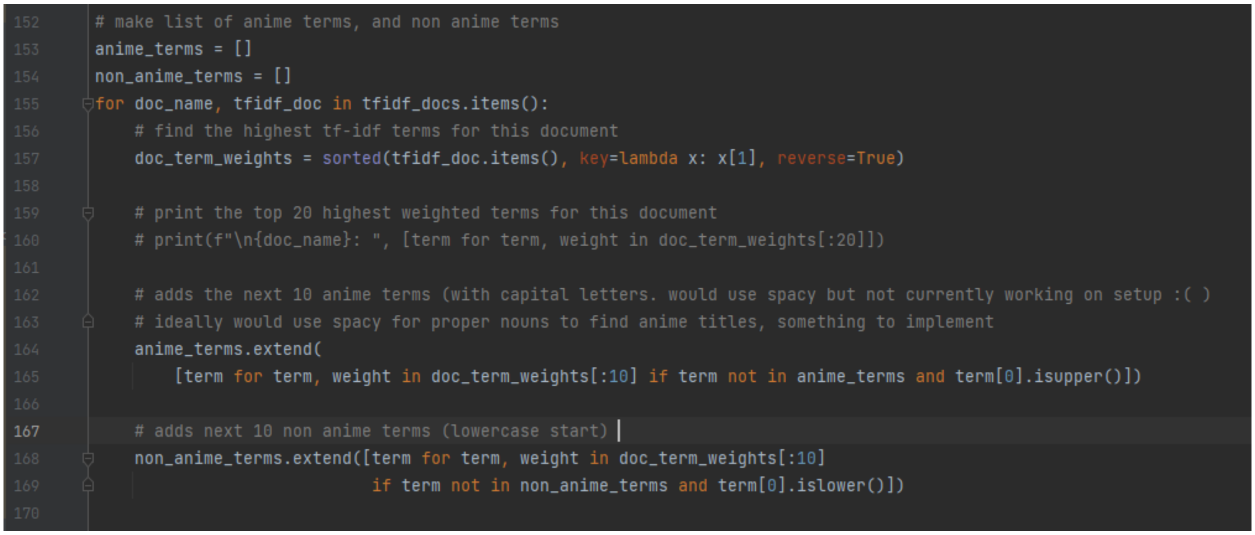
Vicky Nguyen vtn180004, Meinhard Capucao mdc190005
CS4395.001

# Web Crawler

(1) describe how you created your knowledge base, include screen shots of the knowledge base, and indicate your top 10 terms

Our knowledge base was created by using a web crawler to extract content that is from different web pages that are related to anime. Our starter source was a wikipedia article by Hayao Miyazaki. That data would be stored in separate text files for each page/URL. We anyalzed the text files by using TF-IDF.



