ITS-VIP Chatbot

Spring 2020

About Us

Kunal Arora

- C/C++, Python, Java, Scala, C#, HTML, PHP, SQL, Cassandra

Responsibilities

- ML Model Prototyping
- Back-end Integration
- Testing

Samuel Toh

- Java, Python, C/C++, HTML, CSS, jQuery, SQL

Responsibilities

- User-Interface Development
- Testing

Andrew Chen

- Java, Python, C/C++

Responsibilities

- Data Preprocessing
- Testing

Target Problem

- TAs are overwhelmed by high volume of questions before test dates and assignment deadlines
- Some questions may be asked repeatedly even with Piazza forum
- Difficulty in providing customized feedback and resources based on the student's strengths and weaknesses

Solution

Develop an embedded chatbot that will help TAs to handle high volumes of questions before deadlines or exams and provide a more personalized experience

“

Overview

- Our team worked to build a chatbot for the ITS, aiming to provide more data-driven and personalized learning experience and guidance.

- Trained on Piazza and textbook data, the chatbot preprocesses user inputs, builds document vectors with word vectors and term frequency–inverse document frequency (TF-IDF), and finds the closest match using cosine similarity.

Data Cleaning

Remove valueless information

- HTML tags
- Special characters
- Accent characters
- Whitespaces
- Newline chars
- Stopwords

Normalize representation

- Expand contractions
- Convert all text to lowercase
- Lemmatization

Other

- Remove duplicate entries
- Remove unanswered questions and follow-ups
- Remove pure graphics/URL entries

Data Cleaning

Original text

```
'<p>Can someone confirm that 1.1.d is written correctly (with no imaginary part)?</p>\n<p></p>\n<p>$$(e^{\\pi })^{8}
$$</p>'
```

Stripping HTML tags

```
'Can someone confirm that 1.1.d is written correctly (with no imaginary part)?\n\n$$(e^{\\pi })^{8}$$'
```

Remove accented characters

```
'Can someone confirm that 1.1.d is written correctly (with no imaginary part)?\n\n$$(e^{\\pi })^{8}$$'
```

Expand contractions

```
'Can someone confirm that 1.1.d is written correctly (with no imaginary part)?\n\n$$(e^{\\pi })^{8}$$'
```

Remove special characters

```
'Can someone confirm that 11d is written correctly with no imaginary part\n\ne^\\pi ^8'
```

