Assigment-03 Beijing Subway Transfer Agent

AI-For-NLP Course Group Nov.18-2018

Please using the search policy to implement an agent. This agent receives two input, one is @param *start* station and the other is @param *destination*. Your agent should give the optimal route based on Beijing Subway system.

Deadline: 2018-Nov-30
Submit: Submit the source code and result to github.

Dataflow:

1. Get data from web page.

a. Get web page source from: https://baike.baidu.com/item/%E5%8C%97%E4%BA%AC%E5%9C%B0%E9%93%81/408485;
b. You may need @package requests page to get the response via url;
c. You may need save the page source to file system.
d. The target of this step is get station information of all the subway lines;
e. You may need install @package beautiful soup to get the url information, or just use Regular Expression to get the url. Our recommendation is that using the Regular Expression and BeautiflSoup both.
f. You may need BFS to get all the related page url from one url.
Question: Why do we use BFS to traverse web page (or someone said, build a web spider)? Can DFS do this job? which is better?

2. Preprocessing data from page source.

a. Based on the page source gotten from url. You may need some more preprocessing of the page.
b. the Regular Expression you may need to process the text information.
c. You may need @package networkx, @package matplotlib to visualize data.
d. You should build a *dictionary* or *graph* which could represent the connection information of Beijing subway routes.
e. You may need the *defaultdict, set* data structures to implement this procedure.

3. Build the search agent

a. Build the search agent based on the graph we build.
b. As much as you can to use the already implemented search agent. You just need to define the *is_goal, get_successor, strategy* three functions.

4. Create different policies for transfer system.

a. Define different policies for transfer system.
b. Such as Shortest Path Priority（路程最短优先）, Minimum Transfer Priority(最少换乘优先), Comprehensive Priority(综合优先)
c. Implement Continuous transfer. Based on the Agent you implemented, please add this feature: Besides the @param *start* and @param *destination* two stations, add

some more stations, we called @param *by_way*, it means, our path should from the start and end, but also include the @param *by_way* stations.

e.g

1. Input: start=A, destination=B, by_way=[C]
   Output: [A, … .., C, …. B]
2. Input: start=A, destination=B, by_way=[C, D, E]
   Output: [A … C … E … D … B]
   # based on your policy, the E station could be reached firstly.

5. **Test your result with commercial applications.**
   将你的结果和高德地图或者百度地图进行比较，如果有不同，请分析原因