Reg Gonzalez CS 4395.001

Portfolio Project - Finding/Building a Corporus - Report

The starting URL I started with was the Wikipedia page to Henry Selick's and Tim Burton's *The Nightmare Before Christmas*:

```
starting_url = "https://en.wikipedia.org/wiki/The_Nightmare_Before_Christmas"
```

The data I got from the 15 pages was all the visible text on-screen. If there was an element from the webpage that was a style element (i.e., something that didn't provide any visible text, but rather formatted and styled the web page), I didn't include that in the data I got. Originally, I tried to only get text that was labeled with the paragraph tag (…), and while that worked out great for some pages, it didn't work well for all. Below is an example of text I extracted from one of the pages:

```
main.py   textfile1   knowledge_base   urls.txt
Subscribe  Sign In BETA This is a BETA experience. You may opt-out byNBSP clicking here More From Forbes Oct 4, 2022, 05:25pm EDT Chris Cuomo's Return To Prime Time Lands
 Follow this author to stay notified about their latest stories.
 Got it! Oct 15, 2013, 09:00am EDT | This article is more than 8 years old. Share to Facebook Share to Twitter Share to Linkedin

 The Nightmare Before Christmas (Photo credit: Wikipedia)

 Today marks the 20th Anniversary of Tim Burton and Henry Selick's  The Nightmare Before Christmas . If you happen to venture to Disneyland (and I presume Disney World) over the nex
 Now in 2013, Disney's fear of Tim Burton's sketchbook come-to-life (actually directed by Henry Selick) might appear as foolish as Billy Zane dissing Picasso in the opening moments

 While  Batman ReturnsNBSP earned its PG-13 it was also infamously sold to children via McDonalds happy meal tie-ins and the usual kid-friendly marketing. The film basically invente

 The picture went out on just two screens on October 15th, earning an eye-popping $95,000 per-screen and the second-biggest per-screen average of all time behind Disney's 2-screen d
 But Disney had so little faith in the project that they had slotted their big-budget revamp of  The Three Musketeers  for November 12th of that year, meaning that the somewhat surp

 But that's of course not where the story ends. As the film became attained a cult following, with the kids who grew up with it making it the de-facto holiday film for their kids ov
 Via these reissues, the domestic total of  The Nightmare Before Christmas NBSPwent from $50m to $75m. Arguably as important as the film's passionate fanbase is its cultural acceptan
 Walt Disney
  Pictures banner, while the film itself is now a cornerstone of Disney's marketing efforts during a key shopping season. It is now considered not a potentially dangerous horror fil
 If you had told me twenty years ago that  Nightmare Before Christmas  merchandise would be littering the Disney empire while Jack Skellington would take over the Haunted Mansion ev
```

After extracting text from each of the 15 pages, I cleaned them up by removing newlines and tabs. I used the .split() method on the text and then joined them using spaces. The result was basically just one very long line of text that included all the sentences of the webpages' extracted text. Below is an example of what the reformatted file looks like (using the textfile from above):

```
main.py   reformatted_textfile1
Subscribe Sign In BETA This is a BETA experience. You may opt-out by clicking here More From Forbes Oct 4, 2022, 05:25pm EDT Chris Cuomo's Return To Prime Time Lands In Las
```

Portfolio Project - Finding/Building a Corporus - Report

After reformatting the textfiles for all 15 pages, I put them all into one big file called "all_text_reformatted.txt". I then took all that text, tokenized it, lowercased it, and removed stopwords and punctuation. Next, I put that newly tokenized text into a textfile called "all_text_tokenized.txt". From here, I made a unigram list of the words from the file and a unigram dictionary that held the count for each unigram. After sorting that dictionary, I outputted the top 40 terms from the text, which was:

```
Top 40 most common terms in all 15 pages:
1. ("'film", 142)
2. ("'christmas", 137)
3. ("'nightmare", 132)
4. ("'burton", 127)
5. ("'disney", 120)
6. ("'tim", 93)
7. ("'best", 93)
8. ("'new", 90)
9. ("'movies", 87)
10. ("'movie", 85)
11. ("'jack", 68)
12. ("'reviews", 66)
13. ("'time", 57)
14. ("'selick", 56)
15. ("'tv", 55)
16. ("'news", 49)
17. ("'would", 48)
18. ("'first", 48)
19. ("'like", 47)
20. ("'one", 47)
21. ("'back", 45)
22. ("'halloween", 44)
23. ("'ago", 43)
24. ("'games", 42)
25. ("'season", 39)
26. ("'little", 38)
27. ("'us", 37)
28. ("'holiday", 36)
29. ("'see", 32)
```

