John Moreland

Project Overview:

My design goals for this project were to create something simple but useful. I'm an inherently indecisive person, so I decided to design a script that could make a decision for the user. Often times, I will find myself trying to figure out which book on my list I should read next. So, I created a set of functions that will recommend to the user which book is better. After scraping summaries from the internet and performing sentiment analysis, the program is able to determine the “better” book.

Implementation:

On a broad sense, my implementation is very simple. Using two book names as inputs, the functions utilize the pattern API to find the summaries of the books. Still using the pattern API, the comparison function performs sentiment analysis on the two texts in order to generate a relative ranking. Depending on how large the ranking is, the function will recommend books to different degrees (i.e. “They're about the same” vs “If you Give a Mouse a Cookie is *way* better”).

In order to speed up the functions, I added a cache feature which stores a local copy of books that have been searched for. Before checking wikipedia for a summary, the function will see if the summary is already stored on disk. This improves run time by about 10-20 seconds per book.

Results:

Here are some of my favorite outputs:

Book 1	**Book 2**	**Winner**
Of Mice and Men	Moby Dick	Of Mice and Men
If you give a Mouse a Cookie	The Great Gatsby	If you give a Mouse a Cookie
1984	Great Expectations	1984

Reflection:

After creating my function, I realized that I could have effectively done the same thing by creating a digital “coin flip”. I could definitely find ways to pull summaries from other sources on the internet – wikipedia is limited. This was my first experience with using API's, so it was good practice for me to get acquainted with such a powerful tool.