Another Belief Network Model for Information Retrieval

Author：张智卓　　StudentID: 200433101321　　Class: 2004Bilingual

1. Introduction

2006 年的信息检索实验任务概括如下：

在给定的约 1200 个网页文本和 10 个查询，要求在给定网页集中找出各个查询的最相关前 10 个网页文档。

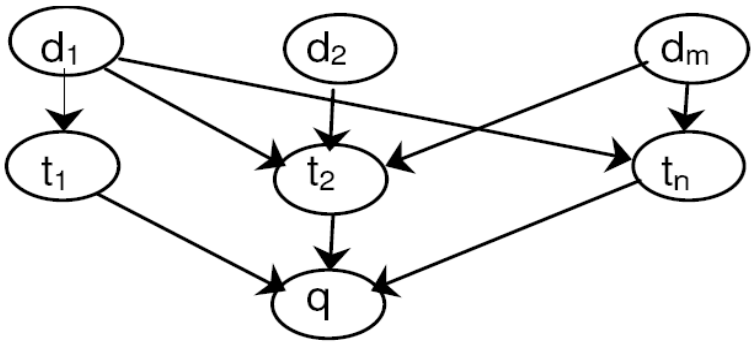
而这份实验报告主要阐述我在解决上述实验问题所用到的 Belief Network 模型和方法。

Belief Network 模型属于图模型[Jordan]中的一种.图模型作为一种表达不同因素之间关系的形式化手段，具有很强的灵活性和把问题清晰地形象化。而用于 IR 模型的图模型主要有两类，分别是 Inference Network 模型 [Fung95], [Turtle90]和 Belief Networks 模型[Ribeiro-Neto96] [Silva2000]。实际上，在 IR 领域中 Belief Networks 模型比 Inference Network 模型表达的内容更广泛，两个模型的关键区别就是在 Belief Networks 模型用到的是 *p(dj|t)*（*t 是索引词，dj 是文档*）。*相反地，*Inference Network 模型用的是 *p(t|dj)，在给定 dj 的情况下各个 t 是独立，这样就无法表示类似向量中 cosine 的方法求相关度。（详细请看[Ribeiro96]）。在* Belief Networks 模型中，每个索引词项 t 表达为“观念“空间中的一个基本“观念”，而查询语句和文档表达为某种“观念“（由基本观念所影

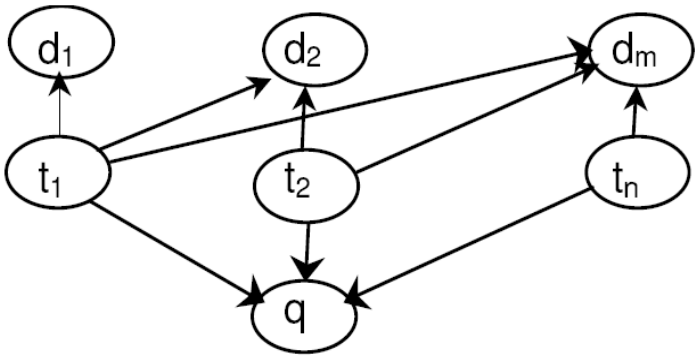
响），值得注意的是在 Belief Networks 中查询语句和文档的地位是等价的，都被映射到“观念”空间中，但在 Inference Network 中它们地位就不是等价了。

这份报告中我将介绍我自己根据具体任务环境设计的另一种 Belief Networks 模型（我称它为 ABN，即 Another Belief Network），在这个模型中查询与文档之间地位不在等价而有直接的联系。下面报告将分为模型介绍，实现过程，实验结果三部分。

