实验思路：

主要分两步：

1. 建立向量空间模型VSM

向量空间模型，可用于计算文本相似度。

1、分词，把20000个文档分成一个个单词，用到Textblob。在分词时去掉a,the等单词。

2、向量用权重表示维度。一个向量表示一个文本。

先初步建立一个字典，所有的词。字典筛选根据权重来筛选。设置一个权重的阈值，低于阈值的单词被过滤掉。首先我们希望将文章向量化。可以将一篇文章表示成从所有文章抽取的关键词及其权重所构成的向量。如何提取文章的关键词及其权重？ TF-IDF（词频-逆文档频率）

当然首先要对文档向量降维：去除文章中的停用词。然后分四步：

1、计算词频

计算每个单词的tf，单个文档中的某个单词出现次数/文档总单词数

2、计算逆文档频率

分母越大，IDF越小

计算每个单词的idf,log[语料库的文档总数/(包含该单词的文档数 + 1)]

3、计算权重

权重 weight = tf \* idf。

自动提取关键词的算法。计算每个词的TF-IDF值，按降序排列，取排在最前面的几个词。

4、对文章进行向量空间建模

对每个文档，用词典中的单词及其权重表示。

于是，每个文档都能被表示成维数为|n|的向量。实际每个文档向量非常稀疏。

二、KNN分类

如何比较文档之间的相似度？使用向量间的余弦值度量相似性。

按照8：2的比例划分训练集和测试集， 计算测试集样本与训练集每个样本的相似度，看相似度靠前的是哪几种类型，选择最多的那种作为训练集的分类。最后可以获得准确率。

在具体的代码实现上没做完，也无法运行。