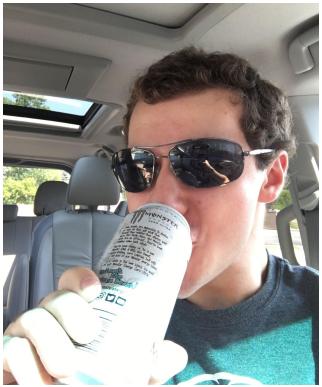


Professor Marvin - Update

AI study tool chatbot.



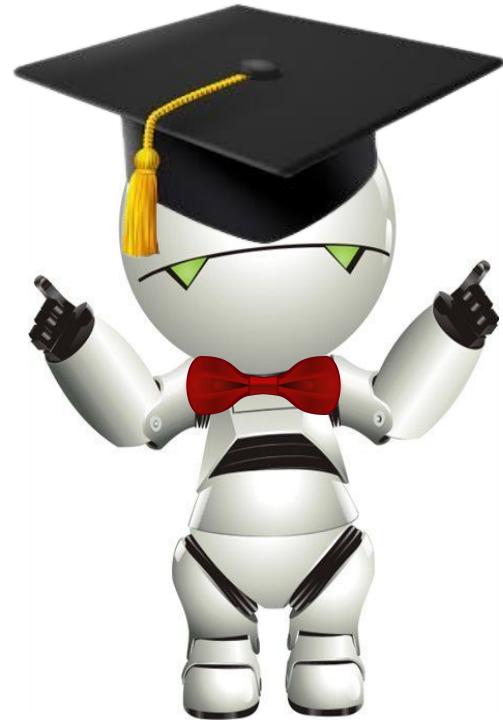


Meet the Martians

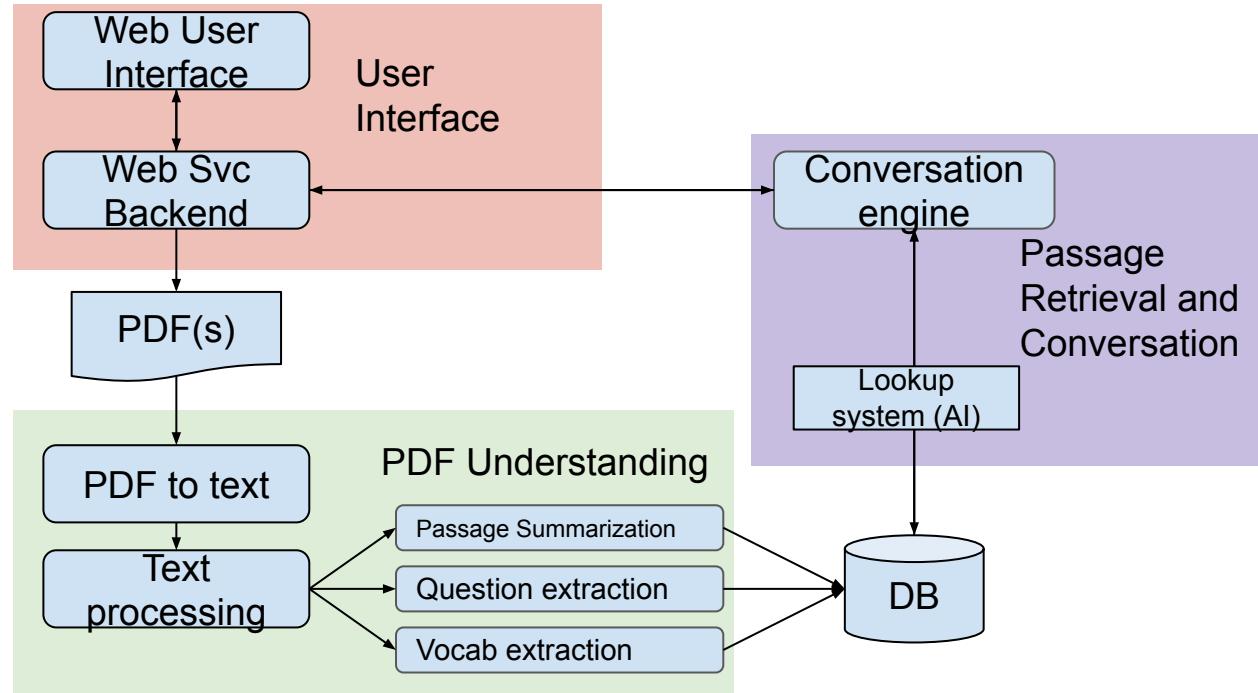


Usecase

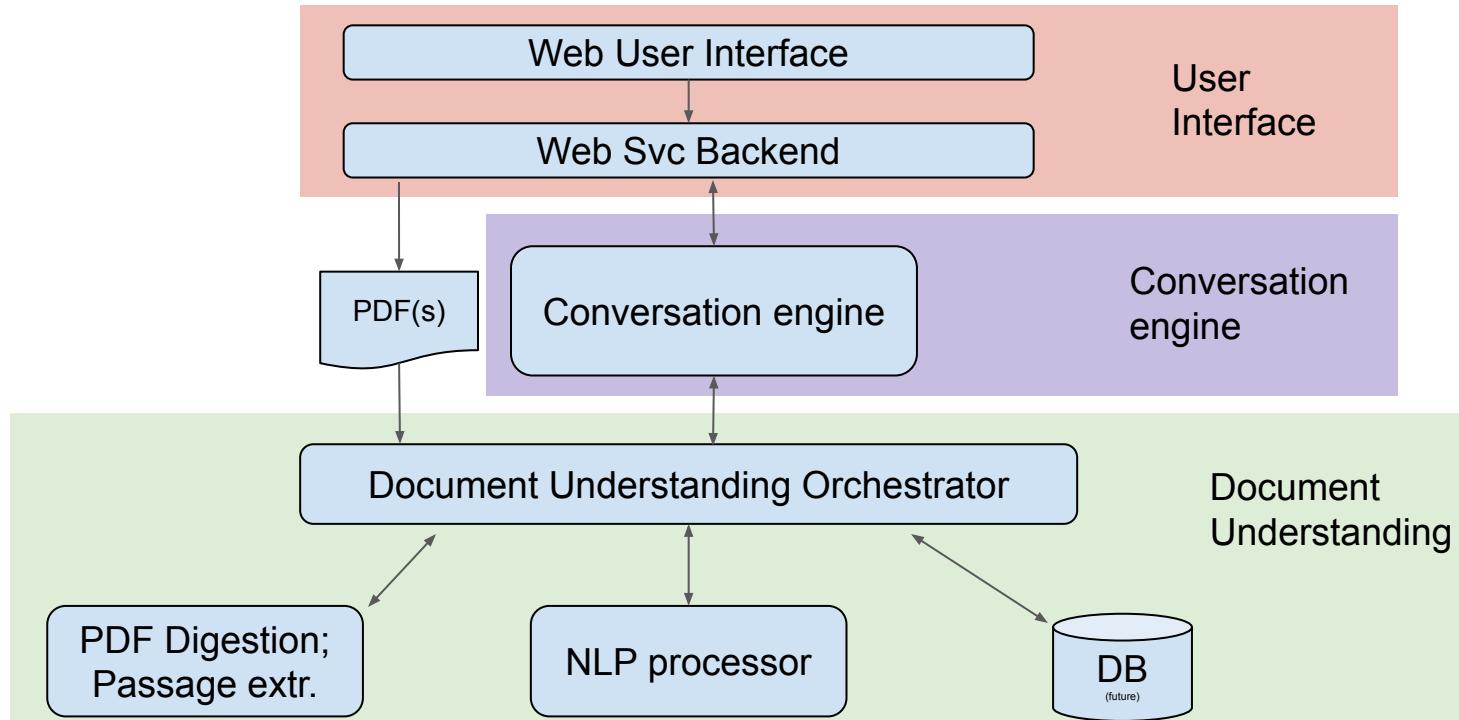
- Students spend too much time just *finding* the information they need
- Upload all your relevant texts to Prof. Marvin
- Automatically summarize the most important parts
- Get asked questions to test your understanding



Architecture changes: Before



