Team Members:

- Blake Oberlander bho180000@utdallas.edu
- Sharar Siddiqui srs200003@utdallas.edu

Chatbot Report

System Description

We created a chatbot with two related objectives:

1. Help the user select a SSD for their laptop. (**Primary Objective**)
2. Provide information about SSDs to the user. (Secondary Objective)

Our system uses a minimal amount of 3rd party code. We only rely on NLTK, Numpy, and a library to help provide the web server.

System Usage

Accessing the web server

We strongly suggest you access the chatbot through the web interface we made:

http://charlesbot.sytes.net/

>> If that doesn't work, try: http://155.138.233.145/ <<

The chatbot should be easily usable through this website. If you, the reviewer(s), have any problems using or accessing the chatbot (ex, because the website is down), please reach out to us by sending an email.

Local Configuration and Setup

This section is here just in case you want to see the code run on your own machine.

The chatbot can either be run as a web application, or it can be run in the terminal. In either case, you'll need your own Google Developer API key on hand (used for Google Search queries). And make sure your Python version is up to date.

1. Unzip the project zip file, and open the unzipped directory
2. Run **pip install -r requirements.txt**
    a. This will install the 3 libraries
3. Create a file called **API_KEY.txt** and paste your Google Developer API key inside
4. Run **python3 main.py**
5. You might see some errors from NLTK if you haven't already downloaded the required NLTK packages. If you see such errors, then open up a python terminal and type nltk.download(...). Then try running it again

Once you do that, you can try running the web interface by entering **python3 web-interface.py**

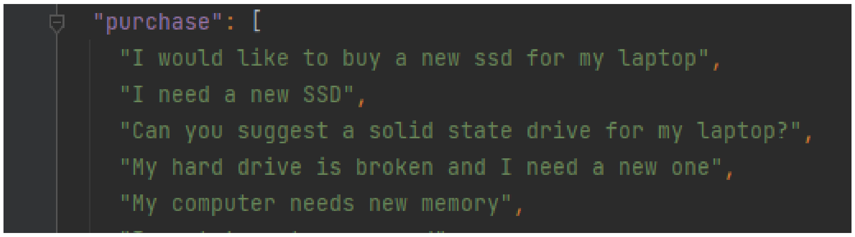
NLP Techniques Used

Below, we discuss various NLP techniques we implemented:

Intent Analysis using Bucket-of-n-gram Vectors, Cosine Similarity

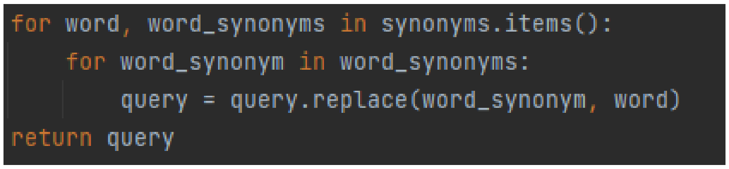
One of the first things the user does is indicate whether they want to buy an SSD or Research facts about SSDs. The system uses a statistical approach to determine which intent the user had. This is accomplished by the following training steps:

1. We created a file called **sample-queries.json** that contains training examples of queries for each intent:



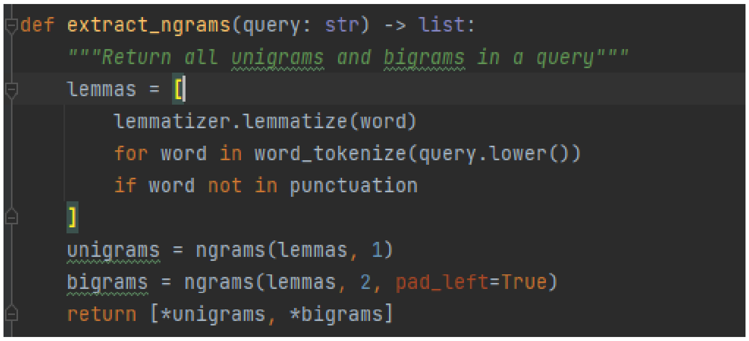
```
"purchase": [
  "I would like to buy a new ssd for my laptop",
  "I need a new SSD",
  "Can you suggest a solid state drive for my laptop?",
  "My hard drive is broken and I need a new one",
  "My computer needs new memory",
```

2. We perform pre-processing on queries by **replacing synonyms, tokenizing, lemmatizing, and finding (1,2)-grams**. This reduces the amount of training data needed to extract user intents. NLTK was used in this part.
    a. *Replacing synonyms*



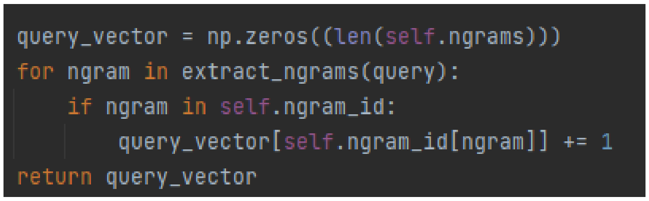
```
for word, word_synonyms in synonyms.items():
    for word_synonym in word_synonyms:
        query = query.replace(word_synonym, word)
return query
```

    b. *Tokenizing, lemmatizing, and finding (1,2)-grams*



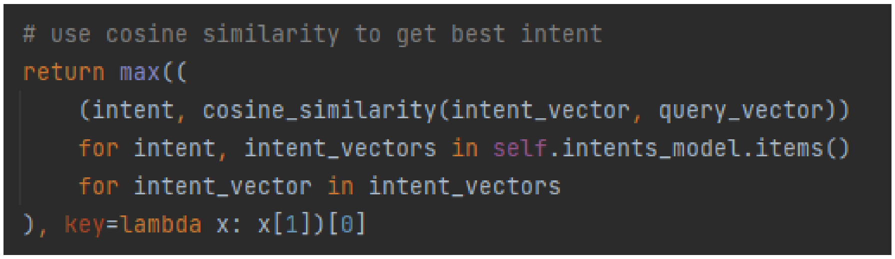
```
def extract_ngrams(query: str) -> list:
    """Return all unigrams and bigrams in a query"""
    lemmas = [
        lemmatizer.lemmatize(word)
        for word in word_tokenize(query.lower())
        if word not in punctuation
    ]
    unigrams = ngrams(lemmas, 1)
    bigrams = ngrams(lemmas, 2, pad_left=True)
    return [*unigrams, *bigrams]
```

