RAGS

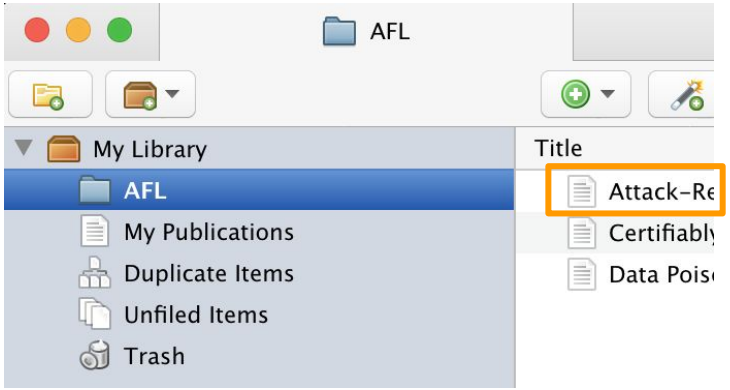
Research Assistant for Graduate Students

Shaobo Cui, Ryan Sullivan, Kamala Varma

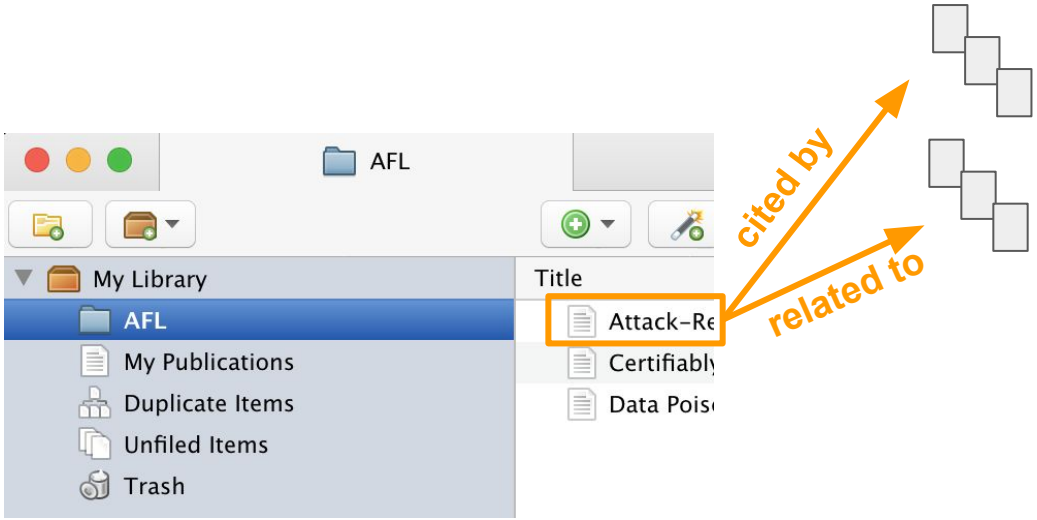
Outline

- Paper data collection and storage
- Recommendation algorithm and data used
- User interface
- Demo
- Limitations and future work