Architecture changes: After

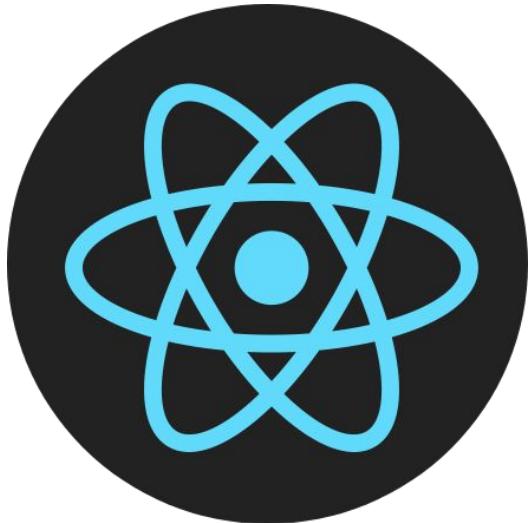


Major design decisions:

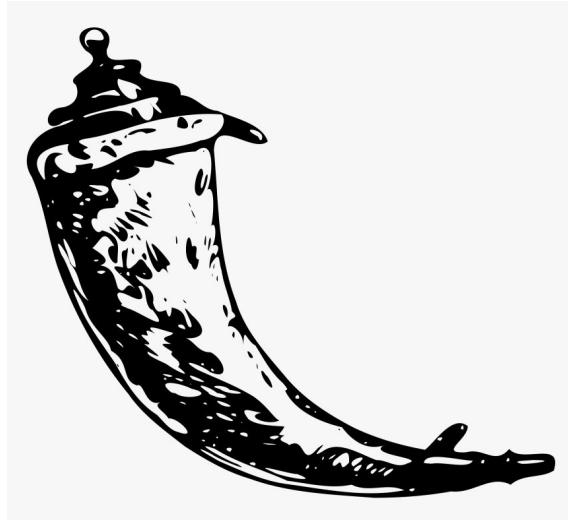
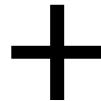
- (almost) everything's a flask server
 - Slow stuff doesn't slow down the rest of stuff
 - IP connections permit smooth interactions
- Orchestrator smooths inter-module communication and simplifies adding new backend features
- Scope-down:
 - Focus on question generation and summarization, and don't use database