3. We **vectorize** the queries using a **bucket-of-ngrams** approach



```
query_vector = np.zeros((len(self.ngrams)))
for ngram in extract_ngrams(query):
    if ngram in self.ngram_id:
        query_vector[self.ngram_id[ngram]] += 1
return query_vector
```

    a.

4. We **classify** intents using **cosine similarity** between query vectors
    a. 

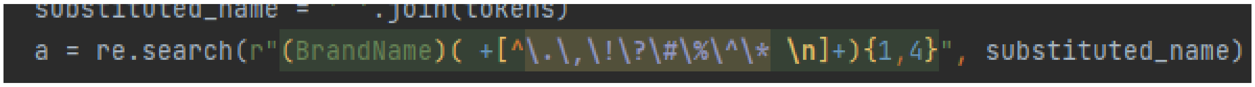
```
# use cosine similarity to get best intent
return max((
    (intent, cosine_similarity(intent_vector, query_vector))
    for intent, intent_vectors in self.intents_model.items()
    for intent_vector in intent_vectors
), key=lambda x: x[1])[0]
```

    b. *We recognize that better approaches for classification exist. This was implemented before the class lectures on machine learning*

Custom Named-Entity Recognition

We use **regular expressions** to parse laptop model names. This is a **named entity recognition** task.

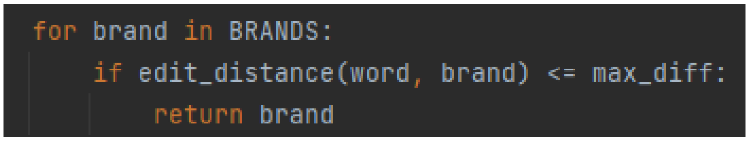
If the user types something like, “My laptop is an Acer Aspire R3 3200U with 8 GB RAM”, the regular expression is smart enough to parse out **“Acer Aspire R3 3200U”**. It has a list of known laptop brands, and it uses that to identify what model the user typed.



```
substituted_name = ' '.join(tokens)
a = re.search(r"(BrandName)( +[^\.\,\!\?\#\%\^\* \n]+){1,4}", substituted_name)
```

Levenshtein Distance

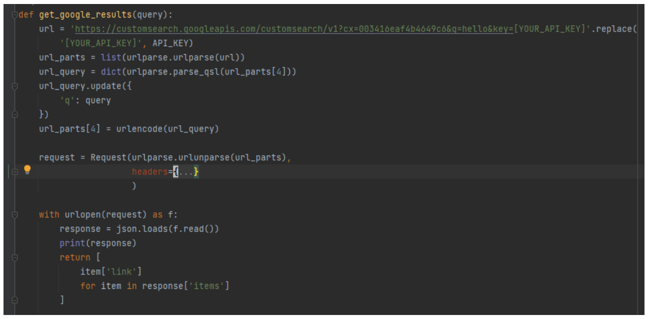
The user might make a typo when entering the laptop model name. Our program can automatically correct small typos using **Levenshtein Distance** (part of NLTK):



```
for brand in BRANDS:
    if edit_distance(word, brand) <= max_diff:
        return brand
```

Web Scraping

We scrape websites to find out specification details about the user’s laptop. First, we search the laptop model on google, and then we parse through each linked page to see which storage interface the user’s laptop has (NVME vs SATA)

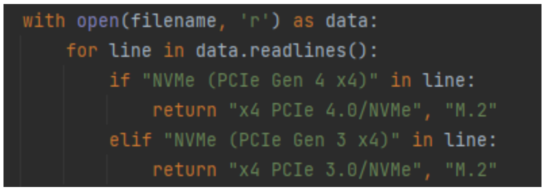


```
def get_google_results(query):
    url = 'https://customsearch.googleapis.com/customsearch/v1?cx=003416eaf4b4649c6&q=hello&key=[YOUR_API_KEY]'.replace(
        '[YOUR_API_KEY]', API_KEY)
    url_parts = list(urlparse.urlparse(url))
    url_query = dict(urlparse.parse_qsl(url_parts[4]))
    url_query.update({
        'q': query
    })
    url_parts[4] = urlencode(url_query)

    request = Request(urlparse.urlunparse(url_parts),
                      headers={...}
                      )

    with urlopen(request) as f:
        response = json.loads(f.read())
        print(response)
        return [
            item['link']
            for item in response['items']
        ]
```