Inference Network



Belief Network



2. *My belief Network Model*

● *2.1Task analysis*

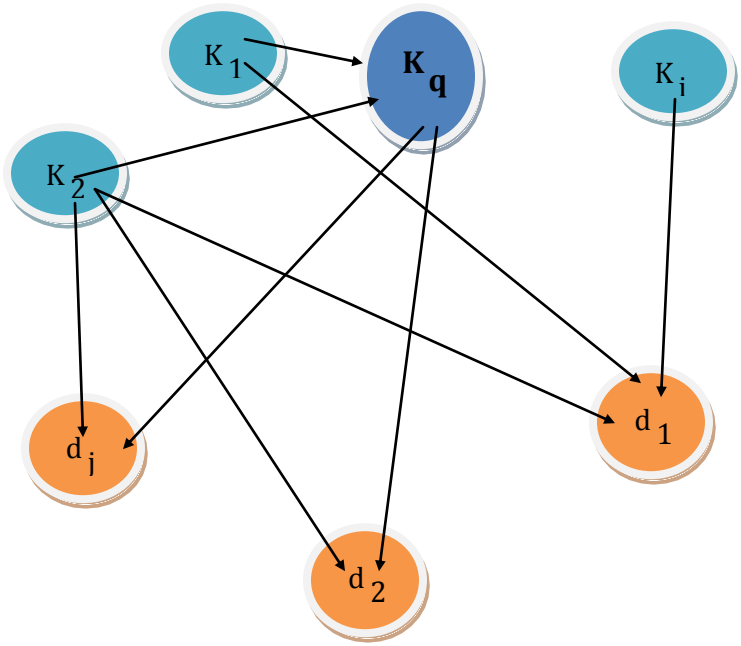
观察我们的查询词，除了 TD174 和 TD197 外，其余都是四字词语。更进一步观察，

TD148：中国历史
TD158：建筑艺术
TD162：电子政务
TD164：北京奥运
TD174：奥斯卡
TD187：中国油画
TD190：中国武术
TD197：消费者权益
TD200：汽车改装
TD213：医疗保险

这些四字词语的都分别是有两个短词组成的，例如“中国历史“由”中国“跟”历史“组成。在传统的 Belief Networks 模型中把查询语句也看作文档一样看待，但在这里我们的直觉更会乐意认为这些四个字的查询更偏向一个词而不是文档，而这个就是我想建立另一种 Belief Networks 模型来描述它们的动机。

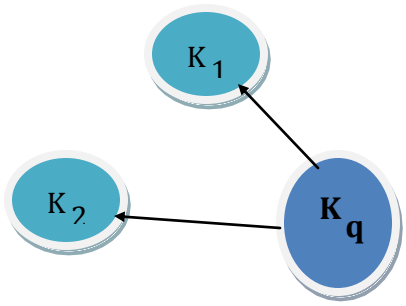
● *2.2ABN Model description*

我建立的 Belief Network 如下图，可以看到查询 q 已经不在与文档 d 有同样的地位，而变成为一个索引项。



另一要指出的就是有两个索引项也指向 Kq，具体例子如 Kq=“中国历史“,那么 K1=”中国”，K2=”历史“。当然如果 Kq=”奥斯卡”的话就没有其他索引项指向它了，这里我假设 Kq 只依赖于它的子词语（虽然你可能会举很多例子来驳斥我这个假设，但为了简化模型我就这样假设）。Kq 依赖它的子词语这个其实很直观，我们可以用 Belief Network 的核心思想就是观念空间来解释。比如“中国历史”这个观念，当“中国”的观念发生了变化（如台湾回归了），“中国历史”这个观念肯定也会相应变化。

这个网络中的箭头方向可能也要解释。为何是 K1 指向 Kq 而不是，Kq 指向 K1 呢？这个是一个关于分布的独立性的问题，如果是 Kq 指向 K1，K2，那么就意味着给定 Kq 下 K1，K2 的概率分布是相互独立的，但这个明显不是。（有关 Bayesian Network 的 d-seperation 详细说明可以参考[Bouckaert 92]）。