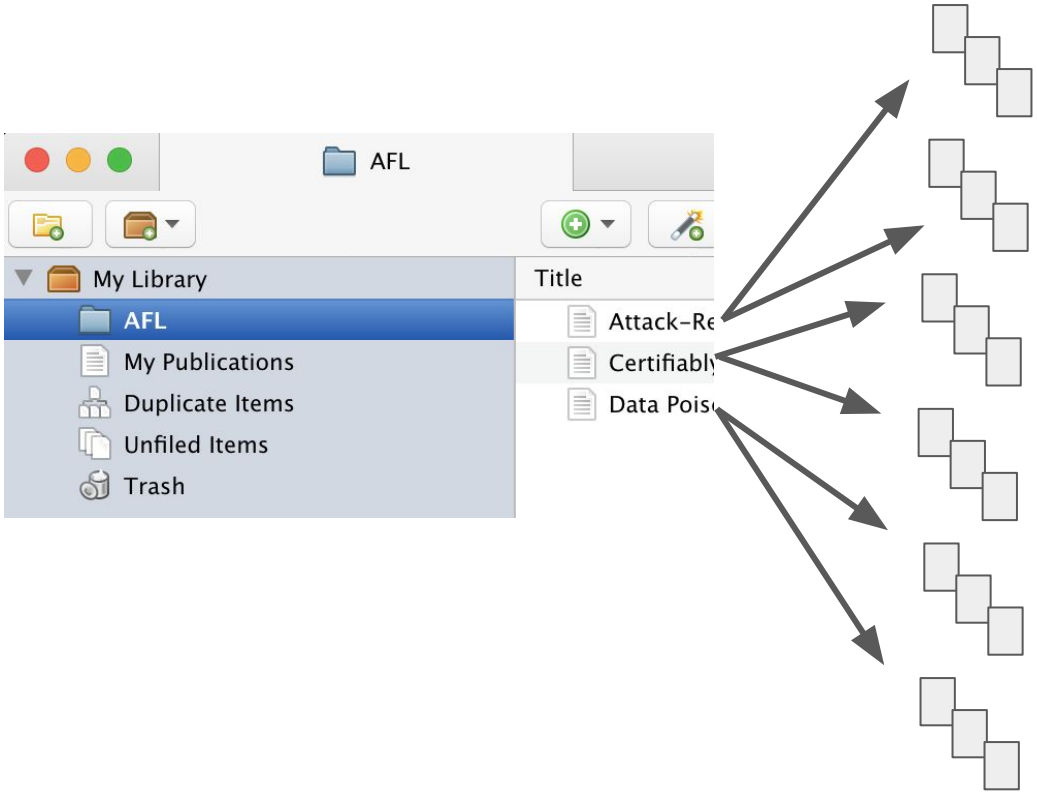
Data Collection - Building a Corpus of Papers



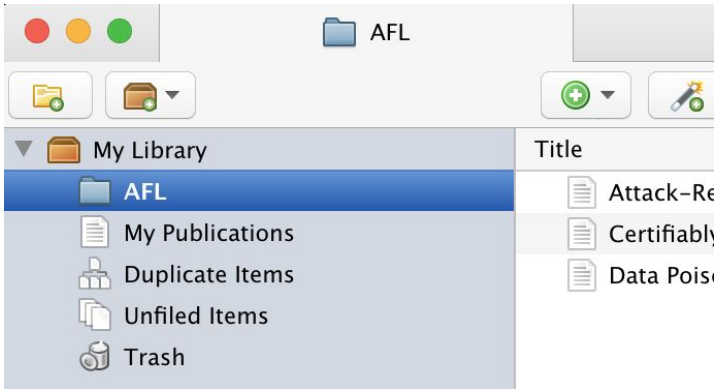
Data Collection - Building a Corpus of Papers