Front End / UI

Tech Stack



Frontend



Backend

Progress

Professor Marvin Study! Chat (REMOVE) How To Use About

Upload

Upload pdfs in this page

Drag and drop file here, or click to select file

Upload

ERROR: File is not a PDF

Copyright © Professor Marvin 2022

Professor Marvin Study! Chat (REMOVE) How To Use About

Upload

Upload pdfs in this page

100%

Drag and drop file here, or click to select file

Upload

'Paper4-T5.pdf' was uploaded successfully

List of Files

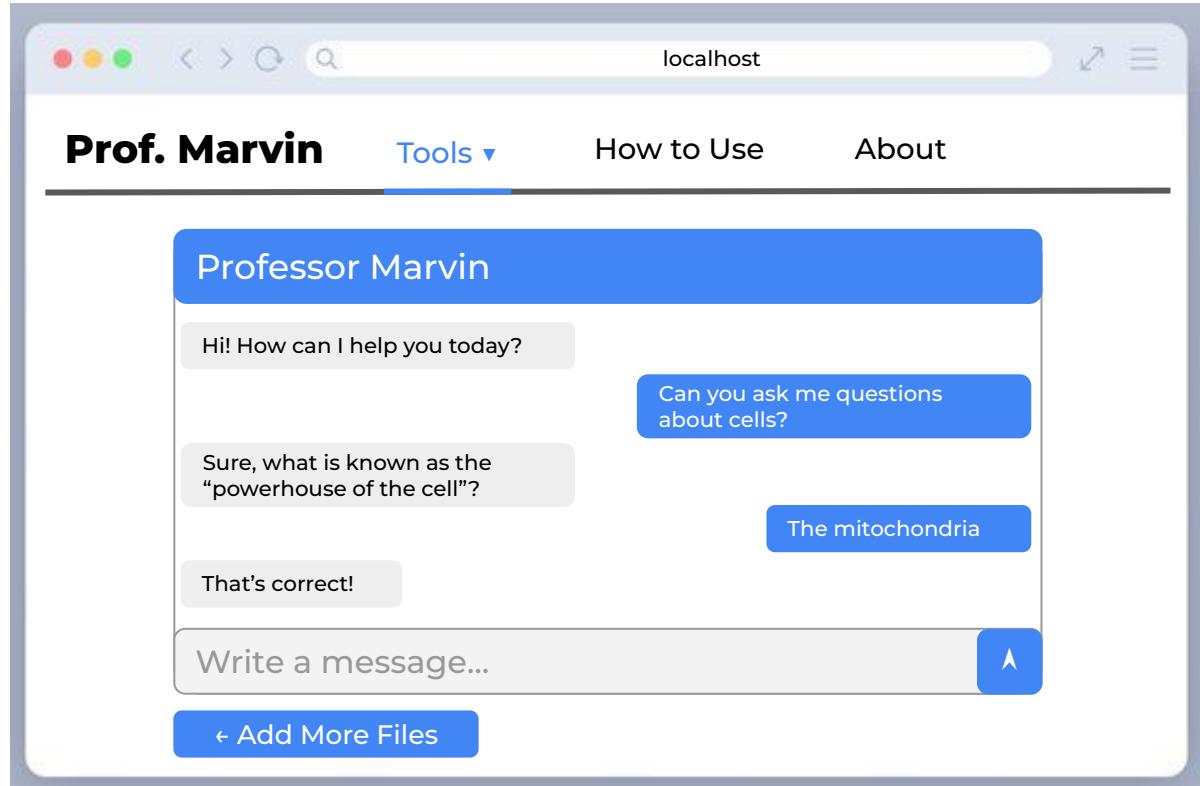
WN21_Final.pdf

worksheet-11.pdf

Paper4-T5.pdf

Copyright © Professor Marvin 2022

Concept



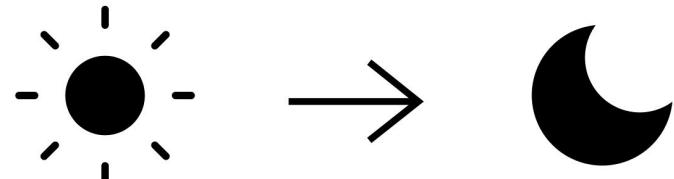
Rest API

- “/” → Home page
- “/upload_page” → Upload page
- “/chat” → Chat page
- “/instructions” → How To Use guide
- “/about” → About page (us and Marvin)
- “/reset” → Reset Marvin, remove files
- “/upload” → Upload pdfs to folder
- “/files” → Get a list of uploaded files
- “/to_user” → Send message to user
- “/to_marvin” → Send message to Marvin
- “/messages” → Get all messages
- “/clear” → Add clear chat functionality



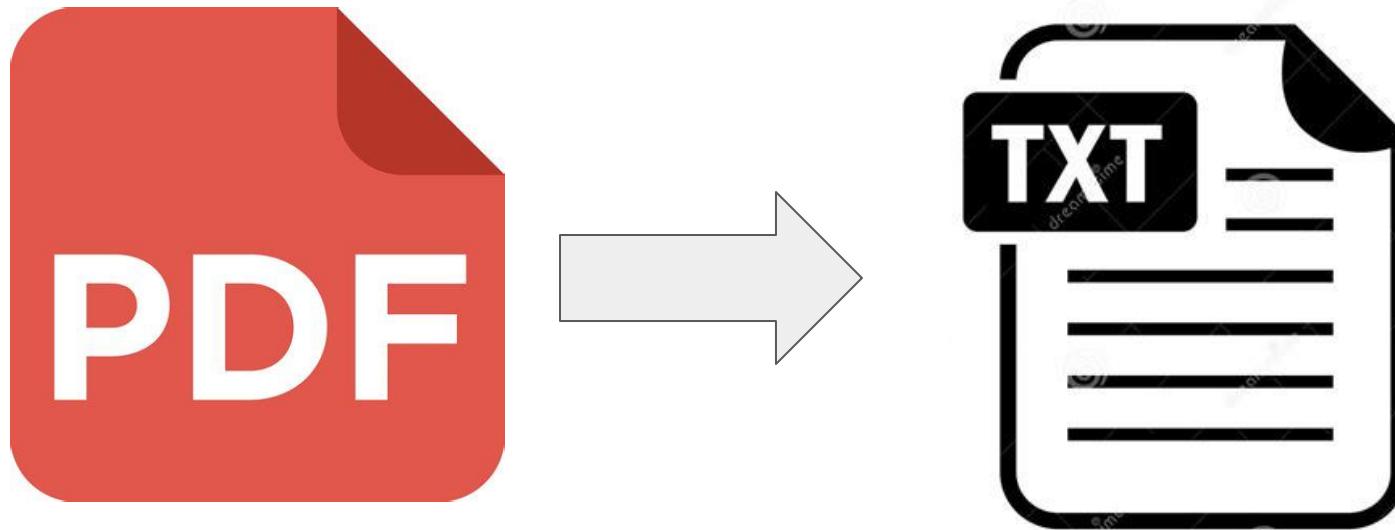
Future Work

- Finish RestAPI endpoints
 - Chat with Professor Marvin
 - Add content to information pages
 - Home, How To Use, About Us
 - CSS Styling
-
- Testing → User Feedback
 - Add User Authentication?
 - UI Improvements
 - Speech to Text, Dark Mode???