- ***2.3ABN Model Evalutation***

下面讲述这个模型的计算相关度的方法：

d_j 为第 j 个文档，K_q为查询词（假设为由两个短词组成的 4 字词语，后面会补充其他情况）。那么概率 P（$d_j|K_q$）~P（d_j,K_q）(正比于联合概率)

$$P（d_j,K_q）=\sum_{\forall \mathbf{k}} P(d_j|K_q,\mathbf{k}) \times P(K_q|\mathbf{k}) \times P(\mathbf{k}) \qquad （1）$$

(**k** 为索引词向量)

因为前面假设了除子词（后面用 K1，K2 代替 Kq 的子词）外，其他索引索引词独立于 Kq，所以（1）式中的 k 可以看成一个 3 维向量（K1，K2，Kq）。又因为根据贝叶斯公式：

$$P(\mathbf{K}|Kq)=\frac{P(Kq \mid \mathbf{k})\times P(\mathbf{k})}{P(Kq)} \qquad （2）$$

因为 P(Kq)对所有文档都一样，所以可以舍掉。

把（2）式代入（1）得：

$$P（d_j,K_q）=\sum_{\forall k} P(d_j|K_q,\mathbf{k}) \times P(\mathbf{k}|K_q) \qquad （3）$$

而我们已经知道 Kq 存在，所以其实索引向量 k＝（X，X，1）只有 4 种可能（001，011，101，111）。我将概率$P(k|K_q)$和$P(d_j|K_q,k)$用如下方法计算：

定义P(k|Kq)为在包含索引词 Kq 的文档集中 k 出现的频率（即是最大似然估计）。举“中国历史”为例：

$P(001|K_q)$	在包含“中国历史”的文档中，没有出现“中国”或“历史”的文档比例
$P(011|K_q)$	在包含“中国历史”的文档中，没有出现“中国”但出现“历史”的文档比例
$P(101|K_q)$	在包含“中国历史”的文档中，没有出现“历史”但出现“中国”的文档比例
$P(111|K_q)$	在包含“中国历史”的文档中，既出现“历史”又出现“中国”的文档比例

接着我定义$P(d_j|K_q,k)$如下：

$$P(d_j|K_q,k)=\sum g_i(k)P'(d_j|k_i) \qquad (4)$$

就是说如果计算 P（d_j|101）的话，就是

P（d_j|101）＝P’（d_j|K1）+P’（d_j|Kq）；

进而定义 P’（$d_j|k_i$），（注：这里定义 P’()是为了不跟前面的 P()概率混淆，p’()是自己定义的一种计算$P(d_j|K_q,k)$方法，跟 Bayesian Network 的概率表达无关）

$$P'(d_j|k_i)=\begin{cases}\dfrac{w_{i,j}}{\sqrt{\sum_{i=0}^{t} w_{i,j}^{2}}} & \text{如果} g_i(d_j)=1 \\ 0 & \text{其他}\end{cases} \qquad (5)$$

这里的$w_{i,j}$为索引项k_i在文档d_j权值，具体算法由特定的权值函数给出与 Belief network 模型无关，这里我采用 tf-idf 的计算权值的方法。

以上就是 ABN 模型的一般计算方法。当K_q没有子词的时候（例如查询“禽流感”），这是（3）式将退化为（5），这样就跟传统的向量模型等价了。而当K_q有 2 个以上的子词时（如“中华人民共和国”），上述公式同样适用，只是向量 **k** 变为 4 维向量，即有 8 种可能的表达。如果有多个查询词，就分别对其求 P（d_j,K_q）的值然后求和，其结果就是这个查询跟文档d_j相关性。

3. ABN model implementation

3.1Word segmentation

分词的工作是我们小组的另一位同学负责（他采用的是根据词典的机械分词），我根据 ABN 模型对分词提出了下面的要求：

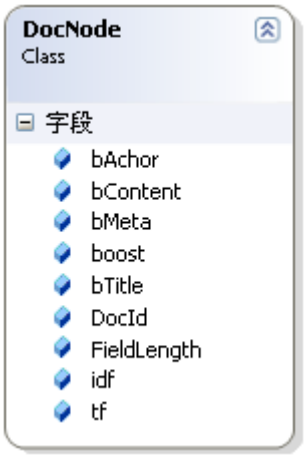
1) 把给出的 10 查询语句中的 8 个四字查询词添加到分词的词典中，也就是当遇到“中国历史”时不会分为“中国”，“历史”两个词，也就是最大匹配原则。