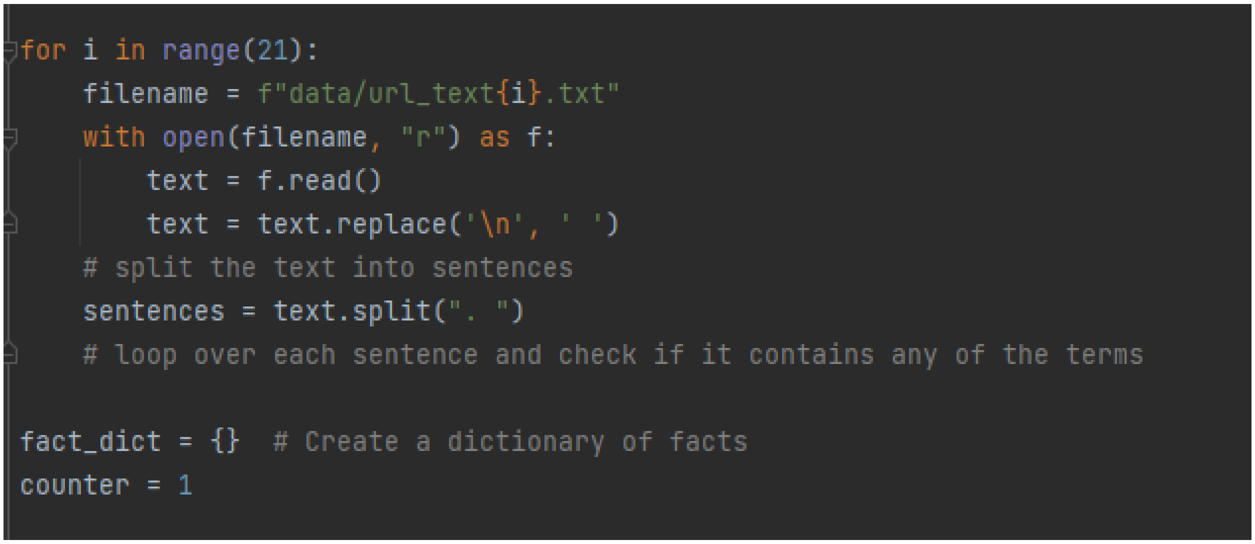
```
# make list of anime terms, and non anime terms
anime_terms = []
non_anime_terms = []
for doc_name, tfidf_doc in tfidf_docs.items():
    # find the highest tf-idf terms for this document
    doc_term_weights = sorted(tfidf_doc.items(), key=lambda x: x[1], reverse=True)

    # print the top 20 highest weighted terms for this document
    # print(f"\n{doc_name}: ", [term for term, weight in doc_term_weights[:20]])

    # adds the next 10 anime terms (with capital letters. would use spacy but not currently working on setup :( )
    # ideally would use spacy for proper nouns to find anime titles, something to implement
    anime_terms.extend(
        [term for term, weight in doc_term_weights[:10] if term not in anime_terms and term[0].isupper()])

    # adds next 10 non anime terms (lowercase start)
    non_anime_terms.extend([term for term, weight in doc_term_weights[:10]
                            if term not in non_anime_terms and term[0].islower()])
```

This function makes a list of terms based on the tf-idf doc that holds each of the 21 tf-idf dictionaries. Since this fact base uses non anime terms (since anime titles are proper nouns and use upper case), it extracts the next lowercase terms that are deemed important by the tf-idf index.

The first list is the important terms pre-evaluation, which prints 38 terms. Here is our list of the 10 most important terms (note: not anime titles).

