信息检索导论--
lucene

授课人：李永可
邮箱：lyk@xjau.edu.cn

相关性

- 如何计算文档和查询语句的相关性呢？
- 不如我们把查询语句看作一片短小的文档，对文档与文档之间的相关性(relevance)进行打分(scoring)，分数高的相关性好，就应该排在前面。
- 那么又怎么对文档之间的关系进行打分呢？

相关性

- 对文档打分不是一件容易的事情。
- **首先**看一个人，往往有很多**要素**，如性格，信仰，爱好，衣着，高矮，胖瘦等等。
- **其次**对于人与人之间的关系，**不同的要素重要性不同**，性格，信仰，爱好可能重要些，衣着，高矮，胖瘦可能就不那么重要了，所以具有相同或相似性格，信仰，爱好的人比较容易成为好的朋友，然而衣着，高矮，胖瘦不同的人，也可以成为好的朋友。

相关性

- 因而判断人与人之间的关系，**首先要找出哪些要素对人与人之间的关系最重要**，比如性格，信仰，爱好。
- **其次要判断两个人的这些要素之间的关系**，比如一个人性格开朗，另一个人性格外向，一个人信仰佛教，另一个信仰伊斯兰教，一个人爱好打篮球，另一个爱好踢足球。我们发现，两个人在性格方面都很积极，信仰方面都很善良，爱好方面都爱运动，因而两个人关系应该会很好。

相关性

- **我们再来看看公司之间的关系。**
- **首先**看一个公司，有很多人组成，如总经理，经理，首席技术官，普通员工，保安，门卫等。

相关性

- **其次对于公司与公司之间的关系，不同的人重要性不同**，总经理，经理，首席技术官可能更重要一些，普通员工，保安，门卫可能较不重要一点。所以如果两个公司总经理，经理，首席技术官之间关系比较好，两个公司容易有比较好的关系。然而一位普通员工就算与另一家公司的一位普通员工有血海深仇，怕也难影响两个公司之间的关系。

相关性

- 因而判断公司与公司之间的关系，**首先要找出哪些人对公司与公司之间的关系最重要**，比如总经理，经理，首席技术官。**其次要判断这些人之间的关系**，不如两家公司的总经理曾经是同学，经理是老乡，首席技术官曾是创业伙伴。我们发现，两家公司无论总经理，经理，首席技术官，关系都很好，因而两家公司关系应该会很好。

相关性

- 分析了两种关系，下面看一下**如何判断文档之间的关系**了。
- **首先，一个文档有很多词(Term)组成**，如search, lucene, full-text, this, a, what等。
- **其次对于文档之间的关系，不同的Term重要性不同**，比如对于本篇文档，search, Lucene, full-text就相对重要一些，this, a , what可能相对不重要一些。所以如果两篇文档都包含search, Lucene，fulltext，这两篇文档的相关性好一些，然而就算一篇文档包含this, a, what，另一篇文档不包含this, a, what，也不能影响两篇文档的相关性。

相关性

- 因而判断文档之间的关系，首先找出哪些词(Term)对文档之间的关系最重要，如search, Lucene, fulltext。然后判断这些词(Term)之间的关系。

权重

- **找出词(Term)对文档的重要性的过程称为计算词的权重(Term weight)的过程。**
- 计算词的权重(term weight)有两个参数，第一个是词(Term)，第二个是文档(Document)。
- 词的权重(Term weight)表示此词(Term)在此文档中的重要程度，越重要的词(Term)有越大的权重(Term weight)，因而在计算文档之间的相关性中将发挥更大的作用。

向量空间模型（vsm）

- **判断词(Term)之间的关系从而得到文档相关性的过程应用一种叫做向量空间模型的算法(Vector Space Model)。**

向量空间模型（vsm）

1. 计算权重(Term weight)的过程。

影响一个词(Term)在一篇文档中的重要性主要有两个因素：

- Term Frequency (tf)：即此Term在此文档中出现了多少次。tf 越大说明越重要。
- Document Frequency (df)：即有多少文档包含次Term。df 越大说明越不重要。

