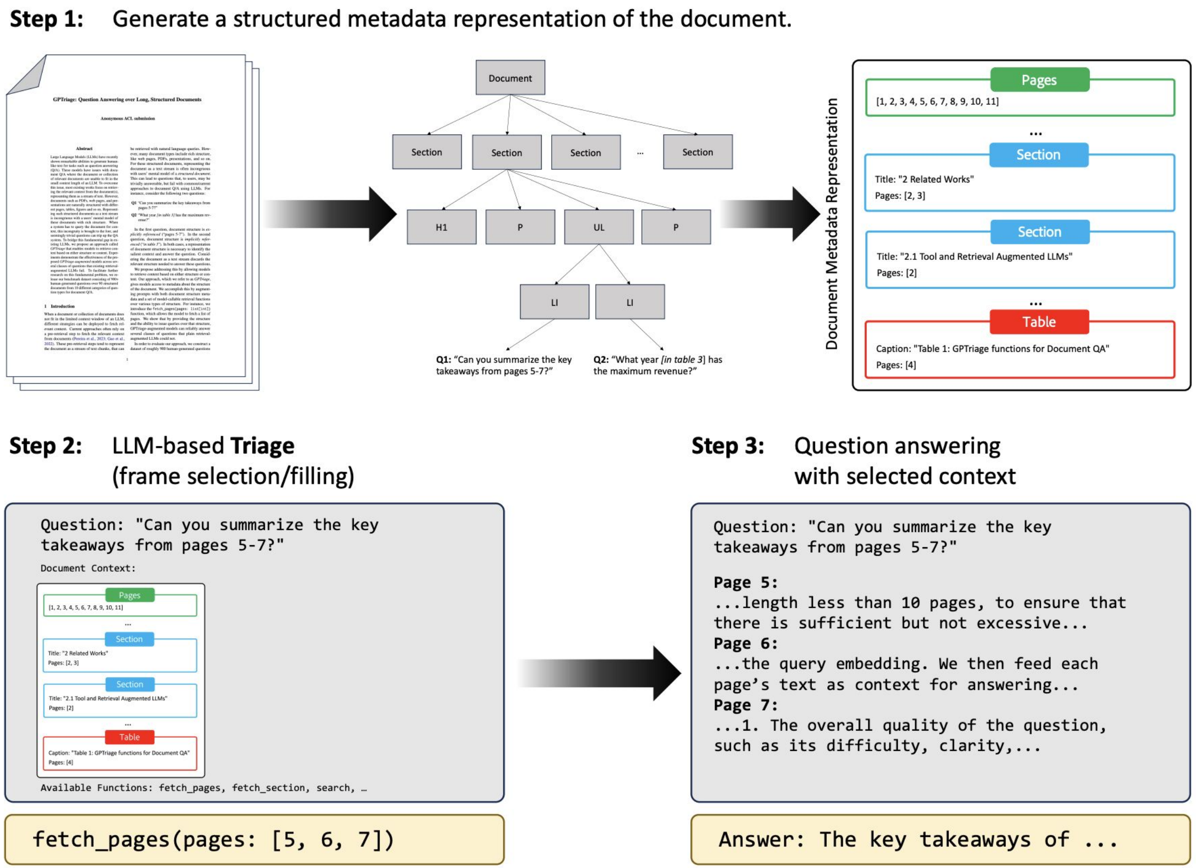
New Approach

Give the LLM its own retrieval tools based on structure

Function	Description
`fetch_pages`	Get the text contained in the pages listed.
`fetch_sections`	Get the text contained in the section listed.
`fetch_figure`	Get the text contained in the figure caption listed.
`fetch_table`	Get the text contained in the table caption listed.
`retrieve`	Issue a natural language query over the document, and fetch relevant chunks.