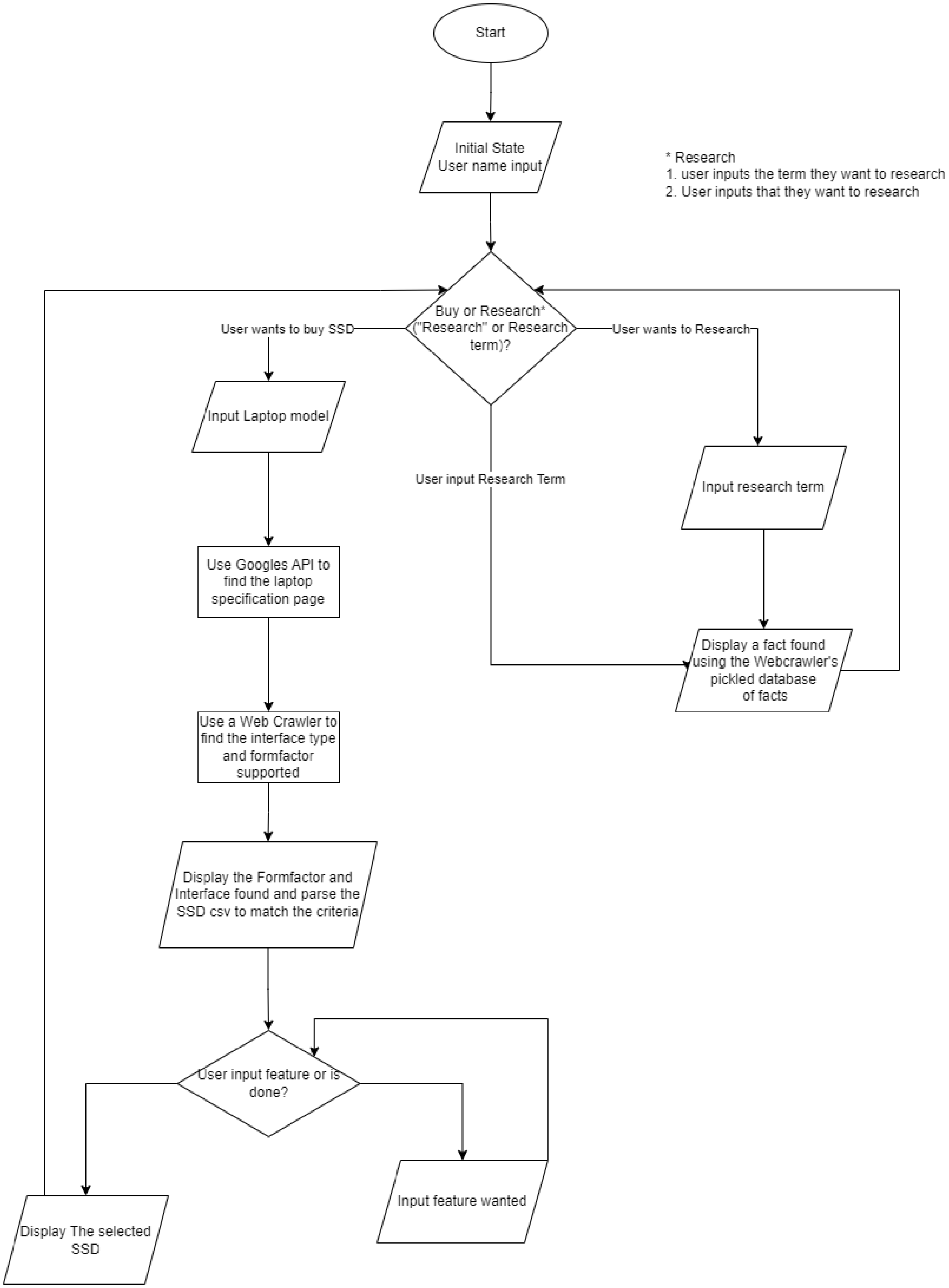
Searching Google



```
with open(filename, 'r') as data:
    for line in data.readlines():
        if "NVMe (PCIe Gen 4 x4)" in line:
            return "x4 PCIe 4.0/NVMe", "M.2"
        elif "NVMe (PCIe Gen 3 x4)" in line:
            return "x4 PCIe 3.0/NVMe", "M.2"
```

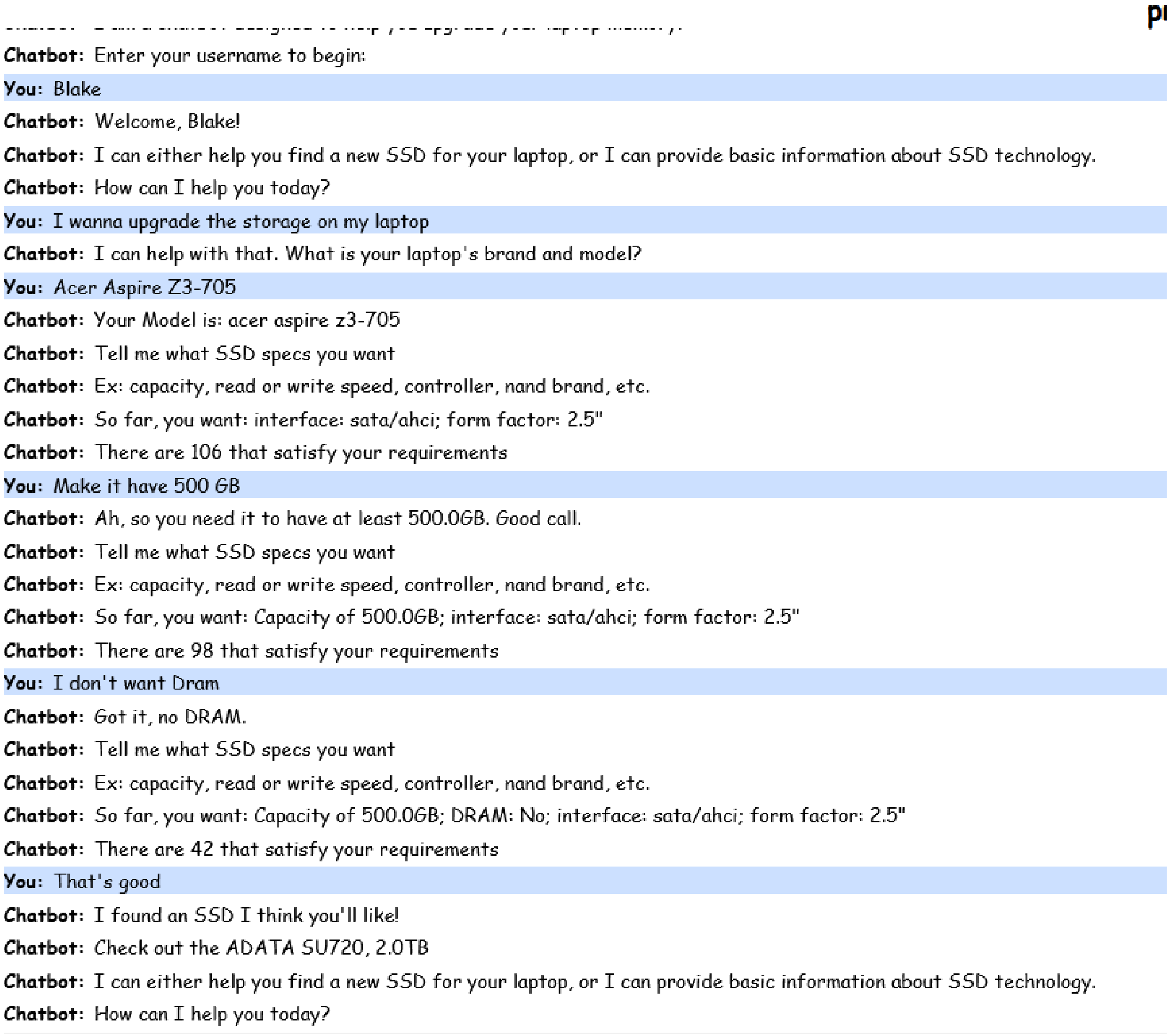
Downloading & Scraping to get SSD interface details