2) 把四字词的子词也添加到分词词典，这样保证遇到“中国”等的单个子词出现时可以正确提取。

3) 分词时需要给出每个词出现的域，例如网页标题、网页自报关键字，网页 Anchor，或者正文。

3.2 Data storage

建立索引数据库的工作也是我们小组的一位同学负责，包括前向索引（即通过文档 ID 索取文档出现的词项）和倒排索引（通过索引词查询出现该词项的文档链表）。在索引数据库中记录最基本数据结构如下（也就是通过指定索引词和文档 ID 后可以查到的信息）



带“b”开头的是 bool 类型，指明该索引词是否在该文档的特定域出现；

DocId 是文档 ID；

Tf 是该索引词在 DocID 文档中的词频（已经归一化）；

Idf 是该索引词的 idf 值（已归一化）；

FieldLength是该文档的索引词种类数；

Boost 是该文档的权重。

计算（5）中$w_{i,j}$权值的方法如下，参考 lucene 权值的计算方法

$$w_{i,j}=\frac{tf_{i,j}\times idf_i\times boost_j\times fieldBoost_{i,j}}{\sqrt{FieldLength}} \quad (6)$$

上式中的$fieldBoost_{i,j}$ 为该词出现的域的权值的乘积。例如该词在文档的 Title 和 Meta，但不在正文出现，假设 Title，Meta 和非正文的权值为 4，1.5，0.1，那么 $fieldBoost_{i,j}=4\times 1.5\times 0.1=0.6$，我们可以发现虽然出现在标题和自报关键字中但出来结果却少于 1，因为这是给它不出现在正文的惩罚。（我们下面可以看到在给出的 1200 多个网页文档集中有很多这样的虚假情况，而有些情况是出乎我当初所料的）。

3.3Standard Vector Model with query extension

为了跟 ABN 模型作比较，我还编写向量模型，但普通的向量模型在这样的分词条件下显得不太公平，因为如果查询“中国历史”，那么倒排文档就不会把没有出现“中国历史”但出现“中国”或“历史”的文档取出来，因此这里需要做简单的查询扩展（ABN 无需查询扩展）。扩展方法如下：

1) 如果出现“中国历史”等复合词，则扩展为“中国”，“历史”，“中国历史“，它们的权值为 0.5，0.5，1。
2) 权值是由所占的词长确定的，例如查询“张家界旅游“，”张家界“的权值为 0.6，而” 旅游“的权值为 0.4。更复杂的例子是“中华人民共和国旅游”，扩展后为（“中华”，$\frac{2}{9}$），（“人民”，$\frac{2}{9}$），（“共和国”，$\frac{3}{9}$），（“中华人民共和国”，$\frac{7}{9}$），（“旅游”，$\frac{2}{9}$）。

那么计算相关度的表达式就是

$$sim(d, q) = \sum w_{i,j} \times queryweight_i \qquad (7)$$

3.3Parameter optimization

当我一切计算方法的准备好的时候，我还觉得有些地方定义得比较主观，例如不同域的权值。我初始定义如下（参考 Nutch）

Title	4
Meta	2
Anchor	1.5
Not Content	0.1

而我手头上有 2005 年的数据和参考答案，因此我决定想办法利用它来优化我的参数。这里的优化方案其实是搜索，以获得较好 R－Precision 为目标，搜索一组较优的参数配搭。我用了两种方法分别是 SA(模拟退火)和 GA(遗传算法+局部爬山),最后获得 10 个查询词的平均 R－Precision 为 90.8%（初始为 87.3%）。对应的权值参数为：

Title	10.376814630006995
Meta	0.18018413080918758
Anchor	8.0091272971486749
Not Content	0.57691624986511747