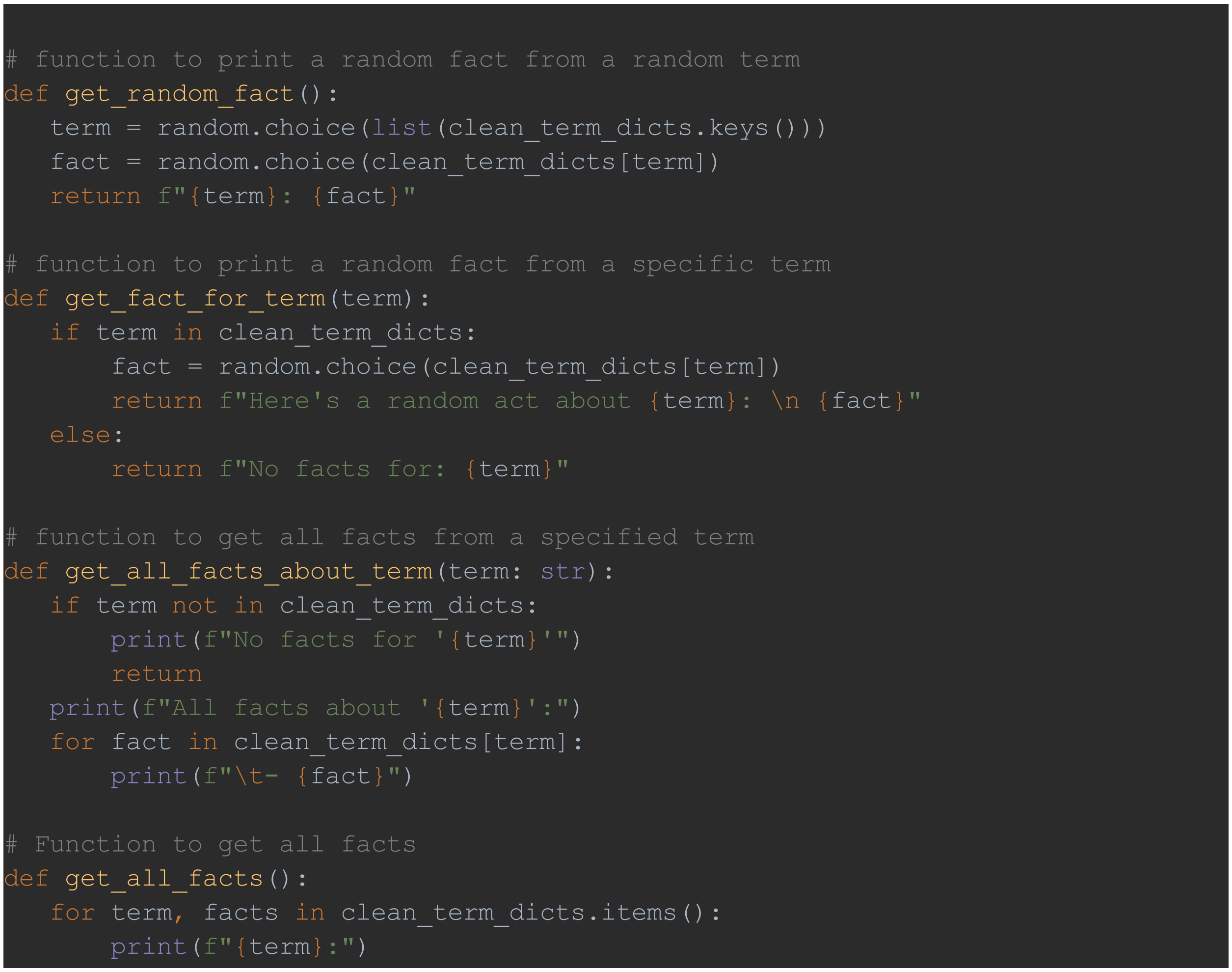
**Our top ten terms are:**

['sushi', 'winner', 'love', 'greatest', 'awards', 'animated', 'cultural', 'highest', 'newest', 'film']



```python
for i in range(21):
    filename = f"data/url_text{i}.txt"
    with open(filename, "r") as f:
        text = f.read()
        text = text.replace('\n', ' ')
    # split the text into sentences
    sentences = text.split(". ")
    # loop over each sentence and check if it contains any of the terms

fact_dict = {}  # Create a dictionary of facts
counter = 1
```

Then, to determine facts, we read in each of the url's again, then created a fact dict. There are four functions specified to get facts:



```python
# function to print a random fact from a random term
def get_random_fact():
    term = random.choice(list(clean_term_dicts.keys()))
    fact = random.choice(clean_term_dicts[term])
    return f"{term}: {fact}"

# function to print a random fact from a specific term
def get_fact_for_term(term):
    if term in clean_term_dicts:
        fact = random.choice(clean_term_dicts[term])
        return f"Here's a random act about {term}: \n {fact}"
    else:
        return f"No facts for: {term}"

# function to get all facts from a specified term
def get_all_facts_about_term(term: str):
    if term not in clean_term_dicts:
        print(f"No facts for '{term}'")
        return
    print(f"All facts about '{term}':")
    for fact in clean_term_dicts[term]:
        print(f"\t- {fact}")

# Function to get all facts
def get_all_facts():
    for term, facts in clean_term_dicts.items():
        print(f"{term}:")
        for fact in facts:
            print(f"\t- {fact}")
```

For our facts, we had two ways of storing it. One is a pickled dictionary that contains all facts, with the keys as the terms and the values as the facts as strings. Another is a text document that stores them based on topic. These are two convenient ways.