Diagram of Logic Tree


Start
Initial State
User name input
* Research
1. user inputs the term they want to research
2. User inputs that they want to research
Buy or Research*
("Research" or Research term)?
User wants to buy SSD
User wants to Research
User input Research Term
Input Laptop model
Input research term
Use Googles API to find the laptop specification page
Display a fact found using the Webcrawler's pickled database of facts
Use a Web Crawler to find the interface type and formfactor supported
Display the Formfactor and Interface found and parse the SSD csv to match the criteria
User input feature or is done?
Input feature wanted
Display The selected SSD

Sample Dialogs

SSD Selection Dialog 1



Chatbot: Enter your username to begin:

You: Blake

Chatbot: Welcome, Blake!

Chatbot: I can either help you find a new SSD for your laptop, or I can provide basic information about SSD technology.

Chatbot: How can I help you today?

You: I wanna upgrade the storage on my laptop

Chatbot: I can help with that. What is your laptop's brand and model?

You: Acer Aspire Z3-705

Chatbot: Your Model is: acer aspire z3-705

Chatbot: Tell me what SSD specs you want

Chatbot: Ex: capacity, read or write speed, controller, nand brand, etc.

Chatbot: So far, you want: interface: sata/ahci; form factor: 2.5"

Chatbot: There are 106 that satisfy your requirements

You: Make it have 500 GB

Chatbot: Ah, so you need it to have at least 500.0GB. Good call.

Chatbot: Tell me what SSD specs you want

Chatbot: Ex: capacity, read or write speed, controller, nand brand, etc.

Chatbot: So far, you want: Capacity of 500.0GB; interface: sata/ahci; form factor: 2.5"

Chatbot: There are 98 that satisfy your requirements

You: I don't want Dram

Chatbot: Got it, no DRAM.

Chatbot: Tell me what SSD specs you want

Chatbot: Ex: capacity, read or write speed, controller, nand brand, etc.

Chatbot: So far, you want: Capacity of 500.0GB; DRAM: No; interface: sata/ahci; form factor: 2.5"