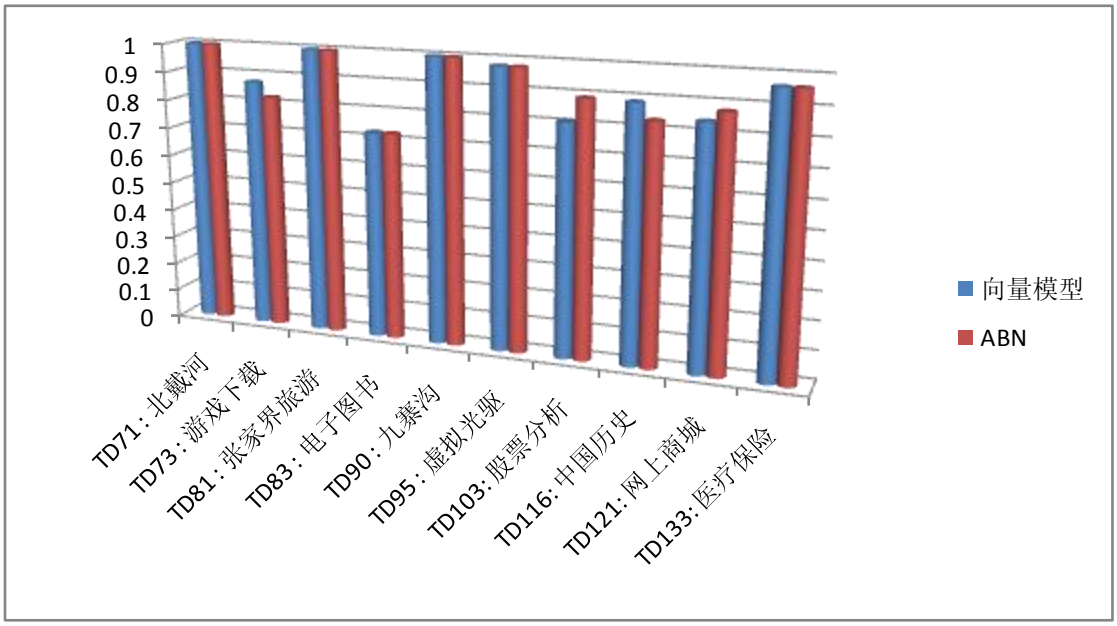
可以看到 Meta 变为 0.18 说明用自报关键字的方法做假的现象十分严重，而 Anchor 的权值提高到 8 或者可以说明一个网页的连出链接可以反应该网页的主题。Title 仍然保持最高的权值。

4. Experiments

我用 ABN 模型跟带查询扩展的向量模型利用 2005 的数据进行了比较，结果如下：

查询词	向量模型	R-Precision	ABN 模型	R-Precision	R
TD71：北戴河	9	1	9	1	9
TD73：游戏下载	70	0.875	66	0.825	80
TD81：张家界旅游	4	1	4	1	4
TD83：电子图书	8	0.727	8	0.7272	11
TD90：九寨沟	15	1	15	1	15
TD103: 股票分析	60	0.810	66	0.891	74
TD116: 中国历史	16	0.888	15	0.833	18
TD121: 网上商城	48	0.842	50	0.877	57
TD133: 医疗保险	27	0.964	27	0.964	28
总体平均准确率		90.83%		90.99%	