向量空间模型（vsm）

- 容易理解吗？词(Term)在文档中出现的次数越多，说明此词(Term)对该文档越重要，如“搜索”这个词，在本文档中出现的次数很多，说明本文档主要就是讲这方面的事的。然而在一篇英语文档中，this出现的次数更多，就说明越重要吗？不是的，这是由第二个因素进行调整，第二个因素说明，有越多的文档包含此词(Term), 说明此词(Term)太普通，不足以区分这些文档，因而重要性越低。

向量空间模型（vsm）

- 这也如我们程序员所学的技术，对于程序员本身来说，这项技术掌握越深越好（掌握越深说明花时间看的越多，tf越大），找工作时越有竞争力。然而对于所有程序员来说，这项技术懂得的人越少越好（懂得的人少df小），找工作越有竞争力。人的价值在于不可替代性就是这个道理。

权重公式

$$w_{t,d} = tf_{t,d} \times \log(n / df_t)$$

$w_{t,d}$ = the weight of the term t in document d

$tf_{t,d}$ = frequency of term t in document d

n = total number of documents

df_t = the number of documents that contain term t

这仅仅只term weight计算公式的简单典型实现。实现全文检索系统的人会有自己的实现，Lucene就与此稍有不同。

vsm

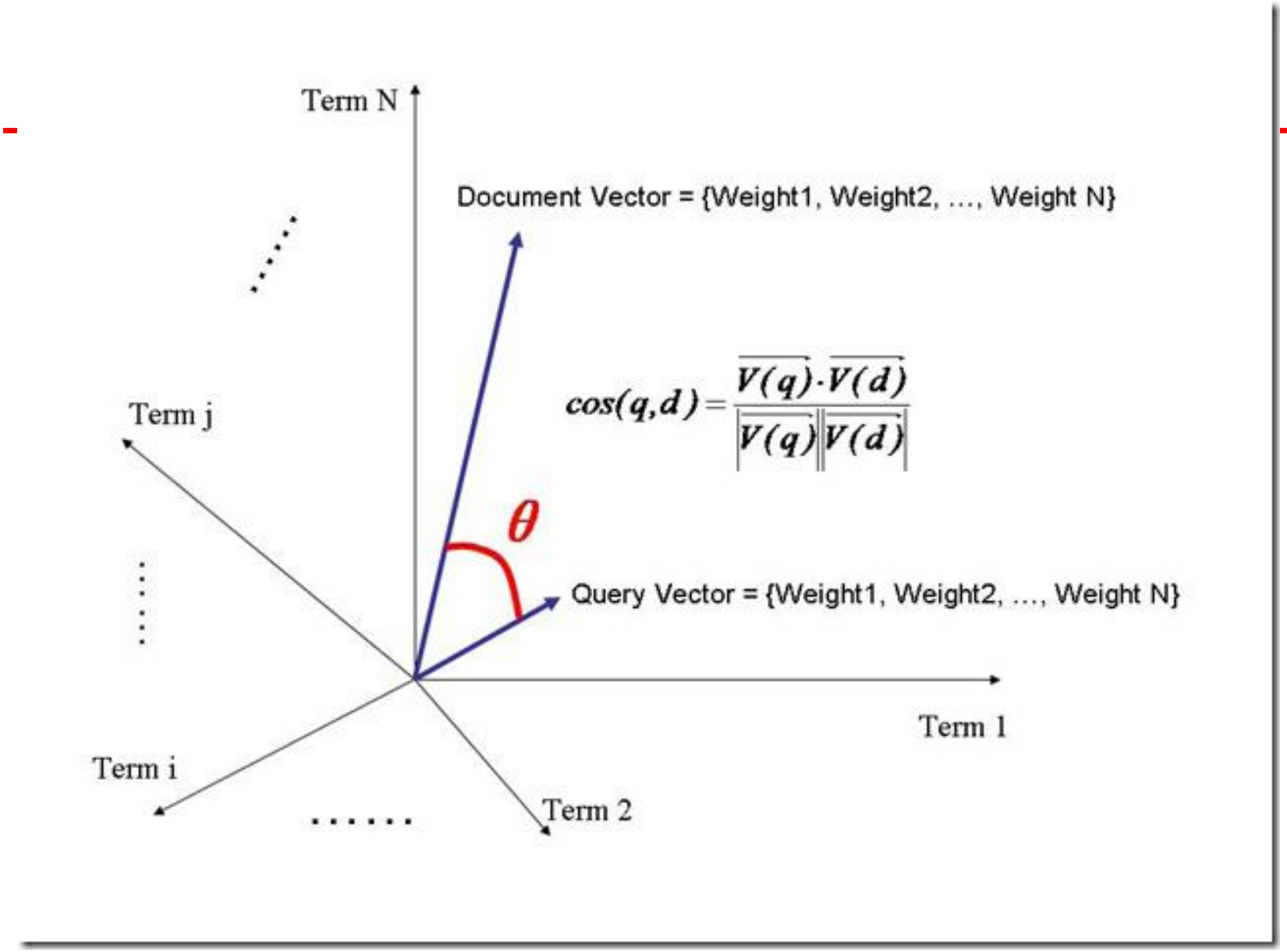
- **2. 判断Term之间的关系从而得到文档相关性的过程，也即向量空间模型的算法(VSM)。**