PDF processing

PDF Processing: Function



PDF Processing: Requirements

Functional

1. Extract text accurately from PDF documents
2. Use text and formatting clues to segment text into passages

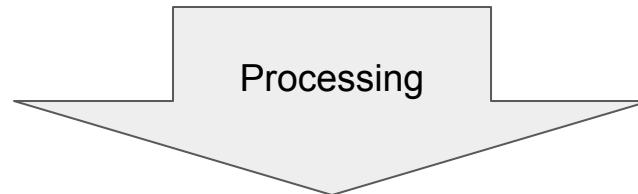
Non-Functional

1. Module acts quickly enough to allow smooth pdf ingestion
2. Extracted text has high fidelity to original text
3. Passage segmentations are logical and/or informative

How does it actually work?

Use PyMuPdf

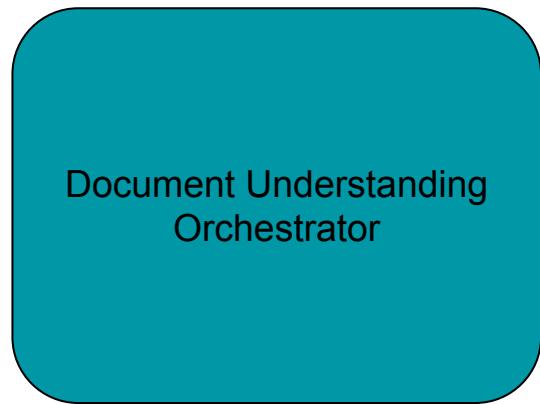
- Use to get granular view of text within document
- Group text by formatting styles



Grouped text chunks, “sub sections”



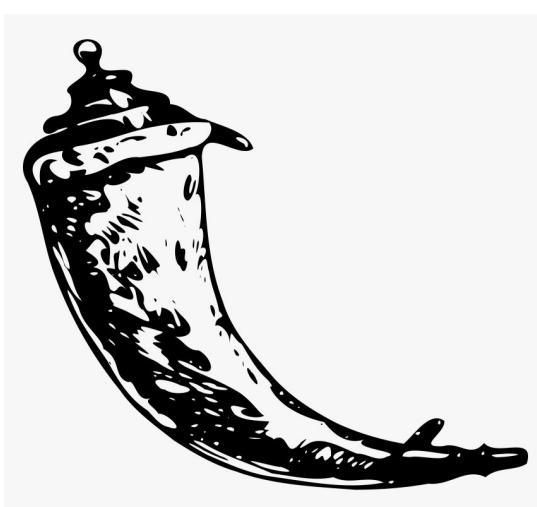
PDF Processing: API



HTTP: what does this
PDF say?



Response: I've translated
it. Here's the grouped text



Current state & future work

Current state:

- Performant, accurate lightweight

Future work

- Improve text filtering
- Improve clustering.

```
[Software Engineering at Google,
'31 Jan 2017',
'Fergus Henderson',
'+fergus@google.com',
'> (work or <',
'+fergus.henderson@gmail.com',
'> (personal))',
'Abstract',
'Me catalog and describe Google's key software engineering practices,
'Biography',
'Fergus Henderson has been a software engineer at Google for over 10 years. He started programming as a kid in 1979, and went on to academic research in programming language design and implementation. With his PhD supervisor, he co-founded a research group at the University of Melbourne that developed the programming language Mercury. He has been a program committee member for eight international conferences, and has released over 500,000 lines of open-source code. He was a former moderator of the Usenet newsgroup comp.std.c++ and was an officially accredited "Technical Expert" to the ISO C and C++ committees.'
```

"He has over 15 years of commercial software industry experience. At Google, he worked on the server-side software behind Google Translate, a build tool in use across Google, and worked on the server-side software behind speech recognition and voice actions (before Siri) and speech synthesis. He currently manages Google's text-to-speech engineering team, but still writes and reviews plenty of code. Software that he has written is installed on over a billion devices, and gets used over a billion times per day."

```
3.8s
test_docs/Rhet_Ops.pdf
Python

Output exceeds the size limit. Open the full output data in a text editor
['WHEN THE WORLD BECAME AWARE OF RHET OPS',
 'The world significantly changed while we were working with authors on the book you now hold. When we first conceived the project, the idea that online discourse might be a kind of weapon wielded by state or non-state actors in conflict was still relatively obscure. As we circulated our call for #RhetOps proposals in the summer of 2016, the US presidential election campaign was nearing its nadir. Both political parties had named presumptive nominees and were engaged in a highly contentious media campaign. There were already allegations in the mainstream press that Russian agents had sought to influence the outcome of the election by hacking, leaking confidential information, and influencing social media (Bessi and Ferrara) with what we label #RhetOps tactics, or rhetorical operations that make use of the latest digital and social networking technology to append, enhance, and amplify state and non-state conflicts. While our project felt important to us when we conceived it several years ago, by mid-summer 2016 the project began to feel urgent as we collected an increasing number of stories under #RhetOps, a hashtag to mark convergences between digital rhetorical theory and military operations, that pointed to broader militarization of social media and the concomitant questions concerning the vulnerability of democracy. As we prepare this book's introduction, Facebook CEO Mark Zuckerberg has just completed his second day of congressional testimony, facing questions from elected officials regarding the potential misuse of their platform's user data by a British company, Cambridge Analytica. Less than two years after we launched the project that would become this book, the idea that words and images—memes, fabricated news stories, lifestyle quizzes, and digital games—can be weapons is now part of our daily public discourse. In the next seventeen chapters, contributors explore how this convergence of digital rhetoric and state/military operations has consequences for our field, pedagogy, and collective future.',
 'VIII Preface',
 'THE ENDLESS APPENDIX COLLECTED HERE: AUTHOR'S PERSONAL LIBRARY OF RHETOPS. DON'T
WHEN THE WORLD BECAME AWARE OF RHET OPS
```