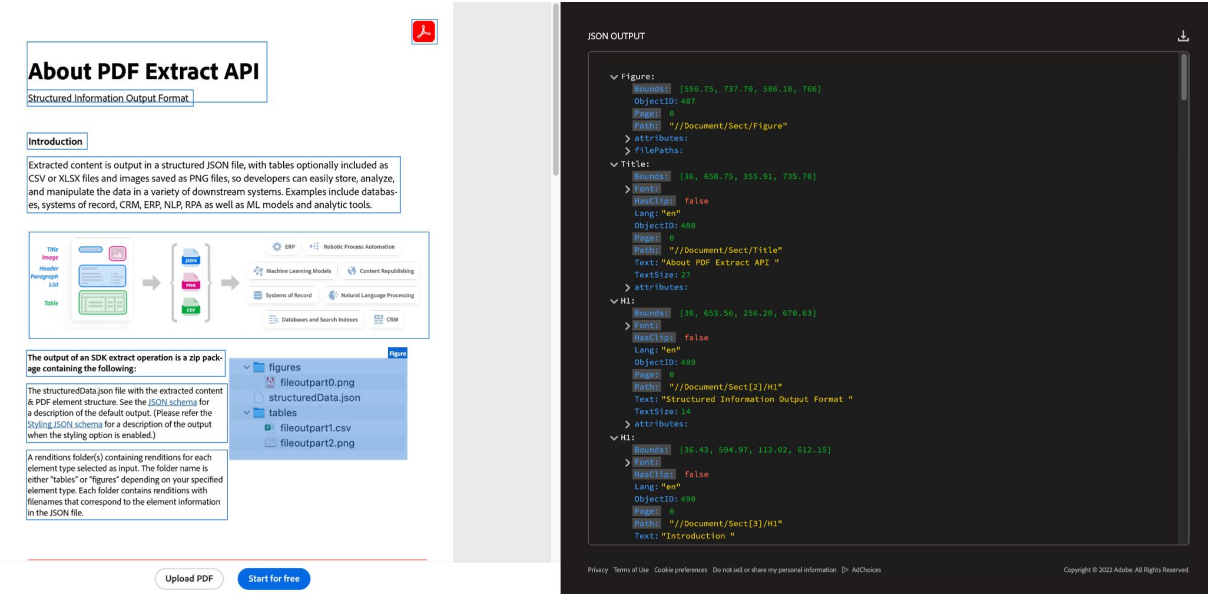
Table 2: PDFTriage Functions for Document QA.

LLM interprets the query and converts it to a function call to a tool that will do the retrieval operation it needs to get the relevant information


Step 1: Generate a structured metadata representation of the document.
GPTriage: Question Answering over Long, Structured Documents
Document
Section
Section
Section
...
Section
H1
P
UL
P
LI
LI
Q1: "Can you summarize the key takeaways from pages 5-7?"
Q2: "What year [in table 3] has the maximum revenue?"
Document Metadata Representation
Pages
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
...
Section
Title: "2 Related Works"
Pages: [2, 3]
Section
Title: "2.1 Tool and Retrieval Augmented LLMs"
Pages: [2]
...
Table
Caption: "Table 1: GPTriage functions for Document QA"
Pages: [4]
Step 2: LLM-based Triage (frame selection/filling)
Question: "Can you summarize the key takeaways from pages 5-7?"
Document Context:
Available Functions: fetch_pages, fetch_section, search, …
fetch_pages(pages: [5, 6, 7])
Step 3: Question answering with selected context
Question: "Can you summarize the key takeaways from pages 5-7?"
Page 5:
...length less than 10 pages, to ensure that there is sufficient but not excessive...
Page 6:
...the query embedding. We then feed each page's text as context for answering...
Page 7:
...1. The overall quality of the question, such as its difficulty, clarity,...
Answer: The key takeaways of ...

New Approach

Use Adobe Extract API to convert from PDF to HTML-like tree, Converted to JSON and then markdown



Evaluation

- Pulled 1k documents from Common Crawl
- Asked people on mechanical turk to write salient questions
- Classified returned questions into various categories

1. **Figure Questions** (6.5%): Ask a question about a figure in the document.
2. **Text Questions** (26.2%): Ask a question about the document.
3. **Table Reasoning** (7.4%): Ask a question about a table in the document.
4. **Structure Questions** (3.7%): Ask a question about the structure of the document.
5. **Summarization** (16.4%): Ask for a summary of parts of the document or the full document.
6. **Extraction** (21.2%): Ask for specific content to be extracted from the document.
7. **Rewrite** (5.2%): Ask for a rewrite of some text in the document.