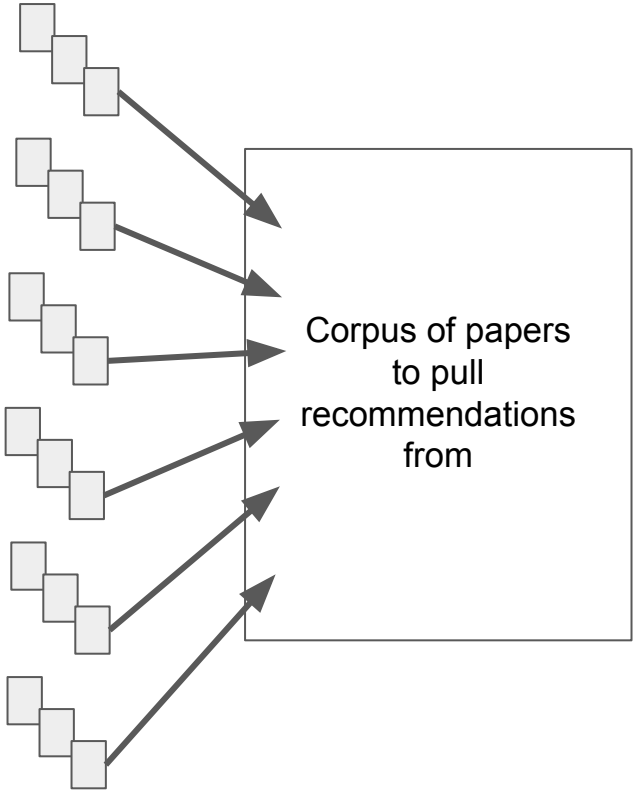


Data Collection - Building a Corpus of Papers



Data Collection - Building a Corpus of Papers





Overview of Paper Data Access-Related Functionality

```
def title_from_scholar_search_result(url):

def find_end_of_title(s, start_ind):

def paper_data_from_zotero(n=-1):

def dictionary_from_google_scholar(title):
```

Get paper info from Zotero library given user ID or from Google Scholar given paper title or url of search result

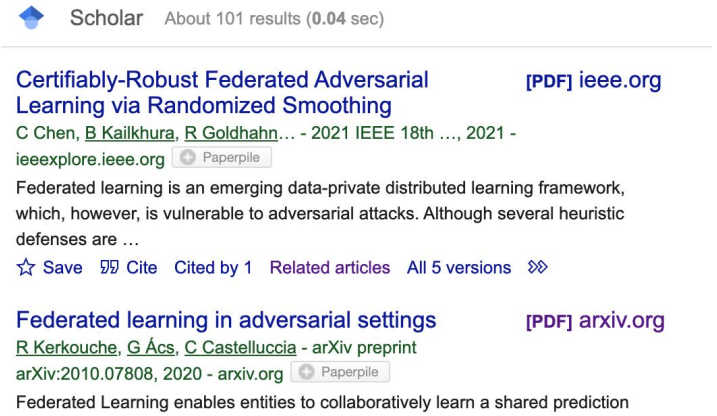
Get full paper text or extract paper abstract

```
def text_from_paper_dict(paper, abstract=True):

def text_from_google_scholar(title, abstract=True):

def full_text_from_pdf(pdf):

def abstract_from_pdf(pdf):

def abstract_from_pub_page(url):

def extract_abstract(text):
```

- problematic

Overview of Paper Data Access-Related Functionality

```
def title_from_scholar_search_result(url):

def find_end_of_title(s, start_ind):

def paper_data_from_zotero(n=-1):

def dictionary_from_google_scholar(title):
```

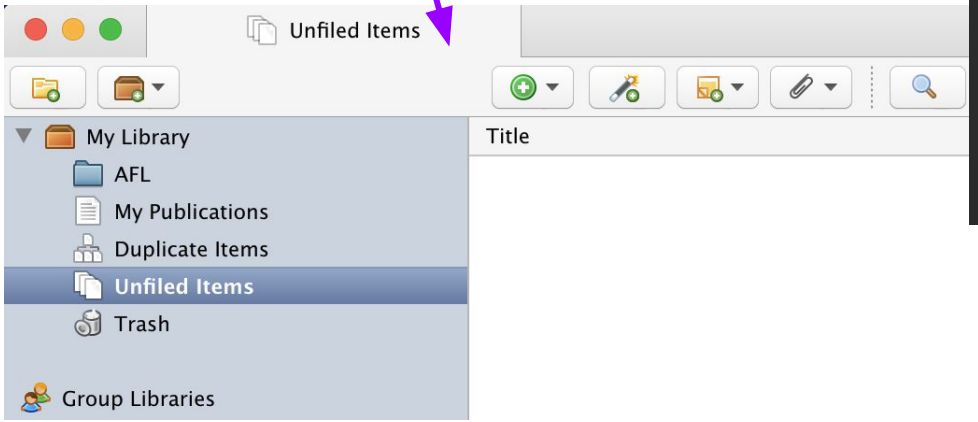


Strict assumptions about what defines the start and end of an abstract

```
def text_from_paper_dict(paper, abstract=True):

def text_from_google_scholar(title, abstract=True):

def full_text_from_pdf(pdf):

def abstract_from_pdf(pdf):

def abstract_from_pub_page(url):

def extract_abstract(text):
```

