词语居间度密度计算：

1. 对于分词之后的的文本，得到 {词语：编码}
2. 计算词语之间的两两语义相似度,设定阈值b=0.66 返回语义相似度网络

语义相似度网络组成形式：

{

"词1": {"词2": "词1和词2的语义相似度",

"词3": "词1和词3的语义相似度",

...}

"词2": {"词1": "词2和词1的语义相似度",

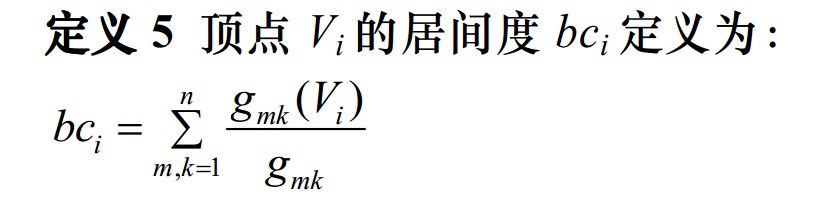
"词3": "词2和词3的语义相似度",

...}

…

}

1. 根据语义相似度网络计算每个词语的居间度



n 网络节点总数，即词语的个数。

m,k分别从1-n遍历，得到任意两个节点（词语）之间的最短路径（Dijkstra算法：求网络中一个节点到其他节点的最短距离） 得到一个字典。

shortestDatas={  
 'V1' : {  
 'V2' : {  
 'path': V1到V2的最短路径 ,  
 'distance': V1到V2的最短路径的距离  
 },  
 'V3' : {  
 #.......  
 }  
 #.......  
 },  
 V2 : {  
 ......  
 },  
 Vn : {  
 ......  
 }  
}

1. 求每个单词的居间度密度 {词语：居间度密度}