Software Engineering at Google

31 Jan 2017

Fergus Henderson

+fergus@google.com (work)
+fergus.henderson@gmail.com (personal)

Abstract

We catalog and describe Google's key software engineering practices.

Biography

Fergus Henderson has been a software engineer at Google for over 10 years. He started programming as a kid in 1979, and went on to academic research in programming language design and implementation. With his PhD supervisor, he co-founded a research group at the University of Melbourne that developed the programming language Mercury. He has been a program committee member for eight international conferences, and has released over 500,000 lines of open-source code. He was a former moderator of the Usenet newsgroup comp.std.c++ and was an officially accredited "Technical Expert" to the ISO C and C++ committees. He has over 15 years of commercial software industry experience. At Google, he was one of the original engineers on Google Translate, a build tool in use across Google, and worked on the server-side software behind speech recognition and voice actions (before Siri) and speech synthesis. He currently manages Google's text-to-speech engineering team, but still writes and reviews plenty of code. Software that he has written is installed on over a billion devices, and gets used over a billion times per day.

WHEN THE WORLD BECAME AWARE OF RHET OPS

The world significantly changed while we were working with authors on the book you now hold. When we first conceived the project, the idea that online discourse might be a kind of weapon wielded by state or non-state actors in conflict was still relatively obscure. As we circulated our call for #RhetOps proposals in the summer of 2016, the US presidential election campaign was nearing its nadir. Both political parties had named presumptive nominees and were engaged in a highly contentious media campaign. There were already allegations in the mainstream press that Russian agents had sought to influence the outcome of the election by hacking, leaking confidential information, and influencing social media (Bessi and Ferrara) with what we label #RhetOps tactics, or rhetorical operations that make use of the latest digital and social networking technology to append, enhance, and amplify state and non-state conflicts. While our project felt important to us when we conceived it several years ago, by mid-summer 2016 the project began to feel urgent as we collected an increasing number of stories under #RhetOps, a hashtag to mark convergences between digital rhetorical theory and military operations, that pointed to broader militarization of social media and the concomitant questions concerning the vulnerability of democracy.

As we prepare this book's introduction, Facebook CEO Mark Zuckerberg has just completed his second day of congressional testimony, facing questions from elected officials regarding the potential misuse of their platform's user data by a British company, Cambridge Analytica. Less than two years after we launched the project that would become this book, the idea that words and images—memes, fabricated news stories, lifestyle quizzes, and digital games—are now part of our daily public discourse. In the next