RAG VS PDFTriage

Compared against page retrieval and chunk retrieval

Manually Evaluated by human annotators on Upwork

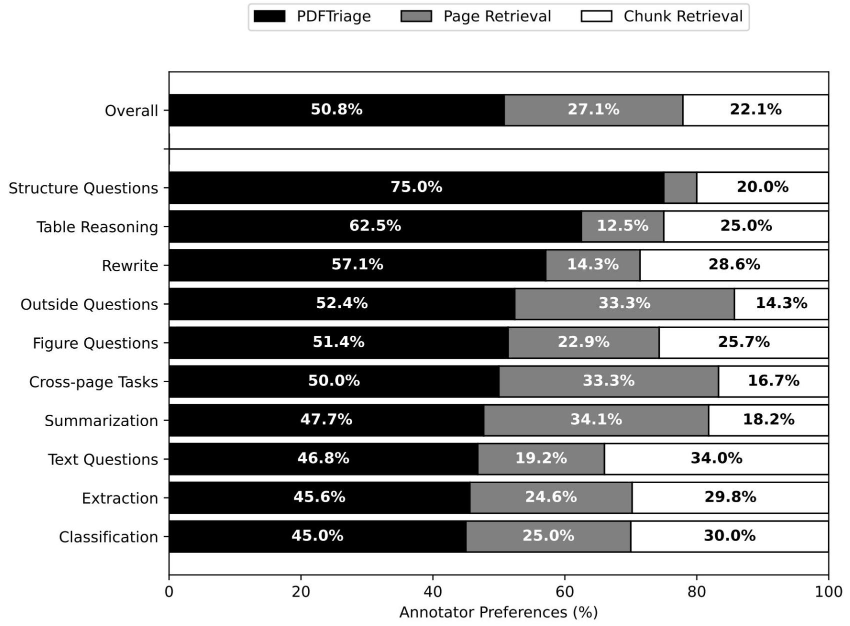


Figure 3: **User Preferences between PDFTriage and Alternate Approaches**: Overall, PDFTriage-generated answers were favored the most by the users, claiming 50.8% of the top-ranked answers overall. Furthermore, PDFTriage answers ranked higher on certain multi-page tasks, such as structure questions and table reasoning, while ranking lower on generalized textual tasks, such as classification and text questions. However, across all the question categories, PDFTriage beat both the Page Retrieval and Chunk Retrieval approaches on a head-to-head ranking.

Segmentation

- Labeling Segmentation is very tedious and time consuming
- Stretching this data further with a semi supervised approach could yield great returns
- Very often people have a lot of data but they don't have much labeled data

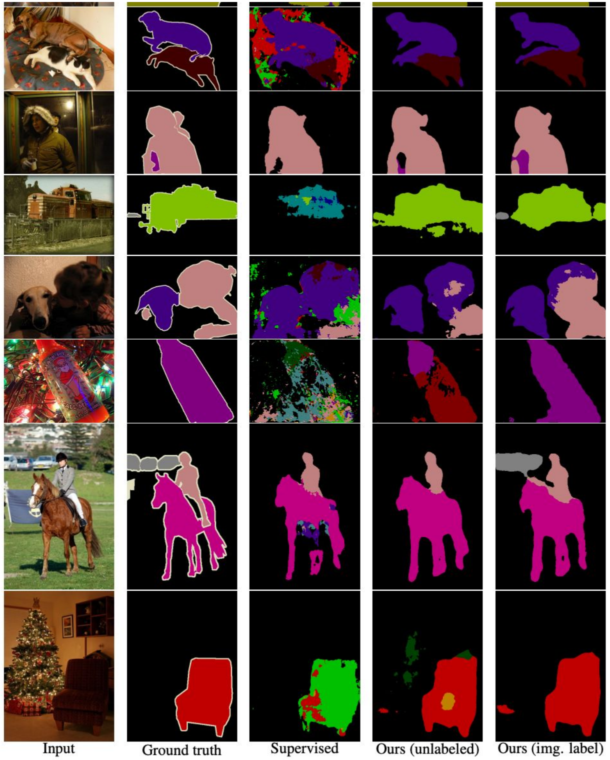


Figure 8: **Qualitative results of PASCAL VOC 2012.** Models are trained with 1/16 pixel-level labeled data in the training set.

What is pseudolabeling?

