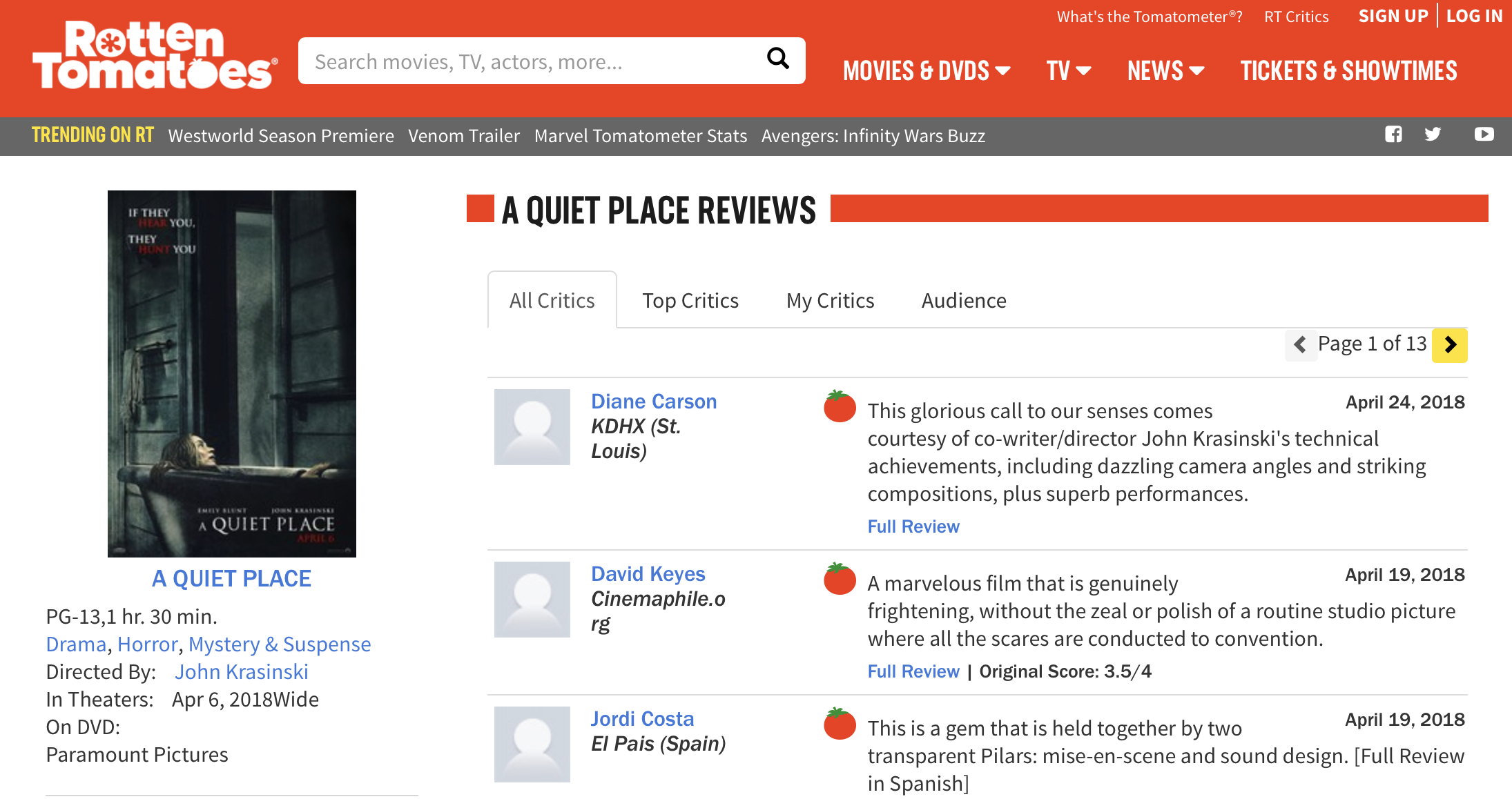
**Exercise 9: Web page mining**

**Assignment Specification**

**Description**: Choose a popular movie you like and find the id of this movie at rottentomatoes.com

* Example: https://www.rottentomatoes.com/m/a\_quiet\_place\_2018/reviews
* Movie id = “a\_quiet\_place\_2018”
* “A quite place” has 19 pages of reviews. You need to collect reviews from all the pages
* Create a list of lists [review, score], where “review” is the review text content and “score” is the rate given for each review. This score can be a nominal grade, fraction or can be missing – see picture
* Rank reviews by score
* Test your program with different movie ids.



**Input**: Data will be collected from the website www.rottentomatoes.com. User can be prompt to insert movie id, but it’s not required.

**Output**: Write a document with a summary of your findings. See details in the Procedure.

**Procedure**:

* Import the needed libraries
* Define the target URL and open it
* Load the page into your “soup”/beautifulsoup
* Identify the total number of pages
* Loop through each page, creating a new soup every time
* Collect review and respective score
* Generate a list of lists for the reviews
* Standardize scores and remove the reviews with no rating. Replace nominal scores (where they appear) with numerical values: A+':5, 'A':4.8, 'A-': 4.6, 'B+':4.45, 'B':4.3, 'B-':4.1, 'C+':3.95, 'C':3.8, 'C-':3.6, 'D+':3.45, 'D':3.3, 'D-':3.1. Normalize the numerical values in a scale 1 to 5.
* Write a document with your findings. This would include:
  + top 5 and bottom 5 reviews lists
  + a word cloud for each of them (top and bottom)
  + your interpretation and comments.

Be sure to use only the text part of the reviews list of lists and to clean stop words using the stopword\_en.txt file.

* Submit both the document (pdf/docx) and your Python script.