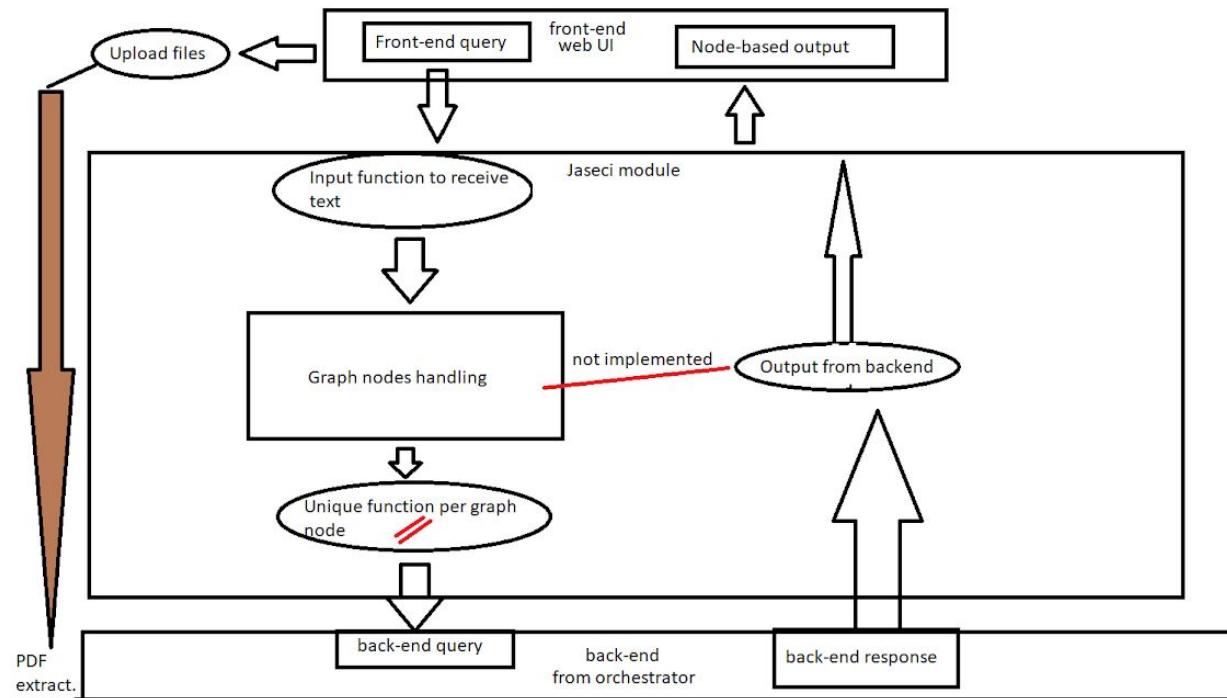
Chatbot

Our Role in Prof. Marvin

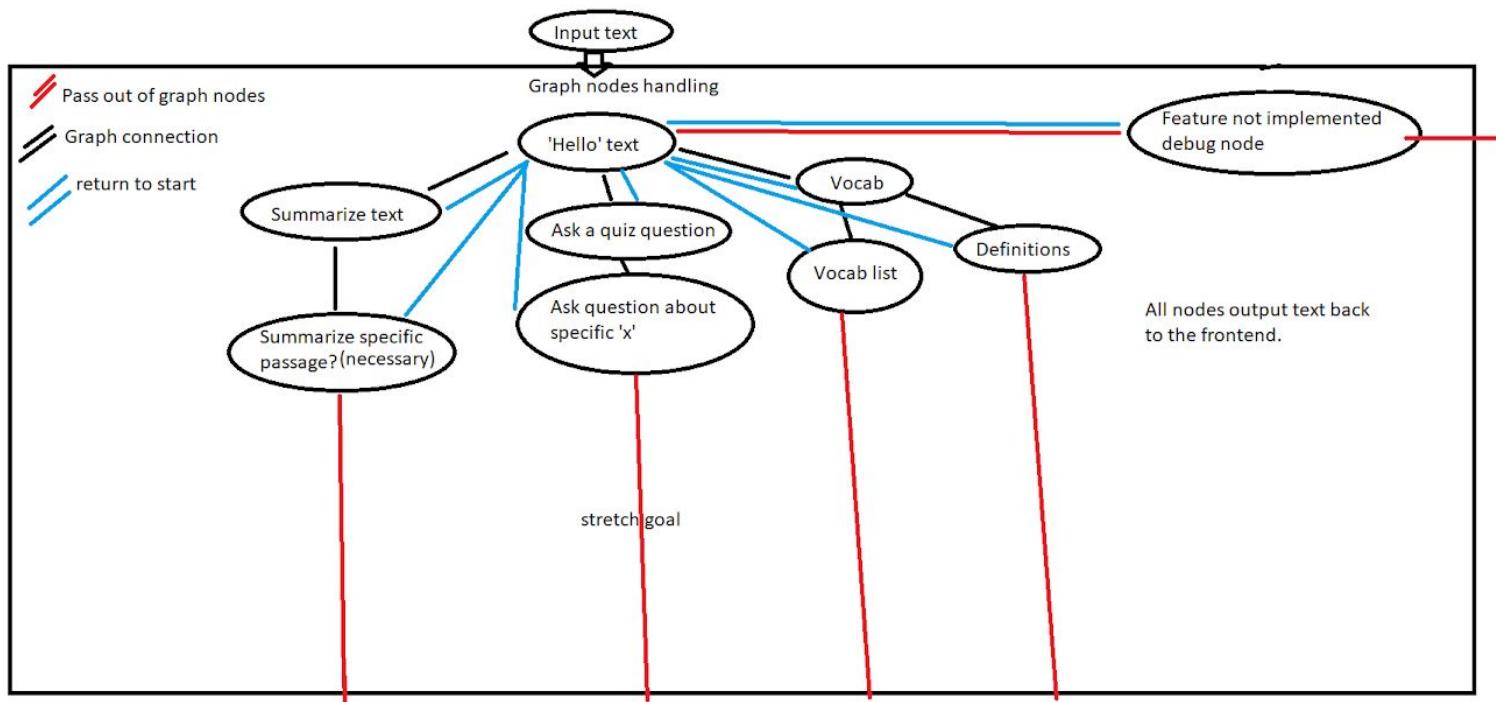
- Handles the conversational aspect of the program
 - Controls what the chatbot will be saying
 - Deduce what the user wants based on their text input
- Communicates with the front and back end
 - Receives text string from user interface
 - Sends tokens to the orchestrator