Chatbot: There are 42 that satisfy your requirements

You: That's good

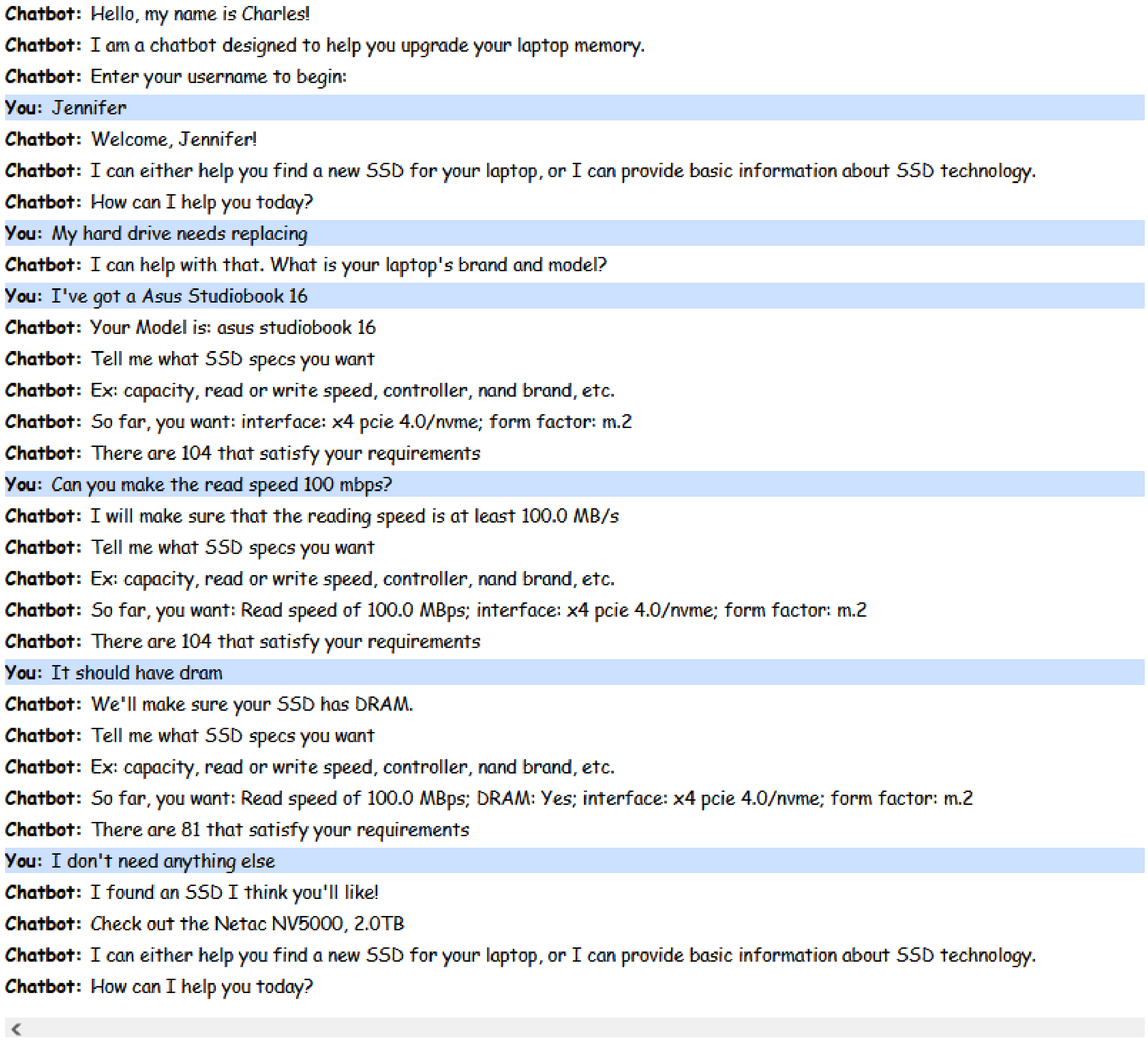
Chatbot: I found an SSD I think you'll like!

Chatbot: Check out the ADATA SU720, 2.0TB

Chatbot: I can either help you find a new SSD for your laptop, or I can provide basic information about SSD technology.

Chatbot: How can I help you today?

SSD Selection Dialog 2



Chatbot: Hello, my name is Charles!

Chatbot: I am a chatbot designed to help you upgrade your laptop memory.

Chatbot: Enter your username to begin:

You: Jennifer

Chatbot: Welcome, Jennifer!

Chatbot: I can either help you find a new SSD for your laptop, or I can provide basic information about SSD technology.

Chatbot: How can I help you today?

You: My hard drive needs replacing

Chatbot: I can help with that. What is your laptop's brand and model?

You: I've got a Asus Studiobook 16

Chatbot: Your Model is: asus studiobook 16

Chatbot: Tell me what SSD specs you want

Chatbot: Ex: capacity, read or write speed, controller, nand brand, etc.

Chatbot: So far, you want: interface: x4 pcie 4.0/nvme; form factor: m.2

Chatbot: There are 104 that satisfy your requirements

You: Can you make the read speed 100 mbps?

Chatbot: I will make sure that the reading speed is at least 100.0 MB/s

Chatbot: Tell me what SSD specs you want

Chatbot: Ex: capacity, read or write speed, controller, nand brand, etc.

Chatbot: So far, you want: Read speed of 100.0 MBps; interface: x4 pcie 4.0/nvme; form factor: m.2

Chatbot: There are 104 that satisfy your requirements

You: It should have dram

Chatbot: We'll make sure your SSD has DRAM.

Chatbot: Tell me what SSD specs you want

Chatbot: Ex: capacity, read or write speed, controller, nand brand, etc.

Chatbot: So far, you want: Read speed of 100.0 MBps; DRAM: Yes; interface: x4 pcie 4.0/nvme; form factor: m.2

Chatbot: There are 81 that satisfy your requirements

You: I don't need anything else

Chatbot: I found an SSD I think you'll like!

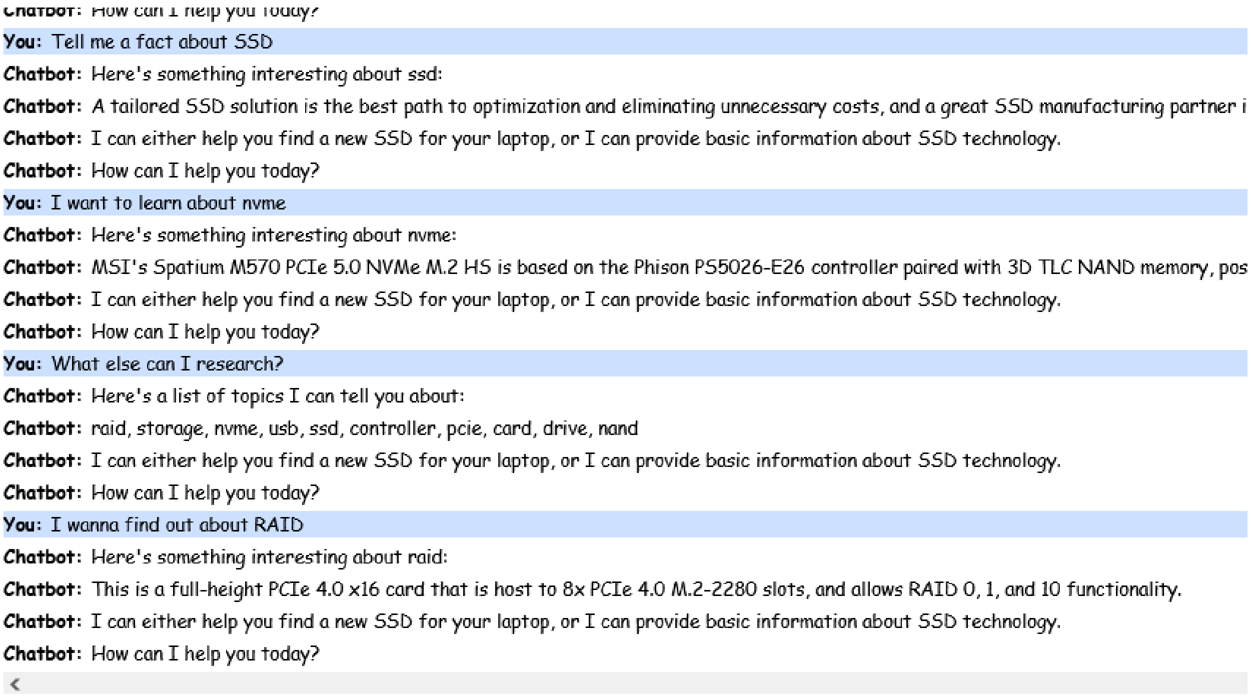
Chatbot: Check out the Netac NV5000, 2.0TB