Paper Data Storage-Related Functionality

```
def save_all_data(papers, fname):

def add_title_to_zotero(titles):
```




```
{
    "papers": {
        "Attack-Resistant Federated Learning with Residual-based Reweighting": {
            "key": "DMCMSRM5",
            "version": 3,
            "library": {
                "type": "user",
                "id": 8601934,
                "name": "kvarma",
                "links": {
                    "alternate": {
                        "href": "https://www.zotero.org/kvarma",
                        "type": "text/html"
                    }
                }
            },
            "links": {
                "self": {
                    "href": "https://api.zotero.org/users/8601934/items/DMCMSRM5",
                    "type": "application/json"
                },
                "alternate": {
                    "href": "https://www.zotero.org/kvarma/items/DMCMSRM5",
                    "type": "text/html"
                }
            },
            "meta": {
                "creatorSummary": "Fu",
                "numChildren": 0
            },
            "data": {
                "key": "DMCMSRM5",
                "version": 3,
                "itemType": "journalArticle",
                "title": "Attack-Resistant Federated Learning with Residual-based Reweighting",
                "creators": [
                    {
                        "creatorType": "author",
                        "firstName": "Shuhao",
                        "lastName": "Fu"
                    }
                ],
```

Recommendation algorithm

- Content-based filtering:
  - Paper vectorized with SciBERT, sentence by sentence
    - Average of per-sentence paper vectors represent the paper
    - BERT summarization performed on text to reduce memory cost
      - Alternatively: use abstract instead of text
  - User profile formation:
    - For each paper added, add a new vector to the profile
  - Recommendation:
    - For each paper in corpus, compute cosine similarity with each paper in profile
    - Aggregate cosine similarities, with optional weight
    - Recommend papers with top similarity

Data used in recommendation

- Paper identifier
  - Using title as identifier
- Vectorized text/abstract representation
- Corpus inclusion information
  - Using keywords here

User Interface

- Simple text-based interface for setting up projects and requesting recommendations
- Users can create projects associated with their Zotero collections and get recommendations for those papers
- Use Zotero to modify papers used for recommendations
- Saves user settings and projects across uses

```
1. Create new project
2. Get recommendations
3. Exit
Select which action you would like to perform: 1
Choose a project name: Context-Aware Systems
```

User Interface

```
1. Create new project
2. Get recommendations
3. Exit
Select which action you would like to perform: 2

1. Context-Aware Systems
2. Recommender Systems
3. Ontologies
Select project id: 1

Recommended:
Effectful Program Distancing
From Enhanced Coinduction towards Enhanced Induction
Oblivious Algebraic Data Types

Successfully added Effectful Program Distancing to Zotero
Successfully added From Enhanced Coinduction towards Enhanced Induction to Zotero
Successfully added Oblivious Algebraic Data Types to Zotero
```

Documentation

RAGS: Research Assistant for Graduate Students

Instructions

Installing necessary packages

- follow setup instructions for pyzotero
- follow setup instuctions for scholarly