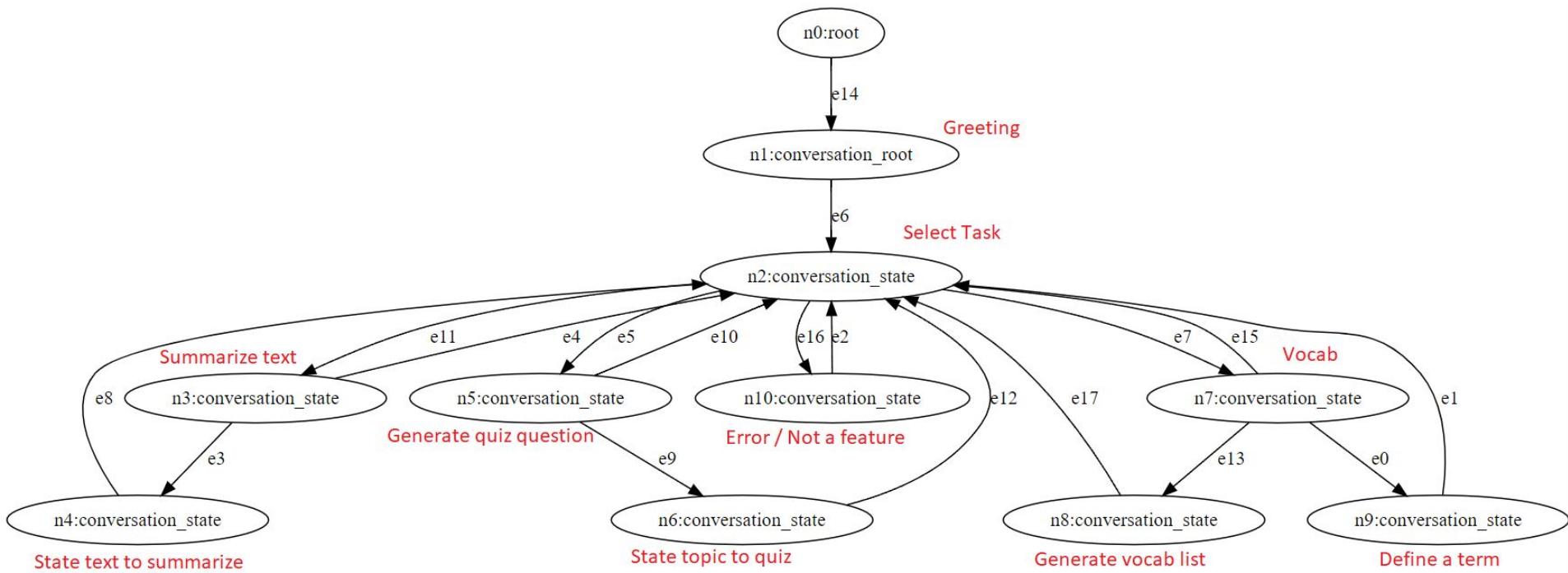
Jaseci: How it works



Jaseci: Graph Nodes



Jaseci: DOT Graph



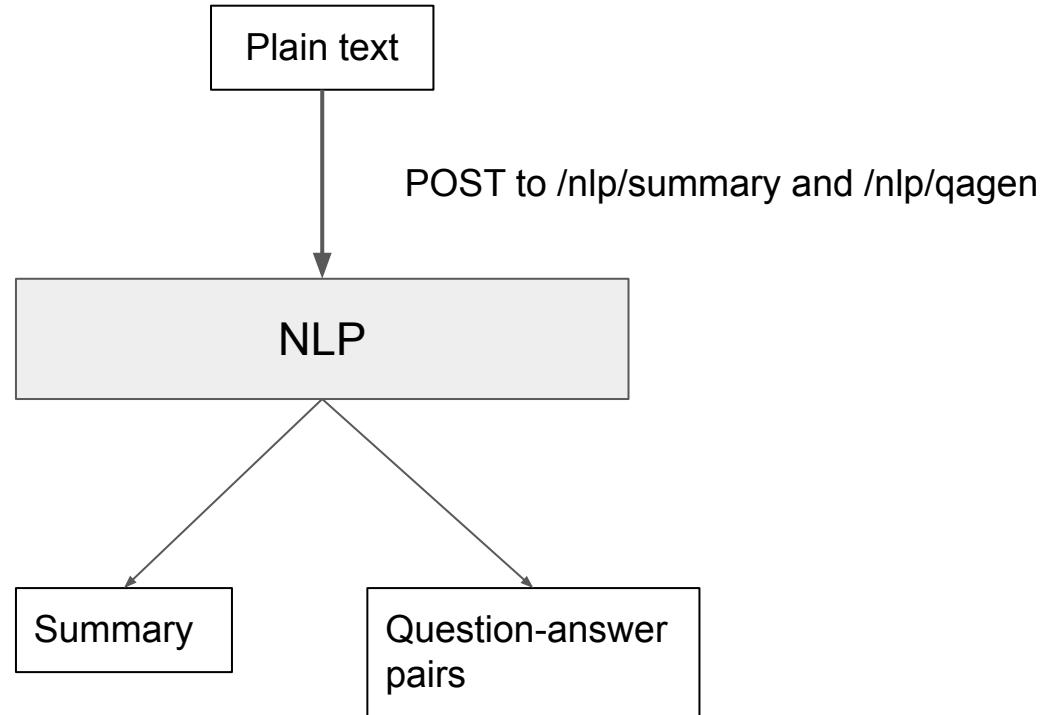
Python Shell

- Use python to communicate with user interface and orchestrator
 - Uses python to read input from the user interface, then sends that information to Jaseci
 - Uses Jaseci to compute which node in our graph to traverse
 - Sends the output back to the python
 - Python will send the output to either the orchestrator or back to the user interface



NLP API

NLP API: Functionality



NLP API: Implementation

- Summarization
 - Extractive using facebook/bart-large-cnn
 - Effective but slow
- QA-generation
 - T5 end-to end answer extraction and question generation using valhalla/t5-base-qa-qg-hl
 - Effective but occasionally breaks
 - Other approaches
 - SpaCy NER to extract answers and fine-tuned T5 model for question generation
 - Extracted answers were low quality and context was difficult to identify for QG

NLP API: Example

Input text:

Jason Mars (Born May 27, 1983) is an American computer scientist, author, and entrepreneur. He is best known for his research into computer architecture and artificial intelligence, particularly in the design and deployment of conversational AI. The best-selling author of *Breaking Bots: Inventing a New Voice in the AI Revolution*, he has been involved in multiple AI initiatives and startups over the course of his career, including ZeroShotBot, Jaseci, Clinc, Myca, and ImpactfulAI.