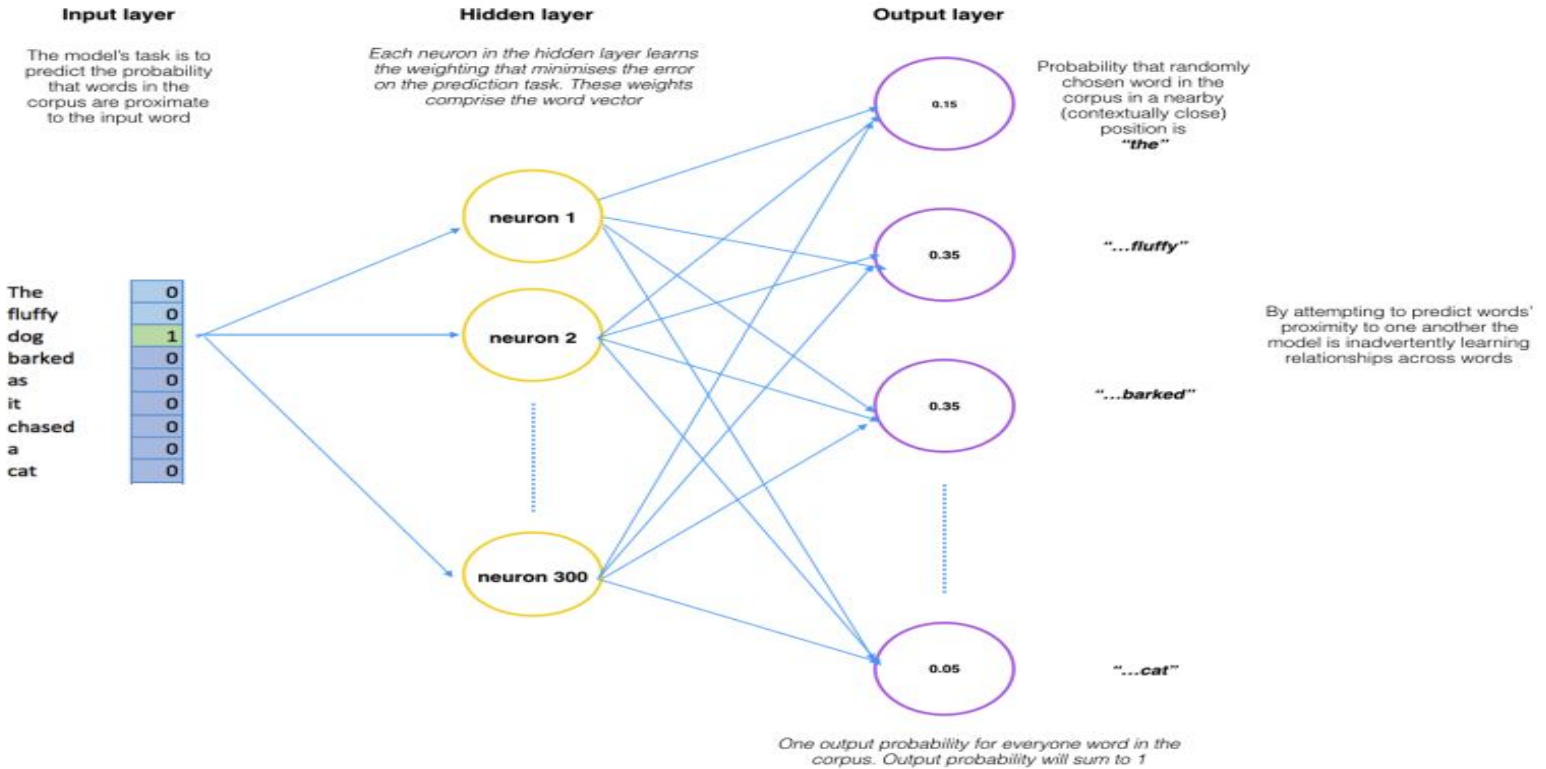
Lemmatize the text

```
'Can someone confirm that 11d be write correctly with no imaginary part \n\n e^\\pi ^8'
```

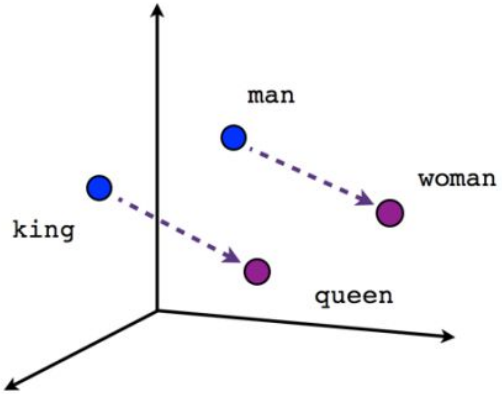
Remove stop words

```
'someone confirm 11d write correctly no imaginary part e^\\pi ^8'
```

