**Problem:**

2. Acquire list of \*possibly\* all tourist destinations in the world

**Idea:**

Thank you for the opportunity to try out this challenge for AnyWhr, but before I get to explain my solution, I want to clarify that I had huge constraints for this task regarding my time, as I have other commitments such as school lessons and assignments. I also had no prior knowledge to scrapping before this task. I researched the basics of scrapping during this short period of time, and I’m able to perform scrapping for a specific page, however, I still lack experience and knowledge regarding traversing of pages etc, more advanced scrapping techniques.

I’ve chosen a very systematic top-down approach to this problem, from continents to cities and then finally to the attractions itself. I think that travellers often do not confine themselves to a city or country while travelling, hence I think that classifying them from continent will be the most logical approach, and it also makes it simple to select our queries if we want to sort them out according to continents or cities.

My website of choice for information is tripadvisor.com, as they have an extremely large pool of resources, given that they operate internationally, have a long history, and can be contributed by anyone. The downsides of using trip advisor is that the website is hard to navigate, as the layout changes depending on which page you are on. Also, some of trip advisor’s listings could very easily be gamed, which is why we might need additional features to protect us from false listings, which I will mention later.

Firstly, I created an attraction filter class, which we could define minimum requirements such as minimum ratings or minimum reviews. This allows us to change our requirements easily without having to change much of the scrapping algorithm in the future. For example, I have set my minimum ratings to 4.5 and minimum reviews to 100. This will allow us to only receive the best attractions for each place. Filter is also expandable, and we could add more conditions into it in the future.

Unfortunately, there was no way of going through the continents on trip advisor systematically, and I had to hardcode the links of the continents. I placed the links in a dictionary with the continent name as the key.

I also created a file to write all the data collected to.

Firstly, on the continent page, we could go through the pages to view all the cities in the continent, which allows us to systematically go through each city and list out its attractions.

On the continent page, I use beautifulsoup4 (bs4) to get the link to each city. Due to timing constraints, I can retrieve the relative url to the individual city’s pages but have yet to come out with a solution to covert it to an absolute link. I face more difficulty here as trip advisor has different formatting for the first page as compared to the rest of the pages. However, I do not think that is that big of a problem as it could be solved with more time.

When it gets the city link, my algorithm will help look for and re-direct to the attractions page, where all the attractions can be viewed as a list.

I programmed the algorithm to read the name of the attraction and the ratings and reviews. The algorithm can be expanded to take in more information in the future, such as the location of the attraction, which may help us to further plan trips based on an attraction‘s proximity to one another. After scanning the first page, the algorithm traverses to the second page using the link as given in page number section of the webpage, and using the max number of pages for the city, as captured by reading the pages at the bottom of the page. Using our defined requirements, we can then filter out the attractions that are deemed to be unreliable.

When the algorithm is done for a city, it will then move on to the next city. When the whole continent is done, it will move on to the next continent until all the attractions for the world are listed!

**Conclusion:**

I think I’ve learnt a lot from this opportunity, and given more time, I would definitely love to improve and complete my solution, such as improve the traversing of page such that it could capture the attractions of all the world accurately. However, even though parts of the algorithm is incomplete due to my lack of experience and knowledge, I believe that the strategy is clear, and could get a very accurate and customizable list of attractions around the world.

If you have any questions regarding me or my solution, feel free to contact me at 90094007 or email me at [e0310209@u.nus.edu](mailto:e0310209@u.nus.edu). Thank you!