Mars holds a PhD in Computer Science from the University of Virginia(UVA), and is currently employed as an Associate Professor of Computer Science and Engineering at the University of Michigan(U-M).^{[4][5]} He is also acting co-director of U-M's Clarity Lab alongside his wife, Professor Lingjia Tang.^[6] There, Mars helps direct advanced research within artificial intelligence, large-scale computing, and coding. Among the lab's most notable projects is the open source Sirius, later rebranded as Lucida.^[7]

A virtual assistant capable of understanding both visual and auditory queries, Lucida was intended by Mars and his colleagues as a sandbox that would help programmers explore the complexities of speech recognition. Mars also hoped that it would act as a foundation for the development of hardware better-suited for conversational AI.^[8] The project was supported by Google, the Defense Advanced Research Projects Agency (DARPA), and the National Science Foundation.^[9]

Jason Mars was one of ten individuals celebrated at the 28th Annual Caribbean American Heritage (CARAH) Awards. Mars received the Vanguard Award from the Institute of Caribbean Studies for his technological impact and "contributions to America and the world."^[1] Other winners include Pfizer Principal Scientist for Viral Vaccines Vidia Roopchand and Grammy-winning songwriter Gordon Chambers. Past honorees of the CARAH Awards include former United States Attorney General Eric Holder, former United States Ambassador to the United Nations Andrew Young and Olympian Usain Bolt.^[10]

NLP API: Example

Q/A Pairs:

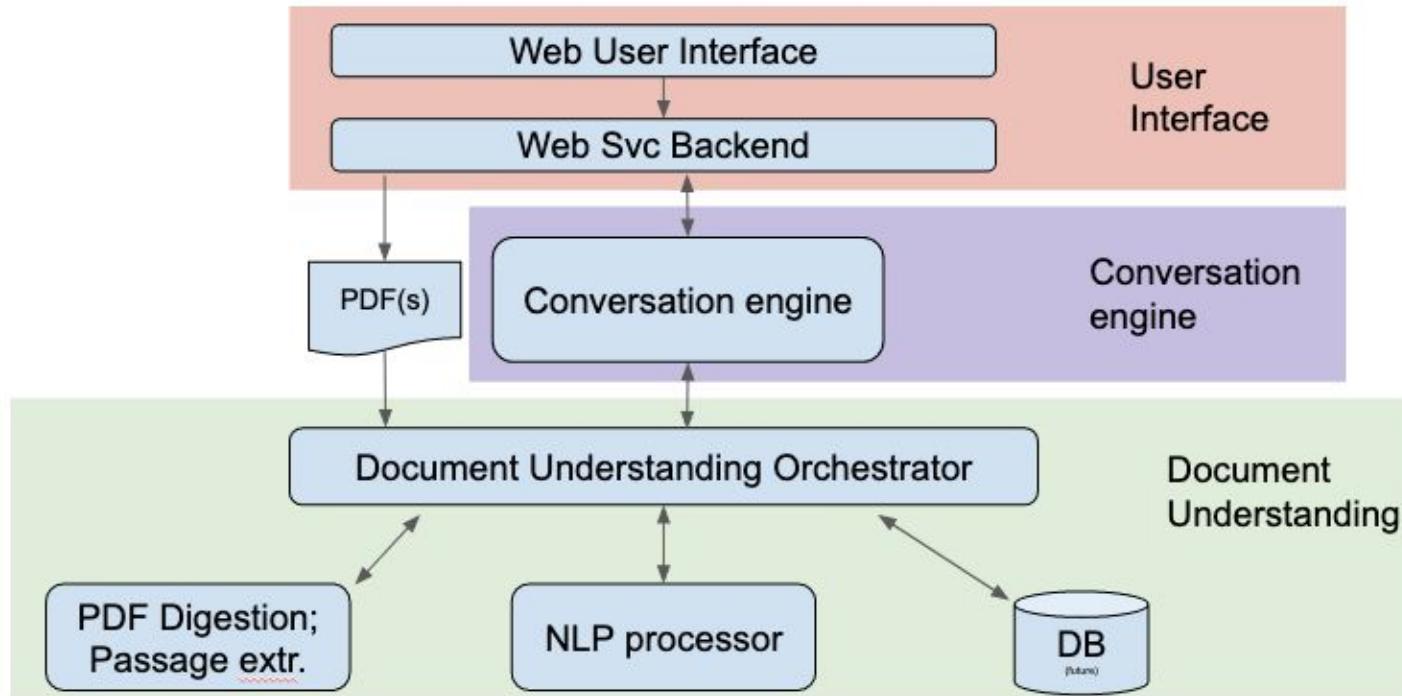
```
"qas": [
  {
    "answer": "Jason Mars",
    "question": "Who is an American computer scientist, author, and entrepreneur?"
  },
  {
    "answer": "computer architecture and artificial intelligence",
    "question": "What is Jason Mars best known for?"
  },
  {
    "answer": "28th Annual Caribbean American Heritage (CARAH) Awards",
    "question": "At what event was Jason Mars honored?"
  },
  {
    "answer": "Vanguard Award",
    "question": "What award did Jason Mars receive from the Institute of Caribbean Studies?"
  },
  {
    "answer": "Gordon Chambers",
    "question": "Which Grammy-winning songwriter was a winner of the CARAH Awards?"
  },
  {
    "answer": "ZeroShotBot, Jaseci, Clinc, Myca, and ImpactfulAI",
    "question": "What AI initiatives has Jason Mars been involved in?"
  },
  {
    "answer": "University of Virginia",
    "question": "Where did Jason Mars receive his PhD in Computer Science?"
  },
  {
    "answer": "Professor Lingjia Tang",
    "question": "Who is Jason Mars' wife?"
  }
]
```

NLP API: Moving Forward

- Test and Improve question-answer pair generation
 - Find error producing input
 - Prevent/remove irrelevant output
- Implement vocabulary/key term extraction
 - Generate vocab lists
 - Define a given key term

Orchestrator

Orchestrator Overview



Orchestrator: next steps

- Store texts and derived data (summaries, QA pairs, etc.) in DB to pull from instead of reprocessing for repeated queries

Overall: Current status

- Q/A no longer stretch goal
- Vocab extraction becomes one
- Web interface & conv engine
1-2 weeks out
- All other modules complete

Key Milestones

- Decide on specific language models 
- Individual components are working separately 
- Key features functioning
 - Text summarization 
 - Question Extraction 
 - Vocab extraction 
- Command line interface
- Front end user interface
- Further development of functionality

