NLP Technology Report

目前国际上大多question answer system都是采用结构化语句进行解析，也就是问题比较符合who, when, where, what and how (4W1H)格式

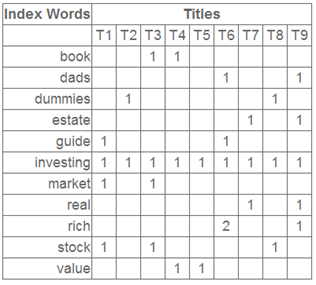
## Latent Semantic Analysis(LSA), 同LSI

* 1. LSA包含分词，TF-IDF, Singular Value Decomposition(SVD)

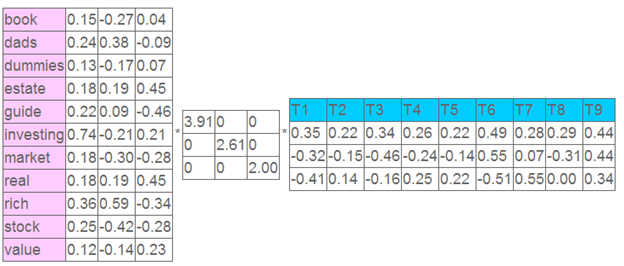
潜在语义索引（Latent Semantic Indexing），至少不是实现了SVD就可以直接用的，不过LSI也是一个严重依赖于SVD的算法，之前吴军老师在矩阵计算与文本处理中的分类问题中谈到：

“三个矩阵有非常清楚的物理含义。第一个矩阵X中的每一行表示意思相关的一类词，其中的每个非零元素表示这类词中每个词的重要性（或者说相关性），数值越大越相关。最后一个矩阵Y中的每一列表示同一主题一类文章，其中每个元素表示这类文章中每篇文章的相关性。中间的矩阵则表示类词和文章雷之间的相关性。因此，我们只要对关联矩阵A进行一次奇异值分解，w 我们就可以同时完成了近义词分类和文章的分类。（同时得到每类文章和每类词的相关性）。”

上面这段话可能不太容易理解，不过这就是LSI的精髓内容，我下面举一个例子来说明一下，下面的例子来自LSA tutorial，具体的网址我将在最后的引用中给出：

[](http://images.cnblogs.com/cnblogs_com/LeftNotEasy/201101/201101192226379109.png)

这就是一个矩阵，不过不太一样的是，这里的一行表示一个词在哪些title中出现了（一行就是之前说的一维feature），一列表示一个title中有哪些词，（这个矩阵其实是我们之前说的那种一行是一个sample的形式的一种转置，这个会使得我们的左右奇异向量的意义产生变化，但是不会影响我们计算的过程）。比如说T1这个title中就有guide、investing、market、stock四个词，各出现了一次，我们将这个矩阵进行SVD，得到下面的矩阵：

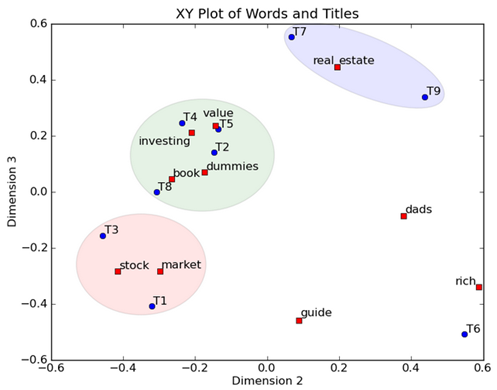
[](http://images.cnblogs.com/cnblogs_com/LeftNotEasy/201101/201101192226389491.png)

左奇异向量表示词的一些特性，右奇异向量表示文档的一些特性，中间的奇异值矩阵表示左奇异向量的一行与右奇异向量的一列的重要程序，数字越大越重要。

继续看这个矩阵还可以发现一些有意思的东西，首先，左奇异向量的第一列表示每一个词的出现频繁程度，虽然不是线性的，但是可以认为是一个大概的描述，比如book是0.15对应文档中出现的2次，investing是0.74对应了文档中出现了9次，rich是0.36对应文档中出现了3次；

其次，右奇异向量中一的第一行表示每一篇文档中的出现词的个数的近似，比如说，T6是0.49，出现了5个词，T2是0.22，出现了2个词。

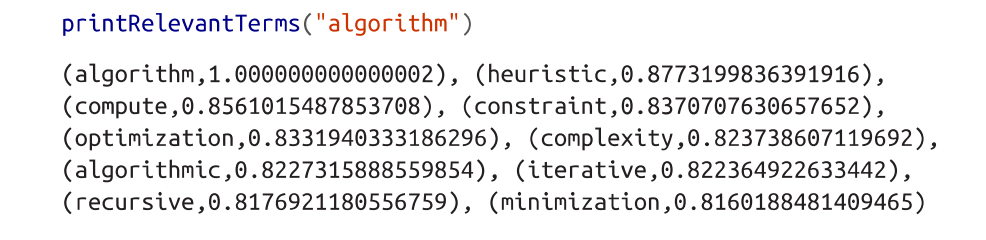
然后我们反过头来看，我们可以将左奇异向量和右奇异向量都取后2维（之前是3维的矩阵），投影到一个平面上，可以得到：