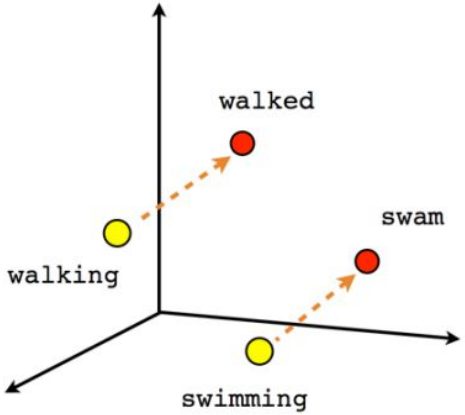
Word embeddings



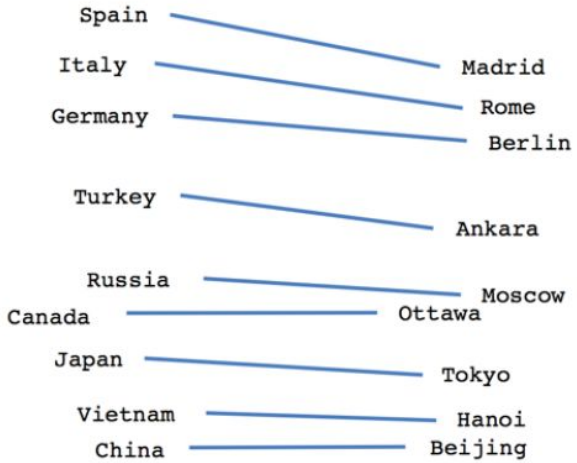
Word embeddings



Male-Female



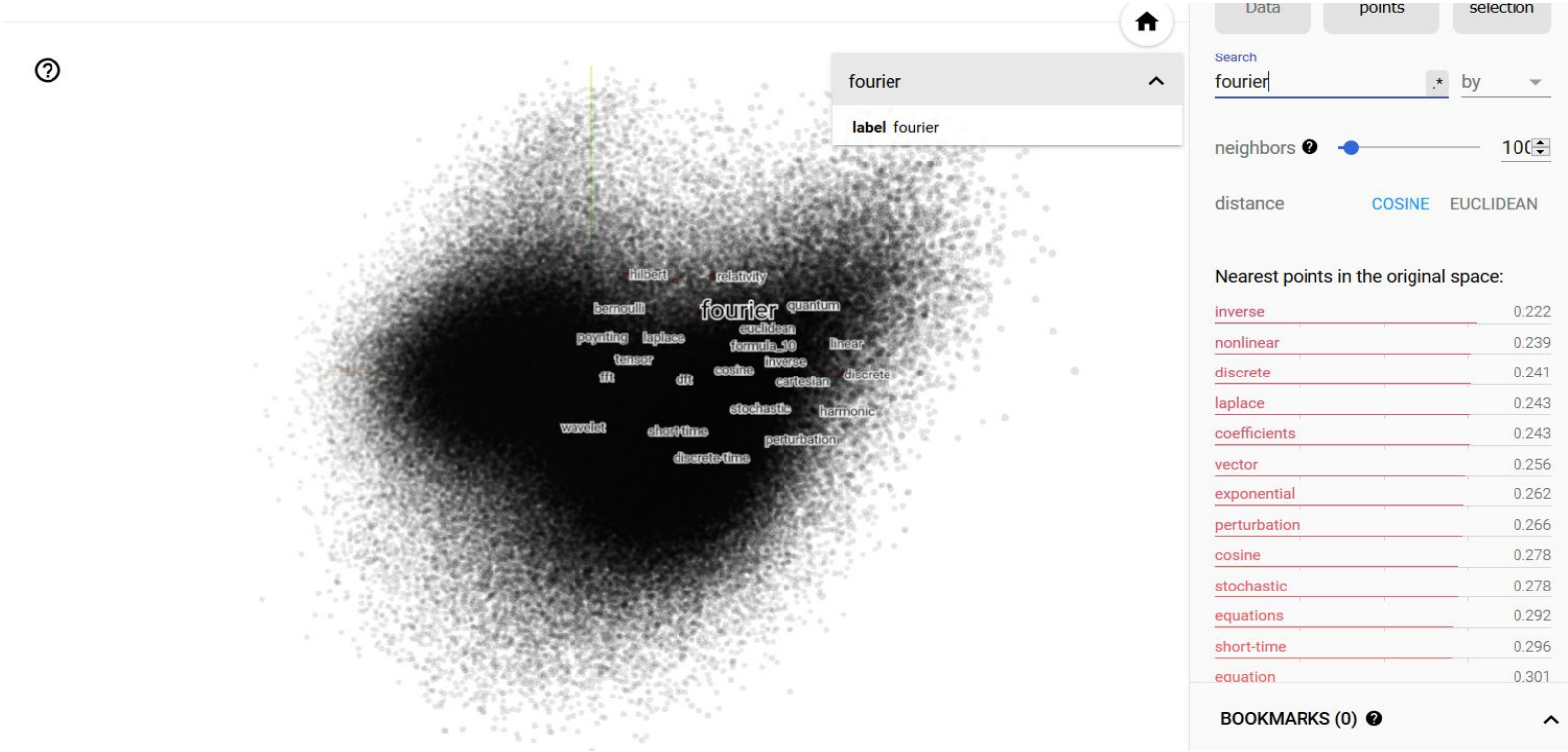
Verb tense



Country-Capital

Word embeddings

Each word vector is of **dimension** : **1 X 100**
Total words= 400,000



Tf-idf : Term frequency-inverse document frequency.

