1. 进行了情感分析（消极、中性、积极）；观点/标签聚类，得到标签权重，生成了词云图。

先分词，再tfidf和textrank结合选取关键词，最后根据关键词的权重绘制词云图。结合了tfidf和textrank，不仅考虑字频，且考虑词与词之间的关系，更为准确率，并进行了可视化观点提取方面：通过依存句法分析，提取重要的“动补结构”，“介宾关系‘，”动宾关系“，根据关键词来找到所在部分，后根据核心词提取观点。可以提取含多个关键词的观点，利用上述重要的关系将提取出的词拼接成观点。使用O(n^2)的方法分别以每个观点为中心判断和其他观点使用difflib的距离进行聚类，并最后根据最高出现次数的字拼接求和各观点的相似度，最相似的作为聚类的代表。

在这之前使用textcnn对情感分类，之后可以针对比如负面评论来提取观点，挖掘出导致负面评论因素。Textcnn使用了评论生成的二元组，三元组，四元组信息，输入到神经网络中，在情感分析方面都卓有成效，但过于复杂的任务则优势不明显。

-----------------------------------------------------------

##效果不好的聚类

对于句子聚类，使用gensim的doc2vec包直接转为句子向量，在用kmeans进行聚类时，效果不好，Silhouette Coefficient接近于0，

对应的是运行doc2vec\_gensim.py。模型保存到了d2v.model

说明了保留标签则是相当重要的

###使用tfidf的AF聚类算法

AP算法不需要先确定聚类的数目，而是把所有的数据点都看成潜在意义上

的聚类中心（exemplar）。这里通过词向量和词向量对应的tf-idf 值构建文本向量表示，文本之间的相似性通过两个文本向

量的余弦相似性得到。最后通过AP聚类算法得到观点数量，并从观点中筛选典型评论

>python tfidf\_ap.py

AP算法虽然不需要指定聚类数量，但由于算法复杂度高，较慢，且有时并不一定会收敛。在此数据集上表现不佳。依赖于每个句子的向量化表示。