开发环境部署

硬件环境：普通台式机或者笔记本一台，可以正常连网，配置不作特别要求

软件环境：win7以上系统，本机采用WIN10 64位系统

开发环境：Pycharm + Python3.6（Anaconda）

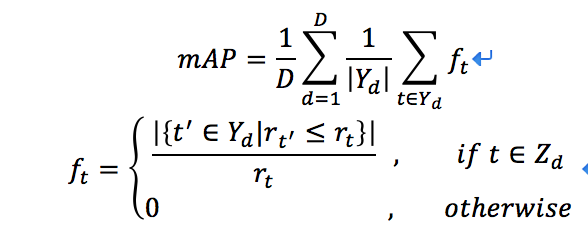
分词工具：采用结巴分词

需求分析：

财经新闻作为重要却海量的投资数据，无时无刻不在影响着投资者们的投资决策，为了更好地提示客户当下新闻事件对应的投资机会和投资风险，本课以研发“历史事件连连看”为目的，旨在根据当前新闻内容从历史事件中**搜索出相似新闻报道**，后期可以结合事件与行情，辅助客户采取相应投资策略。

 该赛题是让参赛者为**每一条测试集**数据寻找其**最相似的TOP 20条新闻**（不包含测试新闻本身），我们会根据参赛者提交的结果和实际的数据进行对比，采用mAP值作为评价指标。

该赛题是让参赛者为每一条测试集数据寻找其最相似的TOP 20条新闻，我们会根据参赛者提交的结果和实际的数据进行对比，采用mAP值作为评价指标，评分公式如下：



其中D表示测试集中新闻的总数量，Yd表示新闻d的n条真实相似新闻集合，无序，Yd={Yd1,Yd2,Yd3,……,Ydn }；Zd表示选手提交m条(赛题中m=20）相似新闻的有序集合Zd={Zd1,Zd2,Zd3,……,Zdm }；Zd中各元素的Rank值分别1,2,3,……,m,记为ri。对于集合K，|K|表示K中元素的个数,即|Yd |=n。

使用向量空间模型计算中文文本相似度的步骤：

（1）中文分词（Python的Jieba），就相当于英文使用天然的空格进行的分词操作。

（2）构建“词袋”（bag-of-words）。对于短文章来说，可能取要比较的两个文本的分词结果集合就行；但是对于长文章来说这显然是不现实的，那就需要使用 TF-IDF 来分别提取有限特征词来组成关键词库。

（3）以“词袋”为基础生成文本向量，一般是使用 TF-IDF 作为词在文档中的权重。

（4）通过计算文本向量之间的余弦来衡量相似度

具体为：

（1）使用TF-IDF算法，找出两篇文章的关键词；

（2）每篇文章各取出若干个关键词（比如20个），合并成一个集合，计算每篇文章对于这个集合中的词的词频（为了避免文章长度的差异，可以使用相对词频）；

（3）生成两篇文章各自的词频向量；

（4）计算两个向量的余弦相似度，值越大就表示越相似。