- It is a numerical statistic that tells how important a word is to a document.

$$\text{Tf-idf}(w, d) = \text{Tf}(w,d) * \text{idf}(w)$$

Tf(w,d)=No. of times the word w appears in document d
idf(w)=log(total documents/total documents containing word w)

Document Vectors

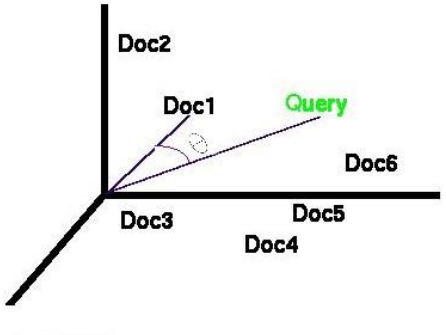
Document = group of words.

Document dataset = Piazza questions, notes, responses, paragraphs from the dsp first book.

Dataset contains ~ 4500 documents.

d = "The fluffy dog barked as it chased a cat"

Doc_vector(d) = **Mean**(word_vector('The')*tf-idf('The', d),
word_vector('fluffy')*tf-idf('fluffy', d),
...
..
word_vector('cat')*tf-idf('cat, d)
)



Dimension of Doc_vector (d) = **1 X 100**

Doc_vector(d) = [0.23, 0.87, 0.39, ... 0.42]

User-Interface development

1) Built using the Flask framework
    a) A micro web framework written in Python
2) Separated into two components:
    a) Front-end implementation (home.html)
    b) Back-end implementation (app.py)

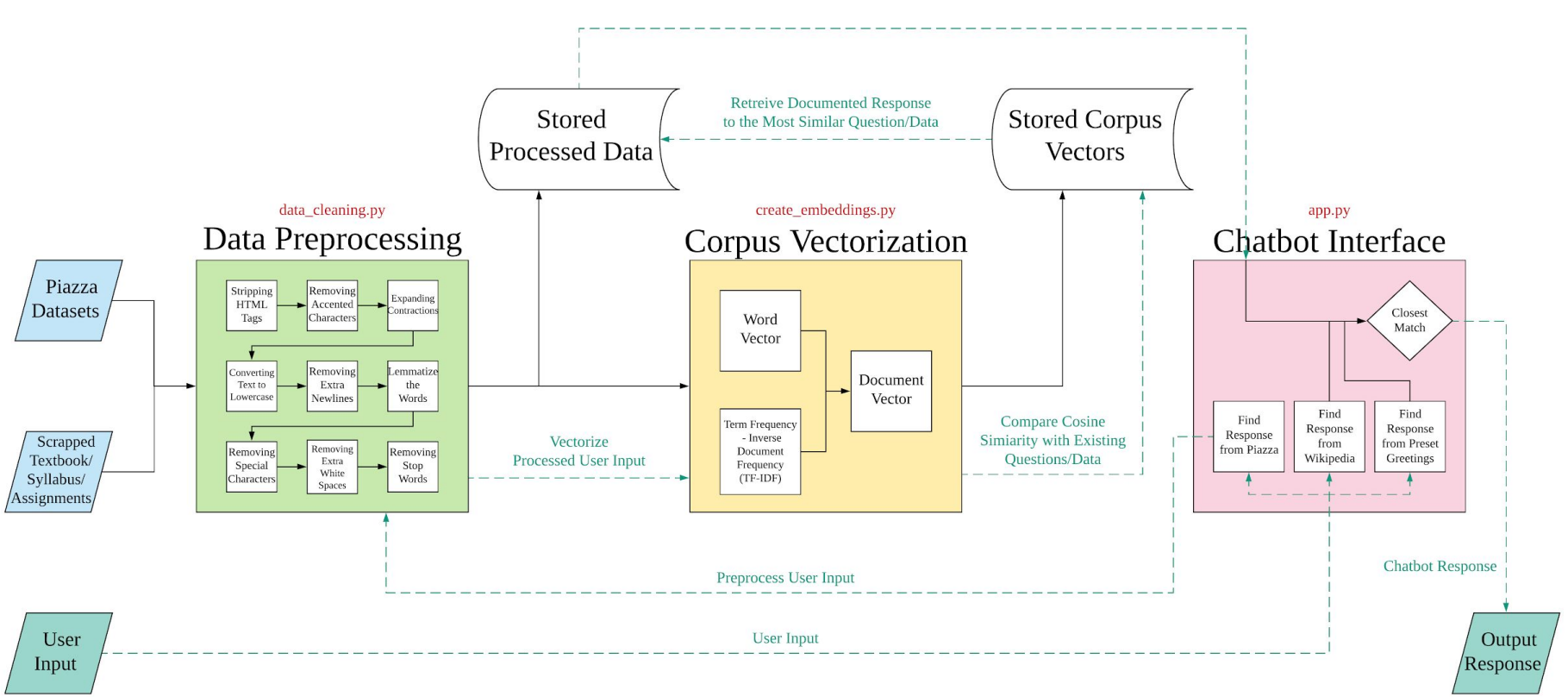
Front-end implementation (home.html)