- Training a model on the data that is already available and labeled
- Using that model to make predictions on a larger pool of unlabeled data
- Retraining a new model on labels + new pseudolabels

Essentially expands your dataset with some weak and uncertain labels

Can work for most problem domains, largely dependent on the skill of the first model

PseudoSeg: Designing Pseudo Labels for Semantic Segmentation

Yuliang Zou[1*] **Zizhao Zhang**[2] **Han Zhang**[3] **Chun-Liang Li**[2]
Xiao Bian[2] **Jia-Bin Huang**[1] **Tomas Pfister**[2]
[1]Virginia Tech [2]Google Cloud AI [3]Google Brain

Pseudolabeling is a bit tricky for segmentation because the labels are much more granular, more complicated than just classification

PseudoSeg proposes a single stage way to utilize the unlabeled data

Hacking out a reliable pseudolabel

Grad-CAM - Class Activation Maps can be mapped to coarse segmentations

https://github.com/jacobgil/pytorch-grad-cam

SGC - Self-guided Attention Grad-cam

Decoder - Direct prediction from the segmentation head of the network

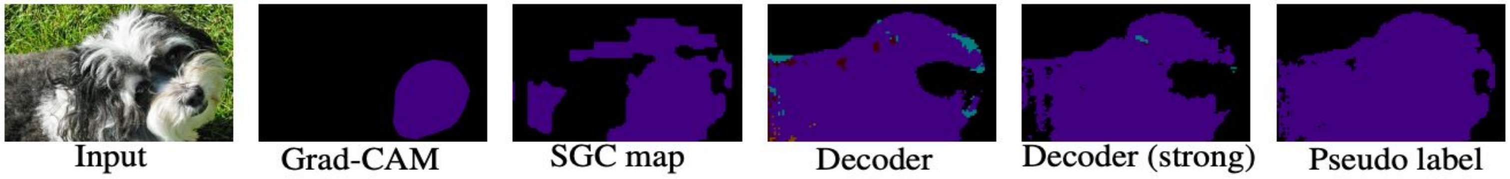


Figure 2: **Visualization of pseudo labels and other predictions.** The generated pseudo label by fusing the predictions from the decoder and SGC map is used to supervise the decoder (strong) predictions of the strongly-augmented counterpart.

Results and ablations

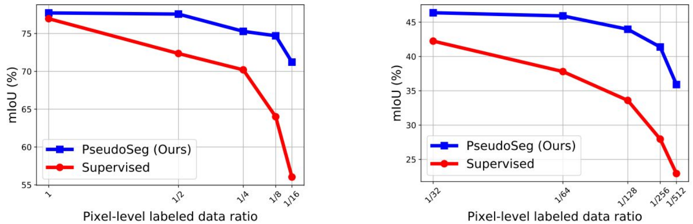
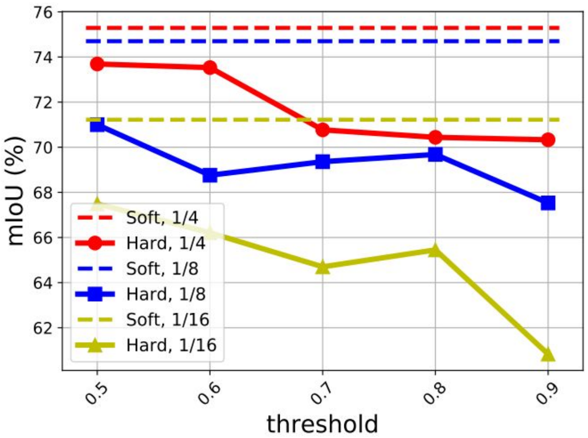


Figure 4: Improvement over the strong supervised baseline, in a semi-supervised setting (w/ image-level labeled data) on VOC12 val (left) and COCO val (right).

Table 5: **Comparison to alternative pseudo labeling strategies.** We conduct experiments using 1/4, 1/8, 1/16 of the pixel-level labeled data, the exact numbers of images are shown in the brackets.

Source	Using image-level labels	1/4 (366)	1/8 (183)	1/16 (92)
Decoder only	-	70.22	69.35	53.20
SGC only	-	67.07	62.61	53.42
Calibrated fusion	-	**73.79**	**73.13**	**67.06**
Decoder only	✓	73.95	73.05	67.54
SGC only	✓	71.73	67.57	64.26
Calibrated fusion	✓	**75.29**	**74.70**	**71.22**



(c) Soft and hard label