Chatbot: I can either help you find a new SSD for your laptop, or I can provide basic information about SSD technology.

Chatbot: How can I help you today?

Random Fact Query

Admittedly, this part of the chatbot isn't as impressive. Note that we put most of our effort into the other parts.



```
Chatbot: How can I help you today?
You: Tell me a fact about SSD
Chatbot: Here's something interesting about ssd:
Chatbot: A tailored SSD solution is the best path to optimization and eliminating unnecessary costs, and a great SSD manufacturing partner i
Chatbot: I can either help you find a new SSD for your laptop, or I can provide basic information about SSD technology.
Chatbot: How can I help you today?
You: I want to learn about nvme
Chatbot: Here's something interesting about nvme:
Chatbot: MSI's Spatium M570 PCIe 5.0 NVMe M.2 HS is based on the Phison PS5026-E26 controller paired with 3D TLC NAND memory, pos
Chatbot: I can either help you find a new SSD for your laptop, or I can provide basic information about SSD technology.
Chatbot: How can I help you today?
You: What else can I research?
Chatbot: Here's a list of topics I can tell you about:
Chatbot: raid, storage, nvme, usb, ssd, controller, pcie, card, drive, nand
Chatbot: I can either help you find a new SSD for your laptop, or I can provide basic information about SSD technology.
Chatbot: How can I help you today?
You: I wanna find out about RAID
Chatbot: Here's something interesting about raid:
Chatbot: This is a full-height PCIe 4.0 x16 card that is host to 8x PCIe 4.0 M.2-2280 slots, and allows RAID 0, 1, and 10 functionality.
Chatbot: I can either help you find a new SSD for your laptop, or I can provide basic information about SSD technology.
Chatbot: How can I help you today?
```

Appendix for the Knowledge Base

Knowledge Base For Fact Queries

Our code for this part comes from a previous assignment.

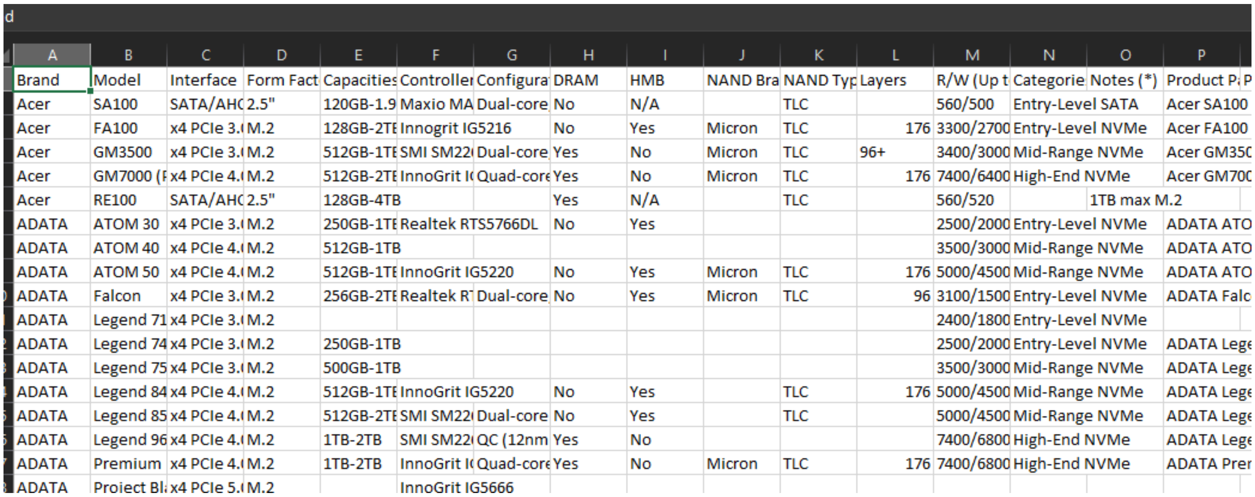
The knowledge base is a pickle file that contains the data scraped from 20 articles relating to SSD. The pickle file contains data as a dictionary of top terms. Here we are limiting the searchable terms as it would require web scraping on a larger scale and is more resource intensive.

Terms we can research/give facts about are:

1. raid,
2. storage,
3. nvme,
4. usb,
5. ssd,
6. controller,
7. pcie,
8. card,
9. drive,
10. nand

Knowledge Base For SSD Data

We found a spreadsheet that lists data about various SSDs.(**SSD_Master_List.csv**). The SSD suggestions given to the user come from this file.