[](http://images.cnblogs.com/cnblogs_com/LeftNotEasy/201101/20110119222640769.png)

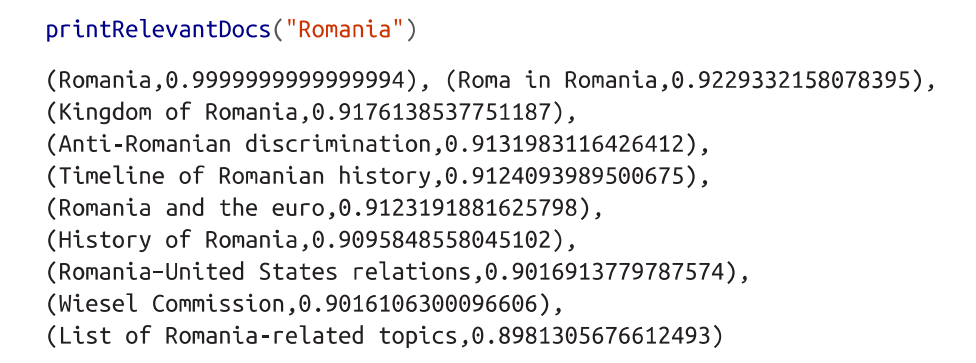
在图上，每一个红色的点，都表示一个词，每一个蓝色的点，都表示一篇文档，这样我们可以对这些词和文档进行聚类，比如说stock 和 market可以放在一类，因为他们老是出现在一起，real和estate可以放在一类，dads，guide这种词就看起来有点孤立了，我们就不对他们进行合并了。按这样聚类出现的效果，可以提取文档集合中的近义词，这样当用户检索文档的时候，是用语义级别（近义词集合）去检索了，而不是之前的词的级别。这样一减少我们的检索、存储量，因为这样压缩的文档集合和PCA是异曲同工的，二可以提高我们的用户体验，用户输入一个词，我们可以在这个词的近义词的集合中去找，这是传统的索引无法做到的。

通过上述生成svd的矩阵，可以实现如下几种匹配模式，

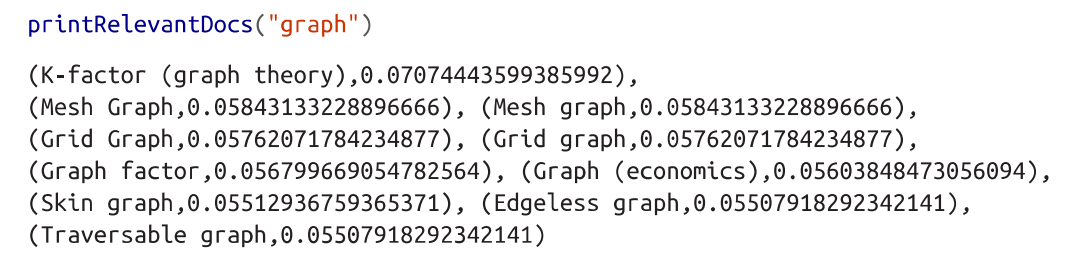
* Term-Term Relevance



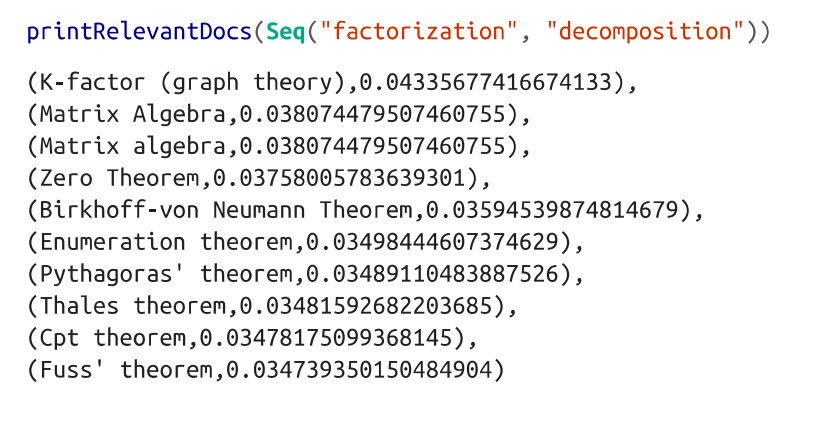
* Document-Document Relevance



* Term-Document Relevance



* Multiple-Term Queries



## Multiple Classification算法分析

* 1. LogisticRegressionWithLBFGS

Split data into training (80%) and test (20%).