1) HTML, CSS, and jQuery
2) jQuery's keypress() method
    a) Event.which = <number>: indicates the specific key or button that was pressed
3) jQuery's $.get(url, [data], [callback]) method
    a) Sends asynchronous http GET request to the server and retrieves the data
    b) Url: request url from which you want to retrieve the data
    c) Data: data to be sent to the server with the request as a query string
    d) Callback: function to be executed when request succeeds
4) jQuery's append method
    a) Appends the response from the Python back-end code to the UI
    b) Appends the user inputs to the UI

Back-end implementation (app.py)

1) def generate_greeting_response (user_input)
   a) Hard-coded greeting inputs to take care of the situation when users greet the bot
2) def generate_wikipedia_search(user_input)
   a) A Python library to access and parse data from wikipedia
3) def generate_piazza_response(user_input)
   a) This function is where the data preprocessing and corpus vectorization take place
4) def get_bot_response()
   a) The user inputs are passed into this function from the front-end code using jQuery's $.get(url, [data],[callback]) method and Python's request.args.get('msg')
   b) The inputs are then passed into different response generators depending on the type of inputs

Overview


Stored Processed Data
Retreive Documented Response to the Most Similar Question/Data
Stored Corpus Vectors
data_cleaning.py
Data Preprocessing
create_embeddings.py
Corpus Vectorization
app.py
Chatbot Interface
Piazza Datasets
Scrapped Textbook/ Syllabus/ Assignments
Stripping HTML Tags
Removing Accented Characters
Expanding Contractions
Converting Text to Lowercase
Removing Extra Newlines
Lemmatize the Words
Removing Special Characters
Removing Extra White Spaces
Removing Stop Words
Word Vector
Term Frequency - Inverse Document Frequency (TF-IDF)
Document Vector
Closest Match
Find Response from Piazza
Find Response from Wikipedia
Find Response from Preset Greetings
Vectorize Processed User Input
Compare Cosine Simiarity with Existing Questions/Data
Preprocess User Input
Chatbot Response
User Input
User Input
Output Response

Demo

Now putting all the work together...

Example 1

Question

“How many crib sheets can we bring to the final exam?”

Ideal Response

Something along the line of:

“You can bring n crib sheets.”