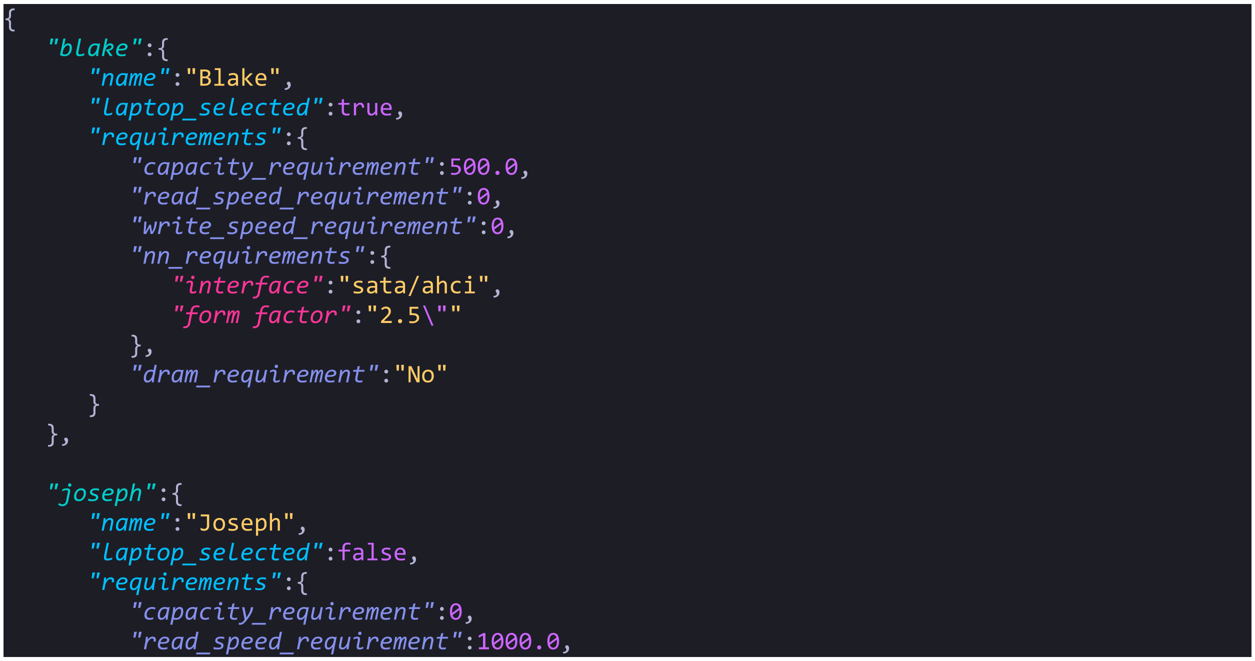
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	
Brand	Model	Interface	Form Fact	Capacities	Controller	Configura	DRAM	HMB	NAND Bra	NAND Typ	Layers	R/W (Up t	Categorie	Notes (*)	Product P	P
Acer	SA100	SATA/AHC	2.5"	120GB-1.9	Maxio MA	Dual-core	No	N/A		TLC		560/500	Entry-Level SATA		Acer SA100	
Acer	FA100	x4 PCIe 3.	M.2	128GB-2TE	Innogrit IG5216		No	Yes	Micron	TLC	176	3300/2700	Entry-Level NVMe		Acer FA100	
Acer	GM3500	x4 PCIe 3.	M.2	512GB-1TE	SMI SM22	Dual-core	Yes	No	Micron	TLC	96+	3400/3000	Mid-Range NVMe		Acer GM35C	
Acer	GM7000 (F	x4 PCIe 4.	M.2	512GB-2TE	InnoGrit I	Quad-core	Yes	No	Micron	TLC	176	7400/6400	High-End NVMe		Acer GM70C	
Acer	RE100	SATA/AHC	2.5"	128GB-4TB			Yes	N/A		TLC		560/520		1TB max M.2		
ADATA	ATOM 30	x4 PCIe 3.	M.2	250GB-1TE	Realtek RTS5766DL		No	Yes				2500/2000	Entry-Level NVMe		ADATA ATO	
ADATA	ATOM 40	x4 PCIe 4.	M.2	512GB-1TB								3500/3000	Mid-Range NVMe		ADATA ATO	
ADATA	ATOM 50	x4 PCIe 4.	M.2	512GB-1TE	InnoGrit IG5220		No	Yes	Micron	TLC	176	5000/4500	Mid-Range NVMe		ADATA ATO	
ADATA	Falcon	x4 PCIe 3.	M.2	256GB-2TE	Realtek R	Dual-core	No	Yes	Micron	TLC	96	3100/1500	Entry-Level NVMe		ADATA Falc	
ADATA	Legend 71	x4 PCIe 3.	M.2									2400/1800	Entry-Level NVMe			
ADATA	Legend 74	x4 PCIe 3.	M.2	250GB-1TB								2500/2000	Entry-Level NVMe		ADATA Lege	
ADATA	Legend 75	x4 PCIe 3.	M.2	500GB-1TB								3500/3000	Mid-Range NVMe		ADATA Lege	
ADATA	Legend 84	x4 PCIe 4.	M.2	512GB-1TE	InnoGrit IG5220		No	Yes		TLC	176	5000/4500	Mid-Range NVMe		ADATA Lege	
ADATA	Legend 85	x4 PCIe 4.	M.2	512GB-2TE	SMI SM22	Dual-core	No	Yes		TLC		5000/4500	Mid-Range NVMe		ADATA Lege	
ADATA	Legend 96	x4 PCIe 4.	M.2	1TB-2TB	SMI SM22	QC (12nm	Yes	No				7400/6800	High-End NVMe		ADATA Lege	
ADATA	Premium	x4 PCIe 4.	M.2	1TB-2TB	InnoGrit I	Quad-core	Yes	No	Micron	TLC	176	7400/6800	High-End NVMe		ADATA Prer	
ADATA	Project Bl	x4 PCIe 5.	M.2		InnoGrit IG5666											

Knowledge Base For Laptop Data

We use a Google Search API and a simple web scraper to look up the storage interface of the user's laptop.