准确度：38.5%

* 1. Random Forest

Split data into training (80%) and test (20%).

100棵树，10层，准确度：18.42

500棵树，10层，准确度：19.28

1000棵树，10层，准确度：19.57

随机森林对多分类支持够好，增加棵树和层级提升有限。

* 1. NaiveBayes

Split data into training (80%) and test (20%).

准确度：11.3%

朴素贝叶斯要求字段间没有相关性，相互对立，才能形成好的计算结果，从分析和实验结果看，完全不适合chatrobot的场景。

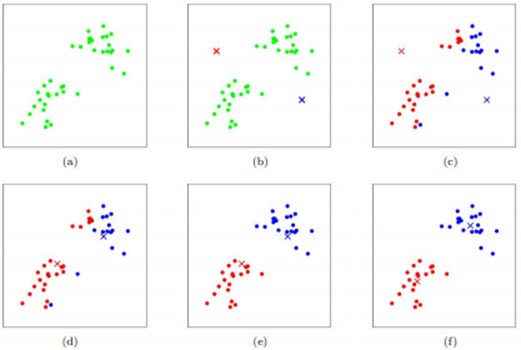
* 1. KNN(K-Nearest Neighbor)待验证。

### Cluster算法验证

系统和用户进行多个问题交互，逐步缩小范围找到正确，需要对问题进行归并，设计聚类算法的应用。

* 1. KMeans算法

K-means算法是聚类分析中使用最广泛的算法之一。它把n个对象根据他们的属性分为k个聚类以便使得所获得的聚类满足：同一聚类中的对象相似度较高；而不同聚类中的对象相似度较小。其聚类过程可以用下图表示：

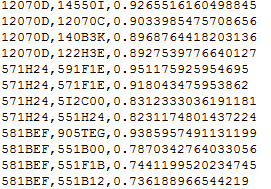


K-means简单易用，每个群的主题很难确认，在我们的应用过程，生成的每个群都包含多个问题类别，不能形成明显的标签，对问题分类进行合并操作。

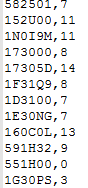
* 1. PowerIterationClustering

Power iteration clustering (PIC) is a scalable and efficient algorithm for clustering vertices of a graph given pairwise similarties as edge properties, described in Lin and Cohen, Power Iteration Clustering. It computes a pseudo-eigenvector of the normalized affinity matrix of the graph via power iteration and uses it to cluster。

应用过程，利用LSA组件的Document-Document Relevance功能，生成所有问题分类的top4相关度，样例如下：



利用问题分类直接的相似度进行聚类，生成k个群。如下样例，列1是问题分类，列2是归属的群号。



* 1. 自动分组后进行分类验证分组效果

用分群ID替换原有分类ID，作为chatrobot的第一层问答，客户在选择进一步提供信息后，对选择的群组内进行子集的分类判定，由于子集包含的分类数明显降低，关键点就存在于第一层的准确度。

验证过程采用全量和子集两种数据进行，最高的验证结果如下图，详细结果参见附件



综述，目前实行的流程简图如下：

## 杂项尝试

Keywords 提取

爬去kb的描述信息，对描述信息做keyword提取操作，提取出2k个keyword

分支工作流

提取2015年gcc的kb信息，抽取出URL和category的对应关系

提供category对应的sn信息，利用sn进行聚类计算

## 未来可尝试方法路径：

* 1. Deep Learning in Natural Language Processing

斯坦福大学的研究是领先的，论文也是比较多，目前比较好的结果是298个问题，top5命中176个(论文里面的最好成绩)，第一，相比较chatrobot，Stanford的问题是完全开放的，换成chatrobot效果会有所提高；第二，Stanford的算法复杂度高，而且需要很大的词库，实施起来难度很大，如果采用普通的deep learning 效果相比Stanford的改良版效果会差，最终的准确度不一定比分类高。

* 1. IBM Watson
* 光环：

代表着IBM在认知计算领域最核心的技术Watson（沃森）的一战成名，是在2011年2月的美国问答节目《Jeopardy！》上。在这次节目中，Watson战胜了这一节目的两位冠军选手，这被和1996年同样来自IBM的“深蓝”战胜国际象棋大师卡斯帕罗夫相提并论，被认为是人工智能历史上的一个里程碑

* 弊端：

watson回答的很多的问题，本身都比较适合机器回答的，比如回答某个人，某个地点之类的。这些只要初步分析语法，用搜索技术就可以限制范围。所以不能算是太大突破，真正的图片，是让机器人理解自然语言，也就是让机器能主动的运用学到的东西。比如，是否能用来解决小学应用题呢？这个恐怕就不能完全靠记忆了，必须理解和推理真正的结合