vsm

- 我们把文档看作一系列词(Term)，每一个词(Term)都有一个权重(Term weight)，不同的词(Term)根据自己在文档中的权重来影响文档相关性的打分计算。于是我们把所有此文档中词(term)的权重(term weight) 看作一个向量。
- Document = {term1, term2, …… ,term N}
- Document Vector = {weight1,weight2, …… ,weight N}

vsm

- 我们把所有搜索出的文档向量及查询向量放到一个N维空间中，每个词(term)是一维。


Term N
Document Vector = {Weight1, Weight2, …, Weight N}
Term j
$cos(q,d) = \frac{\overrightarrow{V(q)} \cdot \overrightarrow{V(d)}}{\left|\overrightarrow{V(q)}\right| \left|\overrightarrow{V(d)}\right|}$
θ
Query Vector = {Weight1, Weight2, …, Weight N}
Term 1
Term i
Term 2

相关性打分公式

$$score(q,d) = \frac{\vec{V}_q \cdot \vec{V}_d}{|\vec{V}_q||\vec{V}_d|} = \frac{\sum_{i=1}^{n} w_{i,q} w_{i,d}}{\sqrt{\sum_{i=1}^{n} w_{i,q}^2}\sqrt{\sum_{i=1}^{n} w_{i,d}^2}}$$

相关性打分

- 举个例子，查询语句有11个Term，共有三篇文档搜索出来。其中各自的权重(Term weight)，如下表格。

	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11
D1	0	0	.477	0	.477	.176	0	0	0	.176	0
D2	0	.176	0	.477	0	0	0	0	.954	0	.176
D3	0	.176	0	0	0	.176	0	0	0	.176	.176
Q	0	0	0	0	0	.176	0	0	.477	0	.176

相关性打分

- 于是计算，三篇文档同查询语句的相关性打分分别为：

$$SC(Q,D_1)=\frac{(0.176)(0.176)}{\sqrt{0.477^2+0.477^2+0.176^2+0.176^2}\sqrt{0.176^2+0.477^2+0.176^2}}\approx 0.08$$

$$SC(Q,D_2)=\frac{(0.954)(0.477)+(0.176)^2}{\sqrt{0.176^2+0.477^2+0.954^2+0.176^2}\sqrt{0.176^2+0.477^2+0.176^2}}\approx 0.825$$

$$SC(Q,D_3)=\frac{(0.176)^2+(0.176)^2}{\sqrt{0.176^2+0.176^2+0.176^2+0.176^2}\sqrt{0.176^2+0.477^2+0.176^2}}\approx 0.327$$

总结

- 在进入Lucene之前，对上述索引创建和搜索过程所一个总结，如图：