We can use these in a chatbot to store topics specifically about anime terms and keywords (not anime themselves. Of course, there is work to be done with cleaning up the fact files since they are text heavy, but with some modification they wouldn't be that difficult to integrate with a chatbot.

### Sample 1:

**User:** Hello.

**Anime/Manga Term Chatbot:** Hello, what would you want to learn about?

**User:** Tell me more about anime that has won awards in the past 20 years.

**Anime Term Chatbot:** Is there any specific anime you want to explore, or just a random anime that has won an award?

**User:** Give me a random one for now.

**Anime Term Chatbot:** Yojiro Takita's Okuribito in 2008 almost swept the rest of the awards with a tally of 10, including the overall Picture of the Year Award.

*Comments*

This was taken from Fact 214: ***"****Yojiro Takita s Okuribito almost swept the rest of the awards with a tally of 10, including the overall Picture of the Year Award)*
We would augment this by searching the anime in fact, then saying the year it was released, which was 2008. This can be done using fact libraries.

### Sample 2:

**User:** Hello.

**Anime/Manga Term Chatbot:** Hello, what would you want to learn about?

**User:** What is the greatest animated film ever made?

**Anime Term Chatbot:** Hayao Miyazaki's Spirited Away is one of the greatest animated films ever made, and might be better than any Disney film I've ever seen.

*Comments*

Having users ask questions about factual statements will be easier with cleanup, as a database can store things about dates of animes, releases, etc. There are libraries for those as well that can help with that. How about questions with opinions?

Generally, while chatbots generally don't give opinionated answers unless trained to, there can be facts in the database which can make it respond in those ways. For example, if we were to search for the term greatest (one of our most important terms), we have these **exact** facts

- Fact 172: Log In Create Account Steven Spielberg Talks About His Meeting with Hayao Miyazaki at "Ready Player One" Talk Event in Tokyo "His Spirited Away is one of the greatest animated films ever made." Mikikazu Komatsu April 20, 2018 2:30am CDT (4/20/18) After the Japanese premiere event for his latest film Ready Player One yesterday, director Steven Spiellberg also attended a talk session along with three cast of the film: Tye Sheridan (Parzival/Wade Watts), Olivia Cooke (Art3mis/Samantha Cook), and Win Morisaki (Daito/Toshiro).

- Fact 178: His Spirited Away is one of the greatest animated films ever made, might be better than any Disney film I've ever seen

- Fact 182: Facebook Pixel Code End Facebook Pixel Code Shows Manga News Forums Store Premium Try Free Login Queue Random Search Search header Steven Spielberg Talks About His Meeting with Hayao Miyazaki at "Ready Player One" Talk Event in Tokyo "His Spirited Away is one of the greatest animated films ever made." Mikikazu Komatsu April 20, 2018 12:30am PDT (6 hours ago) Tweet After the Japanese premiere event for his latest film Ready Player One yesterday, director Steven Spiellberg also attended a talk session along with three cast of the film: Tye Sheridan (Parzival/Wade Watts), Olivia Cooke (Art3mis/Samantha Cook), and Win Morisaki (Daito/Toshiro)

- Fact 188: His Spirited Away is one of the greatest animated films ever made, might be better than any Disney film I've ever seen.

While there are duplicates due to some websites derived from Hayao Miyazaki having similar interviews and information (which would be something we would have to cleanup in use for a chatbot, however, the derived urls from Hayao Miyazaki's Wikipedia Page are very complex anyways), we can clearly see there is a film Steven Spielberg considers the best. This comes from the interview itself. So, we can make the chatbot spit that information out after cleaning up the data for the term 'greatest'.

There are many things to be cleaned up with the amount of HTML tags, symbols, and lines to parse through, but with basic data extraction and web crawling with BeautifulSoup, one can see the potential to make a great information database!