将数据导入code.py

tf2dict=gentf2dict(idf)  
vec2dict=genvec2dict(tf2dict)

tf2dict与vec2dict均为以词为key，对应数值为value的字典，tf2为所有文档中每个词对应的tf（经过去低频词处理），vec2为tf\*idf处理后的数据

在多项query中循环，分别执行以上tf与vec相关操作

tf2dict=gentf2dict(idf)  
vec2dict=genvec2dict(tf2dict)

tf2dict与vec2dict均为以词为key，对应数值为value的字典，tf2为所有文档中每个词对应的tf（经过去低频词处理），vec2为tf\*idf处理后的数据

生成tf和tf\*idf的向量vec

生成idf

vs = genvs(res)

vs为文档中所有词构成的集合（不含重复）

qres=genqres()

data=gendata()  
res=genres()

qres为5个query分词后构成的二维字典，

形为{1:{a,b,c},2:{d,e,f},…}

Data为text.txt中每一行为一个元素构成的列表

res为由text.txt分词后形同qres的二维字典

生成vs

idftemp = genidf(res, vs)  
vs = filtervs(idftemp, vs)  
idf = filteridf(idftemp)

idf是经过去低频词处理的以词为key，频率为value的字典

idftemp为未做取倒数操作之前的数，也就是每个词出现的文档数

并在去低频词处理中去除了vs中的对应词项

注：category.py为构建TF-IDF索引文件，并保在文本文档中，方便再次运行使用，耗时约900s。run.py为读取TF-IDF索引文件，并输出结果文件，耗时1min以内。

gensim(vec1, res, simbase, c, vec2dict)

gensim作用为生成所有文档对query相似度的列表

通过json.dump写入json文件

生成文档和对应相似度的二维列表，并排序