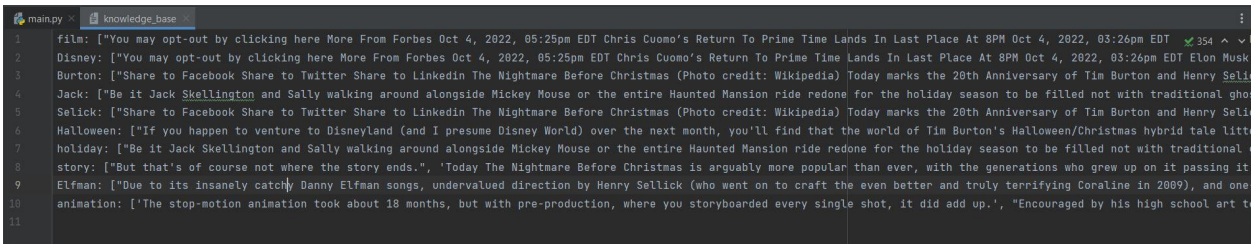
```
30. ("'story", 32)
31. ("'danny", 31)
32. ("'home", 31)
33. ("'review", 30)
34. ("'animation", 30)
35. ("'henry", 30)
36. ("'release", 30)
37. ("'elfman", 29)
38. ("'coming", 29)
39. ("'year", 28)
40. ("'releases", 28)
```

From these 40 terms, I handpicked 10, which were: "film," "Disney," "Burton," "Jack," "Selick," "Halloween," "holiday," "story," "Elfman," and "animation." I picked these 10 terms because I thought they were the terms that were the most broad and covered a lot of different important aspects about the movie.

After picking these 10 terms, I used them to create a knowledge base. To create this knowledge based, first I used the text from "all_text_reformatted.txt". I got the sentences from that text

Portfolio Project - Finding/Building a Corporus - Report

using NLTK's sentence tokenizer. Then I created two new structures: a dictionary and list. For the dictionary, the key is the one of my 10 terms (i.e., "film," "Disney," "Burton," etc.) and the value for that key would be a list the sentences from all 15 pages that contained that term. The list structure would be used to hold those sentences that contained each respective term. After creating that dictionary, I pickled it, read it in, and then wrote the dictionary out to a file called "knowledge_base". Below is a screenshot of that knowledge base:



As you can see, I formatted the file to display it as: term: [list of sentences that contain the term].

Below are sample dialogues of how I might use the data I collected in the knowledge base with a chatbot.

Sample dialogue 1:

- Hi! My name is Zero and I'm a chatbot that has weirdly extensive knowledge about Tim Burton's and Henry Selick's *The Nightmare Before Christmas*. What's your name?
- My name is Reg.
- Hi, Reg! Are you a fan of *The Nightmare Before Christmas*?
- Yeah, I love it
- That's great! Who's your favorite character? Mine is Oogie Boogie.
- Jack's probably my favorite.
- Jack is a great character! Did you know he was voiced by two people? Chris Sarandon did his speaking voice and Danny Elfman did his singing voice.

  .

  .

  .

Portfolio Project - Finding/Building a Corporus - Report

Sample dialogue 2:

- Hi! My name is Zero and I'm a chatbot that has weirdly extensive knowledge about Tim Burton's and Henry Selick's *The Nightmare Before Christmas*. What's your name?
- Hey, my name is Flop.
- Hey, Flop! Are you a fan of *The Nightmare Before Christmas*?
- No, I'm not.
- Then you are of no use to me. Bye, Flop!