**Part1 :**

**(1). Language: Python 2.7.11**

**(2). Tool: Mac + IDLE**

**(3). Data structure:**

dis\_mat: Array[Array[]] (matrix; 10(objects) X 10(objects))

new\_dis\_mat: Array[Array[]] (matrix; 10(objects) X 10(objects))

res: Array[Array[]] (matrix; 2(new dimension, coordinates) X 10(objects))

**(4). Logic:**

Step1: get distance matrix, each element in the matrix means distance between two objects

Step2: find the farthest pair of objects

Step3: get one coordinate (X in the coding)

Step4: update the new distance between objects and get the new distance matrix

Step5: repeat step2 to step 4 to generate k dimensions

Step6: plot the points in 2D space

**(5). Optimization:**

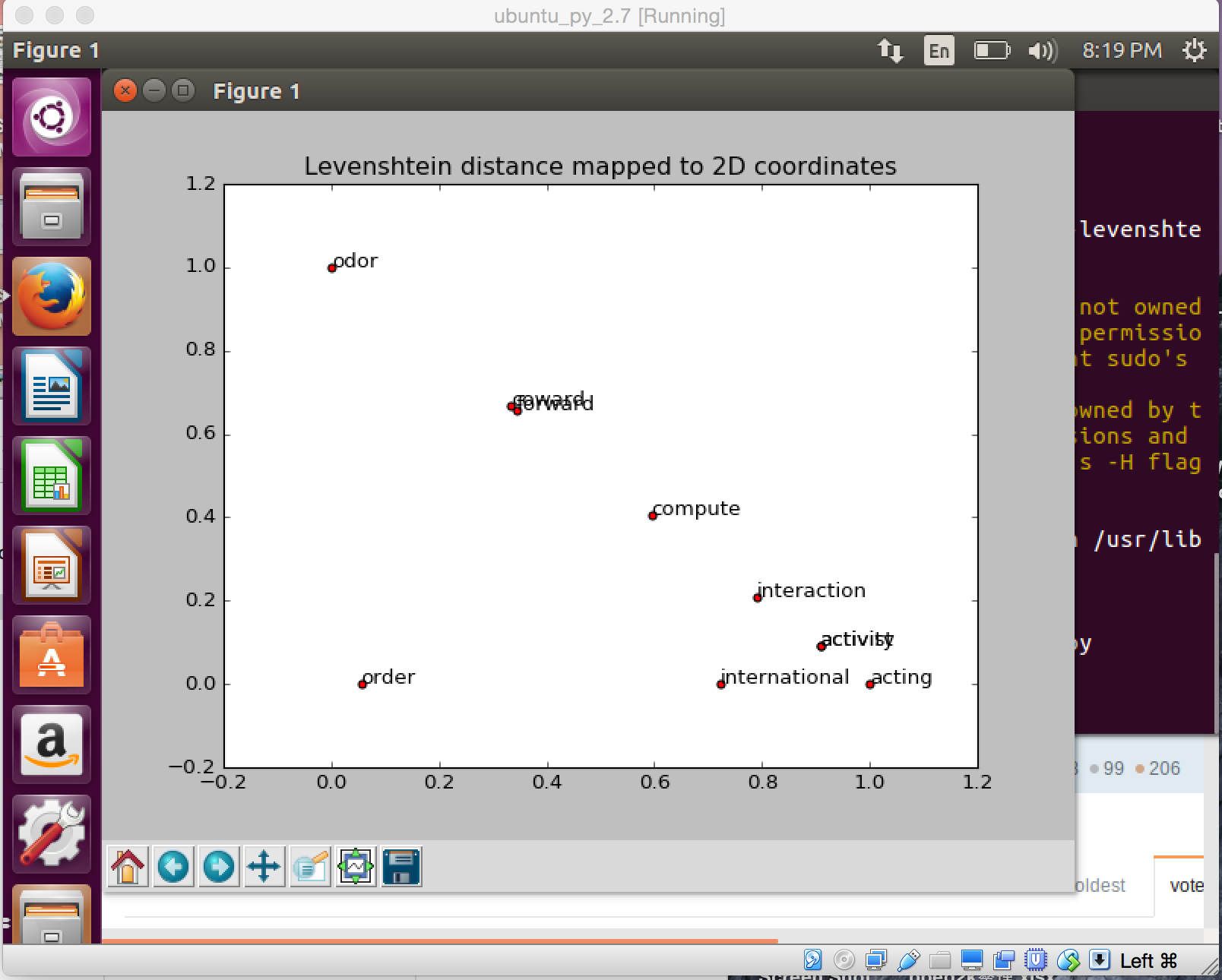
Because the function of python can return two different type of parameter, when we find the pair of farthest points, we also return the distance.

Part 2:

I find a script on the Internet deal with Levenshtein string distance using fastmap algorithm. <http://gromgull.net/blog/2009/08/fastmap-in-python/>

The advantage of the script is convenience because you just need to define the input the string list.

The disadvantage of the script is that: the output picture has text overlapping.



Part3:

For multimedia databases, FastMap algorithm can be used to determine the features for the given dataset, that is: mapping objects into points in k-d space, using k feature-extraction functions.

.