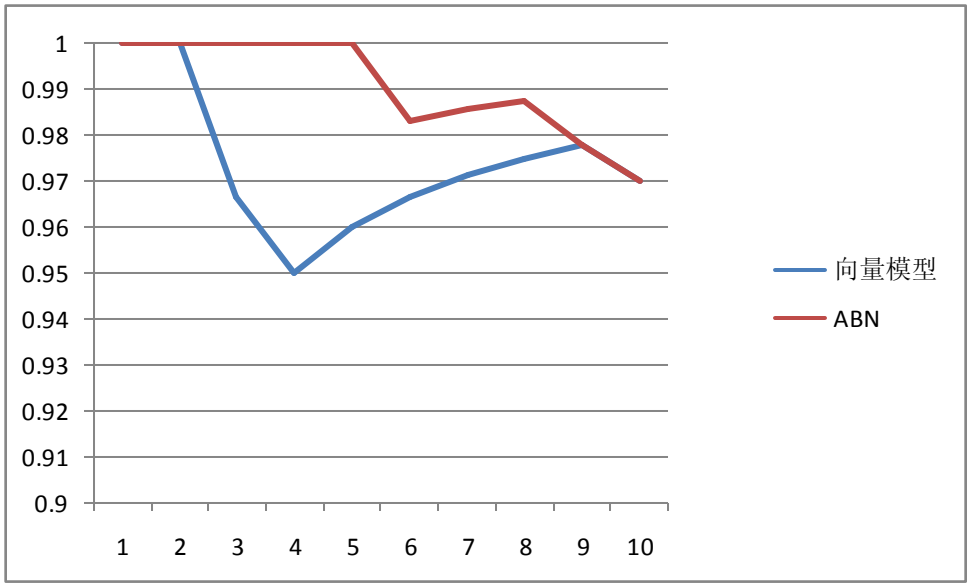
这样看起来两者的性能十分接近。(R 为每个查询的相关文档个数)



但当我们的目标是前几位的 Precision 时，情况就有点不同了，下面是两个模型排名前 10 个文档的 Precision 实验结果：

向量模型	文档数目	ABN 模型
1	1	1
1	2	1
0.966666667	3	1
0.95	4	1
0.96	5	1
0.966666667	6	0.983333333
0.971428571	7	0.985714286
0.975	8	0.9875

0.977777778	9	0.977777778
0.97	10	0.97



可以看到 ABN 模型在前面 10 位排名的标准准确率都高于向量模型。因此我们有理由认为如果结合相关反馈的话 ABN 模型会表现更为突出。

5. *Conclusion*

下面我总结一下 ABN 模型的优点和缺点：

优点：

1) 降低分词的难度，因为如果要标记两个词时相邻的话（用最小匹配原则），在分词必须记下每个词出现的 Position，这样查询时才可以判断两个词的距离来达到判定相邻的目的。

2) Bayesian Network 可以表达向量模型和其他基于向量模型的查询扩展无法表达的信息。例如“医疗保险”这个词，它的两个子词是“医疗”，“保险”。

那么相应概率 P(001)（即两个子词不出现而复合词出现的概率），P(011),P(101)和 P(111)分别为

P(001)	0.2413793103448276
P(101)	0.51724137931034486
P(011)	0.0
P(111)	0.2413793103448276

也就是说三个词都出现的概率比只出现“医疗”和“医疗保险”两个词的概率还要少，所以这种三个词都出现情况相应所得的相关度分数（即（1）式中的概率）就会降低，而扩展查询无论怎样对索引项赋权越多匹配的索引项相关度就越大。还有一种情况就是两个子词是互斥的，即 P(011), P(101)都比较大，但 P(111)很小，这是向量模型也无法表现（不过这种情况很少见）。

缺点：

1) 计算时间复杂度上，ABN 模型是向量模型的2^n倍（n 为子词数目），即如果 2 个子词的话就是 4 倍。而$P(k|Kq)$和归一化项$\sqrt{\sum_{i=0}^{t} w_{i,j}{}^2}$ 可以离线计算。
2) 另一个很致命的缺点就是这里假设查询词必须预先知道，不然的话 3.1 和 3.2 中描述分词和索引数据库的要求无法实现，上面的$P(k|Kq)$也很难预先计算。
3) 这里的模型只考虑了子词的相关性而没有考虑其他索引项跟查询词的相关性，所以性能上跟向量模型比较优势不明显。
4) 这个模型考虑的是简单的查询词，如果是比较长的查询句子就恐怕不太适合。