# **Code Review**

*Reviewer*: **Miranda**

*File Name*: **networkvisual.py**

*Version Name*: **4837580c9244f7afcf1a8a25bb3e7e1c32c9a2f3**

*Location*: **team01-Project/src/**

**Correctness:**

===draw\_graph(...)===

* Manually checked the output image with different input data, the content and layout are set properly based on input.
* Not sure whether it is needed to return imgname, which is exactly as same as one input parameter.
* Line 59: command will show the image, however, it doesn’t make sense for server, it cannot check that. If it is for testing purpose, should be deleted later.

===get\_references\_in\_article()===

* Line 85 - 95: the code use if statement and KeyError exception catching, it can be simplified by using if to check whether has source url, and execute iff if True.
* Line 69 & 81(minor): generated dictionary in different way, better to make them consist.

===get\_references\_to\_source()===

Same KeyError exception issue

===get\_references\_from\_source()===

Same KeyError exception issue

===draw\_references\_functions()===

Those functions will call corresponding data generating functions, however, if the target source/article id is not valid, it will return an empty dictionary and string, in this case, the draw\_references\_functions shouldn’t call graph generating then. However, the draw\_references\_functions() will generate the graph without checking this, which may causes error tp draw\_graph() function.

**Documentation:**

Good use of documentation. However some sections can be combined/ truncated to get more cohesion.

===draw\_graph(...)===

Line 18-21 docstring:

It will be better if there are some description about how the input “graph” represents its edges and nodes, and the types of key and values since they are used separately in function.

For returning name, it will be better to mention that user can input it as a parameter “imgnage”, so that users will know where the “name of graph comes from”.

Line 57:

For variable “graph title”,it is 1 of the input parameters, but there are many input parameters, it might be easy to lose some information. So it will be better to mention here how the user can set this variable.

===get\_references\_in\_article(database, a\_id)===

Line 65: It will be better to indicate how does “the article“ related to input “a\_id”.

===draw\_references\_in\_article(database, imgname, a\_id)===

Line 99: There are 3 input parameter, but (Database, int) only indicate there are w parameters.

Same problem happens in “draw\_references\_()” functions .

Line 101: Not clear about what “Generate a network graph given the database”.

Line 103(minor mistake): Change “outputted” to “output".

===get\_references\_to\_source(database, s\_id)===

Line 139: Could add more details about “Keyerror”, so that user will know what wrong with that.

For the draw\_graph function, there are many input parameters for layout setting, but it is not clear how to use the parameters to set layout, this issue can be improved as well.

**Coding Style:**

* Coding conventions are followed.
* Proper white space are added for making the code structure more clear.
* However, some minor issues can still be improved if checked by “PEP 8”.

**Usability:**

* “draw\_graph” function has too many parameters, not easy to understand how to use them for layout setting.
* The appearance of output images’ arrow can be improved.
* Good level of usability, can generated different types of visualizations.
* Separated functions for partial functionality, which are defined reasonable, and can increase the independency
* And doesn’t have functions with duplicated usability, e.g. use same “draw\_graph()” to generate graph

**Testing:**

* The unit tests cover all data generating functions.
* Basic cases (empty/invalid, single, multiple) are included, but can added more special cases since the real sites’ data could be trivial sometimes.
* There are sample tests for demo in the code file, it should be removed when it is not used.
* No tests for graph generating functions.