Question Supported by Piazza Data

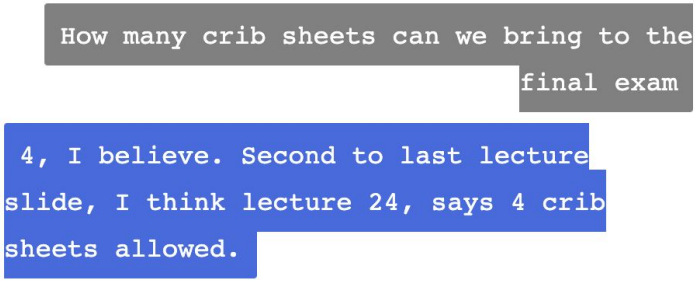
Example 1

Question

"How many crib sheets can we bring to the final exam?"

Actual Response

"4, I believe. Second to last lecture slide, I think lecture 24, says 4 crib sheets allowed."



Question Supported by Piazza Data

Example 2

Question

“What is Digital Signal Processing?”

Ideal Response

A brief explanation like:

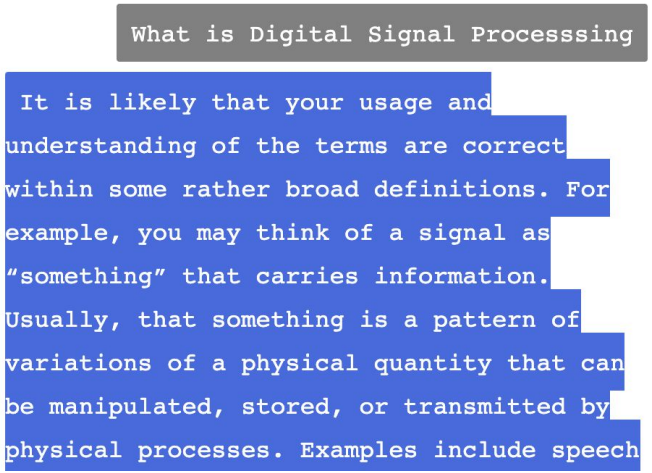
“Digital Signal Processing refers to... ”

Question Supported by Textbook

Example 2

Question

“What is Digital Signal Processing?”



Actual Response

“... you may think of a signal as “something” that carries information... Examples include speech signals, audio signals, video or image signals, biomedical signals, radar signals... ”

Question Supported by Textbook

Example 3

Question

“What is Cubic Spline Interpolation?”

Ideal Response

“Cubic Spline Interpolation is a method that...”

Question Mismatching with Other Data

Example 3

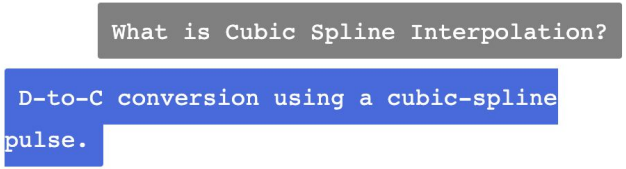
Question

“What is Cubic Spline Interpolation?”

Actual Response

“D-to-C conversion using a cubic-spline pulse.”

Model focuses more on keywords than on semantics



Question Mismatching with Other Data

Example 3

Model focuses more on keywords than on semantics

5 Most Similar Questions Found

1. D-to-C conversion using a cubic-spline pulse.
2. DEMO: Reconstruction via D-to-C
3. The problem of plotting a cosine signal from a set of discrete samples depends on the interpolation method used...
4. ... Constructing the curve between sample points in this way is called linear interpolation. The solid gray curve in the upper plot of Fig. shows the result of linear interpolation...
5. 4-4.7 Cubic Spline Interpolation
   A third pulse shape is shown in the lower left panel of Fig...

Question Mismatching with Other Data

Challenges

- Generalization of assignment/conceptual questions (requires more data)
- Difficulty in autogenerating testing data
- Difficulty in preserving the semantics of LaTeX text

The Roadmap

Fall 2020

- Integrate more data source
  - Textbook
  - Lecture slides
  - Syllabus
  - Wikipedia
- Data augmentation

Spring 2021

- Adopt and compare different models
  - Transfer Learning
  - Deep Learning
- Improve chatbot interface based on user feedbacks

Fall 2021

- Embedding the chatbot into Piazza and/or Canvas webpage
- Incorporate personalized data/cache
- Incorporate more input options

Thank You!