Usage

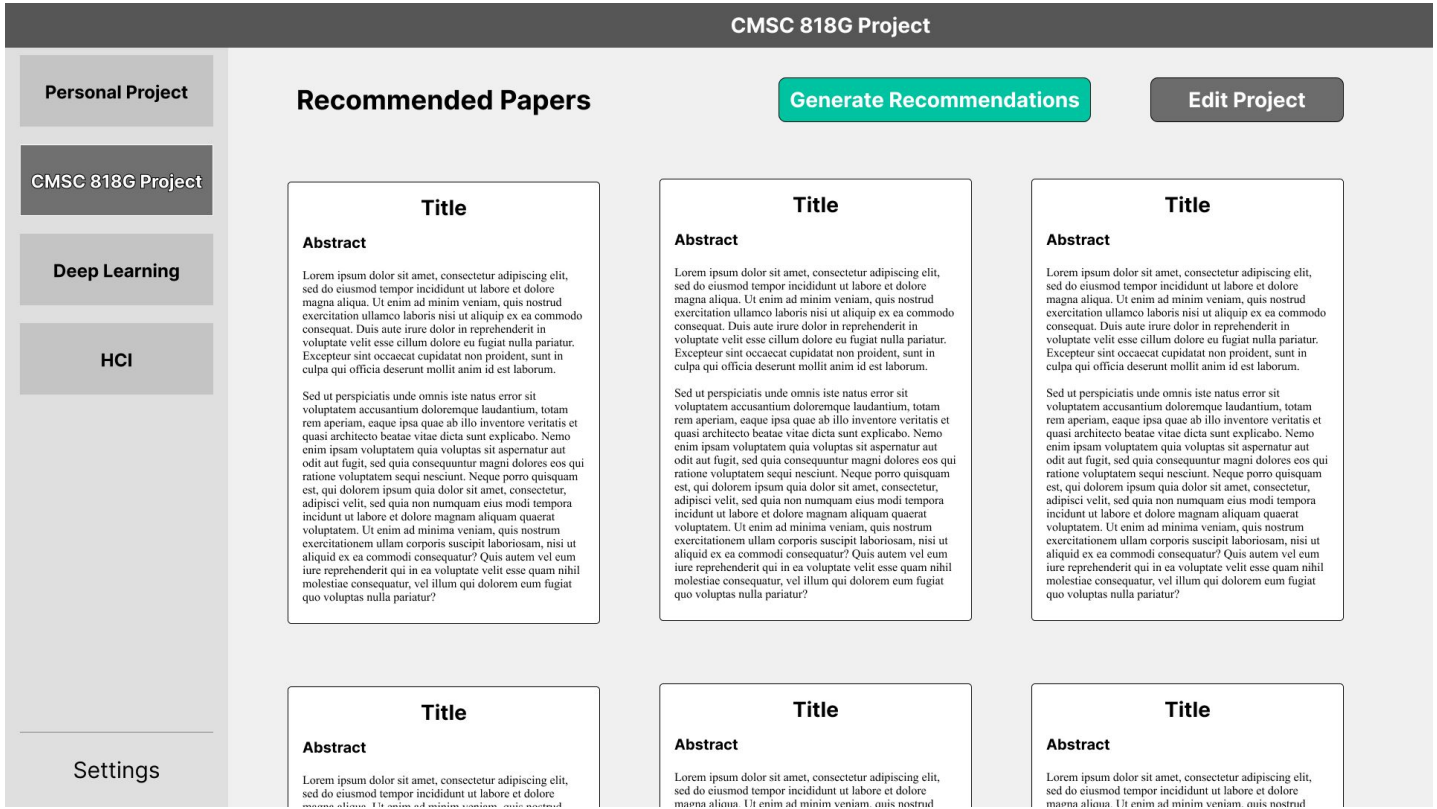
To use the project, run the `run_project` script:

```
1  python run_project.py
```

This will take you through setting up your user profile and first project. You will need to get and API key and the name of your personal zotero library using the instructions for pyzotero. These configurations are stored in `config.ini` and can be edited at any time.

After configuration, the script will provide you with the options of creating a new project, getting recommendations for an existing project, or exiting the application. Any papers searched during the process of producing recommendations will be added to a growing corpus, which is used to provide recommendations more quickly in the future. This corpus is saved in `corpus.pt`.

Graphical User Interface Design



Demo

- Offline store of paper used as corpus as Scholarly is sometimes detected as bot and prevented from Google Scholar access

Limitations and Future Work

- Better paper-related APIs
  - Current manual abstract extraction is unreliable
  - Scholarly (Google Scholar API) is new and unreliable
- More paper-related APIs
  - Storage (eg. Mendeley, Paperpile)
  - Search/access - what if the paper is not available through Google Scholar?
- More context in the recommendation system
  - Activities
- Incorporate other assistance relevant to grad students
  - General planning/scheduling
  - Conference-related planning
  - Note taking/organization