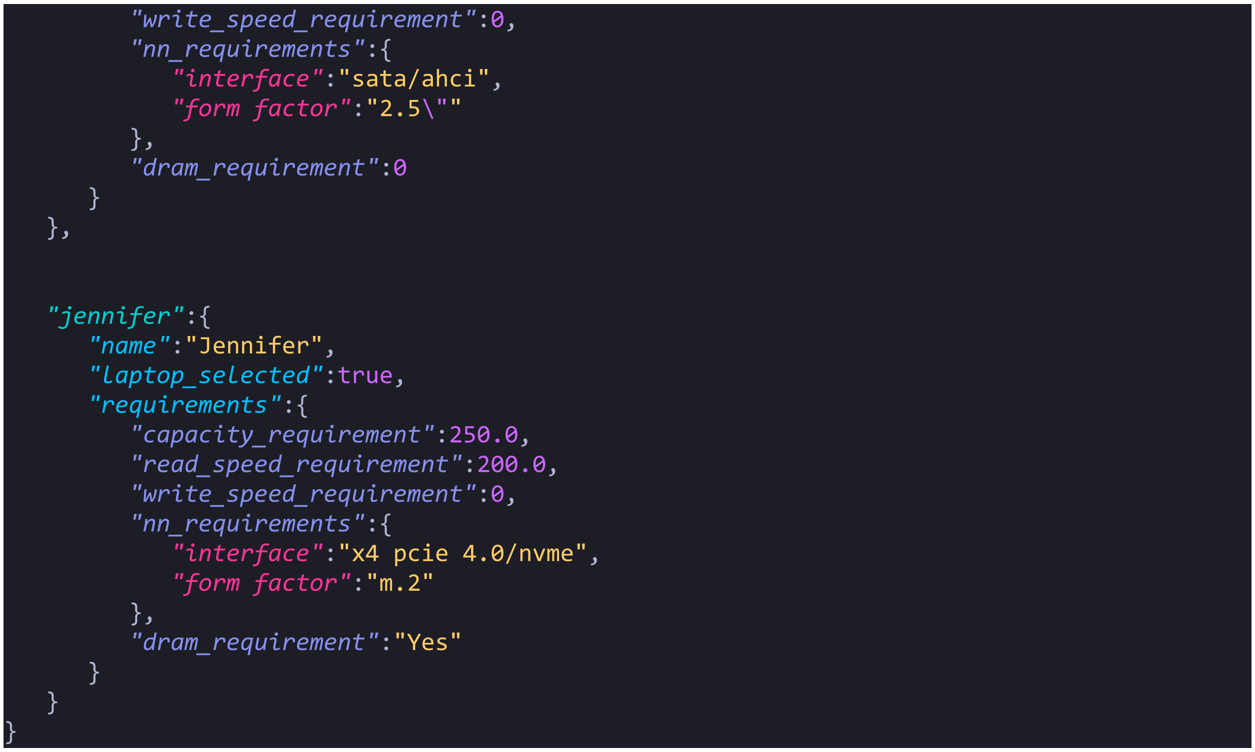
Appendix for Sample User Models

User models are stored as JSON files. Here's some examples



```
{
   "blake":{
      "name":"Blake",
      "laptop_selected":true,
      "requirements":{
         "capacity_requirement":500.0,
         "read_speed_requirement":0,
         "write_speed_requirement":0,
         "nn_requirements":{
            "interface":"sata/ahci",
            "form factor":"2.5\""
         },
         "dram_requirement":"No"
      }
   },

   "joseph":{
      "name":"Joseph",
      "laptop_selected":false,
      "requirements":{
         "capacity_requirement":0,
         "read_speed_requirement":1000.0,
```



```
         "write_speed_requirement":0,
         "nn_requirements":{
            "interface":"sata/ahci",
            "form factor":"2.5\""
         },
         "dram_requirement":0
      }
   },


   "jennifer":{
      "name":"Jennifer",
      "laptop_selected":true,
      "requirements":{
         "capacity_requirement":250.0,
         "read_speed_requirement":200.0,
         "write_speed_requirement":0,
         "nn_requirements":{
            "interface":"x4 pcie 4.0/nvme",
            "form factor":"m.2"
         },
         "dram_requirement":"Yes"
      }
   }
}
```

Analysis

Strengths

The main strength of this chatbot is that it can usually guide the user through a simple SSD selection flow without much issue.

Particularly, it can (usually) extract the correct intent from the user. That is, it can determine whether the user wants to buy an SSD or find a random fact about SSDs. This problem is not as trivial as it seems because there are many different ways a user could say they want to purchase new storage or research a new random fact. In addition to extracting research vs. purchase intents, it also oftenly identifies the correct intent for SSD specification criteria (ex: capacity vs read speed vs write speed) using similar techniques.

Another advantage is that it allows the user to make some errors. For example, typos in the name of the laptop model are usually resolved by a technique described earlier. And the tokenization and word bucket techniques do not require exact matches in order to give an accurate result.

Aside from NLP mechanisms, we believe other aspects of the solution including the web scraper and web interface are robust and suitable for the application domain.

Weaknesses

This chatbot is limited in terms of what it can do. The user model it builds is not very intricate. It isn’t particularly good at “making small talk” or being generally conversational. It isn’t good at providing meaningful research facts in response to user queries, especially since it only recognizes 10 research terms. And the techniques we used for determining user intent are not as powerful or accurate as other machine learning techniques such as deep neural networks / LSTMs. A language transformer model such as GPT-3 would probably be a far superior choice for making a smart chatbot that can provide rich and tailored responses to the user.